Annex 3

Statistical Materials: Source Code

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: biro_se_datastep.r                                                     #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# biro_se_datastep.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_data_format
#   BIRO_loaddata
#
################################################################################


################################################################################
#  BIRO_data_format
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_data_format is used to format variables in a dataset
#
# DEPENDENCIES
#
# biro_se_setup.r
#
################################################################################
#PARAMETERS:
##data [episode] = tareget dataset
#n_var [NULL]  = a vector containing the names of numeric variables
#c_var [NULL]  = a vector containing the names of character variables
#d_var [NULL]  = a vector containing the names of date variables storing as
#                string
#f_var [NULL]  = a vector containing the names of factor variables
#datefmt  !    = input date format
################################################################################


BIRO_data_format<-function(data=episode,
                           n_var=NULL,
                           c_var=NULL,
                           d_var=NULL,
                           f_var=NULL,
                           datefmt) {



 #convert into numeric variables
 if (is.null(n_var)==FALSE) {
  n_var<-na.omit(n_var)
  if (length(n_var)>0) {
   for (i in 1:length(n_var)) {
    data[,pmatch(n_var[i],names(data))]<-
    as.numeric(as.character(data[,pmatch(n_var[i],names(data))]))
   }
  }
 }
 #convert into string variables
 if (is.null(c_var)==FALSE) {
  c_var<-na.omit(c_var)
  if (length(c_var)>0) {
   for (i in 1:length(c_var)) {
    data[,pmatch(c_var[i],names(data))]<-
    as.character(data[,pmatch(c_var[i],names(data))])
   }
  }
 }
 #convert into date variables
 if (is.null(d_var)==FALSE) {
  d_var<-na.omit(d_var)
  if (length(d_var)>0) {
   for (i in 1:length(d_var)) {
    data[,pmatch(d_var[i],names(data))]<-
    as.Date(as.character(data[,pmatch(d_var[i],names(data))]),format=datefmt)
   }
  }
 }
```

```
 #convert into categorical variables
 if (is.null(d_var)==FALSE) {
  f_var<-na.omit(f_var)
  if (length(f_var)>0) {
   for (i in 1:length(f_var)) {
    data[,pmatch(f_var[i],names(data))]<-
     as.factor(as.character(data[,pmatch(f_var[i],names(data))]))
   }
  }
 }
 return(data)
}


################################################################################
#  BIRO_loaddata
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
#-------------------------
# DESCRIPTION
#
# BIRO_loaddata load data from a db or a csv file
#
# DEPENDENCIES
#
################################################################################
#PARAMETERS:
#dbformat        = the format of the database ("postgres","csv")
#drv             = the JDBC driver to connect R to the database
#                  (if not dbformat="csv")
#pathdb          = the path of the database
#user            = a string containing the username
#password        = a string containing the password
#dbname          = a string containing the database name
#dirdatastore    = a string containing the path of the input csv files
#centre_id       = a string containing the centre id
#yearnow         = a numeric value containing the last possible year of the
#                  episode dates
#fromyear        = a numeric value containing the first possible year where
#                  data are ok
#fromyearl       = a numeric value containing the first possible year where
#                  data are ok for the trends
#refyear         = a string containing year to calculate the indicators with
#                  "in the last 12 months" suffix
#refdate         = a string date to calculare age in the statistics about the
#                  refyear interval
################################################################################


BIRO_loaddata<-function(dbformat,
                        drv,
                        pathdb,
                        user,
                        password,
                        dbname,
                        dirdatastore,
                        centre_id,
                        yearnow,
                        fromyear,
                        fromyearl,
                        startdate,
                        enddate,
                        refdate) {

 if (dbformat=="postgres") {
  assign("conn",dbConnect(drv,pathdb,user,password,dbname),envir=.GlobalEnv)

  # Gets data from PGSQL and creates 4 csv s with demographic,
  # clinical, mortality, population and site data

  if (wide==0) {
  ## Import data from Postgres

   x <- "SELECT DISTINCT profile_field_name from profile"

   patient<-as.data.frame(dbGetQuery(conn, x))


   tabpatvar<-as.data.frame(dbGetQuery(conn, x))
   patvar<-tabpatvar$profile_field_name
   rm(tabpatvar)

  x <- (dbSendUpdate(conn, "DROP TABLE IF EXISTS t1"))
  x <- (dbSendUpdate(conn,
                     "CREATE TABLE t1 AS Select DISTINCT patient_id FROM profile"))
  print("First query executed")

  ## Create index
  x <- paste("create index patient_id ON t1 (patient_id)")
  dbSendUpdate(conn, x)
  for (i in 1:length(patvar)) {
   x <- paste("CREATE TEMPORARY TABLE ", patvar[i], " AS ",
              "Select DISTINCT profile.profile_field_value AS ",patvar[i],
              ",profile.patient_id FROM profile WHERE profile_field_name='",
              patvar[i],"'",sep="")
   dbSendUpdate(conn, x)

   x <- paste("create index ", patvar[i], "patient_id ON ", patvar[i], " (patient_id)
", sep = "")
   print(paste("About to create index: ", x))
   dbSendUpdate(conn, x)
   print(paste(i, "th index created",sep=""))

  }

  x <- (dbSendUpdate(conn, "DROP TABLE IF EXISTS xxx"))
  select.part <- NULL
  join.part <- NULL
  commavar=","
  for (i in 1:length(patvar)) {
   if (i == length(patvar)) {
    commavar=""
   }
   select.part <- paste(select.part, patvar[i], ".", patvar[i], commavar, sep = "" )
   join.part <- paste(join.part, "LEFT JOIN ", patvar[i], " ON ",
                      patvar[i], ".patient_id = ", "t1.patient_id ",
```

```
                      sep = "" )
  }
  full.query <- paste("CREATE TABLE xxx AS select t1.patient_id, ", select.part,
                      "from t1 ",
                      join.part)
  dbSendUpdate(conn, full.query)

  x<-"Select * FROM xxx"

  res<-dbSendQuery(conn,x)
  patient<-NULL
  nofinished<-TRUE
  i=0
  while(nofinished) {
    i=i+1
    temp<-as.data.frame(fetch(res,400))
    if ((dim(temp)[1]>0) & i==1) {
     patient<-temp
    }
    if ((dim(temp)[1]>0) & i>1) {
     patient<-rbind(patient,temp)
    }
    if (dim(temp)[1]==0) {
     nofinished<-FALSE
    }
  }
  rm(temp)

  x <- (dbSendUpdate(conn, "DROP TABLE IF EXISTS xxx"))
  x <- "SELECT DISTINCT episode_field_name from data"

  tabepivar<-as.data.frame(dbGetQuery(conn, x))
  epivar<-tabepivar$episode_field_name
  rm(tabepivar)

## Create temporary tables for each variable
  for (i in 1:length(epivar)) {
   x <- paste("CREATE TEMPORARY TABLE ", epivar[i], " AS ",
              "Select DISTINCT data.episode_field_value AS ",epivar[i],
              ",data.episode_data_id FROM data WHERE episode_field_name='",
              epivar[i],"'",sep="")


   dbSendUpdate(conn, x)
   print(paste(i, "th query done",sep=""))
   ## Create indexes
   x <- paste("create index ", epivar[i], "episode_data_id ON ", epivar[i], "
(episode_data_id) ", sep = "")
   print(paste("About to create index: ", x))
   dbSendUpdate(conn, x)
   print(paste(i, "th index created",sep=""))

  }

  x <- (dbSendUpdate(conn, "DROP TABLE IF EXISTS xxx"))
  select.part <- NULL
  join.part <- NULL
  commavar=","
  for (i in 1:length(epivar)) {
   if (i == length(epivar)) {
    commavar=""
   }
   select.part <- paste(select.part, epivar[i], ".", epivar[i], commavar, sep = "" )
   join.part <- paste(join.part, "LEFT JOIN ", epivar[i], " ON ",
                      epivar[i], ".episode_data_id = ", "t2.episode_data_id ",
                      sep = "" )
  }
  full.query <- paste("CREATE TABLE xxx AS select t2.patient_id, t2.episode_data_id, "
, select.part,
                      "from t2 ",
                      join.part)
  dbSendUpdate(conn, full.query)

  x<-"Select * FROM xxx"

  res<-dbSendQuery(conn,x)
  episode<-NULL
  nofinished<-TRUE
  i=0
  while(nofinished) {
    i=i+1
    temp<-as.data.frame(fetch(res,400))
    if ((dim(temp)[1]>0) & i==1){
     episode<-temp
    }
    if ((dim(temp)[1]>0) & i>1){
     episode<-rbind(episode,temp)
    }
    if (dim(temp)[1]==0) {
     nofinished<-FALSE
    }
  }
  rm(temp)

  x <- (dbSendUpdate(conn, "DROP TABLE IF EXISTS xxx"))

 }

 # 17th May 2010 modify anayears if years not prensented

  for (thisyear in anayears) {

partname<-paste(thisyear,substr(as.character(refanadate),1,2),substr(as.character(refa
nadate),4,5),sep="")
    patyear<-paste("profile_wide_",partname,sep="")

    if (dbExistsTable(conn, patyear)==FALSE) {
     anayears[anayears==thisyear]<-NA
   }
  }
  anayears<-as.vector(na.omit(anayears))
  assign("anayears",anayears,envir=.GlobalEnv)
 ############################################################

 if (wide==1) {
  for (thisyear in anayears) {
   if (activitytable==1) {
     if (dbExistsTable(conn, "activity_data")) {
   x<-"Select * FROM profile_wide RIGHT JOIN activity_data ON
profile_wide.patient_id=activity_data.patient_id"
   }
   } else {
#     if (dbExistsTable(conn, "metadata")) {
#      x<-"Select * FROM metadata"
```

```
#       res<-dbSendQuery(conn,x)
#       metadata<-as.data.frame(fetch(res,n=-1))
#       print(metadata)
#
partname<-paste(thisyear,substr(as.character(refanadate),1,2),substr(as.character(refa
nadate),4,5),sep="")
#       patyear<-paste("profile_wide_",partname,sep="")
#       thisprofile<-metadata[metadata$table_name==patyear,]
#
patient<-data.frame(matrix(NA,nrow=thisprofile$row_count,ncol=thisprofile$column_count
))
#       print(dim(patient))
#     }


partname<-paste(thisyear,substr(as.character(refanadate),1,2),substr(as.character(refa
nadate),4,5),sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    print(paste("Importing data from",patyear))
    x<-paste("Select * FROM ",patyear,sep="")
   }


  res<-dbSendQuery(conn,x)
  patient<-NULL
  nofinished<-TRUE
  i=0
  while(nofinished) {
    i=i+1
    temp<-as.data.frame(fetch(res,400))
    if ((dim(temp)[1]>0) & i==1){
     patient<-temp
    }
    if ((dim(temp)[1]>0) & i>1){
     patient<-rbind(patient,temp)
    }
    if (dim(temp)[1]==0) {
     nofinished<-FALSE
    }
   }

  print(paste(dim(patient)[1], "records (patients)"))

  assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

 cat(" <tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
  cat(" <td class=\"reportheader\"> Total No. Subjects: ", dim(patient)[1],
      "</td>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)


  if ("sub_ds_id" %in% patient) {

  assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

 cat(" <tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
  cat(" <td class=\"reportheader\"> Sub Centres: ", unique(patient$sub_ds_id),
      "</td>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)


  }


  assign(patyear,patient,envir=.GlobalEnv)
  rm(patient)

  }
  rm(temp)

  for (thisyear in anayears) {
  if (dbExistsTable(conn, "episode_wide")) {
#   if (dbExistsTable(conn, "metadata")) {
#    x<-"Select * FROM metadata"
#    res<-dbSendQuery(conn,x)
#    metadata<-as.data.frame(fetch(res,n=-1))
#    print(metadata)
#
partname<-paste(thisyear,substr(as.character(refanadate),1,2),substr(as.character(refa
nadate),4,5),sep="")
#    ewyear<-paste("episode_wide_",partname,sep="")
#    thisepisode<-metadata[metadata$table_name==ewyear,]
#
episode<-data.frame(matrix(NA,nrow=thisepisode$row_count,ncol=thisepisode$column_count
))
#    print(dim(episode))
#   }



partname<-paste(thisyear,substr(as.character(refanadate),1,2),substr(as.character(refa
nadate),4,5),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    print(paste("Importing data from",ewyear))
    x<-paste("Select * FROM ",ewyear,sep="")


   res<-dbSendQuery(conn,x)
   episode<-NULL
   nofinished<-TRUE
   i=0
   while(nofinished) {
     i=i+1
     temp<-as.data.frame(fetch(res,400))

     if ((dim(temp)[1]>0) & i==1){
      episode<-temp
     }
     if ((dim(temp)[1]>0) & i>1){
      episode<-rbind(episode,temp)
```

```
    }
    if (dim(temp)[1]==0) {
     nofinished<-FALSE
    }
   }
   rm(temp)
  }
  print(paste(dim(episode)[1], "records (episodes)"))

  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

 cat(" <tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
  cat(" <td class=\"reportheader\"> Total No. Episodes: ", dim(episode)[1],
      "</td>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)

  cat("Total No. Episodes: ", dim(episode)[1],
      file=paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""),fill =
TRUE,append=TRUE)

  assign(ewyear,episode,envir=.GlobalEnv)
  rm(episode)

  }
  }
 }

 allepivar <- c("epi_date","stroke","mi","amput","dialysis","weight","bmi",
                "sbp","dbp","hypertension","hba1c","creat","chol","hdl","ldl",
                "drug_therapy","height","ma_test","smok_stat","retina",
                "ulcer","foot_exam","esrf",
                "retinal_exam","pump_therapy","sulphonylureas","biguanides",
"glucosidase_inhibitors","glitazones","glinides",
                "lipid_therapy","antiplatelet_therapy",
                "self_mon","transplant","hypert_med","microal")

for (thisyear in anayears) {

partname<-paste(thisyear,substr(as.character(refanadate),1,2),substr(as.character(refa
nadate),4,5),sep="")
 ewyear<-paste("episode_wide_",partname,sep="")
 patyear<-paste("profile_wide_",partname,sep="")
 patient<-get(patyear)
 episode<-get(ewyear)

 if (exists(ewyear)==FALSE) {
  #create a data frame of NA
  episode<-matrix(nrow = 1, ncol = length(allepivar)+2)
  episode<-as.data.frame(episode)
  names(episode)<-c("episode_data_id","patient_id",allepivar)
  episode<-na.omit(episode)
  assign(ewyear,episode,envir=.GlobalEnv)
 }

 if (dbExistsTable(conn, "site_profile")) {
  site<-as.data.frame(dbGetQuery(conn, "Select * FROM site_profile"))
 } else site<-NULL
```

```
# if (dbExistsTable(conn, "site_header")) {
#  centre_id=as.character(
#      unique(dbGetQuery(conn, "Select site_header_id from site_header"))[1])
# }
 #Simulating missing RDU variables

 ##########################################

 #Store data read from db
 csvpatient<-paste(dirstoreout,"/patient",thisyear,".csv",sep="")
 csvepisode<-paste(dirstoreout,"/episode",thisyear,".csv",sep="")
 csvpopulation<-paste(dirstoreout,"/pop.csv",sep="")
 csvpopdiab<-paste(dirstoreout,"/popdiab.csv",sep="")

#popdiab

 if (nchar(filepopdiab)>0) {
  popdiab<-read.csv(filepopdiab)
  write.csv(popdiab,csvpopdiab)


  popdiabm<-popdiab[,-pmatch("diabF",names(popdiab))]
  popdiabf<-popdiab[,-pmatch("diabM",names(popdiab))]
  popdiabm$sex=1
  names(popdiabm)[pmatch("diabM",names(popdiabm))]<-"Freqs"
  popdiabf$sex=2
  names(popdiabf)[pmatch("diabF",names(popdiabf))]<-"Freqs"
  popdiab-rbind(popdiabf,popdiabm)


  assign("popdiab",popdiab,envir=.GlobalEnv)
 }

 if (nchar(filepopdiab)==0) {
  if (dbExistsTable(conn, "diabetic_population_table")) {
   popdiab<-dbReadTable(conn, "diabetic_population_table")
   names(popdiab)[pmatch("deaths_f",names(popdiab))]<-"morF"
   names(popdiab)[pmatch("deaths_m",names(popdiab))]<-"morM"
   names(popdiab)[pmatch("pop_d_f",names(popdiab))]<-"popF"
   names(popdiab)[pmatch("pop_d_m",names(popdiab))]<-"popM"
   names(popdiab)[pmatch("pop_d_ageband",names(popdiab))]<-"ageband"
   names(popdiab)[pmatch("pop_d_sub_ds_id",names(popdiab))]<-"centre_id"
   names(popdiab)[pmatch("pop_d_year",names(popdiab))]<-"year"
   names(popdiab)[pmatch("pop_d_type_dm",names(popdiab))]<-"type_dm"
   popdiab$morF<-as.numeric(as.character(popdiab$morF))
   popdiab$morM<-as.numeric(as.character(popdiab$morM))
   popdiab$popF<-as.numeric(as.character(popdiab$popF))
   popdiab$popM<-as.numeric(as.character(popdiab$popM))
   popdiab$ageband<-as.numeric(as.character(popdiab$ageband))
   popdiab$year<-as.numeric(as.character(popdiab$year))
   popdiab$centre_id<-as.character(popdiab$centre_id)
   #write.csv(popdiab,csvpopdiab)
   assign("popdiab",popdiab,envir=.GlobalEnv)
  }
 }



 write.csv(patient,csvpatient)
 write.csv(episode,csvepisode)

 #write.csv(pop,csvpopulation)
```

```
#####################################################
#####################################################
#####################################################
 #Temporarly here (waiting the changes to the XML Schema)
 if (nchar(filepop)>0) {
  pop<-read.csv(filepop)
  write.csv(pop,csvpopulation)
  assign("pop",pop,envir=.GlobalEnv)
 }

 if (nchar(filepop)==0) {
  if (dbExistsTable(conn, "population_table")) {
   pop<-dbReadTable(conn, "population_table")
   if (is.na(pmatch("deaths_f",names(pop)))==FALSE) names(pop)[pmatch("deaths_f"
,names(pop))]<-"morF"
   if (is.na(pmatch("deaths_m",names(pop)))==FALSE) names(pop)[pmatch("deaths_m"
,names(pop))]<-"morM"
   names(pop)[pmatch("pop_f",names(pop))]<-"popF"
   names(pop)[pmatch("pop_m",names(pop))]<-"popM"
   names(pop)[pmatch("pop_ageband",names(pop))]<-"ageband"
   names(pop)[pmatch("pop_sub_ds_id",names(pop))]<-"centre_id"
   names(pop)[pmatch("pop_year",names(pop))]<-"year"
  if (is.na(pmatch("morF",names(pop)))==FALSE)
pop$morF<-as.numeric(as.character(pop$morF))
  if (is.na(pmatch("morM"
,names(pop)))==FALSE)pop$morM<-as.numeric(as.character(pop$morM))
  pop$popF<-as.numeric(as.character(pop$popF))
  pop$popM<-as.numeric(as.character(pop$popM))
  pop$ageband<-as.numeric(as.character(pop$ageband))
  pop$year<-as.numeric(as.character(pop$year))
  #pop$centre_id<-as.character(pop$centre_id)
  #write.csv(pop,csvpopulation)
  assign("pop",pop,envir=.GlobalEnv)
  }
 }

################
#####################################################
#####################################################
#####################################################


 if (dbformat=="csv") {
  if (length(list.files(dirdatastore)) > 0) {
   patient<-read.csv(paste(dirdatastore,"/patient.csv",sep=""))
   episode<-read.csv(paste(dirdatastore,"/episode.csv",sep=""))
   pop<-read.csv(paste(dirdatastore,"/pop.csv"))
  }
 }


 d_var<-c("dob")
 if (is.na(pmatch("dt_diag",names(patient)))==FALSE) {
  d_var<-c(d_var,"dt_diag")
 }
# print(d_var)
  print("formatting patient data")
# print(names(patient))
 patient<-BIRO_data_format(data=patient,
                           n_var="age",
                           c_var=c("patient_id","ds_id"),
                           d_var=d_var,
                           f_var="type_dm",
                           datefmt="%Y-%m-%d")
  print("formatting done")
#  print(dim(patient))
 allnvar<-c("weight",
            "height","sbp","dbp","creat","chol","hdl",
            "ldl","ma_test","bmi","hba1c","stroke","mi",
            "hypertension","smok_stat","retina",
            "retina","ulcer","ulcer","foot_exam",
            "esrf","retinal_exam","pump_therapy",
            "sulphonylureas","biguanides","glucosidase_inhibitors","glitazones",
"glinides",
            "lipid_therapy","antiplatelet_therapy",
            "self_mon","transplant","dialysis","amput",
            "drug_therapy")

 if (dim(episode)[1]>0) {
  print("formatting episode data")

 episode<-BIRO_data_format(data=episode,

n_var=names(episode[na.omit(pmatch(allnvar,names(episode)))]),
                           c_var="patient_id",
                           d_var=c("epi_date"),
                           datefmt="%Y-%m-%d")
  print("formatting done")
  print(paste(dim(episode)[1], "records (episode)"))

 }

 site<-BIRO_data_format(data=site,
                        n_var=c("ds_denom","ds_beds","ds_physicians",
                                "ds_diabetologists","ds_doctors"))

 write("processing file: biro_se_recode.r",file="")
 if (nchar(logfile)>0) {
  write("processing file: biro_se_recode.r",file=logfile,append=TRUE)
 }

 source(paste(dirse,"/source/r/formats/biro_se_recode.r",sep=""))

 write("biro_se_recode.r executed",file="")
 if (nchar(logfile)>0) {
  write("biro_se_recode.r executed",file=logfile,append=TRUE)
 }


####################################
#   Timeframe for analysis         #
####################################



# patient$dob<-as.character(patient$dob)
# patient$dob<-ifelse((as.numeric(substr(patient$dob,1,4))<1900),
# paste("19",substr(patient$dob,3,10),sep=""),patient$dob)

# patient$dob<-as.character(patient$dob)
# patient$dob<-ifelse((as.numeric(substr(patient$dob,1,4))>yearnow),
# paste("19",substr(patient$dob,3,10),sep=""),patient$dob)
```

```r
# patient$dob<-as.Date(as.character(patient$dob))
# print("this")

#episode<-merge(patient[,c("patient_id","dob")],episode,by="patient_id",all.x=FALSE,al
l.y=TRUE)
# print("check")
# print(paste(dim(episode)[1], "records (episode)"))

 #episode$epi_date[as.numeric(episode$epi_date)<as.numeric(episode$dob)]<-NA
 #episode<-episode[,-pmatch("dob",names(episode))]

 #assign patient, episode, population to the global environment

 #replace 0 values with missing values (R bug)
    if ("age" %in% names(patient)) {
     patient$age<-ifelse(patient$age==0,NA,patient$age)
    }
    if ("weight" %in% names(episode)) {
     episode$weight<-ifelse(episode$weight==0,NA,episode$weight)
    }

    if ("height" %in% names(episode)) {
     episode$height<-ifelse(episode$height==0,NA,episode$height)
    }

    if ("sbp" %in% names(episode)) {
     episode$sbp<-ifelse(episode$sbp==0,NA,episode$sbp)
    }

    if ("dbp" %in% names(episode)) {
     episode$dbp<-ifelse(episode$dbp==0,NA,episode$dbp)
    }

    if ("creat" %in% names(episode)) {
     episode$creat<-ifelse(episode$creat==0,NA,episode$creat)
    }

    if ("chol" %in% names(episode)) {
     episode$chol<-ifelse(episode$chol==0,NA,episode$chol)
    }

    if ("hdl" %in% names(episode)) {
     episode$hdl<-ifelse(episode$hdl==0,NA,episode$hdl)
    }

    if ("ldl" %in% names(episode)) {
     episode$ldl<-ifelse(episode$ldl==0,NA,episode$ldl)
    }

    if ("bmi" %in% names(episode)) {
     episode$bmi<-ifelse(episode$bmi==0,NA,episode$bmi)
    }

    if ("hba1c" %in% names(episode)) {
     episode$hba1c<-ifelse(episode$hba1c==0,NA,episode$hba1c)
    }

 assign(patyear,patient,envir=.GlobalEnv)
 assign(ewyear,episode,envir=.GlobalEnv)
 assign("site",site,envir=.GlobalEnv)

  print(paste(dim(episode)[1], "records (episode)"))


 #create table of variable
 #varpat
 #varepi
if (thisyear==anayears[1]) varpat<-as.data.frame(t(rep(FALSE,dim(patient)[2])))


 for (i in 1:dim(patient)[2]) {
  varpat[i]<-ifelse(varpat[1,i]==FALSE,length(na.omit(patient[,i]))>0,varpat[1,i])
 }
names(varpat)<-names(patient)
assign("varpat",varpat,envir=.GlobalEnv)

if (thisyear==anayears[1]) varepi<-as.data.frame(t(rep(FALSE,dim(episode)[2])))

 for (i in 1:dim(episode)[2]) {
  varepi[i]<-ifelse(varepi[1,i]==FALSE,length(na.omit(episode[,i]))>0,varepi[1,i])
 }
names(varepi)<-names(episode)

 for (i in 1:length(allepivar)) {
  if (is.na(pmatch(allepivar[i],names(varepi)))==TRUE) {
    varepi<-cbind(varepi,FALSE)
    names(varepi)<-c(names(varepi[1:(dim(varepi)[2]-1)]),allepivar[i])
  }

 }

assign("varepi",varepi,envir=.GlobalEnv)


}


}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: biro_se.r                                                              #
#                                                                              #
#
# ##############################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# biro_se.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#
# ##############################################################################

# ##############################################################################
#  BIRO_se                                                                     #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_se is the main function of the Statistical Engine
#
# DEPENDENCIES
#
# biro_se_setup.r
# biro_se_datastep.r
#
# ##############################################################################
# PARAMETERS:
# dbformat        !   = the format of the database ("postgres","csv")
# driverClass     !   = a string containing the class of the driver
# classPath       !   = a string containing the path of the driver
# identifier.quote !  = a string containing the identifier of quotes required by
#                       JDBC package
# pathdb          !   = a string containing the path of the database
# user            !   = a string containing the username
# password        !   = a string containing the password
# dbname          !   = a string containing the name of the database
# dirdatastore    !   = a string containing the path of the input csv files
# centre_id       !   = a string containing the centre id
# yearnow         !   = a numeric value containing the last possible year of the
#                       episode dates
# fromyear        !   = a numeric value containing the first possible year where
#                       data are reliable
# fromyearl       !   = a numeric value containing the first possible year where
#                       data are reliable for the trends
# endyearl        !   = a numeric value containing the last possible year where
#                       data are reliable for the trends
# refyear         !   = a numeric containing year to calculate the indicators with
#                       "in the last 12 months" suffix
# refdate         !   = a string date to calculare age in the statistics about the
#                       refyear interval
# logfile   [""]      = a string containing the name of the .log file. If "" only
#                       screen print
# cex                 = A numerical value giving the amount by which
#                       plotting text and symbols should be magnified
#                       relative to the default. Note that some graphics
#                       functions such as plot.default have an argument of
#                       this name which multiplies this graphical
#                       parameter, and some functions such as points accept
#                       a vector of values which are recycled.
#                       Other uses will take just the first value if a
#                       vector of length greater than one is supplied.
# filepop             = population file path
# filepopdiab         = diabetic population file path
# activitytable       = 1 if exists, 0 otherwise
# compiletex          = 1 if you want to compile, 0 otherwise
# ID_Continent   [""] = part of a 7 field center's key indicating continent
# ID_Countries   [""] = part of a 7 field center's key indicating country
# ID_Macroregion [""] = part of a 7 field center's key indicating macro region
# ID_Region      [""] = part of a 7 field center's key indicating region
# ID_Province    [""] = part of a 7 field center's key indicating province
# ID_District    [""] = part of a 7 field center's key indicating district
# ID_Postcode    [""] = part of a 7 field center's key indicating post code
#
# ##############################################################################

BIRO_se<-function(dirse,
                  dirout="",
                  dbformat, # "postgres" if data are stored in a psotgres database,
                            # "csv" if data are stored in csv files
```

```
                driverClass,
                classPath,
                identifier.quote,
                pathdb,
                user,
                password,
                dbname,
                dirdatastore,
                centre_id,
                startdate="01/01/1997", #Bergen meeting this must be a date (the
start date interval for data analysis)
                enddate="01/01/2005",   #Bergen meeting include a final date
interval for data analysis
                yearnow,
                fromyear,
                refanadate,
                refdate,
                logfile="",
                cex,
                wide=1,
                filepop="",
                filepopdiab="",
                activitytable=0,
                compiletex=1,
                ID_Continent="",
                ID_Countries="",
                ID_Macroregion="",
                ID_Region="",
                ID_Province="",
                ID_District="",
                ID_Postcode="") {


 myR<-R.Version()
 ver<-myR$major
 subv<-myR$minor
 strv<-myR$version.string

 if (ver=="2" & (subv=="8.0" | subv=="8.1")) {
  print(paste(strv, "is running"))
 } else {
  print("Statistical engine is not running in a compatibile R version")
  print(paste(strv, "is running"))
  stop()
 }


 if (dirout=="") dirout=dirse

 #2010-09-22

 assign("ID_Continent",ID_Continent,envir=.GlobalEnv)
 assign("ID_Countries",ID_Countries,envir=.GlobalEnv)
 assign("ID_Macroregion",ID_Macroregion,envir=.GlobalEnv)
 assign("ID_Region",ID_Region,envir=.GlobalEnv)
 assign("ID_Province",ID_Province,envir=.GlobalEnv)
 assign("ID_District",ID_District,envir=.GlobalEnv)
 assign("ID_Postcode",ID_Postcode,envir=.GlobalEnv)


 mylog<-logfile
 startdate=as.Date(startdate,format="%Y-%m-%d")
 enddate=as.Date(enddate,format="%Y-%m-%d")
 intervaldd<-difftime(enddate,startdate,units="days")
 assign("intervaldd",intervaldd,envir=.GlobalEnv)
 anayears<-format(startdate,format="%Y"):format(enddate,format="%Y")
 assign("engine","se",envir=.GlobalEnv)
 assign("anayears",anayears,envir=.GlobalEnv)
 assign("wide",wide,envir=.GlobalEnv)
 assign("dirse",dirse,envir=.GlobalEnv)
 assign("dirout",dirout,envir=.GlobalEnv)
 assign("centre_id",centre_id,envir=.GlobalEnv)
 assign("dbname",dbname,envir=.GlobalEnv)
 assign("startdate",startdate,envir=.GlobalEnv)
 assign("enddate",enddate,envir=.GlobalEnv)
 fromyearl=as.character(format(as.Date(startdate,"%Y-%m-%d"),"%Y"))
 endyear=as.character(format(as.Date(enddate,"%Y-%m-%d"),"%Y"))
 assign("yearnow",yearnow,envir=.GlobalEnv)
 assign("fromyearl",startdate,envir=.GlobalEnv)
 assign("endyearl",endyear,envir=.GlobalEnv)
 assign("refanadate",refanadate,envir=.GlobalEnv)
 assign("cex",cex,envir=.GlobalEnv)
 assign("logfile",logfile,envir=.GlobalEnv)
 #Directory Store Input data (aggregated for in patient and episode datasets)
 assign("dirdatastore",dirdatastore,envir=.GlobalEnv)
 assign("filepop",filepop,envir=.GlobalEnv)
 assign("filepopdiab",filepopdiab,envir=.GlobalEnv)
 assign("activitytable",activitytable,envir=.GlobalEnv)

 timestart<-Sys.time()

 write("processing file: biro_se_setup.r",file="")
 source(paste(dirse,"/source/r/main/biro_se_setup.r",sep=""))
 write("biro_se_setup.r executed",file="")
 write("calling BIRO_dircreate",file="")

 BIRO_dircreate(dirse=dirse,
                dirout=dirout,
                centre_id=centre_id)
 write("BIRO_dircreate executed",file="")

 write("processing file: biro_se_datastep.r",file="")
 source(paste(dirse,"/source/r/main/biro_se_datastep.r",sep=""))
 write("biro_se_datastep.r executed",file="")



 logfile<-paste(dirout,"/output/reports/",logfile,sep="")
 assign("logfile",logfile,envir=.GlobalEnv)

 write("File .log created",file="")
 if (nchar(logfile)>0) {
  write("File .log created",file=logfile,append=FALSE)
 }

 write(paste("Indicator Number: ",length(anayears)*73,sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("Indicator Number: ",length(anayears)*73,sep=""
),file=logfile,append=TRUE)
 }

 assign("indnum",0,envir=.GlobalEnv)
 write("calling BIRO_setenv",file="")
 if (nchar(logfile)>0) {
```

```
 write("calling BIRO_setenv",file=logfile,append=TRUE)
 }

 BIRO_setenv(driverClass=driverClass,
             classPath=classPath,
             identifier.quote=identifier.quote,
             pathdb=pathdb,
             user=user,
             password=password,
             dbname=dbname)

 write("BIRO_setenv executed",file="")
 if (nchar(logfile)>0) {
  write("BIRO_setenv executed",file=logfile,append=TRUE)
 }
 if (nchar(logfile)>0) {
  write(paste("Date: ",as.character(Sys.Date()),sep=""),file=logfile,append=TRUE)
  write(paste("Begin ",as.character(timestart),sep=""),file=logfile,append=TRUE)
 }

 write(paste("Directory CSV ",as.character(dirdataout),sep=""
),file=logfile,append=TRUE)

 write("calling BIRO_loaddata",file="")
 if (nchar(logfile)>0) {
  write("calling BIRO_loaddata",file=logfile,append=TRUE)
 }
 BIRO_loaddata(dbformat=dbformat,
               drv=drv,
               pathdb=pathdb,
               user=user,
               password=password,
               dbname=dbname,
               dirdatastore=dirdatastore,
               centre_id=centre_id,
               yearnow=yearnow,
               fromyear=fromyear,
               fromyearl=fromyearl,
               startdate=startdate,
               enddate=enddate,
               refdate=refdate)

 write("BIRO_loaddata executed",file="")
 if (nchar(logfile)>0) {
  write("BIRO_loaddata executed",file=logfile,append=TRUE)
 }


 if (exists("site")) {
  assign("site",site,envir=.GlobalEnv)
  BIRO_dframe(dataname="site",monitor=TRUE,filelog="")
 }

# BIRO_dframe(dataname="patient",monitor=TRUE,filelog="")
# BIRO_dframe(dataname="episode",monitor=TRUE,filelog="")



 for (thisyear in anayears) {

partname<-paste(thisyear,substr(as.character(refanadate),1,2),substr(as.character(refa
nadate),4,5),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
assign("patient",get(patyear),envir=.GlobalEnv)
assign("episode",get(ewyear),envir=.GlobalEnv)

  #cat(paste("output/data/", "#",launchtime,
  #      sep=""),
  #      file=paste(dirout,"/dirout.txt",sep=""),fill = TRUE,append=FALSE)

  #cat(paste("output/reports/", "#",launchtime,
  #      sep=""),
  #      file=paste(dirout,"/dirout.txt",sep=""),fill = TRUE,append=TRUE)

  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

  assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  write(paste("All Outputs in [OUTDIR]:",dirreport),file="")
  if (nchar(logfile)>0) {
   write(paste("All Outputs in [OUTDIR]:",dirreport),file=logfile,append=TRUE)
  }
  write(paste("Statistical Objects saved into [OUTDIR]:",dirdataout),file="")
  if (nchar(logfile)>0) {
   write(paste("Statistical Objects saved into [OUTDIR]:"
,dirdataout),file=logfile,append=TRUE)
  }


  cat("</tbody>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
  cat("</table>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)


  cat("opening .Tex file. year:", thisyear,file="",append=TRUE)
  if (nchar(logfile)>0) {
   cat("opening .Tex file. year:", thisyear,file="",append=TRUE)
  }

  open_tex(title="",
           authors="",
           footnote="",
           time="",
           logo="",
           layout=layout,
           final=texfile)

 # Second Title Page
 cat("\\begin{titlepage}",
     file=texfile,fill = TRUE,append=TRUE)


 cat("\\section*{\\huge Parameters used for the production\\\\of this Statistical
Report}",
     file=texfile,fill = TRUE,append=TRUE)

 cat("{\\bf Author}:  The BIRO Consortium,\\\\",
     file=texfile,fill = TRUE,append=TRUE)
```

```
 cat("{\\bf Date}: ",format(as.Date(launchtime,format="%d%m%y%H%M%S"),format="%d %B
%Y"), "\\\\",
     file=texfile,fill = TRUE,append=TRUE)

 cat("{\\bf Time}: ",format(strptime(launchtime, "%d%m%y%H%M%S"),format="%H:%M:%S"),
"\\\\",
     file=texfile,fill = TRUE,append=TRUE)

 cat("{\\bf EUBIROD source:}",centre_id,"\\\\",
     file=texfile,fill = TRUE,append=TRUE)
 cat(paste("{\\bf Time interval:}",format(as.Date(startdate,format="%d%m%y%H%M%S"
),format="%d %m %Y"),"-",
     format(as.Date(enddate,format="%d%m%y%H%M%S"),format="%d %m %Y")),"\\\\",
     file=texfile,fill = TRUE,append=TRUE)

 cat("{\\bf Output Directory:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\lstset{breaklines=true,basicstyle=\\footnotesize, basewidth=0.5em}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(dirreport,
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\end{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\framebox[3in][l]{\\bf INPUT DATA}\\\\[0.2 cm]",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\bf Database:}",dbname,"\\\\[0.4 cm]",
     file=texfile,fill = TRUE,append=TRUE)



#Waiting Stefano or Valentina for these parameters
#
merge_table=""
urlacttable=""
urlpoptable=""
urldiabtable=""
urldatastruc=""

if (merge_table!="") {
 cat("{\\bf Merge Table:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(merge_table,
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\end{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
}
if (urlacttable!="") {
 cat("{\\bf Activity  Dataset:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(urlacttable,
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\end{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
}
if (urlpoptable!="") {
 cat("{\\bf Population  Table:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(urlpoptable,
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\end{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
}

if (urldiabtable!="") {
 cat("{\\bf Diabetic Population Table:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(urldiabtable,
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\end{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
}

if (urldatastruc!="") {
 cat("{\\bf Data Source Structure:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(urldatastruc,
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\end{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
}

 cat("\\framebox[3in][l]{\\bf CONTENTS} \\\\[0.2 cm]",
     file=texfile,fill = TRUE,append=TRUE)


  cat("Total No. Subjects: ", dim(patient)[1], "\\\\",
      file=texfile,fill = TRUE,append=TRUE)
  if ("sub_ds_id" %in% patient) {
   cat("Sub Centres:  ", unique(patient$sub_ds_id), "\\\\",
       file=texfile,fill = TRUE,append=TRUE)
  }
  cat("Total No. Episodes: ", dim(episode)[1], "\\\\",
      file=paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""),fill =
TRUE,append=TRUE)
 }

cat("\\vfill",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\begin{center}",
    file=texfile,fill = TRUE,append=TRUE)
cat("{\\large Copyright the BIRO Consortium 2010}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\end{center}",
    file=texfile,fill = TRUE,append=TRUE)

 cat("\\end{titlepage}",
     file=texfile,fill = TRUE,append=TRUE)

 # Including Help PDF files
```

```
cat("\\newpage",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\mbox{}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\fancyhf{}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\renewcommand{\\headrulewidth}{0pt}",
    file=texfile,fill = TRUE,append=TRUE)

cat("\\begin{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\includegraphics[width=1\\textwidth]{images/report_structure.pdf}\\\\",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\end{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)

cat("\\begin{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\includegraphics[width=1\\textwidth]{images/simple_oneway_table.pdf}\\\\",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\end{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)

cat("\\begin{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\includegraphics[width=1\\textwidth]{images/simple_oneway_graphs.pdf}\\\\",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\end{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)

cat("\\begin{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\includegraphics[width=1\\textwidth]{images/boxplots_graphs.pdf}\\\\",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\end{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)

cat("\\begin{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\includegraphics[width=1\\textwidth]{images/subdatasource_graphs.pdf}\\\\",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\end{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)

cat("\\begin{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\includegraphics[width=1\\textwidth]{images/simple_twoway_table.pdf}\\\\",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\end{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)

cat("\\begin{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\includegraphics[width=1\\textwidth]{images/trellis_twoway_graphs.pdf}\\\\",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\end{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)

cat("\\begin{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\includegraphics[width=1\\textwidth]{images/riskadjusted.pdf}\\\\",
    file=texfile,fill = TRUE,append=TRUE)
```

```
cat("\\end{titlepage}",
    file=texfile,fill = TRUE,append=TRUE)

cat("\\newpage",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\mbox{}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\fancyhf{}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\renewcommand{\\headrulewidth}{0pt}",
    file=texfile,fill = TRUE,append=TRUE)

cat("\\newpage",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\mbox{}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\fancyhf{}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\renewcommand{\\headrulewidth}{0pt}",
    file=texfile,fill = TRUE,append=TRUE)



 cat("\\vspace{8 cm}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\centerline{}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\centerline{}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\centerline{}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\vspace{8 cm}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\centerline{\\bf }",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\centerline{\\bf }",
     file=texfile,fill = TRUE,append=TRUE)

 cat("\\thispagestyle{empty}",
     file=texfile,fill = TRUE,append=TRUE)


 cat("\\pagestyle{plain}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\newpage",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\pagenumbering{roman}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\tableofcontents",
     file=texfile,fill = TRUE,append=TRUE)



 refanadate<-paste(substr(refanadate,4,5),substr(refanadate,1,2),sep="/")
 assign("refanadate",refanadate,envir=.GlobalEnv)

 # ################################
 #                                #
 #        Create Reports          #
 #                                #
 # ################################
```

```
  #Demographic characteristics

 for (thisyear in anayears) {
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
  assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
  new_chapter(title="Demographic characteristics",texfile=texfile)

  cat("\\setcounter{page}{1}",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\pagenumbering{arabic}",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\pagestyle{plain}",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\mbox{}",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\fancyhf{}",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\renewcommand{\\headrulewidth}{0.5pt}",
     file=texfile,fill = TRUE,append=TRUE)

  cat("\\newpage",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\pagestyle{fancy}",
     file=texfile,fill = TRUE,append=TRUE)

  new_chapter_html(title="Demographic characteristics",htmlfile=htmlfile)
 }

 write("processing file: biro_se_indicator_demographic.r",file="")
 if (nchar(logfile)>0) {
  write("processing file: biro_se_indicator_demographic.r",file=logfile,append=TRUE)
 }
  source(paste(dirse,"/source/r/scripts/","biro_se_indicator_demographic.r",
        sep=""))
 write("biro_se_indicator_demographic.r executed",file="")
 if (nchar(logfile)>0) {
  write("biro_se_indicator_demographic.r executed",file=logfile,append=TRUE)
 }

 #Clinical characteristics
 for (thisyear in anayears) {
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
  assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
  new_chapter(title="Clinical characteristics",texfile=texfile)
  cat("\\newpage",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\pagestyle{fancy}",
     file=texfile,fill = TRUE,append=TRUE)

  new_chapter_html(title="Clinical characteristics",htmlfile=htmlfile)
 }
 write("processing file: biro_se_indicator_clinical.r",file="")
 if (nchar(logfile)>0) {
  write("processing file: biro_se_indicator_clinical.r",file=logfile,append=TRUE)
 }
  source(paste(dirse,"/source/r/scripts/","biro_se_indicator_clinical.r",
        sep=""))
 write("biro_se_indicator_clinical.r executed",file="")
 if (nchar(logfile)>0) {
  write("biro_se_indicator_clinical.r executed",file=logfile,append=TRUE)
 }

 #Health System
 for (thisyear in anayears) {
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
  assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
  new_chapter(title="Health System",texfile=texfile)
  cat("\\newpage",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\pagestyle{fancy}",
     file=texfile,fill = TRUE,append=TRUE)

  new_chapter_html(title="Health System",htmlfile=htmlfile)
 }
 write("processing file: biro_se_indicator_health_system.r",file="")
 if (nchar(logfile)>0) {
  write("processing file: biro_se_indicator_health_system.r",file=logfile,append=TRUE)
 }
  source(paste(dirse,"/source/r/scripts/","biro_se_indicator_health_system.r",
        sep=""))
 write("biro_se_indicator_health_system.r executed",file="")
 if (nchar(logfile)>0) {
  write("biro_se_indicator_health_system.r executed",file=logfile,append=TRUE)
 }

 #Population
 for (thisyear in anayears) {
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
  assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
  new_chapter(title="Population",texfile=texfile)
  cat("\\newpage",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\pagestyle{fancy}",
     file=texfile,fill = TRUE,append=TRUE)

  new_chapter_html(title="Population",htmlfile=htmlfile)
 }
 write("processing file: biro_se_indicator_population.r",file="")
 if (nchar(logfile)>0) {
  write("processing file: biro_se_indicator_population.r",file=logfile,append=TRUE)
 }
  source(paste(dirse,"/source/r/scripts/","biro_se_indicator_population.r",
        sep=""))
 write("biro_se_indicator_population.r executed",file="")
 if (nchar(logfile)>0) {
  write("biro_se_indicator_population.r executed",file=logfile,append=TRUE)
 }
 #Risk Adjusted
```

```
 for (thisyear in anayears) {
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
  assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
  new_chapter(title="Risk Adjusted Indicators",texfile=texfile)
  cat("\\newpage",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\pagestyle{fancy}",
     file=texfile,fill = TRUE,append=TRUE)

  new_chapter_html(title="Risk Adjusted Indicators",htmlfile=htmlfile)
 }
 write("processing file: biro_se_indicator_risk_adjusted.r",file="")
 if (nchar(logfile)>0) {
  write("processing file: biro_se_indicator_risk_adjusted.r",file=logfile,append=TRUE)
 }
  source(paste(dirse,"/source/r/scripts/","biro_se_indicator_risk_adjusted.r",
         sep=""))
 write("biro_se_indicator_risk_adjusted.r executed",file="")
 if (nchar(logfile)>0) {
  write("biro_se_indicator_risk_adjusted.r executed",file=logfile,append=TRUE)
 }

 for (thisyear in anayears) {
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
  close_tex(file=texfile)

  setwd(dirreport)

  assign("layout",paste(dirsourcelatex,"/layout.tex",sep=""),envir=.GlobalEnv)
  assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
  assign("dvifile",paste(dirreport,"/report.dvi",sep=""),envir=.GlobalEnv)
  assign("pdffile",paste(dirreport,"/,report",thisyear,".pdf",sep=""
),envir=.GlobalEnv)

  if (compiletex==1) {

   texfile<-paste("\"",texfile,"\"",sep="")
   command<-paste("pdflatex",texfile)

   assign("diraux",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/pdf",sep=""),envir=.GlobalEnv)
   write("Preparing to compile .Tex File",file="")


   if (nchar(logfile)>0) {
    write("Preparing to compile .Tex File",file=logfile,append=TRUE)
   }

   x<-system(command,intern=TRUE)
   x<-system(command,intern=TRUE)
   x<-system(command,intern=TRUE)

   pathtex=substr(texfile,1,nchar(texfile)-5)
   print(pathtex)
   file.rename(from=paste(dirreport,"/",dbname,"_",thisyear,".aux",sep=""),
to=paste(dirreport,"/pdf/",dbname,"_",thisyear,".aux",sep=""))
   file.rename(from=paste(dirreport,"/",dbname,"_",thisyear,".out",sep=""),
to=paste(dirreport,"/pdf/",dbname,"_",thisyear,".out",sep=""))
   file.rename(from=paste(dirreport,"/",dbname,"_",thisyear,".log",sep=""),
to=paste(dirreport,"/pdf/",dbname,"_",thisyear,".log",sep=""))
   file.rename(from=paste(dirreport,"/",dbname,"_",thisyear,".toc",sep=""),
to=paste(dirreport,"/pdf/",dbname,"_",thisyear,".toc",sep=""))
   file.rename(from=paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""),
to=paste(dirreport,"/pdf/",dbname,"_",thisyear,".tex",sep=""))

   write("Tex File Compiled",file="")
   if (nchar(logfile)>0) {
    write("Tex File Compiled",file=logfile,append=TRUE)
   }
  }

  if (compiletex==0) {
   write(paste("FILETEX:",texfile),file="",append=TRUE)
  }
 }

 timefinish<-Sys.time()

 write(paste("Elapsed time:",timefinish-timestart, "mins",sep=" "),file="")

 cat(" Elapsed time:", timefinish-timestart, "mins",
     file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)

 if (nchar(logfile)>0) {
  write(paste("Elapsed time:",timefinish-timestart, "mins",sep=" "),
        file=logfile,append=TRUE)
 }
 #newlogfile<-paste(dirreport,"/",mylog,sep="")
 #file.copy(from=logfile,to=newlogfile)

 file.remove(logfile)

 for (thisyear in anayears) {
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

  file.copy(from=paste(dirout,"/statisticalEngineSinkFile.txt",sep=""
),to=paste(dirreport,"/",dbname,"_",thisyear,".log",sep=""))

  unlink(paste(dirreport,"/html/temp",sep=""),recursive=TRUE)
  unlink(paste(dirreport,"/pdf/temp",sep=""),recursive=TRUE)


 }
 write("Statistical engine process finished", file="")
}
```

```
# ##############################################################################
#
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_se_setup.r                                                        #
#                                                                              #
#
# ##############################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_se_setup.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
# BIRO_setenv
# BIRO_dircreate
#
# ##############################################################################

# ##############################################################################
#  BIRO_setenv
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_setenv sets the environment of the BIRO Statistical Engine
#
# DEPENDENCIES
#
# R package: RJDBC
#
# ##############################################################################
# PARAMETERS:
# driverClass      !  = a string containing the class of the driver
# classPath        !  = a string containing the path of the driver
# identifier.quote !  = a string containing the identifier of quotes
# pathdb           !  = a string containing the path of the database
# user             !  = a string containing the username
# password         !  = a string containing the password
# dbname           !  = a string containing the name of the database
# ##############################################################################


BIRO_setenv<-function(driverClass,
                      classPath,
                      identifier.quote,
                      pathdb,
                      user,
                      password,
                      dbname) {

#R Packages installation

 mySys<-R.Version()

 if (substr(mySys$os,1,5)=="mingw") {
  write("O.S. Windows",file="")
  if (nchar(logfile)>0) {
   write("O.S. Windows",file=logfile,append=TRUE)
  }
  dirpack<-paste(substr(dirse,1,nchar(dirse)-4),"lib/r/source/packages",sep="")

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Cairo"])==0) {
   install.packages(paste(dirpack,"/win/Cairo_1.4-4.zip",sep=""), repos = NULL)
   write("Cairo package installed from binary zip file... O.S. Windows",file="")
   if (nchar(logfile)>0) {
    write("Cairo package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="rJava"])==0) {
   install.packages(paste(dirpack,"/win/rJava_0.6-0.zip",sep=""), repos = NULL)
   write("rJava package installed from binary zip file... O.S. Windows",file="")
   if (nchar(logfile)>0) {
    write("rJava package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="DBI"])==0) {
```

```
   install.packages(paste(dirpack,"/win/DBI_0.2-4.zip",sep=""), repos = NULL)
   write("DBI package installed from binary zip file... O.S. Windows",file="")
    if (nchar(logfile)>0) {
     write("DBI package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
    }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="RJDBC"])==0) {
   install.packages(paste(dirpack,"/win/RJDBC_0.1-5.zip",sep=""), repos = NULL)
   write("RJDBC package installed from binary zip file... O.S. Windows",file="")
    if (nchar(logfile)>0) {
     write("RJDBC package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
    }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="lattice"])==0) {
   install.packages(paste(dirpack,"/win/lattice_0.17-15.zip",sep=""), repos = NULL)
   write("lattice package installed from binary zip file... O.S. Windows",file="")
    if (nchar(logfile)>0) {
     write("lattice package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
    }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="rmeta"])==0) {
   install.packages(paste(dirpack,"/win/rmeta_2.14.zip",sep=""), repos = NULL)
   write("rmeta package installed from binary zip file... O.S. Windows",file="")
    if (nchar(logfile)>0) {
     write("rmeta package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
    }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="sp"])==0) {
   install.packages(paste(dirpack,"/win/sp_0.9-28.zip",sep=""), repos = NULL)
   write("sp package installed from binary zip file... O.S. Windows",file="")
    if (nchar(logfile)>0) {
     write("sp package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
    }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="maptools"])==0) {
   install.packages(paste(dirpack,"/win/maptools_0.7-15.zip",sep=""), repos = NULL)
   write("maptools package installed from binary zip file... O.S. Windows",file="")
    if (nchar(logfile)>0) {
     write("maptools package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
    }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Hmisc"])==0) {
   install.packages(paste(dirpack,"/win/Hmisc_3.4-4.zip",sep=""), repos = NULL)
   write("Hmisc package installed from binary zip file... O.S. Windows",file="")
    if (nchar(logfile)>0) {
     write("Hmisc package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
    }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="R2HTML"])==0) {
   install.packages(paste(dirpack,"/win/R2HTML_1.59.zip",sep=""), repos = NULL)
   write("R2HTML package installed from binary zip file... O.S. Windows",file="")
    if (nchar(logfile)>0) {
     write("R2HTML package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
    }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Epi"])==0) {
   install.packages(paste(dirpack,"/win/Epi_1.0.8.zip",sep=""), repos = NULL)
   write("Epi package installed from binary zip file... O.S. Windows",file="")
    if (nchar(logfile)>0) {
     write("Epi package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
    }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="epicalc"])==0) {
   install.packages(paste(dirpack,"/win/epicalc_2.8.0.0.zip",sep=""), repos = NULL)
   write("epicalc package installed from binary zip file... O.S. Windows",file="")
    if (nchar(logfile)>0) {
     write("epicalc package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
    }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Rserve"])==0) {
   install.packages(paste(dirpack,"/win/Rserve_0.5-3.zip",sep=""), repos = NULL)
   write("epicalc package installed from binary zip file... O.S. Windows",file="")
    if (nchar(logfile)>0) {
     write("epicalc package installed from binary zip file... O.S. Windows"
,file=logfile,append=TRUE)
    }
  }

 }

 if ((substr(mySys$os,1,5)=="unix") | (substr(mySys$os,1,5)=="Linux")) {
  write("O.S. Linux",file="")
  if (nchar(logfile)>0) {
   write("O.S. Linux",file=logfile,append=TRUE)
  }

  dirpack<-paste(substr(dirse,1,nchar(dirse)-4),"lib/r/source/packages",sep="")

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Cairo"])==0) {
   install.packages(paste(dirpack,"/linux/Cairo_1.4-4.tar.gz",sep=""), repos = NULL)
   write("Cairo package installed from tar.gz file... O.S. Linux",file="")
    if (nchar(logfile)>0) {
     write("Cairo package installed from tar.gz file... O.S. Linux"
```

```
,file=logfile,append=TRUE)
    }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="rJava"])==0) {
   install.packages(paste(dirpack,"/linux/rJava_0.6-0.tar.gz",sep=""), repos = NULL)
   write("rJava package installed from tar.gz file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("rJava package installed from tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="DBI"])==0) {
   install.packages(paste(dirpack,"/linux/DBI_0.2-4.tar.gz",sep=""), repos = NULL)
   write("DBI package installed from tar.gz file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("DBI package installed from tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="RJDBC"])==0) {
   install.packages(paste(dirpack,"/linux/RJDBC_0.1-5.tar.gz",sep=""), repos = NULL)
   write("RJDBC package installed from tar.gz file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("RJDBC package installed from tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="lattice"])==0) {
   install.packages(paste(dirpack,"/linux/lattice_0.17-17.tar.gz",sep=""), repos =
NULL)
   write("lattice package installed from tar.gz file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("lattice package installed from tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="rmeta"])==0) {
   install.packages(paste(dirpack,"/linux/rmeta_2.14.tar.gz",sep=""), repos = NULL)
   write("rmeta package installed from tar.gz file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("rmeta package installed from tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="sp"])==0) {
   install.packages(paste(dirpack,"/linux/sp_0.9-28.tar.gz",sep=""), repos = NULL)
   write("sp package installed from tar.gz file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("sp package installed from tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="maptools"])==0) {
   install.packages(paste(dirpack,"/linux/maptools_0.7-16.tar.gz",sep=""), repos =
NULL)
   write("maptools package installed from tar.gz file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("maptools package installed from tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Hmisc"])==0) {
   install.packages(paste(dirpack,"/linux/Hmisc_3.4-4.tar.gz",sep=""), repos = NULL)
   write("Hmisc package installed from tar.gz file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("Hmisc package installed from tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="R2HTML"])==0) {
   install.packages(paste(dirpack,"/linux/R2HTML_1.59.tar.gz",sep=""), repos = NULL)
   write("R2HTML package installed from tar.gz file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("R2HTML package installed from tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Epi"])==0) {
   install.packages(paste(dirpack,"/linux/Epi_1.0.8.tar.gz",sep=""), repos = NULL)
   write("Epi package installed from tar.gz file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("Epi package installed from tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="epicalc"])==0) {
   install.packages(paste(dirpack,"/linux/epicalc_2.8.0.0.tar.gz",sep=""), repos =
NULL)
   write("epicalc package installed from tar.gz zip file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("epicalc package installed from tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Rserve"])==0) {
   install.packages(paste(dirpack,"/mac/Rserve_0.5-3.tar.gz",sep=""), repos = NULL)
   write("epicalc package installed from binary tar.gz file... O.S. Linux",file="")
   if (nchar(logfile)>0) {
    write("epicalc package installed from binary tar.gz file... O.S. Linux"
,file=logfile,append=TRUE)
```

```
   }
  }

 }

 if ("apple" %in% unlist(strsplit(as.character(mySys$platform),"-"))) {
  write("Apple Computer",file="")
  if (nchar(logfile)>0) {
   write("Apple Computer",file=logfile,append=TRUE)
  }
  dirpack<-paste(substr(dirse,1,nchar(dirse)-4),"lib/r/source/packages",sep="")

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Cairo"])==0) {
   install.packages(paste(dirpack,"/mac/Cairo_1.4-4.tgz",sep=""), repos = NULL)
   write("Cairo package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("Cairo package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="rJava"])==0) {
   install.packages(paste(dirpack,"/mac/rJava_0.6-0.tgz",sep=""), repos = NULL)
   write("rJava package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("rJava package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="DBI"])==0) {
   install.packages(paste(dirpack,"/mac/DBI_0.2-4.tgz",sep=""), repos = NULL)
   write("DBI package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("DBI package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="RJDBC"])==0) {
   install.packages(paste(dirpack,"/mac/RJDBC_0.1-5.tgz",sep=""), repos = NULL)
   write("RJDBC package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("RJDBC package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="lattice"])==0) {
   install.packages(paste(dirpack,"/mac/lattice_0.17-20.tgz",sep=""), repos = NULL)
   write("lattice package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("lattice package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="rmeta"])==0) {
   install.packages(paste(dirpack,"/mac/rmeta_2.14.tgz",sep=""), repos = NULL)
   write("rmeta package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("rmeta package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="sp"])==0) {
   install.packages(paste(dirpack,"/mac/sp_0.9-28.tgz",sep=""), repos = NULL)
   write("sp package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("sp package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="maptools"])==0) {
   install.packages(paste(dirpack,"/mac/maptools_0.7-20.tgz",sep=""), repos = NULL)
   write("maptools package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("maptools package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Hmisc"])==0) {
   install.packages(paste(dirpack,"/mac/Hmisc_3.4-2.tgz",sep=""), repos = NULL)
   write("Hmisc package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("Hmisc package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="R2HTML"])==0) {
   install.packages(paste(dirpack,"/mac/R2HTML_1.59.tgz",sep=""), repos = NULL)
   write("R2HTML package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("R2HTML package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Epi"])==0) {
   install.packages(paste(dirpack,"/mac/Epi_1.0.8.tgz",sep=""), repos = NULL)
   write("Epi package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("Epi package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="epicalc"])==0) {
```

```
   install.packages(paste(dirpack,"/mac/epicalc_2.8.1.1.tgz",sep=""), repos = NULL)
   write("epicalc package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("epicalc package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

  if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available =
TRUE))=="Rserve"])==0) {
   install.packages(paste(dirpack,"/mac/Rserve_0.5-3.tgz",sep=""), repos = NULL)
   write("epicalc package installed from binary tgz file... Apple Computer",file="")
   if (nchar(logfile)>0) {
    write("epicalc package installed from binary tgz file... Apple Computer"
,file=logfile,append=TRUE)
   }
  }

 }

 library(rJava)
 library(RJDBC)
 library(lattice)
 library(rmeta)
 library(maptools)
 library(Hmisc)
 library(R2HTML)
 library(Cairo)
 library(Rserve)


 assign("drv",JDBC(driverClass =driverClass,classPath=classPath,
        identifier.quote=identifier.quote),envir=.GlobalEnv)
}

# ##############################################################################
#  BIRO_dircreate
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_dircreate creates directories according the BIRO directories schema
#
# DEPENDENCIES
#
# r package: base
#
# ##############################################################################
# PARAMETERS:
# dirse      !        = a string containing the path of the Statistical Engine
#                      directory
# centre_id  !        = a string containing the id of the centre
# ##############################################################################
```

```
BIRO_dircreate<-function(dirse,
                         dirout,
                         centre_id) {
 dir.create(dirse, showWarnings = FALSE, recursive = TRUE)
 #Directory Reports
 assign("dirse",dirse,envir=.GlobalEnv)
 assign("dirsourceR",paste(dirse,"/source/r",sep=""),envir=.GlobalEnv)
 assign("dirsourcelatex",paste(substr(dirse,1,nchar(dirse)-4),
        "lib/templates/pdf",sep=""),envir=.GlobalEnv)
 #Load ".../lib/R/source/biro" functions

 assign("dirlibR",paste(substr(dirse,1,nchar(dirse)-4),
        "lib/r/source/biro",sep=""),envir=.GlobalEnv)

 listlib<-c("biro_aggregate.r",
            "biro_demographic.r",
            "biro_explife.r",
            "biro_forest.r",
            "biro_maps.r",
            "biro_plots.r",
            "biro_regression.r",
            "biro_standardize.r",
            "biro_trellis.r",
            "biro_util.r",
            "biro_util.r",
            "biro_webplots.r")

 for (i in 1:length(list.files(dirlibR))) {
  source(paste(dirlibR,"/",listlib[i],sep=""))
 }

 #Load ".../_se_/source/R/include" functions
 assign("dirinclude",paste(dirsourceR,"/include",sep=""),envir=.GlobalEnv)

 #list of files to include
 listinc<-c("biro_se_boxplots.r",
            "biro_se_dispersion.r",
            "biro_se_histograms.r",
            "biro_se_location.r",
            "biro_se_report.r",
            "biro_se_tables.r",
            "biro_se_trend.r")
 for (i in 1:length(listinc)) {
  source(paste(dirinclude,"/",listinc[i],sep=""))
 }

 dir.create(dirsourceR, showWarnings = FALSE, recursive = TRUE)
 dir.create(dirsourcelatex, showWarnings = FALSE, recursive = TRUE)
 #Local Repository of DB data in csv format

 launchtime<-format(Sys.time(),"%d%m%y%H%M%S")
 assign("launchtime",launchtime,envir=.GlobalEnv)
 dirstoreout<-paste(dirout,"/data/", "#",launchtime,"/",centre_id,sep="")
 assign("dirstoreout",dirstoreout,envir=.GlobalEnv)
 dir.create(dirstoreout, showWarnings = FALSE, recursive = TRUE)
 #Directory Local Components
 for (thisyear in anayears) {
  assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
  dir.create(dirdataout, showWarnings = FALSE, recursive = TRUE)

  #write the info file for BIROAdaptor
  infofile<-paste(paste(dirout,"/output/data/", "#",launchtime,sep=""),"/info.csv"
,sep="")

  cat("centre id:",centre_id,file=infofile,fill = TRUE,append=FALSE)
  cat("startdate:",as.character(startdate),file=infofile,fill = TRUE,append=TRUE)
  cat("enddate:",as.character(enddate),file=infofile,fill = TRUE,append=TRUE)

  assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/","graphs",sep=""),envir=.GlobalEnv)
  dir.create(dirgraph, showWarnings = FALSE, recursive = TRUE)
  assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  dir.create(dirtables, showWarnings = FALSE, recursive = TRUE)
  assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

  assign("diraux",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/pdf",sep=""),envir=.GlobalEnv)
  dirimages<-paste(dirout,"/output/reports/", "#",launchtime,
                   "/",thisyear,"/",centre_id,"/images",sep="")
  dir.create(dirimages, showWarnings = FALSE, recursive = TRUE)
  assign("dirlibR",paste(substr(dirse,1,nchar(dirse)-4),
         "lib/r/source/biro",sep=""),envir=.GlobalEnv)
  file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/html/",
"biro-logo01.jpg",sep=""),
            paste(dirimages,"/","biro-logo01.jpg",sep=""))
  file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/pdf/","EUBIROD.png"
,sep=""),
            paste(dirimages,"/","EUBIROD.png",sep=""))
  file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/pdf/","EUBIROD.pdf"
,sep=""),
            paste(dirimages,"/","EUBIROD.pdf",sep=""))

  #Help PDF files

  file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/pdf/",
"boxplots_graphs.pdf",sep=""),
            paste(dirimages,"/","boxplots_graphs.pdf",sep=""))

  file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/pdf/",
"report_structure.pdf",sep=""),
            paste(dirimages,"/","report_structure.pdf",sep=""))

  file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/pdf/",
"riskadjusted.pdf",sep=""),
            paste(dirimages,"/","riskadjusted.pdf",sep=""))

  file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/pdf/",
"simple_oneway_graphs.pdf",sep=""),
            paste(dirimages,"/","simple_oneway_graphs.pdf",sep=""))

  file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/pdf/",
"simple_oneway_table.pdf",sep=""),
            paste(dirimages,"/","simple_oneway_table.pdf",sep=""))

  file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/pdf/",
"simple_twoway_table.pdf",sep=""),
            paste(dirimages,"/","simple_twoway_table.pdf",sep=""))

  file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/pdf/",
"subdatasource_graphs.pdf",sep=""),
            paste(dirimages,"/","subdatasource_graphs.pdf",sep=""))

  file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/pdf/",
"trellis_twoway_graphs.pdf",sep=""),
            paste(dirimages,"/","trellis_twoway_graphs.pdf",sep=""))


  dir.create(diraux, showWarnings = FALSE, recursive = TRUE)
  dir.create(dirhtml, showWarnings = FALSE, recursive = TRUE)
  dir.create(dirwp, showWarnings = FALSE, recursive = TRUE)
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  assign("layout",paste(dirsourcelatex,"/layout.tex",sep=""),envir=.GlobalEnv)
  assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
  assign("dvifile",paste(dirreport,"/report.dvi",sep=""),envir=.GlobalEnv)
  assign("pdffile",paste(dirreport,"/report.pdf",sep=""),envir=.GlobalEnv)

  #create html report index

  file.copy(paste(substr(dirout,1,nchar(dirse)-4),"lib/templates/html/",
"layout_open.html",sep=""),
            paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""))

 cat("<body>",
     file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat("<table class=\"reportheader\" border=\"0\">",
     file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat("<tbody>",
     file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <tr>",
     file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat("  <td><img src=\"images/biro-logo01.jpg\"></td>",
     file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" </tr>",
     file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat("</tbody>",
     file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat("</table>",
     file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat("<table class=\"reportheader\" border=\"0\">",file=paste(dirreport,"/",dbname,"_"
,thisyear,".html",sep=""),fill = TRUE,append=TRUE)
 cat("<tbody>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <td class=\"reportheader\"> Date: ",
     format(as.Date(launchtime,format="%d%m%y%H%M%S"),format="%d %B %Y"),
     "</td>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
```

```
 cat(" </tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <td class=\"reportheader\"> EUBIROD source: ",centre_id,
     "</td>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <td class=\"reportheader\"> Reference Year: ", thisyear,
     "</td>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <td class=\"reportheader\"> Time interval: ", intervaldd,
     "</td>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <td class=\"reportheader\"> Input Database: ", dbname,
     "</td>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <td class=\"reportheader\"> Output Directory Report: ",dirreport,
     "</td>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)

 cat(" </tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 #cat(" <tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 #cat("  <td class=\"reportheader\">Timeframe:
",as.character(format(startdate,format="%d %B
%Y")),"-",as.character(format(enddate,format="%d %B %Y")),"</td>",
 #    file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 #cat(" </tr>",file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)

 }
 assign("dirmap",paste(substr(dirse,1,nchar(dirse)-4),
        "lib/maps",sep=""),envir=.GlobalEnv)
 assign("maploaded","",envir=.GlobalEnv)
 assign("dirgraphlatex","graphs",envir=.GlobalEnv)
 write("BIRO directories created",file="")
 if (nchar(logfile)>0) {
  write("BIRO directories created",file=logfile,append=TRUE)
 }


}
```

```
##############################################################################
#                                                                            #
# Project: BIRO-Project (Funded by European Commission 2005-2008)            #
# File: BIRO_se_indicator_clinical.r                                  #
#                                                                            #
#
##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-24
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_se_indicator_clinical.r is part of WP Statistical Engine of the BIRO
# Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
##############################################################################

  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   new_section(title="2.1 Diabetes Status",texfile=texfile)
   new_section_html(title="2.1 Diabetes Status",htmlfile=htmlfile)

  }
  #
  # 2.1.1 Type of Diabetes (All the patient live at checkdate)
  #

  if (varpat$type_dm==TRUE & varpat$dob==TRUE) {
```

```
    for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)

    data2_1_1<-patient[,c("patient_id","age","sex","type_dm")]
    data2_1_1$type_dm<-ifelse(is.na(data2_1_1$type_dm),99,data2_1_1$type_dm)
    data2_1_1$age_c<-varclass(data2_1_1$age,ageth)

    numlevelcat<-as.numeric(as.character(na.omit(unique(data2_1_1$type_dm))))

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_1_1<-merge(data2_1_1,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_1_1)[pmatch("sub_ds_id",names(data2_1_1))]<-"centre_id"

    BIRO_dframe(dataname="data2_1_1",monitor=TRUE,filelog="")

       BIRO_report(ind="2_1_1",
                   title="Type of diabetes",
                   var_cat="type_dm",
                   class=c("age_c"),
                   lev_var_cat=names(levtype_dm)[sort(numlevelcat)],
                   lev_class=list(a=classlabellist(ageth)),
                   lab_var_cat="Type of Diabetes",
                   lab_class=c("Age"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
```

```
                  bar=1,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex)

    rm(data2_1_1)
    rm(patient)
    }
    } else {
      for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

        BIRO_report_toc(title="2.1.1. Type of diabetes")
      }
    }
  #
  # 2.1.2 Duration of Diabetes
  #


  if (varpat$type_dm==TRUE & varpat$dob==TRUE & varpat$dt_diag==TRUE &
      varpat$sex==TRUE & varepi$hba1c==TRUE) {


    for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

  assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)
    data2_1_2<-patient[,c("patient_id","age","sex","type_dm","diabetes_duration")]

    #patient with at least one hba1c (valid measurement)

    hba1c<-as.data.frame(unique(episode[is.na(episode$hba1c)==FALSE & episode$hba1c>0,
"patient_id"]))
    names(hba1c)<-"patient_id"
    hba1c$hba1c_done<-1
    data2_1_2<-merge(data2_1_2,hba1c,by="patient_id",all.x=TRUE)

    data2_1_2$hba1c_done<-ifelse(is.na(data2_1_2$hba1c_done),2,data2_1_2$hba1c_done)

    data2_1_2$durdiab<-data2_1_2$diabetes_duration

    data2_1_2$durdiab_c<-varclass(data2_1_2$durdiab,durdiabth)

    numlevelcat<-na.omit(unique(data2_1_2$durdiab_c))

    data2_1_2$durdiab_c<-ifelse(is
.na(data2_1_2$durdiab_c)==TRUE,99,data2_1_2$durdiab_c)

    lev_var_cat=classlabel(durdiabth)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_1_2<-merge(data2_1_2,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_1_2)[pmatch("sub_ds_id",names(data2_1_2))]<-"centre_id"

    BIRO_dframe(dataname="data2_1_2",monitor=TRUE,filelog="")

       BIRO_report(ind="2_1_2",
                   title="Duration of diabetes (Classes)",
                   var_cat="durdiab_c",
                   class=c("sex","hba1c_done","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=levsex,
                                  b=list("with hba1c"=1,"without hba1c"=2),
                                  c=levtype_dm),
                   lab_var_cat="Duration of diabetes",
                   lab_class=c("Gender","Hba1c","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex)

    rm(hba1c)
    rm(data2_1_2)
    rm(patient)
    rm(episode)
```

```
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="2.1.2. Duration of diabetes (Classes)")
    }
  }

  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)

   new_section(title="2.2 Risk Factors",texfile=texfile)
   new_section_html(title="2.2 Risk Factors",htmlfile=htmlfile)

   new_section_html(title="2.2.1 Obesity",htmlfile=htmlfile)
  }
  #
  # 2.2.1.1 Weight (the last episode in the reference year)
  #
  if (varpat$type_dm==TRUE & varpat$dob==TRUE &
      varpat$sex==TRUE & varepi$weight==TRUE) {


    for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

  assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)
```

```
    patient$age_c<-varclass(patient$age,ageth)

    episode$weight_c<-varclass(episode$weight,weightth)

    if (dim(episode[is.na(episode$weight)==FALSE & episode$weight>0,])[1]>0) {
    #patient alive at checkdate
    data2_2_1_1<-BIRO_demographic(patient=patient,
                                  episode=episode[is.na(episode$weight)==FALSE &
episode$weight>0,],
                                  var=NULL,
                                  date="last episode date",
                                  startvar="dob",
                                  th=ageth,
                                  hold=TRUE,
                                  varname="age")

    data2_2_1_1<-merge(episode[is.na(episode$weight)==FALSE & episode$weight>0,
                       c("patient_id","epi_date","weight","weight_c")],
                       data2_2_1_1,
                       by.x=c("patient_id","epi_date"),
                       by.y=c("patient_id","last_epi_date"),
                       all.y=TRUE)

    data2_2_1_1<-merge(patient[,c("patient_id","sex","type_dm","age_c")],
                       data2_2_1_1,
                       by.x=c("patient_id"),
                       by.y=c("patient_id"),
                       all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_2_1_1$weight_c))
    data2_2_1_1$weight_c<-ifelse(is.na(data2_2_1_1$weight_c),99,data2_2_1_1$weight_c)

    lev_var_cat=classlabel(weightth)[sort(numlevelcat)]

    BIRO_dframe(dataname="data2_2_1_1",monitor=TRUE,filelog="")

   if (thisyear==anayears[1]) {
    data2_2_1_1l<-data2_2_1_1
    data2_2_1_1l$year_epi=thisyear

   } else {
     if (exists("data2_2_1_1l")) {
      data2_2_1_1$year_epi=thisyear
      data2_2_1_1l<-rbind(data2_2_1_1l,data2_2_1_1)
     } else {
    data2_2_1_1l<-data2_2_1_1
    data2_2_1_1l$year_epi=thisyear
     }
     }

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_2_1_1<-merge(data2_2_1_1,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_2_1_1)[pmatch("sub_ds_id",names(data2_2_1_1))]<-"centre_id"


    printlines=0
    if (thisyear==anayears[length(anayears)]) {
      lines=1
      printlines=1
      if (length(anayears)==1) {
        printlines=0
```

```
    }
  }


     BIRO_report(ind="2_2_1_1",
                 title="Weight (the most recent episode in 12 months)",
                 var="weight",
                 var_cat="weight_c",
                 class=c("sex","age_c","type_dm"),
                 lev_var_cat=lev_var_cat,
                 lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
                 lab_var_cat="Weight",
                 lab_class=c("Gender","Age","Type of Diabetes"),
                 tab=1,
                 tabside="H",
                 tabperc=TRUE,
                 tabwidth=0.95,
                 barbeside=TRUE,
                 barnumber=TRUE,
                 barperc=TRUE,
                 box=1,
                 bar=1,
                 lines=printlines,
                 varlines="weight",
                 timelines="year_epi",
                 trellis=1,
                 pie=0,
                 texfile=texfile,
                 dircsv=dirdataout,
                 dirgraph=dirgraph,
                 dirgraphlatex=dirgraphlatex,
                 cex=cex)


  rm(data2_2_1_1)
  rm(patient)
  rm(episode)
  } else {
    assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="2.2.1.1. Weight (the most recent episode in 12 months)")
  }
 }
  rm(data2_2_1_1l)
} else {
  for (thisyear in anayears) {
 assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="2.2.1.1. Weight (the most recent episode in 12 months)")
  }
 }

#
# 2.2.1.2 BMI (the last episode, not only refyear)
#

if ((varpat$type_dm==TRUE & varpat$dob==TRUE &
     varpat$sex==TRUE & varepi$bmi==TRUE) |
    (varpat$type_dm==TRUE & varpat$dob==TRUE &
     varpat$sex==TRUE & varepi$weight==TRUE &
     varepi$height==TRUE)) {

  for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)

    if (dim(episode[is.na(episode$bmi)==FALSE & episode$bmi>0,])[1]>0) {

    episode$bmi_c<-varclass(episode$bmi,bmith)

    data2_2_1_2<-BIRO_demographic(patient=patient,
                                  episode=episode[is.na(episode$bmi)==FALSE &
episode$bmi>0,],
                                  var=NULL,
                                  date="last episode date",
                                  startvar="dob",
                                  th=ageth,
                                  hold=TRUE,
                                  varname="age")

    data2_2_1_2<-merge(episode[is.na(episode$bmi)==FALSE & episode$bmi>0,
                               c("patient_id","epi_date","bmi","bmi_c")],
                       data2_2_1_2,
                       by.x=c("patient_id","epi_date"),
                       by.y=c("patient_id","last_epi_date"),
                       all.y=TRUE)

    data2_2_1_2<-merge(patient[,c("patient_id","sex","type_dm","age_c")],
                       data2_2_1_2,
```

```
                      by.x=c("patient_id"),
                      by.y=c("patient_id"),
                      all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_2_1_2$bmi_c))
    data2_2_1_2$bmi_c<-ifelse(is.na(data2_2_1_2$bmi_c),99,data2_2_1_2$bmi_c)

    lev_var_cat=classlabel(bmith)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_2_1_2<-merge(data2_2_1_2,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_2_1_2)[pmatch("sub_ds_id",names(data2_2_1_2))]<-"centre_id"

    BIRO_dframe(dataname="data2_2_1_2",monitor=TRUE,filelog="")


   if (thisyear==anayears[1]) {
    data2_2_1_2l<-data2_2_1_2
    data2_2_1_2l$year_epi=thisyear
   } else {
     if (exists("data2_2_1_2l")) {
      data2_2_1_2$year_epi=thisyear
      data2_2_1_2l<-rbind(data2_2_1_2l,data2_2_1_2)
     } else {
      data2_2_1_2l<-data2_2_1_2
      data2_2_1_2l$year_epi=thisyear
     }
    }
    printlines=0
    if (thisyear==anayears[length(anayears)]) {
      lines=1
      printlines=1
      if (length(anayears)==1) {
        printlines=0
      }
    }


       BIRO_report(ind="2_2_1_2",
                   title="BMI (the most recent episode in 12 months)",
                   var="bmi",
                   var_cat="bmi_c",
                   class=c("sex","age_c","type_dm"),
                   lev_var_cat=lev_var_cat,
                  lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
                   lab_var_cat="BMI",
                   lab_class=c("Gender","Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=1,
                   bar=1,
                   lines=printlines,
                   varlines="bmi",
                   timelines="year_epi",
                   trellis=1,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )

  rm(data2_2_1_2)
  rm(patient)
  rm(episode)

} else {
 assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

  BIRO_report_toc(title="2.2.1.2. BMI (the most recent episode in 12 months)")
}
}
  rm(data2_2_1_2l)

} else {
  for (thisyear in anayears) {
 assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="2.2.1.2. BMI (the most recent episode in 12 months)")
  }
}


new_section_html(title="2.2.2 Lifestyle",htmlfile=htmlfile)

#
# 2.2.2.1 Smoking Status (the last episode, not only refyear)
#


if (varpat$type_dm==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE & varepi$smok_stat==TRUE) {

  for (thisyear in anayears) {

 assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

 assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
       "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

 assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
       "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

 assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

 assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

 assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

 assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
```

```
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)

    if (dim(episode[is.na(episode$smok_stat)==FALSE,])[1]>0) {

  data2_2_2_1<-BIRO_demographic(patient=patient,
                                episode=episode[is.na(episode$smok_stat)==FALSE,],
                                var=NULL,
                                date="last episode date",
                                startvar="dob",
                                th=ageth,
                                hold=TRUE,
                                varname="age")

  data2_2_2_1<-merge(episode[is.na(episode$smok_stat)==FALSE,
                       c("patient_id","epi_date","smok_stat")],
                     data2_2_2_1,
                     by.x=c("patient_id","epi_date"),
                     by.y=c("patient_id","last_epi_date"),
                     all.y=TRUE)

    data2_2_2_1<-merge(patient[,c("patient_id","sex","type_dm","age_c")],
                       data2_2_2_1,
                       by.x=c("patient_id"),
                       by.y=c("patient_id"),
                       all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_2_2_1$smok_stat))
    data2_2_2_1$smok_stat<-ifelse(is
.na(data2_2_2_1$smok_stat),99,data2_2_2_1$smok_stat)

    lev_var_cat=names(levsmok_stat)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_2_2_1<-merge(data2_2_2_1,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_2_2_1)[pmatch("sub_ds_id",names(data2_2_2_1))]<-"centre_id"

  BIRO_dframe(dataname="data2_2_2_1",monitor=TRUE,filelog="")


      BIRO_report(ind="2_2_2_1",
                  title="Smoking status (the most recent episode in 12 months)",
                  var_cat="smok_stat",
                  class=c("sex","age_c","type_dm"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
                  lab_var_cat="Smoking status",
                  lab_class=c("Gender","Age","Type of Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=1,
                  bar=1,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,

                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex
                  )

  rm(data2_2_2_1)
    rm(patient)
    rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="2.2.2.1 Smoking status (the most recent episode in 12
months)")
   }
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="2.2.2.1 Smoking status (the most recent episode in 12
months)")
    }
  }

new_section_html(title="2.2.3. Clinical measurements",htmlfile=htmlfile)
  #
  # 2.2.3.1 Systolic BP (the last episode, not only refyear)
  #


  if (varpat$type_dm==TRUE & varpat$dob==TRUE &
      varpat$sex==TRUE & varepi$sbp==TRUE) {

    for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
```

```
   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)

    if (dim(episode[is.na(episode$sbp)==FALSE & episode$sbp>0,])[1]>0) {

    episode$sbp_c<-varclass(episode$sbp,sbpth)

    data2_2_3_1<-BIRO_demographic(patient=patient,
                                  episode=episode[is.na(episode$sbp)==FALSE &
episode$sbp>0,],
                                  var=NULL,
                                  date="last episode date",
                                  startvar="dob",
                                  th=ageth,
                                  hold=TRUE,
                                  varname="age")

    data2_2_3_1<-merge(episode[is.na(episode$sbp)==FALSE & episode$sbp>0,
                         c("patient_id","epi_date","sbp","sbp_c")],
                       data2_2_3_1,
                       by.x=c("patient_id","epi_date"),
                       by.y=c("patient_id","last_epi_date"),
                       all.y=TRUE)

    data2_2_3_1<-merge(patient[,c("patient_id","sex","type_dm","age_c")],
                       data2_2_3_1,
                       by.x=c("patient_id"),
                       by.y=c("patient_id"),
                       all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_2_3_1$sbp_c))
    data2_2_3_1$sbp_c<-ifelse(is.na(data2_2_3_1$sbp_c),99,data2_2_3_1$sbp_c)

    lev_var_cat=classlabel(sbpth)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_2_3_1<-merge(data2_2_3_1,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_2_3_1)[pmatch("sub_ds_id",names(data2_2_3_1))]<-"centre_id"

    BIRO_dframe(dataname="data2_2_3_1",monitor=TRUE,filelog="")
```

```
if (thisyear==anayears[1]) {
 data2_2_3_1l<-data2_2_3_1
 data2_2_3_1l$year_epi=thisyear
} else {
 if (exists("data2_2_1_3l")) {
  data2_2_3_1$year_epi=thisyear
  data2_2_3_1l<-rbind(data2_2_3_1l,data2_2_3_1)
 } else {
  data2_2_3_1l<-data2_2_3_1
  data2_2_3_1l$year_epi=thisyear
 }



 }
 printlines=0
 if (thisyear==anayears[length(anayears)]) {
   lines=1
   printlines=1
   if (length(anayears)==1) {
     printlines=0
   }
 }


    BIRO_report(ind="2_2_3_1",
                title="Systolic BP (the most recent episode in 12 months)",
                var="sbp",
                var_cat="sbp_c",
                class=c("sex","age_c","type_dm"),
                lev_var_cat=lev_var_cat,
                lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
                lab_var_cat="SBP",
                lab_class=c("Gender","Age","Type of Diabetes"),
                tab=1,
                tabside="H",
                tabperc=TRUE,
                tabwidth=0.95,
                barbeside=TRUE,
                barnumber=TRUE,
                barperc=TRUE,
                box=1,
                bar=1,
                lines=0,
                printlines=printlines,
                varlines="sbp",
                timelines="year_epi",
                trellis=1,
                pie=0,
                texfile=texfile,

                dircsv=dirdataout,
                dirgraph=dirgraph,
                dirgraphlatex=dirgraphlatex,
                cex=cex
                )

 rm(data2_2_3_1)
 rm(patient)
 rm(episode)
```

```
  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="2.2.3.1. Systolic BP (the most recent episode in 12
months)")
  }
    }
    rm(data2_2_3_1l)

  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="2.2.3.1. Systolic BP (the most recent episode in 12
months)")
    }
  }

  #
  # 2.2.3.2 Diastolic BP (the most recent episode in 12 months)
  #

  if (varpat$type_dm==TRUE & varpat$dob==TRUE &
      varpat$sex==TRUE & varepi$dbp==TRUE) {

    for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)

    if (dim(episode[is.na(episode$dbp)==FALSE & episode$dbp >0,])[1]>0) {

    episode$dbp_c<-varclass(episode$dbp,dbpth)

    data2_2_3_2<-BIRO_demographic(patient=patient,
                                  episode=episode[is.na(episode$dbp)==FALSE &
episode$dbp >0 ,],
                                  var=NULL,
                                  date="last episode date",
                                  startvar="dob",
                                  th=ageth,
                                  hold=TRUE,
                                  varname="age")

    data2_2_3_2<-merge(episode[is.na(episode$dbp)==FALSE,
                       c("patient_id","epi_date","dbp","dbp_c")],
                       data2_2_3_2,
                       by.x=c("patient_id","epi_date"),
                       by.y=c("patient_id","last_epi_date"),
                       all.y=TRUE)

    data2_2_3_2<-merge(patient[,c("patient_id","sex","type_dm","age_c")],
                       data2_2_3_2,
                       by.x=c("patient_id"),
                       by.y=c("patient_id"),
                       all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_2_3_2$dbp_c))
    data2_2_3_2$dbp_c<-ifelse(is.na(data2_2_3_2$dbp_c),99,data2_2_3_2$dbp_c)

    lev_var_cat=classlabel(dbpth)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_2_3_2<-merge(data2_2_3_2,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_2_3_2)[pmatch("sub_ds_id",names(data2_2_3_2))]<-"centre_id"

    BIRO_dframe(dataname="data2_2_3_2",monitor=TRUE,filelog="")


   if (thisyear==anayears[1]) {
    data2_2_3_2l<-data2_2_3_2
    data2_2_3_2l$year_epi=thisyear
   } else {
    if (exists("data2_2_3_2l")) {
     data2_2_3_2$year_epi=thisyear
     data2_2_3_2l<-rbind(data2_2_3_2l,data2_2_3_2)
    } else {
    data2_2_3_2l<-data2_2_3_2
    data2_2_3_2l$year_epi=thisyear
    }

   }
    printlines=0
    if (thisyear==anayears[length(anayears)]) {
      lines=1
      printlines=1
      if (length(anayears)==1) {
        printlines=0
      }
    }
```

```
      BIRO_report(ind="2_2_3_2",
                  title="Diastolic BP (the most recent episode in 12 months)",
                  var="dbp",
                  var_cat="dbp_c",
                  class=c("sex","age_c","type_dm"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
                  lab_var_cat="DBP",
                  lab_class=c("Gender","Age","Type of Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=1,
                  bar=1,
                  lines=0,
                  printlines=printlines,
                  varlines="dbp",
                  timelines="year_epi",
                  trellis=1,
                  pie=0,
                  texfile=texfile,

                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex
                  )

    rm(data2_2_3_2)
    rm(patient)
    rm(episode)


  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
    BIRO_report_toc(title="2.2.3.2. Diastolic BP (the most recent episode in 12
months)")
  }
  }
    rm(data2_2_3_2l)

  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="2.2.3.2. Diastolic BP (the most recent episode in 12
months)")
  }
  }

  #
  # 2.2.3.2 Total cholesterol (the most recent episode in 12 months)
  #

  if (varpat$type_dm==TRUE & varpat$dob==TRUE &
      varpat$sex==TRUE & varepi$chol==TRUE) {


    for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)

    episode$chol_c<-varclass(episode$chol,cholth)
    levchol<-list("0 - 2.58"=1,"2.59 - 5.17"=2,"5.18 - 7.76"=3,"7.77 + "=4)

    if (dim(episode[is.na(episode$chol)==FALSE & episode$chol >0,])[1]>0) {

    data2_2_3_3<-BIRO_demographic(patient=patient,
                                  episode=episode[is.na(episode$chol)==FALSE &
episode$chol >0,],
                                  var=NULL,
                                  date="last episode date",
                                  startvar="dob",
                                  th=ageth,
                                  hold=TRUE,
                                  varname="age")

    data2_2_3_3<-merge(episode[is.na(episode$chol)==FALSE & episode$chol >0,
                       c("patient_id","epi_date","chol","chol_c")],
                       data2_2_3_3,
                       by.x=c("patient_id","epi_date"),
                       by.y=c("patient_id","last_epi_date"),
                       all.y=TRUE)

    data2_2_3_3<-merge(patient[,c("patient_id","sex","type_dm","age_c")],
                       data2_2_3_3,
```

```
                    by.x=c("patient_id"),
                    by.y=c("patient_id"),
                    all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_2_3_3$chol_c))
    data2_2_3_3$chol_c<-ifelse(is.na(data2_2_3_3$chol_c),99,data2_2_3_3$chol_c)

    lev_var_cat=names(levchol)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_2_3_3<-merge(data2_2_3_3,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_2_3_3)[pmatch("sub_ds_id",names(data2_2_3_3))]<-"centre_id"

    BIRO_dframe(dataname="data2_2_3_3",monitor=TRUE,filelog="")


   if (thisyear==anayears[1]) {
    data2_2_3_3l<-data2_2_3_3
    data2_2_3_3l$year_epi=thisyear
   } else {
    if (exists("data2_2_3_3l")) {
     data2_2_3_3$year_epi=thisyear
     data2_2_3_3l<-rbind(data2_2_3_3l,data2_2_3_3)
    } else {
     data2_2_3_3l<-data2_2_3_3
     data2_2_3_3l$year_epi=thisyear
    }




    }
    printlines=0
    if (thisyear==anayears[length(anayears)]) {
      lines=1
      printlines=1
      if (length(anayears)==1) {
        printlines=0
      }
    }


       BIRO_report(ind="2_2_3_3",
                   title="Total cholesterol (the most recent episode in 12 months)",
                   var="chol",
                   var_cat="chol_c",
                   class=c("sex","age_c","type_dm"),
                   lev_var_cat=lev_var_cat,
                  lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
                   lab_var_cat="Cholesterol",
                   lab_class=c("Gender","Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=1,
                   bar=1,
                   lines=0,
                   printlines=printlines,
                   varlines="chol",
                   timelines="year_epi",
                   trellis=1,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )

    rm(data2_2_3_3)
    rm(patient)
    rm(episode)

    } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
      BIRO_report_toc(title="2.2.3.3. Total cholesterol (the most recent episode in
12 months)")
    }
   }
    rm(data2_2_3_3l)

      } else {
        for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
         BIRO_report_toc(title="2.2.3.3. Total cholesterol (the most recent episode
in 12 months)")
        }
      }

  #
  # 2.2.3.4 HDL-cholesterol (the most recent episode in 12 months)
  #

  if (varpat$type_dm==TRUE & varpat$dob==TRUE &
      varpat$sex==TRUE & varepi$hdl==TRUE) {


    for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)

    episode$hdl_c<-varclass(episode$hdl,hdlth)
    levhdl<-list("0 - 1.294"=1,"1.295 +"=2)
    if (dim(episode[is.na(episode$hdl)==FALSE & episode$hdl>0,])[1]>0) {

    data2_2_3_4<-BIRO_demographic(patient=patient,
                                  episode=episode[is.na(episode$hdl)==FALSE &
episode$hdl>0,],
                                  var=NULL,
                                  date="last episode date",
                                  startvar="dob",
                                  th=ageth,
                                  hold=TRUE,
                                  varname="age")

    data2_2_3_4<-merge(episode[is.na(episode$hdl)==FALSE & episode$hdl>0,
                         c("patient_id","epi_date","hdl","hdl_c")],
                       data2_2_3_4,
                       by.x=c("patient_id","epi_date"),
                       by.y=c("patient_id","last_epi_date"),
                       all.y=TRUE)

    data2_2_3_4<-merge(patient[,c("patient_id","sex","type_dm","age_c")],
                       data2_2_3_4,
                       by.x=c("patient_id"),
                       by.y=c("patient_id"),
                       all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_2_3_4$hdl_c))
    data2_2_3_4$hdl_c<-ifelse(is.na(data2_2_3_4$hdl_c),99,data2_2_3_4$hdl_c)

    lev_var_cat=names(levhdl)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_2_3_4<-merge(data2_2_3_4,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_2_3_4)[pmatch("sub_ds_id",names(data2_2_3_4))]<-"centre_id"

    BIRO_dframe(dataname="data2_2_3_4",monitor=TRUE,filelog="")

   if (thisyear==anayears[1]) {
    data2_2_3_4l<-data2_2_3_4
    data2_2_3_4l$year_epi=thisyear
   } else {
    if (exists("data2_2_3_4l")) {
     data2_2_3_4$year_epi=thisyear
     data2_2_3_4l<-rbind(data2_2_3_4l,data2_2_3_4)
    } else {
     data2_2_3_4l<-data2_2_3_4
     data2_2_3_4l$year_epi=thisyear
    }



    }

    printlines=0
    if (thisyear==anayears[length(anayears)]) {
      lines=1
      printlines=1
      if (length(anayears)==1) {
        printlines=0
      }
    }


       BIRO_report(ind="2_2_3_4",
                   title="HDL-cholesterol (the most recent episode in 12 months)",
                   var="hdl",
                   var_cat="hdl_c",
                   class=c("sex","age_c","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
                   lab_var_cat="HDL",
                   lab_class=c("Gender","Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=1,
                   bar=1,
                   lines=0,
                   printlines=printlines,
                   varlines="hdl",
                   timelines="year_epi",
                   trellis=1,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )


    rm(data2_2_3_4)
    rm(patient)
    rm(episode)

   } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="2.2.3.4. HDL-cholesterol (the most recent episode in 12
months)")
```

```
    }
   }
    rm(data2_2_3_4l)

  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="2.2.3.4. HDL-cholesterol (the most recent episode in 12
months)")
    }
  }

  #
  # 2.2.3.5 creatinine (the last episode, not only refyear)
  #

  if (varpat$type_dm==TRUE & varpat$dob==TRUE &
      varpat$sex==TRUE & varepi$creat==TRUE) {


    for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)

    episode$creat_c<-varclass(episode$creat,creatth)

    if (dim(episode[is.na(episode$creat)==FALSE & episode$creat>0,])[1]>0) {

    data2_2_3_5<-BIRO_demographic(patient=patient,
                                  episode=episode[is.na(episode$creat)==FALSE &
episode$creat>0,],
                                  var=NULL,
                                  date="last episode date",
                                  startvar="dob",
                                  th=ageth,
                                  hold=TRUE,
                                  varname="age")

    data2_2_3_5<-merge(episode[is.na(episode$creat)==FALSE & episode$creat>0,
                         c("patient_id","epi_date","creat","creat_c")],
                       data2_2_3_5,
                       by.x=c("patient_id","epi_date"),
                       by.y=c("patient_id","last_epi_date"),
                       all.y=TRUE)

    data2_2_3_5<-merge(patient[,c("patient_id","sex","type_dm","age_c")],
                       data2_2_3_5,
                       by.x=c("patient_id"),
                       by.y=c("patient_id"),
                       all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_2_3_5$creat_c))
    data2_2_3_5$creat_c<-ifelse(is.na(data2_2_3_5$creat_c),99,data2_2_3_5$creat_c)

    lev_var_cat=classlabel(creatth)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_2_3_5<-merge(data2_2_3_5,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_2_3_5)[pmatch("sub_ds_id",names(data2_2_3_5))]<-"centre_id"

    BIRO_dframe(dataname="data2_2_3_5",monitor=TRUE,filelog="")

   if (thisyear==anayears[1]) {
    data2_2_3_5l<-data2_2_3_5
    data2_2_3_5l$year_epi=thisyear
   } else {
    if (exists("data2_2_3_5l")) {
     data2_2_3_5$year_epi=thisyear
     data2_2_3_5l<-rbind(data2_2_3_5l,data2_2_3_5)
    } else {
    data2_2_3_5l<-data2_2_3_5
    data2_2_3_5l$year_epi=thisyear
    }

    }
    if (thisyear==anayears[length(anayears)]) printlines=1 else printlines=0
    printlines=0
    if (thisyear==anayears[length(anayears)]) {
      lines=1
      printlines=1
      if (length(anayears)==1) {
        printlines=0
      }
    }


       BIRO_report(ind="2_2_3_5",
                   title="Creatinine (the most recent episode in 12 months)",
                   var="creat",
                   var_cat="creat_c",
```

```
                  class=c("sex","age_c","type_dm"),
                  lev_var_cat=lev_var_cat,
                 lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
                  lab_var_cat="CREATININE",
                  lab_class=c("Gender","Age","Type of Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=1,
                  bar=1,
                  lines=0,
                  printlines=printlines,
                  varlines="creat",
                  timelines="year_epi",
                  trellis=1,
                  pie=0,
                  texfile=texfile,

                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex
                  )

    rm(data2_2_3_5)
    rm(patient)
    rm(episode)

    } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="2.2.3.5. Creatinine (the most recent episode in 12
months)")
    }
    }
    rm(data2_2_3_5l)

  } else {

    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="2.2.3.5. Creatinine (the most recent episode in 12
months)")
    }
  }

  #
  # 2.2.3.6 hba1c (the most recent episode in 12 months)
  #


  if (varpat$type_dm==TRUE & varpat$dob==TRUE &
      varpat$sex==TRUE & varepi$hba1c==TRUE) {

    for (thisyear in anayears) {
```

```
   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)

    episode$hba1c<-as.numeric(as.character(episode$hba1c))

    episode$hba1c_c<-varclass(episode$hba1c,hba1cth)
    levhba1cth<-list("0 - 5.9"=1,"6 - 7.9"=2,"8 +"=3)
    if (dim(episode[is.na(episode$hba1c)==FALSE & episode$hba1c>0,])[1]>0) {

    data2_2_3_6<-BIRO_demographic(patient=patient,
                                  episode=episode[is.na(episode$hba1c)==FALSE &
episode$hba1c>0,],
                                  var=NULL,
                                  date="last episode date",
                                  startvar="dob",
                                  th=ageth,
                                  hold=TRUE,
                                  varname="age")


    data2_2_3_6<-merge(episode[is.na(episode$hba1c)==FALSE & episode$hba1c>0,
                         c("patient_id","epi_date","hba1c","hba1c_c")],
                       data2_2_3_6,
                       by.x=c("patient_id","epi_date"),
                       by.y=c("patient_id","last_epi_date"),
                       all.y=FALSE,all.x=FALSE)

    data2_2_3_6<-merge(patient[,c("patient_id","sex","type_dm","age_c")],
                       data2_2_3_6,
                       by.x=c("patient_id"),
                       by.y=c("patient_id"),
```

```
                    all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_2_3_6$hba1c_c))
    data2_2_3_6$hba1c_c<-ifelse(is.na(data2_2_3_6$hba1c_c),99,data2_2_3_6$hba1c_c)

    lev_var_cat=names(levhba1cth)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_2_3_6<-merge(data2_2_3_6,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_2_3_6)[pmatch("sub_ds_id",names(data2_2_3_6))]<-"centre_id"

   if (thisyear==anayears[1]) {
    data2_2_3_6l<-data2_2_3_6
    data2_2_3_6l$year_epi=thisyear
   } else {
    if (exists("data2_2_3_6l")) {
     data2_2_3_6$year_epi=thisyear
     data2_2_3_6l<-rbind(data2_2_3_6l,data2_2_3_6)
    } else {
     data2_2_3_6l<-data2_2_3_6
     data2_2_3_6l$year_epi=thisyear
    }


    BIRO_dframe(dataname="data2_2_3_6",monitor=TRUE,filelog="")
   }

    printlines=0
    if (thisyear==anayears[length(anayears)]) {
      lines=1
      printlines=1
      if (length(anayears)==1) {
        printlines=0
      }
    }


       BIRO_report(ind="2_2_3_6",
                   title="HbA1c (the most recent episode in 12 months)",
                   var="hba1c",
                   var_cat="hba1c_c",
                   class=c("sex","age_c","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
                   lab_var_cat="HbA1c",
                   lab_class=c("Gender","Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=1,
                   bar=1,
                   lines=0,
                   printlines=printlines,
                   varlines="hba1c",
                   timelines="year_epi",
                   trellis=1,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )

    rm(data2_2_3_6)
    rm(patient)
    rm(episode)

    } else {
    assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="2.2.3.6. HbA1c (the most recent episode in 12 months)")
    }
    }
    rm(data2_2_3_6l)

  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="2.2.3.6. HbA1c (the most recent episode in 12 months)")
    }
  }

  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)

   new_section(title="2.3. Diabetes complications",texfile=texfile)
   new_section_html(title="2.3. Diabetes complications",htmlfile=htmlfile)
  }
  #
  # 2.3.1 retinopathy (the most recent episode in 12 months)
  #

  if ((varpat$dt_diag==TRUE) & (varepi$retina==TRUE)) {


  for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
```

```
   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

  if (dim(episode[episode$retina %in% c(1,2,3) & is.na(episode$retina)==FALSE
,])[1]>0) {



  data2_3_1<-BIRO_demographic(patient=patient,
                              episode=episode[episode$retina %in% c(1,2,3) & is
.na(episode$retina)==FALSE,],
                              var=NULL,
                              date="last episode date",
                              startvar="dt_diag",
                              th=durdiabth,
                              hold=TRUE,
                              varname="durdiab")

  data2_3_1<-merge(data2_3_1,
                   episode[episode$retina %in% c(1,2,3) & is
.na(episode$retina)==FALSE,],
                   by.x=c("patient_id","last_epi_date"),
                   by.y=c("patient_id","epi_date"),
                   all.x=TRUE,all.y=FALSE)

  data2_3_1<-merge(patient[,c("patient_id","diabetes_duration")],
                       data2_3_1,
                       by.x=c("patient_id"),
                       by.y=c("patient_id"),
                       all.x=TRUE)

  data2_3_1$durdiab_c<-varclass(data2_3_1$diabetes_duration,durdiabth)

  numlevelcat<-na.omit(unique(data2_3_1$retina))
  data2_3_1$retina<-ifelse(is.na(data2_3_1$retina),99,data2_3_1$retina)

  lev_var_cat=names(levretin)[sort(numlevelcat)]

  data2_3_1<-data2_3_1[,c("retina","durdiab_c")]


      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_3_1<-merge(data2_3_1,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_3_1)[pmatch("sub_ds_id",names(data2_3_1))]<-"centre_id"

      BIRO_report(ind="2_3_1",
                  title="Retinopathy (the most recent episode in 12 months)",
                  var_cat="retina",
                  class=c("durdiab_c"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=classlabellist(durdiabth)),
                  lab_var_cat="Retinopathy",
                  lab_class=c("Duration"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  bar=1,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex
                  )


rm(data2_3_1)
  rm(patient)
  rm(episode)

}
}
} else {for (thisyear in anayears) {
 assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

  BIRO_report_toc(title="2.3.1. Retinopathy (the most recent episode in 12 months)")
}
}

#
# 2.3.2 esrf
#

if (varpat$dt_diag==TRUE & varepi$esrf==TRUE) {

for (thisyear in anayears) {

 assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

 assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
       "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

 assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
       "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

 assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
```

```
   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    data2_3_2<-patient[,c("patient_id","diabetes_duration")]

    #patient with esrf

    esrf<-as.data.frame(unique(episode[is.na(episode$esrf)==FALSE & episode$esrf==1,
"patient_id"]))
    names(esrf)<-"patient_id"

    if (dim(esrf)[1]>0) {
     esrf$esrf<-1
    }

     esrf2<-as.data.frame(unique(episode[is.na(episode$esrf)==FALSE & episode$esrf==0,
"patient_id"]))
     names(esrf2)<-"patient_id"
     if (dim(esrf2)[1]>0) {
      esrf2$esrf<-2
     }

    esrf<-merge(esrf,esrf2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("esrf.x" %in% names(esrf)) {
     esrf$esrf<-ifelse(is.na(esrf$esrf.x),esrf$esrf.y,esrf$esrf.x)
    }
    data2_3_2<-merge(data2_3_2,esrf,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_3_2$esrf))

    data2_3_2$esrf<-ifelse(is.na(data2_3_2$esrf),99,data2_3_2$esrf)

    lev_var_cat=names(levesrf)[sort(numlevelcat)]

    data2_3_2$durdiab_c<-varclass(data2_3_2$diabetes_duration,durdiabth)


      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_3_2<-merge(data2_3_2,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_3_2)[pmatch("sub_ds_id",names(data2_3_2))]<-"centre_id"

  if ("esrf.x" %in% names(data2_3_2)) {
   data2_3_2<-data2_3_2[,-pmatch(c("esrf.x","esrf.y"),names(data2_3_2))]
  }

BIRO_dframe(dataname="data2_3_2",monitor=TRUE,filelog="")

     BIRO_report(ind="2_3_2",
                 title="End stage renal failure",
                 var_cat="esrf",
                 class=c("durdiab_c"),
                 lev_var_cat=lev_var_cat,
                 lev_class=list(a=classlabellist(durdiabth)),
                 lab_var_cat="E.S.R.F.",lab_class="Duration",
                 tab=1,
                 tabside="H",
                 tabperc=TRUE,
                 tabwidth=0.95,
                 barbeside=TRUE,
                 barnumber=TRUE,
                 barperc=TRUE,
                 box=0,
                 bar=1,
                 lines=0,
                 trellis=0,
                 pie=0,
                 texfile=texfile,

                 dircsv=dirdataout,
                 dirgraph=dirgraph,
                 dirgraphlatex=dirgraphlatex,
                 cex=cex
                 )

rm(data2_3_2)
rm(patient)
rm(episode)
rm(esrf)

}
} else {
  for (thisyear in anayears) {
 assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="2.3.2. End stage renal failure")
  }
 }


#
# 2.3.3 foot ulcer
#

if (((varpat$dt_diag)==TRUE) & (varepi$ulcer==TRUE)) {


for (thisyear in anayears) {

 assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

 assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
       "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

 assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
       "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
```

```
   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)



    data2_3_3<-patient[,c("patient_id","diabetes_duration")]

    #patient with foot ulcer

    ulcer<-as.data.frame(unique(episode[is.na(episode$ulcer)==FALSE &
episode$ulcer==1,"patient_id"]))
    names(ulcer)<-"patient_id"

    if (dim(ulcer)[1]>0) {
     ulcer$ulcer<-1
    }

     ulcer2<-as.data.frame(unique(episode[is.na(episode$ulcer)==FALSE &
episode$ulcer==0,"patient_id"]))
     names(ulcer2)<-"patient_id"
     if (dim(ulcer2)[1]>0) {
      ulcer2$ulcer<-2
     }

    ulcer<-merge(ulcer,ulcer2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("ulcer.x" %in% names(ulcer)) {
     ulcer$ulcer<-ifelse(is.na(ulcer$ulcer.x),ulcer$ulcer.y,ulcer$ulcer.x)
    }

    data2_3_3<-merge(data2_3_3,ulcer,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_3_3$ulcer))

    data2_3_3$ulcer<-ifelse(is.na(data2_3_3$ulcer),99,data2_3_3$ulcer)

    lev_var_cat=names(levulcer)[sort(numlevelcat)]

    data2_3_3$durdiab_c<-varclass(data2_3_3$diabetes_duration,durdiabth)

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_3_3<-merge(data2_3_3,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
names(data2_3_3)[pmatch("sub_ds_id",names(data2_3_3))]<-"centre_id"

if ("ulcer.x" %in% names(data2_3_3)) {
 data2_3_3<-data2_3_3[,-pmatch(c("ulcer.x","ulcer.y"),names(data2_3_3))]
}

BIRO_dframe(dataname="data2_3_3",monitor=TRUE,filelog="")

if (dim(episode[is.na(episode$ulcer)==FALSE,])[1]>0) {

     BIRO_report(ind="2_3_3",
                 title="Foot ulcer",
                 var_cat="ulcer",
                 class=c("durdiab_c"),
                 lev_var_cat=lev_var_cat,
                 lev_class=list(a=classlabellist(durdiabth)),
                 lab_var_cat="Foot Ulcer",lab_class="Duration",
                 tab=1,
                 tabside="H",
                 tabperc=TRUE,
                 tabwidth=0.95,
                 barbeside=TRUE,
                 barnumber=TRUE,
                 barperc=TRUE,
                 box=0,
                 bar=1,
                 lines=0,
                 trellis=0,
                 pie=0,
                 texfile=texfile,

                 dircsv=dirdataout,
                 dirgraph=dirgraph,
                 dirgraphlatex=dirgraphlatex,
                 cex=cex
                 )



rm(data2_3_3)
rm(ulcer)
  rm(patient)
  rm(episode)
}
}
} else {for (thisyear in anayears) {
 assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

  BIRO_report_toc(title="2.3.3. Foot ulcer")
  }
 }

#
# 2.3.4 amputation
#

if (varpat$dt_diag==TRUE & varepi$amput==TRUE) {

for (thisyear in anayears) {

 assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
```

```
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)


    data2_3_4<-patient[,c("patient_id","diabetes_duration")]


    amput<-as.data.frame(unique(episode[is.na(episode$amput)==FALSE &
episode$amput==1,"patient_id"]))
    names(amput)<-"patient_id"


    if (dim(amput)[1]>0) {
     amput$amput<-1
    }

     amput2<-as.data.frame(unique(episode[is.na(episode$amput)==FALSE &
episode$amput==0,"patient_id"]))
     names(amput2)<-"patient_id"
     if (dim(amput2)[1]>0) {
      amput2$amput<-2
     }

    amput<-merge(amput,amput2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("amput.x" %in% names(amput)) {
     amput$amput<-ifelse(is.na(amput$amput.x),amput$amput.y,amput$amput.x)
    }


    data2_3_4<-merge(data2_3_4,amput,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_3_4$amput))

    data2_3_4$amput<-ifelse(is.na(data2_3_4$amput),99,data2_3_4$amput)


    lev_var_cat=names(levamput)[sort(numlevelcat)]

    data2_3_4$durdiab_c<-varclass(data2_3_4$diabetes_duration,durdiabth)

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_3_4<-merge(data2_3_4,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_3_4)[pmatch("sub_ds_id",names(data2_3_4))]<-"centre_id"

  if ("amput.x" %in% names(data2_3_4)) {
   data2_3_4<-data2_3_4[,-pmatch(c("amput.x","amput.y"),names(data2_3_4))]
  }

  BIRO_dframe(dataname="data2_3_4",monitor=TRUE,filelog="")

  if (dim(episode[is.na(episode$amput)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,])[1]>0) {

      BIRO_report(ind="2_3_4",
                  title="Amputation",
                  var_cat="amput",
                  class=c("durdiab_c"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=classlabellist(durdiabth)),
                  lab_var_cat="Amputation",lab_class="Duration",
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  bar=1,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,

                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex
                  )



  rm(data2_3_4)
   rm(amput)
    rm(patient)
    rm(episode)

  }
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="2.3.4. Amputation")
    }
```

```
  }


  #
  # 2.3.5 stroke
  #

  if (varpat$dt_diag==TRUE & varepi$stroke==TRUE) {

  for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    data2_3_5<-patient[,c("patient_id","diabetes_duration")]


    stroke<-as.data.frame(unique(episode[is.na(episode$stroke)==FALSE &
episode$stroke==1,"patient_id"]))
    names(stroke)<-"patient_id"

    if (dim(stroke)[1]>0) {
     stroke$stroke<-1
    }

     stroke2<-as.data.frame(unique(episode[is.na(episode$stroke)==FALSE &
episode$stroke==0,"patient_id"]))
     names(stroke2)<-"patient_id"
     if (dim(stroke2)[1]>0) {
      stroke2$stroke<-2
     }

    stroke<-merge(stroke,stroke2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("stroke.x" %in% names(stroke)) {
     stroke$stroke<-ifelse(is.na(stroke$stroke.x),stroke$stroke.y,stroke$stroke.x)
    }


    data2_3_5<-merge(data2_3_5,stroke,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_3_5$stroke))

    data2_3_5$stroke<-ifelse(is.na(data2_3_5$stroke),99,data2_3_5$stroke)

    lev_var_cat=names(levstroke)[sort(numlevelcat)]

    data2_3_5$durdiab_c<-varclass(data2_3_5$diabetes_duration,durdiabth)

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_3_5<-merge(data2_3_5,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_3_5)[pmatch("sub_ds_id",names(data2_3_5))]<-"centre_id"

  if ("stroke.x" %in% names(data2_3_5)) {
   data2_3_5<-data2_3_5[,-pmatch(c("stroke.x","stroke.y"),names(data2_3_5))]
  }

  BIRO_dframe(dataname="data2_3_5",monitor=TRUE,filelog="")

       BIRO_report(ind="2_3_5",
                   title="Stroke",
                   var_cat="stroke",
                   class=c("durdiab_c"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(durdiabth)),
                   lab_var_cat="Stroke",
                   lab_class="Duration",
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )

  rm(data2_3_5)
   rm(stroke)
    rm(patient)
    rm(episode)
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
      BIRO_report_toc(title="2.3.5. Stroke")
    }
   }

  #
  # 2.3.6 myocardial infarctions
  #

  if (varpat$dt_diag==TRUE & varepi$mi==TRUE) {

  for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

  if (length(na.omit(episode$mi[episode$mi==1]))>0 & is.na(episode$epi_date)==FALSE) {

    data2_3_6<-patient[,c("patient_id","diabetes_duration")]


    mi<-as.data.frame(unique(episode[is.na(episode$mi)==FALSE & episode$mi==1,
"patient_id"]))
    names(mi)<-"patient_id"

    if (dim(mi)[1]>0) {
     mi$mi<-1
    }
     mi2<-as.data.frame(unique(episode[is.na(episode$mi)==FALSE & episode$mi==0,
"patient_id"]))
     names(mi2)<-"patient_id"
     if (dim(mi2)[1]>0) {
      mi2$mi<-2
     }

    mi<-merge(mi,mi2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("mi.x" %in% names(mi)) {
     mi$mi<-ifelse(is.na(mi$mi.x),mi$mi.y,mi$mi.x)
    }


    data2_3_6<-merge(data2_3_6,mi,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_3_6$mi))

    data2_3_6$mi<-ifelse(is.na(data2_3_6$mi),99,data2_3_6$mi)

    lev_var_cat=names(levmi)[sort(numlevelcat)]

    data2_3_6$durdiab_c<-varclass(data2_3_6$diabetes_duration,durdiabth)

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_3_6<-merge(data2_3_6,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_3_6)[pmatch("sub_ds_id",names(data2_3_6))]<-"centre_id"

  if ("mi.x" %in% names(data2_3_6)) {
   data2_3_6<-data2_3_6[,-pmatch(c("mi.x","mi.y"),names(data2_3_6))]
  }

  BIRO_dframe(dataname="data2_3_6",monitor=TRUE,filelog="")

       BIRO_report(ind="2_3_6",
                   title="Myocardal infarction",
                   var_cat="mi",
                   class=c("durdiab_c"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(durdiabth)),
                   lab_var_cat="Myocardial Infarction",
                   lab_class="Duration",
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )

  rm(data2_3_6)
   rm(mi)
    rm(patient)
    rm(episode)
  }
```

```
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="2.3.6. Myocardal infarction")
    }
   }

  #
  # 2.3.7 hypertension
  #

  if (varpat$dt_diag==TRUE & varepi$hypertension==TRUE) {

  for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    data2_3_7<-patient[,c("patient_id","diabetes_duration")]


    hypertension<-as.data.frame(unique(episode[is.na(episode$hypertension)==FALSE &
episode$hypertension==1,"patient_id"]))
    names(hypertension)<-"patient_id"

    if (dim(hypertension)[1]>0) {
     hypertension$hypertension<-1
    }
     hypertension2<-as.data.frame(unique(episode[is.na(episode$hypertension)==FALSE &
episode$hypertension==0,"patient_id"]))
     names(hypertension2)<-"patient_id"
     if (dim(hypertension2)[1]>0) {
      hypertension2$hypertension<-2
     }

    hypertension<-merge(hypertension,hypertension2,by="patient_id"
,all.x=TRUE,all.y=TRUE)
    if ("hypertension.x" %in% names(hypertension)) {
     hypertension$hypertension<-ifelse(is
.na(hypertension$hypertension.x),hypertension$hypertension.y,hypertension$hypertension
.x)
    }

    data2_3_7<-merge(data2_3_7,hypertension,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data2_3_7$hypertension))

    data2_3_7$hypertension<-ifelse(is
.na(data2_3_7$hypertension),99,data2_3_7$hypertension)

    lev_var_cat=names(levhypertension)[sort(numlevelcat)]

    data2_3_7$durdiab_c<-varclass(data2_3_7$diabetes_duration,durdiabth)

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data2_3_7<-merge(data2_3_7,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data2_3_7)[pmatch("sub_ds_id",names(data2_3_7))]<-"centre_id"

  if ("hypertension.x" %in% names(data2_3_7)) {
   data2_3_7<-data2_3_7[,-pmatch(c("hypertension.x","hypertension.y"
),names(data2_3_7))]
  }

  BIRO_dframe(dataname="data2_3_7",monitor=TRUE,filelog="")

       BIRO_report(ind="2_3_7",
                   title="Hypertension",
                   var_cat="hypertension",
                   class=c("durdiab_c"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(durdiabth),b=NA,c=NA),
                   lab_var_cat="Hypertension",
                   lab_class="Duration",
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )
```

```
rm(data2_3_7)
  rm(patient)
  rm(episode)


}
} else {
  for (thisyear in anayears) {
 assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="2.3.7. Hypertension")
  }
 }
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_se_indicator_demographic.r                                      #
#                                                                              #
#
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-24
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_se_indicator_clinical.r is part of WP Statistical Engine of the BIRO
# Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
################################################################################

print("demographic")
  #1.1 Age of patients (all the patient in the db)


  if (varpat$sex==TRUE & varpat$dob==TRUE) {

    for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirxml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   new_section(title="1.1 Basic demographics",texfile=texfile)
   new_section_html(title="1.1 Basic demographics",htmlfile=htmlfile)


    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    data1_1_1<-get(patyear)
    patient<-get(patyear)
    #data1_1_1$age<-(checkdate-patient$dob)/365.25
    data1_1_1<-as.data.frame(data1_1_1)
    data1_1_1$age_c<-varclass(data1_1_1$age,ageth)
    numlevelcat<-na.omit(unique(data1_1_1$age_c))
    data1_1_1$age_c<-ifelse(is.na(data1_1_1$age_c),99,data1_1_1$age_c)

    lev_var_cat=classlabel(ageth)[sort(numlevelcat)]

#    data1_1_1<-BIRO_demographic(patient=patient[patient],
#                                episode=episode,
#                                var=c("sex","age_c"),
#                                date=refdate,
#                                startvar="dob",
#                                th=ageth,
#                                hold=FALSE,
#                                varname="age")

    #data1.2 is the same of data1.1


      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data1_1_1<-merge(data1_1_1,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data1_1_1)[pmatch("sub_ds_id",names(data1_1_1))]<-"centre_id"

    BIRO_dframe(dataname="data1_1_1",monitor=TRUE,filelog="")

    BIRO_report(ind="1_1_1",
                title="Age (Classes)",
                var_cat="age_c",
                class=c("sex"),
                lev_var_cat=lev_var_cat,
                lev_class=list(levsex),
                lab_var_cat="Age",
                lab_class=c("Gender"),
                tab=1,
```

```
                tabside="H",
                tabperc=TRUE,
                tabwidth=0.95,
                barbeside=TRUE,
                barnumber=TRUE,
                barperc=TRUE,
                box=0,
                bar=1,
                lines=0,
                trellis=0,
                pie=0,
                texfile=texfile,
                dircsv=dirdataout,
                dirgraph=dirgraph,
                dirgraphlatex=dirgraphlatex,
                cex=cex,
                chisq="sex")
  rm(patient)
  rm(episode)
  rm(data1_1_1)
}
} else {
  for (thisyear in anayears) {
 assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="1.1. Age (Classes)")
  }
}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_se_indicator_health_system.r                                      #
#                                                                              #
#
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-24
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_se_indicator_health_system.r is part of WP Statistical Engine of the
# BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
################################################################################


  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   new_section(title="3.1. Structure (provider level)",texfile=texfile)
   new_section_html(title="3.1. Structure (provider level)",htmlfile=htmlfile)
  }

names<-c("GP"
,"Hospital Clinic (Internal Medicine)"
,"Hospital Clinic (Diabetes)"
,"Regional Shared-data Register"
,"Regional Primary Care Project"
,"Disease Management Programme"
,"Hospital Discharge Information"
,"Insurance Programme"
,"Retinal Screening Programme"
,"Diabetes Specialist Nurse Clinic"
,"National Data ? Complete"
,"National Data ? Sample"
,"Regional Data ? Sample")
codes<-1:13
legend_type<-data.frame(cbind(codes,names))

#
# 3.1.1 Type of Provider
#
  if (is.null(site$ds_type)==FALSE & dim(site)[1]>0) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   data3_1_1<-as.data.frame(cbind(as.character(centre_id),as.character(site$ds_type)))
   names(data3_1_1)<-c("centre_id","type")
   data3_1_1$centre_id<-as.character(data3_1_1$centre_id)
   data3_1_1csv<-data3_1_1
   data3_1_1csv$dbname=dbname
   data3_1_1csv$start=startdate
   data3_1_1csv$end=enddate

   write.csv(data3_1_1csv,paste(dirdataout,"/","i3_1_1d1_3a.csv",sep=""
),row.names=FALSE)
   rm(data3_1_1csv)
   data3_1_1<-merge(data3_1_1,legend_type,by.x="type",by.y="codes",all=FALSE)
   data3_1_1<-data3_1_1[,-1]
   names(data3_1_1)[2]<-'Type of Provider'


       BIRO_report(ind="3_1_1",
                   title="Type of Provider",
                   var=NULL,
```

```
                var_cat=NULL,
                lev_var_cat=NULL,
                lev_class=NULL,
                lab_var=NULL,
                lab_var_cat=NULL,
                lab_class=NULL,
                tab=0,
                box=0,
                bar=0,
                lines=0,
                trellis=0,
                pie=0,
                texfile=texfile,

                dircsv=dirdataout,
                dirgraph=dirgraph,
                dirgraphlatex=dirgraphlatex,
                cex=cex,
                stand=0)
 }
 } else {
   for (thisyear in anayears) {
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="3.1.1 Type of Provider")
   }
 }


if (exists("popdiab")==TRUE) {

 for (thisyear in anayears) {

  data3_1_2<-as.data.frame(popdiab[popdiab$year==thisyear,])
  data3_1_2$sex<-factor(data3_1_2$sex)
  data3_1_2$type_dm<-factor(as.numeric(as.character(data3_1_2$type_dm)))
  data3_1_2$ageband<-factor(data3_1_2$ageband)
  #levels(data3_1_2$sex)<-levsex
  #levels(data3_1_2$type_dm)<-levtype_dm
  #levels(data3_1_2$ageband)<-classlabellist(ageth5)
  data3_1_2$centre_id=centre_id
  names(data3_1_2)[pmatch("Freq",names(data3_1_2))]<-"n"


  write.csv(data3_1_2,paste(dirdataout,"/","i3_1_2d1_3a.csv",sep=""),row.names=FALSE)

  ind<-rep(1:dim(data3_1_2)[1], times=data3_1_2$n)
  data3_1_2<-data3_1_2[ind,-pmatch("n",names(data3_1_2))]

  data3_1_2<-data3_1_2[,pmatch(c("sex","ageband","type_dm","n"),names(data3_1_2))]

      BIRO_report(ind="3_1_2",
                title="Average diabetes population",
                var_cat="age_c",
                class=c("sex",NA,NA,"type_dm"),
                lev_var_cat=classlabel(ageth10),
                lev_class=list(a=levsex,b="",c="",d=levtype_dm),
                lab_var_cat="Age",
                lab_class=c("Gender","Type of Diabetes"),
                tab=1,
                tabside="H",
                tabperc=TRUE,
                tabwidth=0.95,
                barbeside=TRUE,
                barnumber=TRUE,
                barperc=TRUE,
                box=0,
                bar=0,
                lines=0,
                trellis=0,
                pie=0,
                texfile=texfile,
                dircsv=dirdataout,
                dirgraph=dirgraph,
                dirgraphlatex=dirgraphlatex,
                cex=cex
                )

 }
 } else {
   for (thisyear in anayears) {

  if (varpat$dob==TRUE & varpat$type_dm==TRUE &
      varpat$sex==TRUE) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    ageth10<-c(10,20,30,40,50,60,70,80)
    patient$age_c<-varclass(patient$age,ageth10)


   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
    data3_1_2<-patient[,c("patient_id","type_dm","sex","age_c")]
    #data3_1_2$sex<-as.factor(data3_1_2$sex)
    #data3_1_2$type_dm<-as.factor(data3_1_2$type_dm)
```

```
    #data3_1_2$age_c<-as.factor(data3_1_2$age_c)

    numlevelcat<-as.numeric(as.character(na.omit(unique(data3_1_2$age_c))))
    data3_1_2$age_c<-ifelse(is.na(data3_1_2$age_c),99,data3_1_2$age_c)

    lev_var_cat=classlabel(ageth10)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_1_2<-merge(data3_1_2,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_1_2)[pmatch("sub_ds_id",names(data3_1_2))]<-"centre_id"

       BIRO_report(ind="3_1_2",
                   title="Average diabetes population",
                   var_cat="age_c",
                   class=c("sex",NA,"type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=levsex,b=NA,c=levtype_dm),
                   lab_var_cat="Age",
                   lab_class=c("Gender","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )

 }
  }
  }

 # else BIRO_report_toc(title="3.1.2 Average diabetes population per center")

  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)

   new_section(title="3.2. Structural quality",texfile=texfile)
   new_section_html(title="3.2. Structural quality",htmlfile=htmlfile)
  }
#
# 3.2.1 Hospital beds per 100,000 population
#

if (dim(site)[1]>0 & exists("pop")) {
 if (is.null(site$ds_beds)==FALSE) {

 if (dim(pop)[1]>0) {

  for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   totpop<-sum(pop[pop$year==thisyear,"popM"])+sum(pop[pop$year==thisyear,"popF"])


data3_2_1<-as.data.frame(cbind(thisyear,site$ds_beds,totpop,round((site$ds_beds/totpop
)*100000,digits=2)))
   names(data3_2_1)<-c("year","beds","pop","rate")
   data3_2_1$centre_id=centre_id
   data3_2_1csv<-data3_2_1
   data3_2_1csv$dbname=dbname
   data3_2_1csv$start=startdate
   data3_2_1csv$end=enddate

   write.csv(data3_2_1csv,paste(dirdataout,"/","i3_2_1d1_3a.csv",sep=""
),row.names=FALSE)
   rm(data3_2_1csv)

   data3_2_1<-data3_2_1[,pmatch(c("year","beds","pop","rate"),names(data3_2_1))]

       BIRO_report(ind="3_2_1",
                   title="Hospital beds per 100,000 population",
                   var=NULL,
                   var_cat=NULL,
                   lev_var_cat=NULL,
                   lev_class=NULL,
                   lab_var=NULL,
                   lab_var_cat=NULL,
                   lab_class=NULL,
                   tab=0,
                   box=0,
                   bar=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
```

```
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  chisq=NULL)
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.2.1 Hospital beds per 100,000 population")
    }
   }
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="3.2.1 Hospital beds per 100,000 population")
   }
  }
} else {
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="3.2.1 Hospital beds per 100,000 population")
  }
 }

#
# 3.2.2 Physicians employed per 100,000 population
#
  if ((length(is.na(site$ds_physicians))>0) & exists("pop")) {

   for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   totpop<-sum(pop[pop$year==thisyear,"popM"])+sum(pop[pop$year==thisyear,"popF"])
data3_2_2<-as.data.frame(cbind(thisyear,centre_id,site$ds_physicians,totpop,round((sit
e$ds_physicians/totpop)*100000,digits=2)))
   names(data3_2_2)<-c("year","centre_id","ds_physicians","pop","rate")
   data3_2_2$centre_id=centre_id
   data3_2_2csv<-data3_2_2
   data3_2_2csv$dbname=dbname
   data3_2_2csv$start=startdate
   data3_2_2csv$end=enddate


   write.csv(data3_2_2,paste(dirdataout,"/","i3_2_2d1_3a.csv",sep=""),row.names=FALSE)
   rm(data3_2_2csv)
   data3_2_2<-data3_2_2[,pmatch(c("year","ds_physicians","pop","rate"
),names(data3_2_2))]

       BIRO_report(ind="3_2_2",
                  title="Physicians employed per 100,000 population",
                  var=NULL,
                  var_cat=NULL,
                  lev_var_cat=NULL,
                  lev_class=NULL,
                  lab_var=NULL,
                  lab_var_cat=NULL,
                  lab_class=NULL,
                  tab=0,
                  box=0,
                  bar=0,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex)

   }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.2.2 Physicians employed per 100,000 population")
    }
  }


  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)

   new_section_html(title="3.3 Processes (individual level)",htmlfile=htmlfile)
   new_section(title="3.3 Processes (individual level)",texfile=texfile)

  }
#
# 3.3.1.1 Foot examination
#
```

```
  if (varpat$dob==TRUE & varpat$type_dm==TRUE &
      varepi$foot_exam==TRUE) {


   for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   new_sub_section_html(title="3.3.1 Foot examination",htmlfile=htmlfile)
   new_sub_section(title="3.3.1 Foot examination",texfile=texfile)

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)
    patient$age_c<-varclass(patient$age,ageth)


    if (dim(episode[is.na(episode$foot_exam)==FALSE,])[1]>0) {

    data3_3_1_1<-patient[,c("patient_id","age_c","type_dm")]

    foot_exam<-as.data.frame(unique(episode[is.na(episode$foot_exam)==FALSE &
episode$foot_exam==1,"patient_id"])) #patient with
    names(foot_exam)<-"patient_id"
    if (dim(foot_exam)[1]>0) {
     foot_exam$foot_exam<-1
    }

     foot_exam2<-as.data.frame(unique(episode[is.na(episode$foot_exam)==FALSE &
episode$foot_exam==0,"patient_id"]))
     names(foot_exam2)<-"patient_id"
     if (dim(foot_exam2)[1]>0) {
      foot_exam2$foot_exam<-2
     }

    foot_exam<-merge(foot_exam,foot_exam2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("foot_exam.x" %in% names(foot_exam)) {
     foot_exam$foot_exam<-ifelse(is
.na(foot_exam$foot_exam.x),foot_exam$foot_exam.y,foot_exam$foot_exam.x)
    }

    data3_3_1_1<-merge(data3_3_1_1,foot_exam,by="patient_id",all.x=TRUE,all.y=FALSE)
#Only people with foot examination

    numlevelcat<-na.omit(unique(data3_3_1_1$foot_exam))

    data3_3_1_1$foot_exam<-ifelse(is
.na(data3_3_1_1$foot_exam),99,data3_3_1_1$foot_exam)

    lev_var_cat=names(levfoot_done)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_1_1<-merge(data3_3_1_1,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_3_1_1)[pmatch("sub_ds_id",names(data3_3_1_1))]<-"centre_id"

  if ("foot_exam.x" %in% names(data3_3_1_1)) {
   data3_3_1_1<-data3_3_1_1[,-pmatch(c("foot_exam.x","foot_exam.y"
),names(data3_3_1_1))]
  }

       BIRO_report(ind="3_3_1_1",
                   title="Foot Examination Done",
                   var_cat="foot_exam",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Foot Examination",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   what="examinations",
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex)

  rm(data3_3_1_1)
    rm(patient)
    rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
      BIRO_report_toc(title="3.3.1.1 Foot Examination Done")
  }
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.1.1 Foot Examination Done")
    }
  }


 # 3.3.2.1 Eye examination


   if (varpat$dob==TRUE & varpat$type_dm==TRUE &
       varepi$retinal_exam==TRUE) {


   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)


   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   new_sub_section_html(title="3.3.2 Eye examination",htmlfile=htmlfile)
   new_sub_section(title="3.3.2 Eye examination",texfile=texfile)
```

```
   if (dim(episode[is.na(episode$retinal_exam)==FALSE,])[1]>0) {

    data3_3_2_1<-patient[,c("patient_id","age_c","type_dm")]

    eye_exam<-as.data.frame(unique(episode[is.na(episode$retinal_exam)==FALSE &
episode$retinal_exam==1,"patient_id"])) #patient with
    names(eye_exam)<-"patient_id"
    if (dim(eye_exam)[1]>0) {
     eye_exam$eye_exam<-1
    }

     eye_exam2<-as.data.frame(unique(episode[is.na(episode$retinal_exam)==FALSE &
episode$retinal_exam==0,"patient_id"]))
     names(eye_exam2)<-"patient_id"
     if (dim(eye_exam2)[1]>0) {
      eye_exam2$eye_exam<-2
     }

    eye_exam<-merge(eye_exam,eye_exam2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("eye_exam.x" %in% names(eye_exam)) {
     eye_exam$eye_exam<-ifelse(is
.na(eye_exam$eye_exam.x),eye_exam$eye_exam.y,eye_exam$eye_exam.x)
    }

    data3_3_2_1<-merge(data3_3_2_1,eye_exam,by="patient_id",all.x=TRUE,all.y=FALSE)

    numlevelcat<-na.omit(unique(data3_3_2_1$eye_exam))
    data3_3_2_1$eye_exam<-ifelse(is.na(data3_3_2_1$eye_exam),99,data3_3_2_1$eye_exam)

    lev_var_cat=names(leveye_done)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_2_1<-merge(data3_3_2_1,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_3_2_1)[pmatch("sub_ds_id",names(data3_3_2_1))]<-"centre_id"

  if ("eye_exam.x" %in% names(data3_3_2_1)) {
   data3_3_2_1<-data3_3_2_1[,-pmatch(c("eye_exam.x","eye_exam.y"),names(data3_3_2_1))]
  }

       BIRO_report(ind="3_3_2_1",
                   title="Eye Examination Done",
                   var_cat="eye_exam",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Eye Examination",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   what="examinations",
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
```

```
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex)

  rm(data3_3_2_1)
    rm(patient)
    rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.2.1 Done")
  }
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.2.1 Done (last episode in 12 months)")
    }
  }

# Measurements done

   for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)

   new_sub_section_html(title="3.3.3 Measurements done",htmlfile=htmlfile)
   new_sub_section(title="3.3.3 Measurements done",texfile=texfile)

  }
  # 3.3.3.1 BP

if ((varpat$dob==TRUE & varpat$type_dm==TRUE &
     varepi$sbp==TRUE) |
    (varpat$dob==TRUE & varpat$type_dm==TRUE &
     varepi$dbp==TRUE)) {
```

```
   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)
    patient$age_c<-varclass(patient$age,ageth)


     episode$bp<-ifelse((is.na(episode$sbp)==FALSE & episode$sbp>0)|
                        (is.na(episode$dbp)==FALSE & episode$dbp>0),1,2)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   if (dim(episode[is.na(episode$bp)==FALSE,])[1]>0) {

    data3_3_3_1<-patient[,c("patient_id","age_c","type_dm")]

    bp<-as.data.frame(unique(episode[is.na(episode$bp)==FALSE & episode$bp==1,
"patient_id"]))
    names(bp)<-"patient_id"
    if (dim(bp)[1]>0) {
     bp$bp<-1
    }

     bp2<-as.data.frame(unique(episode[is.na(episode$bp)==FALSE & episode$bp==2,
"patient_id"]))
     names(bp2)<-"patient_id"
     if (dim(bp2)[1]>0) {
      bp2$bp<-2
     }
```

```
    bp<-merge(bp,bp2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("bp.x" %in% names(bp)) {
     bp$bp<-ifelse(is.na(bp$bp.x),bp$bp.y,bp$bp.x)
    }

    data3_3_3_1<-merge(data3_3_3_1,bp,by="patient_id",all.x=TRUE,all.y=FALSE)

    numlevelcat<-na.omit(unique(data3_3_3_1$bp))

    data3_3_3_1$bp<-ifelse(is.na(data3_3_3_1$bp)==TRUE,99,data3_3_3_1$bp)

    lev_var_cat=names(levbp_done)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_3_1<-merge(data3_3_3_1,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_3_3_1)[pmatch("sub_ds_id",names(data3_3_3_1))]<-"centre_id"

  if ("bp.x" %in% names(data3_3_3_1)) {
   data3_3_3_1<-data3_3_3_1[,-pmatch(c("bp.x","bp.y"),names(data3_3_3_1))]
  }
       BIRO_report(ind="3_3_3_1",
                   title="BP (at least one measurement in 12 months)",
                   var_cat="bp",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="BP measurements",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex)

  rm(data3_3_3_1)
    rm(patient)
    rm(episode)

   } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.3.1 BP (at least one measurement in 12 months)")
   }
     }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.3.1 BP (at least one measurement in 12 months)")
    }
  }

  #3.3.3.2 Lipids

  if ((varpat$dob==TRUE & varpat$type_dm==TRUE &
       varepi$chol==TRUE) |
     (varpat$dob==TRUE & varpat$type_dm==TRUE &
      varepi$hdl==TRUE)) {


#  if (length(na.omit(episode$lipids))>0) {
   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)
    patient$age_c<-varclass(patient$age,ageth)


    episode$lipids<-ifelse((is.na(episode$chol)==FALSE & episode$chol>0)|
                           (is.na(episode$hdl)==FALSE  & episode$hdl>0) ,1,2)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   if (dim(episode[is.na(episode$lipids)==FALSE,])[1]>0) {

    data3_3_3_2<-patient[,c("patient_id","age_c","type_dm")]

    lipids<-as.data.frame(unique(episode[is.na(episode$lipids)==FALSE &
episode$lipids==1,"patient_id"]))
    names(lipids)<-"patient_id"
    if (dim(lipids)[1]>0) {
```

```
     lipids$lipids<-1
    }

     lipids2<-as.data.frame(unique(episode[is.na(episode$lipids)==FALSE &
episode$lipids==2,"patient_id"]))
     names(lipids2)<-"patient_id"
     if (dim(lipids2)[1]>0) {
      lipids2$lipids<-2
     }

    lipids<-merge(lipids,lipids2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("lipids.x" %in% names(lipids)) {
     lipids$lipids<-ifelse(is.na(lipids$lipids.x),lipids$lipids.y,lipids$lipids.x)
    }


    data3_3_3_2<-merge(data3_3_3_2,lipids,by="patient_id",all.x=TRUE,all.y=FALSE)

    numlevelcat<-na.omit(unique(data3_3_3_2$lipids))
    data3_3_3_2$lipids<-ifelse(is.na(data3_3_3_2$lipids),99,data3_3_3_2$lipids)

    lev_var_cat=names(levlipids)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_3_2<-merge(data3_3_3_2,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_3_3_2)[pmatch("sub_ds_id",names(data3_3_3_2))]<-"centre_id"

  if ("lipids.x" %in% names(data3_3_3_2)) {
   data3_3_3_2<-data3_3_3_2[,-pmatch(c("lipids.x","lipids.y"),names(data3_3_3_2))]
  }

       BIRO_report(ind="3_3_3_2",
                   title="Lipids (at least one measurement in 12 months)",
                   var_cat="lipids",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),NA,c=levtype_dm),
                   lab_var_cat="Lipids",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex)

  rm(data3_3_3_2)
    rm(patient)
    rm(episode)
   } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.3.2 Lipids (at least one measurement in 12 months)")
   }
   }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.3.2 Lipids (at least one measurement in 12 months)")
    }
   }

  #3.3.3.3 MA_test


 if (varpat$dob==TRUE & varpat$type_dm==TRUE &
     varepi$ma_test==TRUE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)
    patient$age_c<-varclass(patient$age,ageth)


   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   if (dim(episode[is.na(episode$ma_test)==FALSE,])[1]>0) {

    data3_3_3_3<-patient[,c("patient_id","age_c","type_dm")]
```

```
    ma_test<-as.data.frame(unique(episode[is.na(episode$ma_test)==FALSE &
episode$ma_test %in% c(1,2),"patient_id"])) #All ma test
    names(ma_test)<-"patient_id"
    ma_test$ma_test<-1

    data3_3_3_3<-merge(data3_3_3_3,ma_test,by="patient_id",all.x=TRUE,all.y=FALSE)
    numlevelcat<-na.omit(unique(data3_3_3_3$ma_test))
    data3_3_3_3$ma_test<-ifelse(is
.na(data3_3_3_3$ma_test)==TRUE,99,data3_3_3_3$ma_test)

    lev_var_cat=names(levma_testdone)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_3_3<-merge(data3_3_3_3,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_3_3_3)[pmatch("sub_ds_id",names(data3_3_3_3))]<-"centre_id"

       BIRO_report(ind="3_3_3_3",
                   title="Microalbumin (at least one measurement in 12 months)",
                   var_cat="ma_test",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Microalbumin",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex)


    rm(data3_3_3_3)
    rm(patient)
    rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.3.3 Microalbumin (at least one measurement in 12
months)")
  }
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.3.3 Microalbumin (at least one measurement in 12
months)")
    }
   }

 #
 # 3.3.3.4 Hba1c
 #
 if (varpat$dob==TRUE & varpat$type_dm==TRUE &
     varepi$hba1c==TRUE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   if (dim(episode[is.na(episode$hba1c)==FALSE & episode$hba1c>0,])[1]>0) {

  data3_3_3_4<-patient[,c("patient_id","age_c","type_dm")]


    hba1c_done<-as.data.frame(unique(episode[is.na(episode$hba1c)==FALSE &
episode$hba1c>0,"patient_id"])) #All ma test
    names(hba1c_done)<-"patient_id"
    hba1c_done$hba1c_done<-1

    data3_3_3_4<-merge(data3_3_3_4,hba1c_done,by="patient_id",all.x=TRUE,all.y=FALSE)

    data3_3_3_4$hba1c_done<-ifelse(is
.na(data3_3_3_4$hba1c_done)==TRUE,2,data3_3_3_4$hba1c_done)
```

```
      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_3_4<-merge(data3_3_3_4,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_3_3_4)[pmatch("sub_ds_id",names(data3_3_3_4))]<-"centre_id"

       BIRO_report(ind="3_3_3_4",
                   title="HbA1c (at least one measurement in 12 months)",
                   var_cat="hba1c_done",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=names(levhba1c_done),
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="HbA1c done",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex)

  rm(data3_3_3_4)
    rm(patient)
    rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.3.4 HbA1c (at least one measurement in 12 months)")
  }
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.3.4 HbA1c (at least one measurement in 12 months)")
    }
   }

 ##Treatment


  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   new_sub_section_html(title="3.3.4 Treatment",htmlfile=htmlfile)
   new_sub_section(title="3.3.4 Treatment",texfile=texfile)
  }
 #
 # 3.3.4.1 Antihypertensive Medication
 #


 if (varpat$dob==TRUE & varpat$type_dm==TRUE &
     varepi$hypert_med==TRUE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   if (dim(episode[is.na(episode$hypert_med)==FALSE,])[1]>0) {

    data3_3_4_1<-patient[,c("patient_id","age_c","type_dm")]

    hypert_med<-as.data.frame(unique(episode[is.na(episode$hypert_med)==FALSE &
episode$hypert_med==1,"patient_id"]))
    names(hypert_med)<-"patient_id"
    if (dim(hypert_med)[1]>0) {
     hypert_med$hypert_med<-1
    }
     hypert_med2<-as.data.frame(unique(episode[is.na(episode$hypert_med)==FALSE &
episode$hypert_med==0,"patient_id"]))
     names(hypert_med2)<-"patient_id"
     if (dim(hypert_med2)[1]>0) {
```

```
      hypert_med2$hypert_med<-2
     }

    hypert_med<-merge(hypert_med,hypert_med2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("hypert_med.x" %in% names(hypert_med)) {
     hypert_med$hypert_med<-ifelse(is
.na(hypert_med$hypert_med.x),hypert_med$hypert_med.y,hypert_med$hypert_med.x)
    }


    data3_3_4_1<-merge(data3_3_4_1,hypert_med,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data3_3_4_1$hypert_med))
    data3_3_4_1$hypert_med<-ifelse(is
.na(data3_3_4_1$hypert_med),99,data3_3_4_1$hypert_med)

    lev_var_cat=names(levhypert_med)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_4_1<-merge(data3_3_4_1,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_3_4_1)[pmatch("sub_ds_id",names(data3_3_4_1))]<-"centre_id"

  if ("hypert_med.x" %in% names(data3_3_4_1)) {
   data3_3_4_1<-data3_3_4_1[,-pmatch(c("hypert_med.x","hypert_med.y"
),names(data3_3_4_1))]
  }

       BIRO_report(ind="3_3_4_1",
                   title="Antihypertensive Medication (at least one medication in 12
months)",
                   var_cat="hypert_med",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Antihypertensive Medication",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex)


  rm(data3_3_4_1)
    rm(patient)
    rm(episode)

  } else {
    assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

       BIRO_report_toc(title="3.3.4.1 Antihypertensive Medication (at least one
medication in 12 months)")
   }
   }
   } else {
     for (thisyear in anayears) {
    assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

       BIRO_report_toc(title="3.3.4.1 Antihypertensive Medication (at least one
medication in 12 months)")
    }
   }

 #
 # 3.3.4.2 Lipid Lowering Treatment
 #

 if (varpat$dob==TRUE & varpat$type_dm==TRUE &
     varepi$lipid_therapy==TRUE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)


   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
```

```
   if (dim(episode[is.na(episode$lipid_med)==FALSE,])>0) {

    data3_3_4_2<-patient[,c("patient_id","age_c","type_dm")]

    lipid_med<-as.data.frame(unique(episode[is.na(episode$lipid_med)==FALSE &
episode$lipid_therapy==1,"patient_id"]))
    names(lipid_med)<-"patient_id"

    if (dim(lipid_med)[1]>0) {
     lipid_med$lipid_med<-1
    }
     hypert_med2<-as.data.frame(unique(episode[is.na(episode$lipid_med)==FALSE &
episode$lipid_therapy==0,"patient_id"]))
     names(lipid_med2)<-"patient_id"
     if (dim(lipid_med2)[1]>0) {
      lipid_med2$lipid_med<-2
     }

    lipid_med<-merge(lipid_med,lipid_med2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("lipid_med.x" %in% names(lipid_med)) {
     lipid_med$lipid_med<-ifelse(is
.na(lipid_med$lipid_med.x),lipid_med$lipid_med.y,lipid_med$lipid_med.x)
    }

    data3_3_4_2<-merge(data3_3_4_2,lipid_med,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data3_3_4_2$lipid_med))
    data3_3_4_2$lipid_med<-ifelse(is
.na(data3_3_4_2$lipid_med),99,data3_3_4_2$lipid_med)

    lev_var_cat=names(levlipid_med)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_4_2<-merge(data3_3_4_2,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_3_4_2)[pmatch("sub_ds_id",names(data3_3_4_2))]<-"centre_id"

  if ("lipids_med.x" %in% names(data3_3_4_2)) {
   data3_3_4_2<-data3_3_4_2[,-pmatch(c("lipids_med.x","lipids_med.y"
),names(data3_3_4_2))]
  }

       BIRO_report(ind="3_3_4_2",
                   title="Lipid Lowering Treatment (at least one medication in 12
months)",
                   var_cat="lipid_med",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Lipid Lowering Medication",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )


  rm(data3_3_4_2)
    rm(patient)
    rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.4.2 Lipid Lowering Treatment (at least one
medication in 12 months)")
  }
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.4.2 Lipid Lowering Treatment (at least one
medication in 12 months)")
    }
   }
 #
 # 3.3.4.3 ASA Treatment
 #



 if (varpat$dob==TRUE & varpat$type_dm==TRUE &
     varepi$antiplatelet_therapy==TRUE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)


   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
```

```
   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   if (dim(episode[is.na(episode$asa_med)==FALSE,])[1]>0) {

    data3_3_4_3<-patient[,c("patient_id","age_c","type_dm")]

    asa_med<-as.data.frame(unique(episode[is.na(episode$asa_med)==FALSE &
episode$antiplatelet_therapy==1,"patient_id"]))
    names(asa_med)<-"patient_id"

    if (dim(asa_med)[1]>0) {
     asa_med$asa_med<-1
    }

     asa_med2<-as.data.frame(unique(episode[is.na(episode$asa_med)==FALSE &
episode$antiplatelet_therapy==0,"patient_id"]))
     names(asa_med2)<-"patient_id"
     if (dim(asa_med2)[1]>0) {
      asa_med2$asa_med<-2
     }

    asa_med<-merge(asa_med,asa_med2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("asa_med.x" %in% names(asa_med)) {
     asa_med$asa_med<-ifelse(is
.na(asa_med$asa_med.x),asa_med$asa_med.y,asa_med$asa_med.x)
    }

    data3_3_4_3<-merge(data3_3_4_3,asa_med,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data3_3_4_3$asa_med))
    data3_3_4_3$asa_med<-ifelse(is.na(data3_3_4_3$asa_med),99,data3_3_4_3$asa_med)

    lev_var_cat=names(levasa_med)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_4_3<-merge(data3_3_4_3,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_3_4_3)[pmatch("sub_ds_id",names(data3_3_4_3))]<-"centre_id"

  if ("asa_med.x" %in% names(data3_3_4_3)) {
   data3_3_4_3<-data3_3_4_3[,-pmatch(c("asa_med.x","asa_med.y"),names(data3_3_4_3))]
  }

       BIRO_report(ind="3_3_4_3",
                   title="ASA Medication (at least one medication in 12 months)",
                   var_cat="asa_med",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="ASA Medication",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )


  rm(data3_3_4_3)
    rm(patient)
    rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.4.3 ASA Medication (at least one medication in 12
months)")
  }
  }
  } else {
    for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

      BIRO_report_toc(title="3.3.4.3 ASA Medication (at least one medication in 12
months)")
    }
   }

   new_sub_section_html(title="3.3.4.4 Glucose Lowering Treatment",htmlfile=htmlfile)
   new_sub_section(title="3.3.4.4 Glucose Lowering Treatment",texfile=texfile)

  #
  # 3.3.4.4 Glucose Lowering Treatment (only epirefyear)
  #
  if ((varpat$dob==TRUE) & (varpat$type_dm==TRUE) &
      (varepi$drug_therapy==TRUE)) {

#  episode$drug_therapy=as.numeric(as.character(episode$drug_therapy))

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
```

```
partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)


   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
  if (dim(episode[episode$drug_therapy==4 & is
.na(episode$drug_therapy)==FALSE,])[1]>0) {

    data3_3_4_4_1<-patient[,c("patient_id","age_c","type_dm")]
    diet<-as.data.frame(unique(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy==4,"patient_id"]))
    names(diet)<-"patient_id"

    if (dim(diet)[1]>0) {
     diet$diet<-1
    }
    if (dim(diet)[1]==0) {
     diet<-as.data.frame(unique(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy!=4,"patient_id"]))
     names(diet)<-"patient_id"
     if (dim(diet)[1]>0) {
      diet$diet<-2
     }
    }

    data3_3_4_4_1<-merge(data3_3_4_4_1,diet,by="patient_id",all.x=TRUE,all.y=FALSE)

    data3_3_4_4_1$diet<-ifelse(is.na(data3_3_4_4_1$diet),2,data3_3_4_4_1$diet)

    numlevelcat<-na.omit(unique(data3_3_4_4_1$diet))

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_4_4_1<-merge(data3_3_4_4_1,patient[c("patient_id","sub_ds_id")],by=
"patient_id",all.x=TRUE,all.y=FALSE)
  names(data3_3_4_4_1)[pmatch("sub_ds_id",names(data3_3_4_4_1))]<-"centre_id"

      BIRO_report(ind="3_3_4_4_1",
                  title="Glucose Lowering: Diet Only ",
                  var_cat="diet",
                  class=c("age_c",NA,"type_dm"),
                  lev_var_cat=names(levdiet)[sort(numlevelcat)],
                  lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                  lab_var_cat="Diet Only",
                  lab_class=c("Age","Type of Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  bar=1,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,

                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex
                  )


 rm(data3_3_4_4_1)
 } else {
      assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   BIRO_report_toc(title="3.3.4.4.1 Glucose Lowering: Diet Only ")
 }

  if (dim(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy==2,])[1]>0) {

    data3_3_4_4_2<-patient[,c("patient_id","age_c","type_dm")]
    tab<-as.data.frame(unique(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy==2,"patient_id"]))
    names(tab)<-"patient_id"

   if (dim(tab)[1]>0) {
     tab$tab<-1
    }
    if (dim(tab)[1]==0) {
     tab<-as.data.frame(unique(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy!=2,"patient_id"]))
     names(tab)<-"patient_id"
     if (dim(tab)[1]>0) {
      tab$tab<-2
     }
    }



    data3_3_4_4_2<-merge(data3_3_4_4_2,tab,by="patient_id",all.x=TRUE)
```

```
    data3_3_4_4_2$tab<-ifelse(is.na(data3_3_4_4_2$tab),2,data3_3_4_4_2$tab)

    numlevelcat<-na.omit(unique(data3_3_4_4_2$tab))

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_4_4_2<-merge(data3_3_4_4_2,patient[c("patient_id","sub_ds_id")],by=
"patient_id",all.x=TRUE,all.y=FALSE)
  names(data3_3_4_4_2)[pmatch("sub_ds_id",names(data3_3_4_4_2))]<-"centre_id"

       BIRO_report(ind="3_3_4_4_2",
                   title="Glucose Lowering: Tablets Only ",
                   var_cat="tab",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=names(levtablet)[sort(numlevelcat)],
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Tablets Only",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )

 rm(data3_3_4_4_2)
  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="3.3.4.4.2 Glucose Lowering: Tablets Only ")
  }
  if (dim(episode[episode$drug_therapy==1 & is
.na(episode$drug_therapy)==FALSE,])[1]>0) {

    data3_3_4_4_3<-patient[,c("patient_id","age_c","type_dm")]
    ins<-as.data.frame(unique(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy==1,"patient_id"]))
    names(ins)<-"patient_id"

    if (dim(ins)[1]>0) {
     ins$ins<-1
    }
    if (dim(ins)[1]==0) {
     ins<-as.data.frame(unique(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy!=1,"patient_id"]))
     names(ins)<-"patient_id"
     if (dim(ins)[1]>0) {
      ins$ins<-2
     }
    }

    data3_3_4_4_3<-merge(data3_3_4_4_3,ins,by="patient_id",all.x=TRUE)

    data3_3_4_4_3$ins<-ifelse(is.na(data3_3_4_4_3$ins),2,data3_3_4_4_3$ins)

    numlevelcat<-na.omit(unique(data3_3_4_4_3$ins))

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_4_4_3<-merge(data3_3_4_4_3,patient[c("patient_id","sub_ds_id")],by=
"patient_id",all.x=TRUE,all.y=FALSE)
  names(data3_3_4_4_3)[pmatch("sub_ds_id",names(data3_3_4_4_3))]<-"centre_id"

       BIRO_report(ind="3_3_4_4_3",
                   title="Glucose Lowering: Insulin Only ",
                   var_cat="ins",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=names(levinsulin)[sort(numlevelcat)],
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Insulin Only",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )

 rm(data3_3_4_4_3)
  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="3.3.4.4.3 Glucose Lowering: Insulin Only ")
  }

  if (dim(episode[episode$drug_therapy==3 &is
.na(episode$drug_therapy)==FALSE,])[1]>0) {

    data3_3_4_4_4<-patient[,c("patient_id","age_c","type_dm")]
    instab<-as.data.frame(unique(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy==3,"patient_id"]))
    names(instab)<-"patient_id"

    if (dim(instab)[1]>0) {
     instab$instab<-1
    }
    if (dim(instab)[1]==0) {
```

```
     instab<-as.data.frame(unique(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy!=3,"patient_id"]))
     names(instab)<-"patient_id"
     if (dim(instab)[1]>0) {
      instab$instab<-2
     }
    }

    data3_3_4_4_4<-merge(data3_3_4_4_4,instab,by="patient_id",all.x=TRUE)

    data3_3_4_4_4$instab<-ifelse(is.na(data3_3_4_4_4$instab),2,data3_3_4_4_4$instab)

    numlevelcat<-na.omit(unique(data3_3_4_4_4$instab))

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_4_4_4<-merge(data3_3_4_4_4,patient[c("patient_id","sub_ds_id")],by=
"patient_id",all.x=TRUE,all.y=FALSE)
  names(data3_3_4_4_4)[pmatch("sub_ds_id",names(data3_3_4_4_4))]<-"centre_id"

       BIRO_report(ind="3_3_4_4_4",
                   title="Glucose Lowering: Insulin and Tablets ",
                   var_cat="instab",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=names(levinstab)[sort(numlevelcat)],
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Insulin And Tablets",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )

 rm(data3_3_4_4_4)
} else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

  BIRO_report_toc(title="3.3.4.4.4 Glucose Lowering: Insulin and Tablets ")
}
}
} else {
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   BIRO_report_toc(title="3.3.4.4.1 Glucose Lowering: Diet Only ")
   BIRO_report_toc(title="3.3.4.4.2 Glucose Lowering: Tablets Only ")
   BIRO_report_toc(title="3.3.4.4.3 Glucose Lowering: Insulin Only ")
   BIRO_report_toc(title="3.3.4.4.4 Glucose Lowering: Insulin and Tablets ")
  }

 }

  if (varpat$dob==TRUE & varpat$type_dm==TRUE &
      varepi$pump_therapy==TRUE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)


   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
  episode$pump_therapy<-as.numeric(as.character(episode$pump_therapy))

  if (dim(episode[is.na(episode$pump_therapy)==FALSE,])[1]>0) {

    data3_3_4_4_5<-patient[,c("patient_id","age_c","type_dm")]
    pump_therapy<-as.data.frame(unique(episode[is.na(episode$pump_therapy)==FALSE &
episode$pump_therapy==1,"patient_id"]))
    names(pump_therapy)<-"patient_id"

    if (dim(pump_therapy)[1]>0) {
     pump_therapy$pump_therapy<-1
    }

     pump_therapy2<-as.data.frame(unique(episode[is.na(episode$pump_therapy)==FALSE &
episode$pump_therapy==0,"patient_id"]))
```

```
     names(pump_therapy2)<-"patient_id"
     if (dim(pump_therapy2)[1]>0) {
      pump_therapy2$pump_therapy<-2
     }

    pump_therapy<-merge(pump_therapy,pump_therapy2,by="patient_id"
,all.x=TRUE,all.y=TRUE)
    if ("pump_therapy.x" %in% names(pump_therapy)) {
     pump_therapy$pump_therapy<-ifelse(is
.na(pump_therapy$pump_therapy.x),pump_therapy$pump_therapy.y,pump_therapy$pump_therapy
.x)
    }

    data3_3_4_4_5<-merge(data3_3_4_4_5,pump_therapy,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data3_3_4_4_5$pump_therapy))
    data3_3_4_4_5$pump_therapy<-ifelse(is
.na(data3_3_4_4_5$pump_therapy),99,data3_3_4_4_5$pump_therapy)

    lev_var_cat=names(levpump_med)[sort(numlevelcat)]

  if ("pump_therapy.x" %in% names(data3_3_4_4_5)) {
   data3_3_4_4_5<-data3_3_4_4_5[,-pmatch(c("pump_therapy.x","pump_therapy.y"
),names(data3_3_4_4_5))]
  }

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_4_4_5<-merge(data3_3_4_4_5,patient[c("patient_id","sub_ds_id")],by=
"patient_id",all.x=TRUE,all.y=FALSE)
  names(data3_3_4_4_5)[pmatch("sub_ds_id",names(data3_3_4_4_5))]<-"centre_id"

       BIRO_report(ind="3_3_4_4_5",
                   title="Glucose Lowering: Insulin Pump ",
                   var_cat="pump_therapy",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Pump Therapy",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                  )


  rm(data3_3_4_4_5)
 } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="3.3.4.4.5 Glucose Lowering: Insulin Pump ")
 }
 }
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="3.3.4.4.5 Glucose Lowering: Insulin Pump ")
   }
  }

if (exists("patient")) rm(patient)
if (exists("episode")) rm(episode)

 ##Management

   for (thisyear in anayears) {
    assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
    assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)

    new_sub_section_html(title="3.3.5 Management",htmlfile=htmlfile)
    new_sub_section(title="3.3.5 Management",texfile=texfile)
   }
 #
 # 3.3.5.1 Selfmonitoring (only epirefyear)
 #

  if (varpat$dob==TRUE & varpat$type_dm==TRUE &
      varepi$self_mon==TRUE) {


   for (thisyear in anayears) {

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
```

```
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)

    episode$self_mon=as.numeric(as.character(episode$self_mon))

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   if (dim(episode[is.na(episode$self_mon)==FALSE,])[1]>0) {

    data3_3_5_1<-BIRO_demographic(patient=patient,
                                  episode=episode[is.na(episode$self_mon)==FALSE,],
                                  var=NULL,
                                  date="last episode date",
                                  startvar="dob",
                                  th=ageth,
                                  hold=TRUE,
                                  varname="age")

    data3_3_5_1<-merge(episode[is.na(episode$self_mon)==FALSE,
                           c("patient_id","epi_date","self_mon")],
                       data3_3_5_1,
                       by.x=c("patient_id","epi_date"),
                       by.y=c("patient_id","last_epi_date"),
                       all.y=TRUE,
                       all.x=FALSE)

    data3_3_5_1<-merge(patient[,c("patient_id","type_dm","age_c")],
                       data3_3_5_1,
                       by.x=c("patient_id"),
                       by.y=c("patient_id"),
                       all.x=TRUE)

    numlevelcat<-na.omit(unique(data3_3_5_1$self_mon))
    data3_3_5_1$self_mon<-ifelse(is.na(data3_3_5_1$self_mon),99,data3_3_5_1$self_mon)

    lev_var_cat=names(levself_mon)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_5_1<-merge(data3_3_5_1,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_3_5_1)[pmatch("sub_ds_id",names(data3_3_5_1))]<-"centre_id"

      BIRO_report(ind="3_3_5_1",
                  title="Self monitoring",
                  var_cat="self_mon",
                  class=c("age_c",NA,"type_dm"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                  lab_var_cat="Self Monitoring",
                  lab_class=c("Age","Type of Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  bar=1,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex
                  )

 rm(data3_3_5_1)
 rm(patient)
 rm(episode)

 } else {
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="3.3.5.1 Self monitoring")
 }
 }
 } else {
   for (thisyear in anayears) {
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="3.3.5.1 Self monitoring")
   }
  }

#
# 3.3.5.2  Visit Frequency
# URGENT: this must be specified in the Dictionary and DATABASE
# TODO Create variable visit_freq (1=Low,2=High)



if (varpat$dob==TRUE & varpat$type_dm==TRUE &
    varpat$sex==TRUE) {

  for (thisyear in anayears) {

   checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
   oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
```

```
partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age_c<-varclass(patient$age,ageth)


       sixmbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")+180

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
  data3_3_5_2<-patient[,c("patient_id","age_c","type_dm")]

  episode$weight<-1


visit_freq<-as.data.frame(aggregate(episode$weight,by=list(episode$patient_id),FUN=sum
))
  names(visit_freq)<-c("patient_id","visit_freq")
  visit_freq$visit_freq<-ifelse(visit_freq$visit_freq>=2,2,1)

  data3_3_5_2<-merge(data3_3_5_2,
                     visit_freq,
                     by="patient_id",
                     all.x=FALSE,
                     all.y=FALSE)

  numlevelcat<-na.omit(unique(data3_3_5_2$visit_freq))
  data3_3_5_2<-data3_3_5_2[,c("patient_id","age_c","type_dm","visit_freq")]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data3_3_5_2<-merge(data3_3_5_2,patient[c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data3_3_5_2)[pmatch("sub_ds_id",names(data3_3_5_2))]<-"centre_id"

       BIRO_report(ind="3_3_5_2",
                   title="Visit Frequency",
                   var_cat="visit_freq",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=names(levvisit_freq)[sort(numlevelcat)],
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Visit Frequency",
                   lab_class=c("Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex
                   )


rm(data3_3_5_2)
rm(patient)
rm(episode)


}
   } else {
     for (thisyear in anayears) {
 assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="3.3.5.2 Visit Frequency")
     }
    }
```

```
##############################################################################
#                                                                            #
# Project: BIRO-Project (Funded by European Commission 2005-2008)            #
# File: BIRO_se_indicator_population.r                                       #
#                                                                            #
#
##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-24
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_se_indicator_population.r is part of WP Statistical Engine of the BIRO
# Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
##############################################################################


# 4.1.1. Total population

if (exists("pop")) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   new_section(title="4.1 Vital Statistics",texfile=texfile)
   new_section_html(title="4.1 Vital Statistics",htmlfile=htmlfile)


  data4_1_1<-pop[pop$year==thisyear,pmatch(c("ageband","popM","popF"),names(pop))]
  names(data4_1_1)<-c('Age','Males (N)','Females (N)')
  data4_1_1[,pmatch('Age',names(data4_1_1))]<-classlabel(c(15,25,35,45,55,65,75,85))
  data4_1_1<-aggregate(data4_1_1[,-pmatch('Age',names(data4_1_1))],by =list('Age' =
data4_1_1$'Age'),sum)
  data4_1_1$'Total (N)'<-data4_1_1$'Males (N)'+data4_1_1$'Females (N)'

  csvdata<-data4_1_1
  csvdata$centre_id<-centre_id
  csvdata$dbname<-dbname
  write.csv(csvdata,paste(dirdataout,"/","i","4_1_1","_d1_1.csv",sep=""
),row.names=FALSE)
  rm(csvdata)

  lastrow<-c("Overall",sum(data4_1_1$'Males (N)'),sum(data4_1_1$'Females (N)'
),sum(data4_1_1$'Total (N)'))
  data4_1_1<-rbind(data4_1_1,lastrow)

  BIRO_dframe(dataname="data4_1_1",monitor=TRUE,filelog="")

       BIRO_report(ind="4_1_1",
                   title="Total Population",
                   var=NULL,
                   var_cat=NULL,
                   lev_var_cat=NULL,
                   lev_class=NULL,
                   lab_var=NULL,
                   lab_var_cat=NULL,
                   lab_class=NULL,
                   tab=0,
                   box=0,
                   bar=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   catcol=1
                   )
```

```
  rm(data4_1_1)
 }
} else for (thisyear in anayears) BIRO_report_toc(title="4.1.1 Total Population")

# 4.1.2. Life expectancy

if (exists("pop")) {
 if (is.na(pmatch("morF",names(pop)))==FALSE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  #print(pop[pop$year==thisyear,])
  exm<-BIRO_explife(pop=pop[pop$year==thisyear,],
                    mor=pop[pop$year==thisyear,],
                    agewidth=c(15,10,10,10,10,10,10,10,12.5),
                    ageth=c(15,25,35,45,55,65,75,85),
                    sex="M",
                    morname="mor",
                    popname="pop",
                    labsex='Males')

  exf<-BIRO_explife(pop=pop[pop$year==thisyear,],
                    mor=pop[pop$year==thisyear,],
                    agewidth=c(15,10,10,10,10,10,10,10,12.5),
                    ageth=c(15,25,35,45,55,65,75,85),
                    sex="F",
                    morname="mor",
                    popname="pop",
                    labsex='Females')
  data4_1_2<-as.data.frame(merge(exm,exf))
  data4_1_2$Males<-round(as.numeric(as.character(data4_1_2$Males,2)))
  data4_1_2$Females<-round(as.numeric(as.character(data4_1_2$Females,2)))
  data4_1_2<-na.omit(data4_1_2)
  names(data4_1_2)[pmatch("Males",names(data4_1_2))]<-'Males (years)'
  names(data4_1_2)[pmatch("Females",names(data4_1_2))]<-'Females (years)'

  csvdata<-data4_1_2
  csvdata$centre_id<-centre_id
  csvdata$dbname<-dbname
  write.csv(csvdata,paste(dirdataout,"/","i","4_1_2","_d1_1.csv",sep=""
),row.names=FALSE)
  rm(csvdata)

  BIRO_dframe(dataname="data4_1_2",monitor=TRUE,filelog="")

       BIRO_report(ind="4_1_2",
                   title="Life expectancy",
                   var=NULL,
                   var_cat=NULL,
                   lev_var_cat=NULL,
                   lev_class=NULL,
                   lab_var=NULL,
                   lab_var_cat=NULL,
                   lab_class=NULL,
                   tab=0,
                   box=0,
                   bar=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,

                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   catcol=1
                   )

  rm(list=c("exm","exf","data4_1_2"))
  }
 } else for (thisyear in anayears) BIRO_report_toc(title="4.1.2 Life expenctancy")
} else for (thisyear in anayears) BIRO_report_toc(title="4.1.2 Life expenctancy")
# 4.1.3. Mortality data

if (exists("pop")) {
 if (is.na(pmatch("morF",names(pop)))==FALSE) {

  for (thisyear in anayears) {

     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")

    assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

    assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

    assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

    assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

    assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
    assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   data4_1_3<-pop[pop$year==thisyear,pmatch(c("ageband","morM","morF"),names(pop))]
   names(data4_1_3)<-c('Age','Males (N)','Females (N)')
   data4_1_3[,pmatch('Age',names(data4_1_3))]<-classlabel(c(15,25,35,45,55,65,75,85))
   data4_1_3<-aggregate(data4_1_3[,-pmatch('Age',names(data4_1_3))],by =list('Age' =
data4_1_3$'Age'),sum)
   data4_1_3$'Total (N)'<-data4_1_3$'Males (N)'+data4_1_3$'Females (N)'

   csvdata<-data4_1_3
   csvdata$centre_id<-centre_id
   csvdata$dbname<-dbname
   write.csv(csvdata,paste(dirdataout,"/","i","4_1_3","_d1_1.csv",sep=""
),row.names=FALSE)
   rm(csvdata)

   lastrow<-c("Overall",sum(data4_1_3$'Males (N)'),sum(data4_1_3$'Females (N)'
),sum(data4_1_3$'Total (N)'))
   data4_1_3<-rbind(data4_1_3,lastrow)

   BIRO_dframe(dataname="data4_1_3",monitor=TRUE,filelog="")

        BIRO_report(ind="4_1_3",
                    title="Mortality Data",
                    var=NULL,
                    var_cat=NULL,
                    lev_var_cat=NULL,
                    lev_class=NULL,
                    lab_var=NULL,
                    lab_var_cat=NULL,
                    lab_class=NULL,
                    tab=0,
                    box=0,
                    bar=0,
                    lines=0,
                    trellis=0,
                    pie=0,
                    texfile=texfile,

                    dircsv=dirdataout,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    catcol=1
                    )

  rm(data4_1_3)
   }
  } else for (thisyear in anayears) BIRO_report_toc(title="4.1.3 Mortality Data")
 } else for (thisyear in anayears) BIRO_report_toc(title="4.1.3 Mortality Data")
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_se_indicator_risk_adjusted.r                                      #
#                                                                              #
#
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_se_indicator_risk_adjusted.r is part of WP Statistical Engine of the
# BIRO Project
#
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# ##############################################################################

 # 5.1 Epidemiology

  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)

   new_section(title="5.1. Epidemiology",texfile=texfile)
   new_section_html(title="5.1. Epidemiology",htmlfile=htmlfile)
  }

#  spider_data<-as.data.frame(NULL)
  #Create Spider_data

#    if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE) {
#     for (thisyear in anayears) {
#      checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
#      oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
#
partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
#      patyear<-paste("profile_wide_",partname,sep="")
#      patient<-get(patyear)
#      spider_data_year<-as.data.frame(cbind(unique(patient$sub_ds_id)))
#      names(spider_data_year)<-"Strata"
#      rm(patient)
#     }
#     spider_data<-as.data.frame(na.omit(rbind(spider_data,spider_data_year)))
#     names(spider_data)<-"Strata"

#    }
   ##############################

 # 5.1.1 Prevalence of diabetes mellitus per 1,000

if (exists("pop")) {

   for (thisyear in anayears) {

  if (varpat$dob==TRUE & varpat$type_dm==TRUE &
      varpat$sex==TRUE) {


    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  pop$pop=pop$popM+pop$popF
```

```
  popyear<-pop[pop$year==thisyear,]
#  age<35 type 1 else type 2
#  if (is.null(pop$type)==TRUE) {
#   poptype_dm<-ifelse(popyear$ageband<=3,1,2)
#  }
  pop1<-popyear
  pop2<-popyear
  popyear<-rbind(pop1,pop2)
  popyear$type_dm<-c(rep(1,dim(pop1)[1]),rep(2,dim(pop2)[1]))

  if (("sub_ds_id" %in% names(patient))==FALSE) {
   popyear$centre_id<-centre_id
  }

  if (("centre_id" %in% names(popyear))==FALSE & ("sub_ds_id" %in% names(patient))) {
   allcentre<-unique(patient$sub_ds_id)
   newpopyear<-NULL
   for (h in 1:length(allcentre)) {
    temp<-popyear
    temp$centre_id<-allcentre[h]
    newpopyear<-rbind(newpopyear,temp)
   }
  popyear<-newpopyear
  }


poptype<-aggregate(popyear$pop,by=list(centre_id=popyear$centre_id,type_dm=popyear$type_dm),FUN=sum)
  names(poptype)<-c("centre_id","type_dm","pop")

  BIRO_dframe(dataname="poptype",monitor=TRUE,filelog="")


    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refanadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)

    patient$age_c<-varclass(patient$age,ageth)

  data5_1_1<-patient[,c("patient_id","type_dm","age")]
  data5_1_1$age_c<-varclass(data5_1_1$age,ageth5)
  data5_1_1$diabetes=1
  data5_1_1<-data5_1_1[,c("patient_id","diabetes","type_dm")]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_1_1<-merge(data5_1_1,patient[,c("patient_id","sub_ds_id")],by="patient_id",all.x=TRUE,all.y=FALSE)
  names(data5_1_1)[pmatch("sub_ds_id",names(data5_1_1))]<-"centre_id"

  BIRO_dframe(dataname="data5_1_1",monitor=TRUE,filelog="")

       BIRO_report(ind="5_1_1",
                   title="Prevalence of diabetes mellitus per 1,000",
                   var=NULL,
                   var_cat=NULL,
                   lev_var_cat=NULL,
                   lev_class=NULL,
                   lab_var=NULL,
                   lab_var_cat=NULL,
                   lab_class=NULL,
                   tab=0,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   catcol=1,
                   stand=1,
                   outcome="diabetes",
                   cov=c("type_dm"),
                   factcov=c(TRUE),
                   strata="centre_id",
                   levstd=list(a=list("Diabetic"=1),b=levtype_dmst),
                   sucvalue=1,
                   per=1000,
                   pop=poptype,
                   caption="5_1_1",
                   chisq=NULL)

  rm(data5_1_1)
      } else {
         for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

          BIRO_report_toc(title="5.1.1 Prevalence of diabetes mellitus per 1,000")
         }
      }
   }
}

#else {
#  for (thisyear in anayears) {
#   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
#          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
#    BIRO_report_toc(title="5.1.1 Prevalence of diabetes mellitus per 1,000")
#  }
#}


#


  if (varpat$dob==TRUE & varpat$dt_diag==TRUE &
      varpat$sex==TRUE & varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {


    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
```

```
partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  data5_1_2<-patient[,-pmatch("age",names(patient))]
  data5_1_2$age=round((data5_1_2$dt_diag-data5_1_2$dob)/365)

  ageth10<-c(10,20,30,40,50,60,70,80)

  data5_1_2$age_c<-varclass(data5_1_2$age,ageth10)
  data5_1_2$diabetes=1
  data5_1_2<-data5_1_2[,c("patient_id","diabetes","type_dm","sex","age_c")]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_1_2<-merge(data5_1_2,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_1_2)[pmatch("sub_ds_id",names(data5_1_2))]<-"centre_id"
  data5_1_2<-data5_1_2[data5_1_2$type_dm %in% c("1","2"),]

  numlevelcat<-na.omit(unique(data5_1_2$age_c))

  data5_1_2$age_c<-ifelse(is.na(data5_1_2$age_c),99,data5_1_2$age_c)

  lev_var_cat=classlabel(ageth10)[sort(numlevelcat)]

  BIRO_dframe(dataname="data5_1_2",monitor=TRUE,filelog="")


      BIRO_report(ind="5_1_2",
                  title="Age at diagnosis by 10 year age bands",
                var_cat="age_c",
                   class=c("sex",NA,"type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=levsex,b=NA,c=levtype_dmst),
                lab_var_cat="Age",
                lab_class=c("Gender","Type of Diabetes"),
                tab=1,
                tabside="H",
                tabperc=TRUE,
                tabwidth=0.95,
                barbeside=TRUE,
                barnumber=TRUE,
                barperc=TRUE,
                box=0,
                bar=1,
                lines=0,
                trellis=0,
                pie=0,
                texfile=texfile,
                dircsv=dirdataout,
                dirgraph=dirgraph,
                dirgraphlatex=dirgraphlatex,
                cex=cex,
                chisq=NULL)

  rm(data5_1_2)
  rm(patient)
   }
} else for (thisyear in anayears) BIRO_report_toc(title="5.1.2. Age at diagnosis by
10 year age bands")
#

  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)
   new_section(title="5.2. Process Quality",texfile=texfile)
   new_section_html(title="5.2. Process Quality",htmlfile=htmlfile)
  }
  #5.2.1.
  #% subjects with 1+ HbA1c tests
  #during the last 12 months


if (varepi$hba1c==TRUE & varpat$dob==TRUE &
    varpat$dt_diag==TRUE & varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {


    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   if (dim(episode[is.na(episode$hba1c)==FALSE & episode$hba1c>0,])[1]>0) {

    data5_2_1<-patient[,c("patient_id","age2_c","age_c","sex","type_dm")]
    hba1c_done<-as.data.frame(unique(episode[is.na(episode$hba1c)==FALSE &
episode$hba1c>0,"patient_id"]))
    names(hba1c_done)<-"patient_id"
    hba1c_done$hba1c_done<-1

    data5_2_1<-merge(data5_2_1,hba1c_done,by="patient_id",all.x=TRUE)

    data5_2_1$hba1c_done<-ifelse(is.na(data5_2_1$hba1c_done),2,data5_2_1$hba1c_done)


    data5_2_1$agesex<-ifelse(data5_2_1$sex=="1",data5_2_1$age_c,0)

        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_1<-merge(data5_2_1,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_1)[pmatch("sub_ds_id",names(data5_2_1))]<-"centre_id"

    data5_2_1<-data5_2_1[data5_2_1$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_2_1",monitor=TRUE,filelog="")

       BIRO_report(ind="5_2_1",
                   title="% subjects with 1+ HbA1c tests during the last 12 months",
                   var_cat="hba1c_done",
                   class=c("age2_c","sex","type_dm"),
                   lev_var_cat=names(levhba1c_test),
                   lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                   lab_var_cat="HbA1c done",
                   lab_class=c("Age","Gender","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="hba1c_done",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
                   caption="% subjects with 1+ HbA1c tests during the last 12 months")

  rm(data5_2_1)
  rm(patient)
  rm(episode)
  rm(noNAepirefyear)
 } else BIRO_report_toc(title="5.2.1 % subjects with 1+ HbA1c tests during the last
12 months")
 }
 } else for (thisyear in anayears) BIRO_report_toc(title="5.2.1 % subjects with 1+
HbA1c tests during the last 12 months")

  #5.2.2.
  #5.2.2 % subjects with at least one test for microalbuminuria during the last 12
months


if (varepi$ma_test==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {
   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
```

```
   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if (dim(episode[is.na(episode$ma_test)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,])[1]>0) {

    data5_2_2<-patient[,c("patient_id","age_c","age2_c","sex","type_dm")]

    ma_done<-as.data.frame(unique(episode[is.na(episode$ma_test)==FALSE &
episode$ma_test %in% c(1,2),"patient_id"]))
    names(ma_done)<-"patient_id"
    if (dim(ma_done)[1]>0) {
     ma_done$ma_done<-1
    }

     ma_done2<-as.data.frame(unique(episode[is.na(episode$ma_test)==FALSE &
episode$ma_test==0,"patient_id"]))
     names(ma_done2)<-"patient_id"
     if (dim(ma_done2)[1]>0) {
      ma_done2$ma_done<-2
     }

    ma_done<-merge(ma_done,ma_done2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("ma_done.x" %in% names(ma_done)) {
     ma_done$ma_done<-ifelse(is
.na(ma_done$ma_done.x),ma_done$ma_done.y,ma_done$ma_done.x)
    }

    data5_2_2<-merge(data5_2_2,ma_done,by="patient_id",all.x=TRUE)

    data5_2_2$ma_done<-ifelse(is.na(data5_2_2$ma_done),2,data5_2_2$ma_done)


    data5_2_2$agesex<-ifelse(data5_2_2$sex=="1",data5_2_2$age_c,NA)

        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_2<-merge(data5_2_2,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_2)[pmatch("sub_ds_id",names(data5_2_2))]<-"centre_id"

    data5_2_2$agesex<-ifelse(data5_2_2$sex=="1",data5_2_2$age_c,0)

    data5_2_2<-data5_2_2[data5_2_2$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_2_2",monitor=TRUE,filelog="")

   BIRO_report(ind="5_2_2",
                    title="% subjects with at least one test for microalbuminuria
during the last 12 months",
                    var_cat="ma_done",
                    class=c("age2_c","sex","type_dm"),
                    lev_var_cat=names(levma_test),
                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                    lab_var_cat="Microalbumin done",
                    lab_class=c("Age","Gender","Type of Diabetes"),
                    tab=1,
                    tabside="H",
                    tabperc=TRUE,
                    tabwidth=0.95,
                    bar=1,
                    barbeside=TRUE,
                    barnumber=TRUE,
                    barperc=TRUE,
                    box=0,
                    lines=0,
                    trellis=0,
                    pie=0,
                    texfile=texfile,
                    dircsv=dirdataout,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    stand=1,
                    catcol=c(2,3,4),
                    outcome="ma_done",
                    cov=c("sex","age_c","agesex","type_dm"),
                    chisq=NULL,
                    factcov=c(TRUE,TRUE,TRUE,TRUE),
                    strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                    sucvalue=1,
                    per=100,
                    caption="% subjects with at least one test for microalbuminuria
during the last 12 months")

  rm(data5_2_2)
  rm(patient)
  rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.2.2 % subjects with at least one test for
microalbuminuria during the last 12 months")
  }
   }
} else {
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.2.2 % subjects with at least one test for
microalbuminuria during the last 12 months")
  }
}


 #5.2.3 Percentage of diabetes patients who received a dilated eye examination or
evaluation of retinal photography by a trained caregiver during the 12 months


 if (varepi$retinal_exam==TRUE & varpat$dob==TRUE &
```

```
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if (dim(episode[is.na(episode$retinal_exam)==FALSE & episode$epi_date >= oneyearbf
& episode$epi_date <= checkdate,])[1]>0) {

    data5_2_3<-patient[,c("patient_id","age_c","age2_c","sex","type_dm")]
    eye_done<-as.data.frame(unique(episode[is.na(episode$retinal_exam)==FALSE &
episode$retinal_exam==1,"patient_id"]))
    names(eye_done)<-"patient_id"
    eye_done$eye_done<-1

    eye_done2<-as.data.frame(unique(episode[is.na(episode$retinal_exam)==FALSE &
episode$retinal_exam==0,"patient_id"]))
    names(eye_done2)<-"patient_id"
    if (dim(eye_done2)[1]>0) {
     eye_done2$eye_done<-2
    }

    eye_done<-merge(eye_done,eye_done2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("eye_done.x" %in% names(eye_done)) {
     eye_done$eye_done<-ifelse(is
.na(eye_done$eye_done.x),eye_done$eye_done.y,eye_done$eye_done.x)
    }

    data5_2_3<-merge(data5_2_3,eye_done,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data5_2_3$eye_done))
    data5_2_3$eye_done<-ifelse(is.na(data5_2_3$eye_done),99,data5_2_3$eye_done)

    lev_var_cat=names(leveye_done)[sort(numlevelcat)]

    data5_2_3$agesex<-ifelse(data5_2_3$sex=="1",data5_2_3$age_c,NA)

    if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_3$centre_id<-patient$sub_ds_id
  names(data5_2_3)[pmatch("sub_ds_id",names(data5_2_3))]<-"centre_id"

    data5_2_3$agesex<-ifelse(data5_2_3$sex=="1",data5_2_3$age_c,0)

    data5_2_3<-data5_2_3[data5_2_3$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_2_3",monitor=TRUE,filelog="")

       BIRO_report(ind="5_2_3",
                   title="Percentage of diabetes patients who received a dilated eye
examination or evaluation of retinal photography by a trained caregiver during the 12
months",
                   var_cat="eye_done",
                   class=c("age2_c","sex","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                   lab_var_cat="Eye examination",
                   lab_class=c("Age","Gender","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="eye_done",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
                   caption="Percentage of diabetes patients who received a dilated
eye examination or evaluation of retinal photography by a trained caregiver during
the 12 months")
```

```
  rm(data5_2_3)
  rm(patient)
  rm(episode)


 } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.3 Percentage of diabetes patients who received a
dilated eye examination or evaluation of retinal photography by a trained caregiver
during the 12 months")
 }
 }
 } else {
   for (thisyear in anayears) {

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.3 Percentage of diabetes patients who received a
dilated eye examination or evaluation of retinal photography by a trained caregiver
during the 12 months")
   }
 }


  #5.2.4 Percentage of diabetes patients receiving at least one foot examination
during the 12 months

 if (varepi$foot_exam==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   if(dim(episode[is.na(episode$foot_exam)==FALSE,])[1]>0) {

    data5_2_4<-patient[,c("patient_id","sex","age_c","age2_c","type_dm")]

    footexam<-as.data.frame(unique(episode[is.na(episode$foot_exam)==FALSE &
episode$foot_exam==1,"patient_id"])) #patient with
    names(footexam)<-"patient_id"
    if (dim(footexam)[1]>0) {
     footexam$footexam<-1
    }

     footexam2<-as.data.frame(unique(episode[is.na(episode$foot_exam)==FALSE &
episode$foot_exam==0,"patient_id"]))
     names(footexam2)<-"patient_id"
     if (dim(footexam2)[1]>0) {
      footexam2$footexam<-2
     }

    footexam<-merge(footexam,footexam2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("footexam.x" %in% names(footexam)) {
     footexam$footexam<-ifelse(is
.na(footexam$footexam.x),footexam$footexam.y,footexam$footexam.x)
    }


    data5_2_4<-merge(data5_2_4,footexam,by="patient_id",all.x=TRUE,all.y=FALSE) #Only
people with foot examination

    numlevelcat<-na.omit(unique(data5_2_4$footexam))

    data5_2_4$footexam<-ifelse(is.na(data5_2_4$footexam),99,data5_2_4$footexam)

    lev_var_cat=names(levfoot_done)[sort(numlevelcat)]


        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_4<-merge(data5_2_4,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_4)[pmatch("sub_ds_id",names(data5_2_4))]<-"centre_id"

    data5_2_4$agesex<-ifelse(data5_2_4$sex=="1",data5_2_4$age_c,0)

    data5_2_4<-data5_2_4[data5_2_4$type_dm %in% c("1","2"),]

  if ("footexam.x" %in% names(data5_2_4)) {
   data5_2_4<-data5_2_4[,-pmatch(c("footexam.x"),names(data5_2_4))]
  }
  if ("footexam.y" %in% names(data5_2_4)) {
   data5_2_4<-data5_2_4[,-pmatch(c("footexam.y"),names(data5_2_4))]
  }

  BIRO_dframe(dataname="data5_2_4",monitor=TRUE,filelog="")
```

```
       BIRO_report(ind="5_2_4",
                   title="Percentage of diabetes patients receiving at least one foot
examination during the 12 months",
                   var_cat="footexam",
                   class=c("age2_c","sex","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                   lab_var_cat="Foot examination",
                   lab_class=c("Age","Gender","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="footexam",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
                   caption="Percentage of diabetes patients receiving at least one
foot examination during the 12 months")

  rm(data5_2_4)
  rm(patient)
  rm(episode)

 } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.4 Percentage of diabetes patients receiving at least
one foot examination during the 12 months")
 }
 }
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.4 Percentage of diabetes patients receiving at least
one foot examination during the 12 months")
   }
 }
```

```
  #5.2.5 Percentage of diabetes patients whose smoking status was ascertained and
documented during the 12 months

 if (varepi$smok_stat==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if(dim(episode[is.na(episode$smok_stat)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,])[1]>0) {


    data5_2_5<-patient[,c("patient_id","age_c","age2_c","sex","type_dm")]
    smok_stat<-as.data.frame(unique(episode[is.na(episode$smok_stat)==FALSE &
episode$smok_stat %in% c(1,2,3),"patient_id"]))
    names(smok_stat)<-"patient_id"
    smok_stat$smok_stat<-1

    data5_2_5<-merge(data5_2_5,smok_stat,by="patient_id",all.x=TRUE)

    data5_2_5$smok_stat<-ifelse(is.na(data5_2_5$smok_stat),2,data5_2_5$smok_stat)


        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
```

```
data5_2_5<-merge(data5_2_5,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_5)[pmatch("sub_ds_id",names(data5_2_5))]<-"centre_id"

    data5_2_5$agesex<-ifelse(data5_2_5$sex=="1",data5_2_5$age_c,0)

    data5_2_5<-data5_2_5[data5_2_5$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_2_5",monitor=TRUE,filelog="")

       BIRO_report(ind="5_2_5",
                   title="Percentage of diabetes patients whose smoking status was
ascertained and documented during the 12 months",
                   var_cat="smok_stat",
                   class=c("age2_c","sex","type_dm"),
                   lev_var_cat=names(levsmoke_doc),
                   lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                   lab_var_cat="Smoke Status",
                   lab_class=c("Age","Gender","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="smok_stat",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
                   caption="Percentage of diabetes patients whose smoking status was
ascertained and documented during the 12 months")

  rm(data5_2_5)
  rm(patient)
  rm(episode)

 } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.5 Percentage of diabetes patients whose smoking
status was ascertained and documented during the 12 months")
 }
 }
```

```
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.5 Percentage of diabetes patients whose smoking
status was ascertained and documented during the 12 months")
   }
  }

  #5.2.6 Percent with serum creatinine tested in last 12 months



 if (varepi$creat==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   if (dim(episode[is.na(episode$creat)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,])[1]>0) {

    data5_2_6<-patient[,c("patient_id","age_c","age2_c","sex","type_dm")]
    creat<-as.data.frame(unique(episode[is.na(episode$creat)==FALSE & episode$creat>0,
"patient_id"]))
```

```
    names(creat)<-"patient_id"
    creat$creat<-1

    data5_2_6<-merge(data5_2_6,creat,by="patient_id",all.x=TRUE)

    data5_2_6$creat<-ifelse(is.na(data5_2_6$creat),2,data5_2_6$creat)


        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_6<-merge(data5_2_6,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_6)[pmatch("sub_ds_id",names(data5_2_6))]<-"centre_id"

    data5_2_6$agesex<-ifelse(data5_2_6$sex=="1",data5_2_6$age_c,0)

    data5_2_6<-data5_2_6[data5_2_6$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_2_6",monitor=TRUE,filelog="")

       BIRO_report(ind="5_2_6",
                   title="% subjects with serum creatinine tested in last 12 months",
                   var_cat="creat",
                   class=c("age2_c","sex","type_dm"),
                   lev_var_cat=names(levcreat_done),
                   lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                   lab_var_cat="Creatinine",
                   lab_class=c("Age","Gender","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="creat",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
                   caption="% subjects with serum creatinine tested in last 12 months"
)

  rm(data5_2_6)
  rm(patient)
  rm(episode)
  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.6 % subjects with serum creatinine tested in last 12
months")
   }
  }
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.6 % subjects with serum creatinine tested in last 12
months")
   }
  }

  #5.2.7 % subjects with 1+ BP measurements during the 12 months
  #



 if ((varepi$sbp==TRUE | varepi$dbp==TRUE) & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  episode$bp<-ifelse(is.na(episode$sbp)==FALSE & episode$sbp>0 &
                     is.na(episode$dbp)==FALSE & episode$dbp>0,1,2)

  if (dim(episode[is.na(episode$bp)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,])[1]>0) {

    data5_2_7<-patient[,c("patient_id","age_c","age2_c","sex","type_dm")]
    bp<-as.data.frame(unique(episode[is.na(episode$bp)==FALSE & episode$bp==1,
"patient_id"]))
    names(bp)<-"patient_id"
    bp$bp<-1

    data5_2_7<-merge(data5_2_7,bp,by="patient_id",all.x=TRUE)

    data5_2_7$bp<-ifelse(is.na(data5_2_7$bp),2,data5_2_7$bp)

        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_7<-merge(data5_2_7,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_7)[pmatch("sub_ds_id",names(data5_2_7))]<-"centre_id"

    data5_2_7$agesex<-ifelse(data5_2_7$sex=="1",data5_2_7$age_c,0)

    data5_2_7<-data5_2_7[data5_2_7$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_2_7",monitor=TRUE,filelog="")

      BIRO_report(ind="5_2_7",
                  title="% subjects with 1+ BP measurements during the 12 months",
                  var_cat="bp",
                  class=c("age2_c","sex","type_dm"),
                  lev_var_cat=names(levbp12),
                  lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                  lab_var_cat="BP measurements",
                  lab_class=c("Age","Gender","Type of Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  bar=1,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="bp",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                  sucvalue=1,
                  per=100,
                  caption="% subjects with 1+ BP measurements during the 12 months")

   rm(data5_2_7)
  rm(patient)
  rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.7 % subjects with 1+ BP measurements during the 12
months")
  }
   }
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.7 % subjects with 1+ BP measurements during the 12
months")
   }
  }

  #5.2.8 % subjects with hypertension who receive antihypertensive medication
  #

 if (varepi$hypert_med==TRUE & varepi$hypertension==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {


   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

    episode$hypert_med<-as.numeric(as.character(episode$hypert_med))

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
```

```
   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if(dim(episode[is.na(episode$hypert_med)==FALSE & is
.na(episode$hypertension)==FALSE & episode$hypertension==1,])[1]>0) {

    #data5_2_8<-patient[,c("patient_id","age_c","age2_c","sex","type_dm")]
    #Numerator
    hypertension<-as.data.frame(unique(episode[is.na(episode$hypertension)==FALSE &
episode$hypertension==1,"patient_id"]))
    names(hypertension)<-"patient_id"
    hypertension$hypertension<-1

    #Cohort
    hypert_med<-as.data.frame(unique(episode[is.na(episode$hypert_med)==FALSE &
episode$hypert_med==1,"patient_id"]))
    names(hypert_med)<-"patient_id"
    if (dim(hypert_med)[1]>0) {
     hypert_med$hypert_med<-1
    }

    hypert_med2<-as.data.frame(unique(episode[is.na(episode$hypert_med)==FALSE &
episode$hypert_med==0,"patient_id"]))
    names(hypert_med2)<-"patient_id"
    if (dim(hypert_med2)[1]>0) {
     hypert_med2$hypert_med<-2
    }

    hypert_med<-merge(hypert_med,hypert_med2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("hypert_med.x" %in% names(hypert_med)) {
     hypert_med$hypert_med<-ifelse(is
.na(hypert_med$hypert_med.x),hypert_med$hypert_med.y,hypert_med$hypert_med.x)
    }

    data5_2_8<-merge(hypertension,hypert_med,by="patient_id",all.x=TRUE,all.y=FALSE)

    numlevelcat<-na.omit(unique(data5_2_8$hypert_med))
    data5_2_8$hypert_med<-ifelse(is.na(data5_2_8$hypert_med),99,data5_2_8$hypert_med)

    lev_var_cat=names(levhypert)[sort(numlevelcat)]

    if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE) {
    data5_2_8<-merge(data5_2_8,patient[,c("patient_id","age_c","age2_c","sex",
"type_dm","sub_ds_id")],by="patient_id",all.x=FALSE,all.y=FALSE)
    names(data5_2_8)[pmatch("sub_ds_id",names(data5_2_8))]<-"centre_id"
    }

    if (is.na(pmatch("sub_ds_id",names(patient)))==TRUE) {
    data5_2_8<-merge(data5_2_8,patient[,c("patient_id","age_c","age2_c","sex",
"type_dm")],by="patient_id",all.x=FALSE,all.y=FALSE)
    }

    data5_2_8$agesex<-ifelse(data5_2_8$sex=="1",data5_2_8$age_c,0)

    data5_2_8<-data5_2_8[data5_2_8$type_dm %in% c("1","2"),]
    if ("hypert_med.x" %in% names(data5_2_8)) {
     data5_2_8<-data5_2_8[,-pmatch(c("hypert_med.x","hypert_med.y"),names(data5_2_8))]
    }
  BIRO_dframe(dataname="data5_2_8",monitor=TRUE,filelog="")

      BIRO_report(ind="5_2_8",
                  title="% subjects with hypertension who receive antihypertensive
medication",
                  var_cat="hypert_med",
                  class=c("age2_c","sex","type_dm"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                  lab_var_cat="Hypertension",
                  lab_class=c("Age","Gender","Type of Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  bar=1,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="hypert_med",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                  sucvalue=1,
                  per=100,
                  caption="% subjects with hypertension who receive antihypertensive
medication")

   rm(data5_2_8)
   rm(patient)
   rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.8 % subjects with hypertension who receive
antihypertensive medication")
  }
```

```
   }
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.8 % subjects with hypertension who receive
antihypertensive medication")
   }
  }

  #5.2.9.1 Type of oral therapy (distribution of agents) in patients with diabetes
type 2
  #sulphonylureas

  if (varepi$sulphonylureas==TRUE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,ageth)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   if (dim(episode[is.na(episode$sulphonylureas)==FALSE,])[1]>0) {

   data5_2_9_1<-patient[,c("patient_id","sex","type_dm","age_c","age2_c")]
   #Patient with Sulphonylureas (1)in 12 months

   sulphonylureas<-as.data.frame(NULL)

   sulphonylureas<-as.data.frame(unique(episode[is.na(episode$sulphonylureas)==FALSE
& episode$sulphonylureas==1,"patient_id"]))
   names(sulphonylureas)<-"patient_id"
   if (dim(sulphonylureas)[1]>0) {
    sulphonylureas$sulphonylureas<-1
   }

   sulphonylureas2<-as.data.frame(unique(episode[is.na(episode$sulphonylureas)==FALSE
& episode$sulphonylureas==0,"patient_id"]))
   names(sulphonylureas2)<-"patient_id"
   if (dim(sulphonylureas2)[1]>0) {
    sulphonylureas2$sulphonylureas<-2
   }

    sulphonylureas<-merge(sulphonylureas,sulphonylureas2,by="patient_id"
,all.x=TRUE,all.y=TRUE)
    if ("sulphonylureas.x" %in% names(sulphonylureas)) {
     sulphonylureas$sulphonylureas<-ifelse(is
.na(sulphonylureas$sulphonylureas.x),sulphonylureas$sulphonylureas.y,sulphonylureas$su
lphonylureas.x)
    }


    data5_2_9_1<-merge(data5_2_9_1,sulphonylureas,by="patient_id"
,all.x=TRUE,all.y=FALSE) #Only people with Sulphonylureas

    numlevelcat<-na.omit(unique(data5_2_9_1$sulphonylureas))

    data5_2_9_1$sulphonylureas<-ifelse(is
.na(data5_2_9_1$sulphonylureas),99,data5_2_9_1$sulphonylureas)

    data5_2_9_1<-data5_2_9_1[data5_2_9_1$type_dm %in% c("2"),]

    data5_2_9_1<-data5_2_9_1[,-pmatch(c("sulphonylureas.x","sulphonylureas.y"
),names(data5_2_9_1))]

    data5_2_9_1$agesex<-ifelse(data5_2_9_1$sex=="1",data5_2_9_1$age_c,0)

  BIRO_dframe(dataname="data5_2_9_1",monitor=TRUE,filelog="")

      BIRO_report(ind="5_2_9_1",
                  title="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Sulphonylureas",
                  var_cat="sulphonylureas",
                  class=c("sex","age2_c"),
                  lev_var_cat=names(levoral_th12),
                  lev_class=list(a=levsex,b=classlabellist(ageth)),
                  lab_var_cat="Sulphonylureas",
                  lab_class=c("Gender","Age"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  bar=1,
                  lines=0,
```

```
                  printlines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="sulphonylureas",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                  sucvalue=1,
                  per=100,
                  caption="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Sulphonylureas")
  rm(data5_2_9_1)
  rm(patient)
  rm(episode)
  rm(sulphonylureas,sulphonylureas2)

   } else BIRO_report_toc(title="5.2.9.1 Type of oral therapy (distribution of
agents) in patients with diabetes type 2: Sulphonylureas")
  }
 } else for (thisyear in anayears) BIRO_report_toc(title="5.2.9.1 Type of oral
therapy (distribution of agents) in patients with diabetes type 2: Sulphonylureas")

  #5.2.9.2 Type of oral therapy (distribution of agents) in patients with diabetes
type 2
  #biguanides

  if (varepi$sulphonylureas==TRUE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,ageth)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   if (dim(episode[is.na(episode$biguanides)==FALSE,])[1]>0) {

   data5_2_9_2<-patient[,c("patient_id","sex","type_dm","age_c","age2_c")]
   #Patient with biguanides (1)in 12 months

   biguanides<-as.data.frame(NULL)

   biguanides<-as.data.frame(unique(episode[is.na(episode$biguanides)==FALSE &
episode$biguanides==1,"patient_id"]))
   names(biguanides)<-"patient_id"
   if (dim(biguanides)[1]>0) {
    biguanides$biguanides<-1
   }

   biguanides2<-as.data.frame(unique(episode[is.na(episode$biguanides)==FALSE &
episode$biguanides==0,"patient_id"]))
   names(biguanides2)<-"patient_id"
   if (dim(biguanides2)[1]>0) {
    biguanides2$biguanides<-2
   }

    biguanides<-merge(biguanides,biguanides2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("biguanides.x" %in% names(biguanides)) {
     biguanides$biguanides<-ifelse(is
.na(biguanides$biguanides.x),biguanides$biguanides.y,biguanides$biguanides.x)
    }


    data5_2_9_2<-merge(data5_2_9_2,biguanides,by="patient_id",all.x=TRUE,all.y=FALSE)
#Only people with biguanides

    numlevelcat<-na.omit(unique(data5_2_9_2$biguanides))

    data5_2_9_2$biguanides<-ifelse(is
.na(data5_2_9_2$biguanides),99,data5_2_9_2$biguanides)

    data5_2_9_2<-data5_2_9_2[data5_2_9_2$type_dm %in% c("2"),]

    data5_2_9_2<-data5_2_9_2[,-pmatch(c("biguanides.x","biguanides.y"
),names(data5_2_9_2))]

    data5_2_9_2$agesex<-ifelse(data5_2_9_2$sex=="1",data5_2_9_2$age_c,0)

  BIRO_dframe(dataname="data5_2_9_2",monitor=TRUE,filelog="")

      BIRO_report(ind="5_2_9_2",
                  title="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Biguanides",
```

```
                  var_cat="biguanides",
                  class=c("sex","age2_c"),
                  lev_var_cat=names(levoral_th12),
                  lev_class=list(a=levsex,b=classlabellist(ageth)),
                  lab_var_cat="Biguanides",
                  lab_class=c("Gender","Age"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  bar=1,
                  lines=0,
                  printlines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="biguanides",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                  sucvalue=1,
                  per=100,
                  caption="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Biguanides")
  rm(data5_2_9_2)
  rm(patient)
  rm(episode)
  rm(biguanides,biguanides2)

   } else BIRO_report_toc(title="5.2.9.2 Type of oral therapy (distribution of
agents) in patients with diabetes type 2: Biguanides")
  }
 } else for (thisyear in anayears) BIRO_report_toc(title="5.2.9.2 Type of oral
therapy (distribution of agents) in patients with diabetes type 2: Biguanides")

  #5.2.9.3 Type of oral therapy (distribution of agents) in patients with diabetes
type 2
  #glucosidase_inhibitors

  if (varepi$sulphonylureas==TRUE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,ageth)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   if (dim(episode[is.na(episode$glucosidase_inhibitors)==FALSE,])[1]>0) {

   data5_2_9_3<-patient[,c("patient_id","sex","type_dm","age_c","age2_c")]
   #Patient with glucosidase_inhibitors (1)in 12 months

   glucosidase_inhibitors<-as.data.frame(NULL)

   glucosidase_inhibitors<-as.data.frame(unique(episode[is
.na(episode$glucosidase_inhibitors)==FALSE & episode$glucosidase_inhibitors==1,
"patient_id"]))
   names(glucosidase_inhibitors)<-"patient_id"
   if (dim(glucosidase_inhibitors)[1]>0) {
    glucosidase_inhibitors$glucosidase_inhibitors<-1
   }

   glucosidase_inhibitors2<-as.data.frame(unique(episode[is
.na(episode$glucosidase_inhibitors)==FALSE & episode$glucosidase_inhibitors==0,
"patient_id"]))
   names(glucosidase_inhibitors2)<-"patient_id"
   if (dim(glucosidase_inhibitors2)[1]>0) {
    glucosidase_inhibitors2$glucosidase_inhibitors<-2
   }

    glucosidase_inhibitors<-merge(glucosidase_inhibitors,glucosidase_inhibitors2,by=
"patient_id",all.x=TRUE,all.y=TRUE)
    if ("glucosidase_inhibitors.x" %in% names(glucosidase_inhibitors)) {
     glucosidase_inhibitors$glucosidase_inhibitors<-ifelse(is
.na(glucosidase_inhibitors$glucosidase_inhibitors.x),glucosidase_inhibitors$glucosidas
e_inhibitors.y,glucosidase_inhibitors$glucosidase_inhibitors.x)
    }
```

```
    data5_2_9_3<-merge(data5_2_9_3,glucosidase_inhibitors,by="patient_id"
,all.x=TRUE,all.y=FALSE) #Only people with glucosidase_inhibitors

    numlevelcat<-na.omit(unique(data5_2_9_3$glucosidase_inhibitors))

    data5_2_9_3$glucosidase_inhibitors<-ifelse(is
.na(data5_2_9_3$glucosidase_inhibitors),99,data5_2_9_3$glucosidase_inhibitors)

    data5_2_9_3<-data5_2_9_3[data5_2_9_3$type_dm %in% c("2"),]

    data5_2_9_3<-data5_2_9_3[,-pmatch(c("glucosidase_inhibitors.x",
"glucosidase_inhibitors.y"),names(data5_2_9_3))]

    data5_2_9_3$agesex<-ifelse(data5_2_9_3$sex=="1",data5_2_9_3$age_c,0)

  BIRO_dframe(dataname="data5_2_9_3",monitor=TRUE,filelog="")

      BIRO_report(ind="5_2_9_3",
                  title="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Glucosidase",
                  var_cat="glucosidase_inhibitors",
                  class=c("sex","age2_c"),
                  lev_var_cat=names(levoral_th12),
                  lev_class=list(a=levsex,b=classlabellist(ageth)),
                  lab_var_cat="Glucosidase Inhibitors",
                  lab_class=c("Gender","Age"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  bar=1,
                  lines=0,
                  printlines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="glucosidase_inhibitors",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                  sucvalue=1,
                  per=100,
                  caption="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Glucosidase")
  rm(data5_2_9_3)
  rm(patient)
  rm(episode)
  rm(glucosidase_inhibitors,glucosidase_inhibitors2)
   } else BIRO_report_toc(title="5.2.9.3 Type of oral therapy (distribution of
agents) in patients with diabetes type 2: Glucosidase")
  }
 } else for (thisyear in anayears) BIRO_report_toc(title="5.2.9.3 Type of oral
therapy (distribution of agents) in patients with diabetes type 2: Glucosidase")

  #5.2.9.4 Type of oral therapy (distribution of agents) in patients with diabetes
type 2
  #sulphonylureas

  if (varepi$glitazones==TRUE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,ageth)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   if (dim(episode[is.na(episode$glitazones)==FALSE,])[1]>0) {

   data5_2_9_4<-patient[,c("patient_id","sex","type_dm","age_c","age2_c")]
   #Patient with glitazones (1)in 12 months

   glitazones<-as.data.frame(NULL)

   glitazones<-as.data.frame(unique(episode[is.na(episode$glitazones)==FALSE &
episode$glitazones==1,"patient_id"]))
   names(glitazones)<-"patient_id"
   if (dim(glitazones)[1]>0) {
    glitazones$glitazones<-1
```

```
   }

   glitazones2<-as.data.frame(unique(episode[is.na(episode$glitazones)==FALSE &
episode$glitazones==0,"patient_id"]))
   names(glitazones2)<-"patient_id"
   if (dim(glitazones2)[1]>0) {
    glitazones2$glitazones<-2
   }

    glitazones<-merge(glitazones,glitazones2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("glitazones.x" %in% names(glitazones)) {
     glitazones$glitazones<-ifelse(is
.na(glitazones$glitazones.x),glitazones$glitazones.y,glitazones$glitazones.x)
    }


    data5_2_9_4<-merge(data5_2_9_4,glitazones,by="patient_id",all.x=TRUE,all.y=FALSE)
#Only people with glitazones

    numlevelcat<-na.omit(unique(data5_2_9_4$glitazones))

    data5_2_9_4$glitazones<-ifelse(is
.na(data5_2_9_4$glitazones),99,data5_2_9_4$glitazones)

    data5_2_9_4<-data5_2_9_4[data5_2_9_4$type_dm %in% c("2"),]

    data5_2_9_4<-data5_2_9_4[,-pmatch(c("glitazones.x","glitazones.y"
),names(data5_2_9_4))]

    data5_2_9_4$agesex<-ifelse(data5_2_9_4$sex=="1",data5_2_9_4$age_c,0)

  BIRO_dframe(dataname="data5_2_9_4",monitor=TRUE,filelog="")

      BIRO_report(ind="5_2_9_4",
                  title="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Glitazones",
                  var_cat="glitazones",
                  class=c("sex","age2_c"),
                  lev_var_cat=names(levoral_th12),
                  lev_class=list(a=levsex,b=classlabellist(ageth)),
                  lab_var_cat="Glitazones",
                  lab_class=c("Gender","Age"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  bar=1,
                  lines=0,
                  printlines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="glitazones",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                  sucvalue=1,
                  per=100,
                  caption="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Glitazones")
  rm(data5_2_9_4)
  rm(patient)
  rm(episode)
  rm(glitazones,glitazones2)

   } else BIRO_report_toc(title="5.2.9.4 Type of oral therapy (distribution of
agents) in patients with diabetes type 2: Glitazones")
  }
 } else for (thisyear in anayears) BIRO_report_toc(title="5.2.9.4 Type of oral
therapy (distribution of agents) in patients with diabetes type 2: Glitazones")

  #5.2.9.5 Type of oral therapy (distribution of agents) in patients with diabetes
type 2
  #glinides

  if (varepi$glinides==TRUE) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,ageth)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   if (dim(episode[is.na(episode$glinides)==FALSE,])[1]>0) {

   data5_2_9_5<-patient[,c("patient_id","sex","type_dm","age_c","age2_c")]
   #Patient with glinides (1)in 12 months

   glinides<-as.data.frame(NULL)

   glinides<-as.data.frame(unique(episode[is.na(episode$glinides)==FALSE &
episode$glinides==1,"patient_id"]))
   names(glinides)<-"patient_id"
   if (dim(glinides)[1]>0) {
    glinides$glinides<-1
   }

   glinides2<-as.data.frame(unique(episode[is.na(episode$glinides)==FALSE &
episode$glinides==0,"patient_id"]))
   names(glinides2)<-"patient_id"
   if (dim(glinides2)[1]>0) {
    glinides2$glinides<-2
   }

    glinides<-merge(glinides,glinides2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("glinides.x" %in% names(glinides)) {
     glinides$glinides<-ifelse(is
.na(glinides$glinides.x),glinides$glinides.y,glinides$glinides.x)
    }


    data5_2_9_5<-merge(data5_2_9_5,glinides,by="patient_id",all.x=TRUE,all.y=FALSE)
#Only people with glinides

    numlevelcat<-na.omit(unique(data5_2_9_5$glinides))

    data5_2_9_5$glinides<-ifelse(is.na(data5_2_9_5$glinides),99,data5_2_9_5$glinides)

    data5_2_9_5<-data5_2_9_5[data5_2_9_5$type_dm %in% c("2"),]

    data5_2_9_5<-data5_2_9_5[,-pmatch(c("glinides.x","glinides.y"
),names(data5_2_9_5))]

    data5_2_9_5$agesex<-ifelse(data5_2_9_5$sex=="1",data5_2_9_5$age_c,0)

  BIRO_dframe(dataname="data5_2_9_5",monitor=TRUE,filelog="")

      BIRO_report(ind="5_2_9_5",
                  title="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Glinides",
                  var_cat="glinides",
                  class=c("sex","age2_c"),
                  lev_var_cat=names(levoral_th12),
                  lev_class=list(a=levsex,b=classlabellist(ageth)),
                  lab_var_cat="Glinides",
                  lab_class=c("Gender","Age"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  bar=1,
                  lines=0,
                  printlines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="glinides",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                  sucvalue=1,
                  per=100,
                  caption="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Glinides")
  rm(data5_2_9_5)
  rm(patient)
  rm(episode)
  rm(glinides,glinides2)

   } else BIRO_report_toc(title="5.2.9.5 Type of oral therapy (distribution of
agents) in patients with diabetes type 2: Glinides")
  }
 } else for (thisyear in anayears) BIRO_report_toc(title="5.2.9.5 Type of oral
therapy (distribution of agents) in patients with diabetes type 2: Glinides")



  #5.2.10 % subjects treated with insulin


  if (varepi$drug_therapy==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)
```

```
    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if(dim(episode[is.na(episode$drug_therapy)==FALSE  & episode$epi_date >= oneyearbf
& episode$epi_date <= checkdate,])[1]>0) {

  #noNAepirefyear<-episode[is.na(episode$drug_therapy)==FALSE  & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,]

    data5_2_10<-patient[,c("patient_id","age_c","age2_c","sex","type_dm")]
    ins<-as.data.frame(unique(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy %in% c(1,3),"patient_id"])) #ins only and tablets
    names(ins)<-"patient_id"
    if (dim(ins)[1]>0) {
     ins$ins<-1
    }

    data5_2_10<-merge(data5_2_10,ins,by="patient_id",all.x=TRUE)

    data5_2_10$ins<-ifelse(is.na(data5_2_10$ins),2,data5_2_10$ins)

        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_10<-merge(data5_2_10,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_10)[pmatch("sub_ds_id",names(data5_2_10))]<-"centre_id"

    data5_2_10$agesex<-ifelse(data5_2_10$sex=="1",data5_2_10$age_c,0)

    data5_2_10<-data5_2_10[data5_2_10$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_2_10",monitor=TRUE,filelog="")

      BIRO_report(ind="5_2_10",
                  title="% subjects treated with insulin",
                  var_cat="ins",
                  class=c("age2_c","sex","type_dm"),
                  lev_var_cat=names(levinsulin12),
                  lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                  lab_var_cat="Insulin",
                  lab_class=c("Age","Gender","Type of Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  bar=1,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="ins",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                  sucvalue=1,
                  per=100,
                  caption="% subjects treated with insulin")

  rm(data5_2_10)
  rm(patient)
  rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.10 % subjects treated with insulin")
  }
   }
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.10 % subjects treated with insulin")
   }
  }

  #5.2.11 Portion of patients treated with insulin in combination with OADs among
patients with diabetes

  if ((varepi$drug_therapy==TRUE) & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
```

```
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

   if(dim(episode[is.na(episode$oral_therapy)==FALSE  & episode$epi_date >= oneyearbf
& episode$epi_date <= checkdate,])[1]>0) {

    data5_2_11<-patient[,c("patient_id","age_c","age2_c","sex","type_dm")]
    insOAD<-as.data.frame(unique(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy==3,"patient_id"])) #ins only and tablets
    names(insOAD)<-"patient_id"
    if (dim(insOAD)[1]>0) {
     insOAD$insOAD<-1
    }

    data5_2_11<-merge(data5_2_11,insOAD,by="patient_id",all.x=TRUE)

    data5_2_11$insOAD<-ifelse(is.na(data5_2_11$insOAD),2,data5_2_11$insOAD)

        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_11<-merge(data5_2_11,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_11)[pmatch("sub_ds_id",names(data5_2_11))]<-"centre_id"

    data5_2_11$agesex<-ifelse(data5_2_11$sex=="1",data5_2_11$age_c,0)

    data5_2_11<-data5_2_11[data5_2_11$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_2_11",monitor=TRUE,filelog="")

       BIRO_report(ind="5_2_11",
                   title="Portion of patients treated with insulin in combination
with OADs among patients with diabetes",
                   var_cat="insOAD",
                   class=c("age2_c","sex","type_dm"),
                   lev_var_cat=names(levinsOAD),
                   lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                   lab_var_cat="Insulin and OAD",
                   lab_class=c("Age","Gender","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="insOAD",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
                   caption="Portion of patients treated with insulin in combination
with OADs among patients with diabetes")

  rm(data5_2_11)
  rm(patient)
  rm(episode)

   } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.2.11 Portion of patients treated with insulin in
combination with OADs among patients with diabetes")
   }
   }
} else {
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.2.11 Portion of patients treated with insulin in
combination with OADs among patients with diabetes")
  }
 }

  #5.2.12 Percentage of insulin treated persons with diabetes with insulin pump
therapy

 if ((varepi$drug_therapy==TRUE) & (varepi$pump_therapy==TRUE) & varpat$dob==TRUE &
    varpat$sex==TRUE &
```

```
    varpat$type_dm==TRUE ) {


   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if(dim(episode[is.na(episode$drug_therapy)==FALSE  & episode$drug_therapy %in%
c(1,3) & is.na(episode$pump_therapy)==FALSE,])[1]>0) {

  instreat<-as.data.frame(unique(episode[is.na(episode$drug_therapy)==FALSE &
episode$drug_therapy %in% c(1,3),"patient_id"])) #ins only and tablets
  names(instreat)<-"patient_id"
  pump<-as.data.frame(unique(episode[is.na(episode$pump_therapy)==FALSE &
episode$pump_therapy ==1,"patient_id"])) #ins only and tablets
  names(pump)<-"patient_id"
  if (dim(pump)[1]>0) {
   pump$pump<-1
  }

  pump2<-as.data.frame(unique(episode[is.na(episode$pump_therapy)==FALSE &
episode$pump_therapy ==0,"patient_id"])) #ins only and tablets
  names(pump2)<-"patient_id"
  if (dim(pump2)[1]>0) {
   pump2$pump<-2
  }

    pump<-merge(pump,pump2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("pump.x" %in% names(pump)) {
     pump$pump<-ifelse(is.na(pump$pump.x),pump$pump.y,pump$pump.x)
    }

  data5_2_12<-merge(instreat,pump,by="patient_id",all.x=TRUE)

  data5_2_12<-merge(data5_2_12,patient,by="patient_id",all.x=FALSE,all.y=FALSE)

  numlevelcat<-na.omit(unique(data5_2_12$pump))
  data5_2_12$pump<-ifelse(is.na(data5_2_12$pump),99,data5_2_12$pump)
  lev_var_cat=names(levinsulin12)[sort(numlevelcat)]


       if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_12<-merge(data5_2_12,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_12)[pmatch("sub_ds_id",names(data5_2_12))]<-"centre_id"

  data5_2_12$agesex<-ifelse(data5_2_12$sex=="1",data5_2_12$age_c,0)

  data5_2_12<-data5_2_12[data5_2_12$type_dm %in% c("1","2"),]
  if ("pump.x" %in% names(data5_2_12)) {
   data5_2_12<-data5_2_12[,-pmatch(c("pump.x","pump.y"),names(data5_2_12))]
  }

  BIRO_dframe(dataname="data5_2_12",monitor=TRUE,filelog="")

      BIRO_report(ind="5_2_12",
                  title="Percentage of insulin treated persons with diabetes with
insulin pump therapy",
                  var_cat="pump",
                  class=c("age2_c","sex","type_dm"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                  lab_var_cat="Pump Therapy",lab_class=c("Age","Gender","Type of
Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  bar=1,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  box=0,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="pump",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
```

```
                  sucvalue=1,
                  per=100,
                  caption="Percentage of insulin treated persons with diabetes with
insulin pump therapy")

  rm(data5_2_12)
  rm(patient)
  rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.2.12 Percentage of insulin treated persons with diabetes
with insulin pump therapy")
  }
   }
} else {
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.2.12 Percentage of insulin treated persons with diabetes
with insulin pump therapy")
  }
 }

  #5.2.13 Portion of diabetes patients with anti hypertensive treatment

 if ((varepi$hypert_med==TRUE) & (varepi$hypertension==TRUE) & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if(dim(episode[is.na(episode$hypert_med)==FALSE  & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,])[1]>0) {

    data5_2_13<-patient[,c("patient_id","age_c","age2_c","sex","type_dm")]
    hypert_med<-as.data.frame(unique(episode[is.na(episode$hypert_med)==FALSE &
episode$hypert_med==1,"patient_id"])) #ins only and tablets
    names(hypert_med)<-"patient_id"
    if (dim(hypert_med)[1]>0) {
     hypert_med$hypert_med<-1
    }

    hypert_med2<-as.data.frame(unique(episode[is.na(episode$hypert_med)==FALSE &
episode$hypert_med==0,"patient_id"])) #ins only and tablets
    names(hypert_med2)<-"patient_id"
    if (dim(hypert_med2)[1]>0) {
     hypert_med2$hypert_med<-2
    }

    hypert_med<-merge(hypert_med,hypert_med2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("hypert_med.x" %in% names(hypert_med)) {
     hypert_med$hypert_med<-ifelse(is
.na(hypert_med$hypert_med.x),hypert_med$hypert_med.y,hypert_med$hypert_med.x)
    }

    data5_2_13<-merge(data5_2_13,hypert_med,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data5_2_13$hypert_med))
    data5_2_13$hypert_med<-ifelse(is
.na(data5_2_13$hypert_med),99,data5_2_13$hypert_med)
    lev_var_cat=names(levinsulin12)[sort(numlevelcat)]

        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_13<-merge(data5_2_13,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_13)[pmatch("sub_ds_id",names(data5_2_13))]<-"centre_id"

    data5_2_13$agesex<-ifelse(data5_2_13$sex=="1",data5_2_13$age_c,0)

    data5_2_13<-data5_2_13[data5_2_13$type_dm %in% c("1","2"),]
    if ("hypert_med.x" %in% names(data5_2_13)) {
     data5_2_13<-data5_2_13[,-pmatch(c("hypert_med.x","hypert_med.y"
),names(data5_2_13))]
    }
  BIRO_dframe(dataname="data5_2_13",monitor=TRUE,filelog="")

      BIRO_report(ind="5_2_13",
                  title="Portion of diabetes patients with anti hypertensive
treatment",
                  var_cat="hypert_med",
                  class=c("age2_c","sex","type_dm"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                  lab_var_cat="Hypertension",
```

```
                    lab_class=c("Age","Gender","Type of Diabetes"),
                    tab=1,
                    tabside="H",
                    tabperc=TRUE,
                    tabwidth=0.95,
                    bar=1,
                    barbeside=TRUE,
                    barnumber=TRUE,
                    barperc=TRUE,
                    box=0,
                    lines=0,
                    trellis=0,
                    pie=0,
                    texfile=texfile,
                    dircsv=dirdataout,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    stand=1,
                    catcol=c(2,3,4),
                    outcome="hypert_med",
                    cov=c("sex","age_c","agesex","type_dm"),
                    chisq=NULL,
                    factcov=c(TRUE,TRUE,TRUE,TRUE),
                    strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                    sucvalue=1,
                    per=100,
                    caption="Portion of diabetes patients with anti hypertensive
treatment")

  rm(data5_2_13)
  rm(patient)
  rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.13 Portion of diabetes patients with anti
hypertensive treatment")
  }
   }
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.13 Portion of diabetes patients with anti
hypertensive treatment")
   }
  }

  # 5.2.14 Portion of diabetes patients with lipid lowering treatment
  #No variable in XML Schema

  if (varepi$lipid_therapy==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if(dim(episode[is.na(episode$lipid_therapy)==FALSE  & episode$epi_date >= oneyearbf
& episode$epi_date <= checkdate,])[1]>0) {

    data5_2_14<-patient[,c("patient_id","age_c","age2_c","sex","type_dm")]
    lipid_med<-as.data.frame(unique(episode[is.na(episode$lipid_therapy)==FALSE &
episode$lipid_therapy==1,"patient_id"])) # ins and tablets
    names(lipid_med)<-"patient_id"
    if (dim(lipid_med)[1]>0) {
     lipid_med$lipid_med<-1
    }

    lipid_med2<-as.data.frame(unique(episode[is.na(episode$lipid_therapy)==FALSE &
episode$lipid_therapy==0,"patient_id"])) # ins and tablets
    names(lipid_med2)<-"patient_id"
    if (dim(lipid_med2)[1]>0) {
     lipid_med2$lipid_med<-2
    }

    lipid_med<-merge(lipid_med,lipid_med2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("lipid_med.x" %in% names(lipid_med)) {
     lipid_med$lipid_med<-ifelse(is
.na(lipid_med$lipid_med.x),lipid_med$lipid_med.y,lipid_med$lipid_med.x)
    }

    data5_2_14<-merge(data5_2_14,lipid_med,by="patient_id",all.x=TRUE)
```

```
    numlevelcat<-na.omit(unique(data5_2_14$lipid_med))
    data5_2_14$lipid_med<-ifelse(is.na(data5_2_14$lipid_med),99,data5_2_14$lipid_med)

    lev_var_cat=names(levlipids)[sort(numlevelcat)]

        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_14<-merge(data5_2_14,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_14)[pmatch("sub_ds_id",names(data5_2_14))]<-"centre_id"

    data5_2_14$agesex<-ifelse(data5_2_14$sex=="1",data5_2_14$age_c,0)

    data5_2_14<-data5_2_14[data5_2_14$type_dm %in% c("1","2"),]
    if ("lipid_med.x" %in% names(data5_2_14)) {
     data5_2_14<-data5_2_14[,-pmatch(c("lipid_med.x","lipid_med.y"
),names(data5_2_14))]
    }
  BIRO_dframe(dataname="data5_2_14",monitor=TRUE,filelog="")

       BIRO_report(ind="5_2_14",
                   title="Portion of diabetes patients with lipid lowering treatment",
                   var_cat="lipid_med",
                   class=c("age2_c","sex","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                   lab_var_cat="Lipid lowering treatment",
                   lab_class=c("Age","Gender","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="lipid_med",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
                   caption="Portion of diabetes patients with anti hypertensive
treatment")

  rm(data5_2_14)
  rm(patient)
  rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.14 Portion of diabetes patients with lipid lowering
treatment")
  }
   }
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.14 Portion of diabetes patients with lipid lowering
treatment")
   }
  }

  # 5.2.15 Portion of diabetes patients with ASA treatment
  #No variable in XML Schema

  if (varepi$antiplatelet_therapy==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
```

```
),envir=.GlobalEnv)

  if(dim(episode[is.na(episode$antiplatelet_therapy)==FALSE,])[1]>0) {

    data5_2_15<-patient[,c("patient_id","age_c","age2_c","sex","type_dm")]
    asa_med<-as.data.frame(unique(episode[is.na(episode$antiplatelet_therapy)==FALSE
& episode$antiplatelet_therapy==1,"patient_id"])) #asa
    names(asa_med)<-"patient_id"
    if (dim(asa_med)[1]>0) {
     asa_med$asa_med<-1
    }

    asa_med2<-as.data.frame(unique(episode[is.na(episode$antiplatelet_therapy)==FALSE
& episode$antiplatelet_therapy==0,"patient_id"])) #asa
    names(asa_med2)<-"patient_id"
    if (dim(asa_med2)[1]>0) {
     asa_med2$asa_med<-2
    }

    asa_med<-merge(asa_med,asa_med2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("asa_med.x" %in% names(asa_med)) {
     asa_med$asa_med<-ifelse(is
.na(asa_med$asa_med.x),asa_med$asa_med.y,asa_med$asa_med.x)
    }

    data5_2_15<-merge(data5_2_15,asa_med,by="patient_id",all.x=TRUE)

    numlevelcat<-na.omit(unique(data5_2_15$asa_med))
    data5_2_15$asa_med<-ifelse(is.na(data5_2_15$asa_med),99,data5_2_15$asa_med)

    lev_var_cat=names(levasa_med)[sort(numlevelcat)]

        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_15<-merge(data5_2_15,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_15)[pmatch("sub_ds_id",names(data5_2_15))]<-"centre_id"

    data5_2_15$agesex<-ifelse(data5_2_15$sex=="1",data5_2_15$age_c,0)

    data5_2_15<-data5_2_15[data5_2_15$type_dm %in% c("1","2"),]
    if ("asa_med.x" %in% names(data5_2_15)) {
     data5_2_15<-data5_2_15[,-pmatch(c("asa_med.x","asa_med.y"),names(data5_2_15))]
    }
  BIRO_dframe(dataname="data5_2_15",monitor=TRUE,filelog="")

       BIRO_report(ind="5_2_15",
                   title="Portion of diabetes patients with ASA treatment",
                   var_cat="asa_med",
                   class=c("age2_c","sex","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                   lab_var_cat="ASA treatment",
                   lab_class=c("Age","Gender","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="asa_med",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
                   caption="Portion of diabetes patients with anti hypertensive
treatment")

  rm(data5_2_15)
  rm(patient)
  rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.15 Portion of diabetes patients with ASA treatment")
  }
   }
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.15 Portion of diabetes patients with ASA treatment")
   }
  }

  # 5.2.16 Percent of patients with diabetes performing selfmonitoring
  # of blood glucose/ urine testing** Svein questions the
  # urine at least in some countries

  if (varepi$self_mon==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)
```

```
    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if(dim(episode[is.na(episode$self_mon)==FALSE  & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,])[1]>0) {

  noNAepirefyear<-episode[is.na(episode$self_mon)==FALSE  & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,]

  data5_2_16_1<-noNAepirefyear[noNAepirefyear$self_mon==1,pmatch(c("patient_id",
"self_mon"),names(noNAepirefyear))]
  names(data5_2_16_1)<-c("patient_id","self_mon1")
  data5_2_16_2<-noNAepirefyear[noNAepirefyear$self_mon==2,pmatch(c("patient_id",
"self_mon"),names(noNAepirefyear))]
  names(data5_2_16_2)<-c("patient_id","self_mon2")
  data5_2_16_3<-noNAepirefyear[noNAepirefyear$self_mon==3,pmatch(c("patient_id",
"self_mon"),names(noNAepirefyear))]
  names(data5_2_16_3)<-c("patient_id","self_mon3")

  data5_2_16_12<-merge(data5_2_16_1,data5_2_16_2,by="patient_id",all=TRUE)
  data5_2_16<-merge(data5_2_16_12,data5_2_16_3,by="patient_id",all=TRUE)

  data5_2_16$self_mon12<-data5_2_16$self_mon3
  data5_2_16$self_mon12<-ifelse((data5_2_16$self_mon1==1 & is
.na(data5_2_16$self_mon12)),1,data5_2_16$self_mon12)
  data5_2_16$self_mon12<-ifelse((data5_2_16$self_mon2==2 & is
.na(data5_2_16$self_mon12)),2,data5_2_16$self_mon12)
  data5_2_16$self_mon12<-ifelse((data5_2_16$self_mon1==1) & (data5_2_16$self_mon2==2)
& is.na(data5_2_16$self_mon12),3,data5_2_16$self_mon12)

  data5_2_16<-aggregate(data5_2_16$self_mon12,list(data5_2_16$patient_id),FUN="max")
  names(data5_2_16)<-c("patient_id","self_mon12")

  data5_2_16<-merge(data5_2_16,patient,
                    by="patient_id",all=TRUE)

  data5_2_16$age=(checkdate-data5_2_16$dob)/365.25
  data5_2_16$age_c=varclass(data5_2_16$age,agethst)

  numlevelcat<-na.omit(unique(data5_2_16$self_mon12))
  data5_2_16$self_mon12<-ifelse(is.na(data5_2_16$self_mon12),99,data5_2_16$self_mon12)

  lev_var_cat=names(levself_mon)[sort(numlevelcat)]

  data5_2_16<-data5_2_16[,c("self_mon12","age_c","age2_c","sex","type_dm")]

        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_2_16<-merge(data5_2_16,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_2_16)[pmatch("sub_ds_id",names(data5_2_16))]<-"centre_id"

    data5_2_16$agesex<-ifelse(data5_2_16$sex=="1",data5_2_16$age_c,0)

  data5_2_16<-data5_2_16[data5_2_16$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_2_16",monitor=TRUE,filelog="")

       BIRO_report(ind="5_2_16",
                   title="Percent of patients with diabetes performing selfmonitoring
of blood glucose/ urine testing",
                   var_cat="self_mon12",
                   class=c("age2_c","sex","type_dm"),
                   lev_var_cat=names(levself_mon),
                   lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dmst),
                   lab_var_cat="selfmon. blood glucose/ urine",
                   lab_class=c("Age","Gender","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="self_mon12",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
                   caption="Portion of diabetes patients with anti hypertensive
treatment")

  rm(data5_2_16)
  rm(patient)
  rm(episode)
```

```
  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.16 Percent of patients with diabetes performing
selfmonitoring of blood glucose/ urine testing")
  }
   }
 } else {
   for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

     BIRO_report_toc(title="5.2.16 Percent of patients with diabetes performing
selfmonitoring of blood glucose/ urine testing")
   }
  }

  #5.3 Outcome quality-intermediate outcomes

  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)

   new_section("5.3. Outcome quality - intermediate outcomes",texfile=texfile)
   new_section_html("5.3. Outcome quality - intermediate outcomes",htmlfile=htmlfile)
  }
  #5.3.1 % subjects with most recent HbA1c >9.0% (poor control)

 if (varepi$hba1c==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  hba1cth9<-9
  episode$hba1c_c9<-ifelse(as.numeric(as.character(episode$hba1c))>9,1,2)

  if (dim(episode[is.na(episode$hba1c_c9)==FALSE,])[1]>0) {

  last_hba1c<-BIRO_aggregate(data=episode[is.na(episode$hba1c)==FALSE,],
                             request=c("patient_id"),
                             fun="max",  #the greatest (1) value is taken
                             vard=c("epi_date"),
                             isdate=TRUE) #Select the last valid measurement

  names(last_hba1c)<-c("patient_id","epi_date")

  data5_3_1<-merge(last_hba1c,episode[,c("patient_id","epi_date","hba1c_c9")],by=c(
"patient_id","epi_date"),all.x=TRUE,all.y=FALSE)
  rm(last_hba1c)

  data5_3_1<-merge(data5_3_1,patient[,c("patient_id","age_c","age2_c","type_dm","sex"
)],by=c("patient_id"),all.x=TRUE,all.y=TRUE)

  data5_3_1<-data5_3_1[,c("patient_id","hba1c_c9","age_c","age2_c","sex","type_dm")]


       if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_3_1<-merge(data5_3_1,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_3_1)[pmatch("sub_ds_id",names(data5_3_1))]<-"centre_id"

  numlevelcat<-na.omit(unique(data5_3_1$hba1c_c9))
  data5_3_1$hba1c_c9[is.na(data5_3_1$hba1c_c9)]<-99

  lev_var_cat=c("(9 + )","(0 - 9]")[sort(numlevelcat)]

  data5_3_1<-data5_3_1[data5_3_1$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_3_1",monitor=TRUE,filelog="")

    data5_3_1$agesex<-ifelse(data5_3_1$sex=="1",data5_3_1$age_c,0)

      BIRO_report(ind="5_3_1",
                  title="% subjects with most recent HbA1c > 9.0 pct (poor control)",
                  var=NULL,
                  var_cat="hba1c_c9",
                  class=c("sex","age2_c","type_dm"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dmst),
                  lab_var_cat="HbA1c",
                  lab_class=c("Gender","Age","Type of Diabetes"),
                  tab=1,
```

```
                    tabside="H",
                    tabperc=TRUE,
                    tabwidth=0.95,
                    box=0,
                    bar=1,
                    barbeside=TRUE,
                    barnumber=TRUE,
                    barperc=TRUE,
                    lines=0,
                    trellis=0,
                    pie=0,
                    texfile=texfile,
                    dircsv=dirdataout,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    stand=1,
                    catcol=c(2,3,4),
                    outcome="hba1c_c9",
                    cov=c("sex","age_c","agesex","type_dm"),
                    chisq=NULL,
                    factcov=c(TRUE,TRUE,TRUE,TRUE),
                    strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                    sucvalue=1,
                    per=100,
                    caption="% subjects with most recent HbA1c > 9.0 pct (poor
control)")

  rm(data5_3_1)
  rm(patient)
  rm(episode)
  rm(noNAepirefyear)
  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.3.1 % subjects with most recent HbA1c > 9.0 pct (poor
control)")
  }
   }
} else {
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.3.1 % subjects with most recent HbA1c > 9.0 pct (poor
control)")
  }
 }

  #5.3.2 % subjects with most recent HbA1c >7,5%

 if (varepi$hba1c==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {


   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

  hba1cth75<-7.5
  episode$hba1c_c75<-ifelse(as.numeric(as.character(episode$hba1c))>7.5,1,2)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if (dim(episode[is.na(episode$hba1c_c75)==FALSE,])[1]>0) {

  last_hba1c<-BIRO_aggregate(data=episode[is.na(episode$hba1c)==FALSE,],
                                  request=c("patient_id"),
                                  fun="max",  #the greatest (1) value is taken
                                  vard=c("epi_date"),
                                  isdate=TRUE) #Select the last valid measurement

  names(last_hba1c)<-c("patient_id","epi_date")

  data5_3_2<-merge(last_hba1c,episode[,c("patient_id","epi_date","hba1c_c75")],by=c(
"patient_id","epi_date"),all.x=TRUE,all.y=FALSE)
  rm(last_hba1c)

  data5_3_2<-merge(data5_3_2,patient[,c("patient_id","age_c","age2_c","type_dm","sex"
)],by=c("patient_id"),all.x=TRUE,all.y=TRUE)

  data5_3_2<-data5_3_2[,c("patient_id","hba1c_c75","age_c","age2_c","sex","type_dm")]


       if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_3_2<-merge(data5_3_2,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=TRUE)
  names(data5_3_2)[pmatch("sub_ds_id",names(data5_3_2))]<-"centre_id"
```

```
    data5_3_2$agesex<-ifelse(data5_3_2$sex=="1",data5_3_2$age_c,0)

    numlevelcat<-na.omit(unique(data5_3_2$hba1c_c75))
    data5_3_2$hba1c_c75[is.na(data5_3_2$hba1c_c75)]<-99
    lev_var_cat=c("(7.5 + )","(0 - 7.5]")[sort(numlevelcat)]

  data5_3_2<-data5_3_2[data5_3_2$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_3_2",monitor=TRUE,filelog="")


       BIRO_report(ind="5_3_2",
                   title="% subjects with most recent HbA1c > 7,5 pct",
                   var=NULL,
                   var_cat="hba1c_c75",
                   class=c("sex","age2_c","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dmst),
                   lab_var=NULL,
                   lab_var_cat="HbA1c",
                   lab_class=c("Gender","Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   box=0,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="hba1c_c75",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
                   caption="% subjects with most recent HbA1c > 7,5 pct ")


  rm(data5_3_2)
  rm(patient)
  rm(episode)
  rm(noNAepirefyear)
  } else {
        assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

        BIRO_report_toc(title="5.3.2 % subjects with most recent HbA1c > 7,5 pct")
  }
   }
  } else {
    for (thisyear in anayears) {
         assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

         BIRO_report_toc(title="5.3.2 % subjects with most recent HbA1c > 7,5 pct")
    }
  }

  #5.3.3 % subjects with most recent BP <140/90 mmHg

 if (varepi$sbp==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {


   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

  sbpth140<-140
  episode$sbp_c140<-ifelse(as.numeric(as.character(episode$sbp))<140,1,2)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
        "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if (dim(episode[is.na(episode$sbp_c140)==FALSE,])[1]>0) {

  last_sbp<-BIRO_aggregate(data=episode[is.na(episode$sbp)==FALSE,],
```

```
                              request=c("patient_id"),
                              fun="max",  #the greatest (1) value is taken
                              vard=c("epi_date"),
                              isdate=TRUE) #Select the last valid measurement

  names(last_sbp)<-c("patient_id","epi_date")

  data5_3_3<-merge(last_sbp,episode[,c("patient_id","epi_date","sbp_c140")],by=c(
"patient_id","epi_date"),all.x=TRUE,all.y=FALSE)
  rm(last_sbp)

  data5_3_3<-merge(data5_3_3,patient[,c("patient_id","age_c","age2_c","type_dm","sex"
)],by=c("patient_id"),all.x=TRUE,all.y=TRUE)

  data5_3_3<-data5_3_3[,c("patient_id","sbp_c140","age_c","age2_c","sex","type_dm")]


       if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_3_3<-merge(data5_3_3,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_3_3)[pmatch("sub_ds_id",names(data5_3_3))]<-"centre_id"

  numlevelcat<-na.omit(unique(data5_3_3$sbp_c140))
  data5_3_3$sbp_c140[is.na(data5_3_3$sbp_c140)]<-99

  lev_var_cat=c("[0 - 140)","[140 +)")[sort(numlevelcat)]

    data5_3_3$agesex<-ifelse(data5_3_3$sex=="1",data5_3_3$age_c,0)

  data5_3_3<-data5_3_3[data5_3_3$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_3_3",monitor=TRUE,filelog="")


      BIRO_report(ind="5_3_3",
                  title="% subjects with most recent BP < 140/90 mmHg",
                  var=NULL,
                  var_cat="sbp_c140",
                  class=c("sex","age2_c","type_dm"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dmst),
                  lab_var_cat="SBP",
                  lab_class=c("Gender","Age","Type of Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  box=0,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  bar=1,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="sbp_c140",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                  sucvalue=1,
                  per=100,
                  caption="% subjects with most recent BP < 140/90 mmHg")

  rm(data5_3_3)
  rm(patient)
  rm(episode)
  rm(noNAepirefyear)
  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.3.3 % subjects with most recent BP < 140/90 mmHg")
  }
   }
} else {
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.3.3 % subjects with most recent BP < 140/90 mmHg")
  }
 }


  #5.3.4 % subjects with most recent BMI > 30

if (varepi$bmi==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {


   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

  bmith30<-30
  episode$bmi_c30<-ifelse(episode$bmi>30,1,2)
  #episode$bmi_c30<-ifelse(is.na(episode$bmi_c30),2,episode$bmi_c30)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if (dim(episode[is.na(episode$bmi_c30)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,])[1]>0) {

  last_bmi<-BIRO_aggregate(data=episode[is.na(episode$bmi)==FALSE,],
                           request=c("patient_id"),
                           fun="max",  #the greatest (1) value is taken
                           vard=c("epi_date"),
                           isdate=TRUE) #Select the last valid measurement

  names(last_bmi)<-c("patient_id","epi_date")

  data5_3_4<-merge(last_bmi,episode[,c("patient_id","epi_date","bmi_c30")],by=c(
"patient_id","epi_date"),all.x=TRUE,all.y=FALSE)
  rm(last_bmi)

  data5_3_4<-merge(data5_3_4,patient[,c("patient_id","age_c","age2_c","type_dm","sex"
)],by=c("patient_id"),all.x=TRUE,all.y=TRUE)

  data5_3_4<-data5_3_4[,c("patient_id","bmi_c30","age_c","age2_c","sex","type_dm")]

  numlevelcat<-na.omit(unique(data5_3_4$bmi_c30))
  data5_3_4$bmi_c30[is.na(data5_3_4$bmi_c30)]<-99

  lev_var_cat=c("[30 + )","(0 - 30)")[sort(numlevelcat)]

       if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_3_4<-merge(data5_3_4,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_3_4)[pmatch("sub_ds_id",names(data5_3_4))]<-"centre_id"

    data5_3_4$agesex<-ifelse(data5_3_4$sex=="1",data5_3_4$age_c,0)

  data5_3_4<-data5_3_4[data5_3_4$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_3_4",monitor=TRUE,filelog="")

       BIRO_report(ind="5_3_4",
                   title="% subjects with most recent BMI > 30",
                   var=NULL,
                   var_cat="bmi_c30",
                   class=c("sex","age2_c","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dmst),
                   lab_var_cat="BMI",
                   lab_class=c("Gender","Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   box=0,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="bmi_c30",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
                   caption="% subjects with most recent BMI > 30")

  rm(data5_3_4)
  rm(patient)
  rm(episode)
  rm(noNAepirefyear)
  } else {
    BIRO_report_toc(title="5.3.4 % subjects with most recent BMI > 30")
  }
   }
} else for (thisyear in anayears) BIRO_report_toc(title="5.3.4 % subjects with most
recent BMI > 30")

  #5.3.5 Percentage with microalbuminuria in last 12 months (among those who have
been tested)


if (varepi$ma_test==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {


   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
```

```
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if (dim(episode[is.na(episode$ma_test)==FALSE ,])[1]>0) {

    last_matest<-BIRO_aggregate(data=episode[is.na(episode$ma_test)==FALSE &
episode$ma_test!=0,],
                                request=c("patient_id"),
                                fun="max",  #the greatest (1) value is taken
                                vard=c("epi_date"),
                                isdate=TRUE) #Select the last valid measurement


    names(last_matest)<-c("patient_id","epi_date")

    last_matest<-merge(last_matest,episode[,c("patient_id","epi_date","ma_test"
)],by=c("patient_id","epi_date"),all.x=TRUE,all.y=FALSE)

    #Cohort
    normal<-as.data.frame(unique(last_matest[is.na(last_matest$ma_test)==FALSE &
last_matest$ma_test==1,"patient_id"]))
    names(normal)<-"patient_id"
    if (dim(normal)[1]>0) {
     normal$ma_test<-1
    }

    abnormal<-as.data.frame(unique(last_matest[is.na(last_matest$ma_test)==FALSE &
last_matest$ma_test==2,"patient_id"]))
    names(abnormal)<-"patient_id"
    if (dim(abnormal)[1]>0) {
     abnormal$ma_test<-2
    }

    data5_3_5<-merge(normal,abnormal,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("ma_test.x" %in% names(data5_3_5)) {
     data5_3_5$ma_test<-ifelse(is
.na(data5_3_5$ma_test.x),data5_3_5$ma_test.y,data5_3_5$ma_test.x)
    }

    data5_3_5<-merge(data5_3_5,patient,by="patient_id",all.x=TRUE,all.y=FALSE)
    numlevelcat<-na.omit(unique(data5_3_5$ma_test))

    lev_var_cat=names(levma_test)[c(-0,sort(numlevelcat))]

        if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_3_5<-merge(data5_3_5,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_3_5)[pmatch("sub_ds_id",names(data5_3_5))]<-"centre_id"

    data5_3_5$agesex<-ifelse(data5_3_5$sex=="1",data5_3_5$age_c,0)

    data5_3_5<-data5_3_5[data5_3_5$type_dm %in% c("1","2"),]

    if ("ma_test.x" %in% names(data5_3_5)) {
     data5_3_5<-data5_3_5[,-pmatch(c("ma_test.x","ma_test.y"),names(data5_3_5))]
     }



  BIRO_dframe(dataname="data5_3_5",monitor=TRUE,filelog="")

       BIRO_report(ind="5_3_5",
                   title="Percentage with normal value of recent microalbuminuria in
last 12 months (among those who have been tested)",
                   var=NULL,
                   var_cat="ma_test",
                   class=c("sex","age2_c","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dmst),
                   lab_var_cat="microalbuminuria",
                   lab_class=c("Gender","Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   box=0,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="ma_test",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100,
```

```
                caption="Percentage with microalbuminuria in last 12 months (among
those who have been tested)")

  rm(data5_3_5)
  rm(patient)
  rm(episode)
  rm(noNAepirefyear)
  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.3.5 Percentage with normal value of recent
microalbuminuria in last 12 months (among those who have been tested)")
  }
   }
} else {
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.3.5 Percentage with normal value of recent
microalbuminuria in last 12 months (among those who have been tested)")
  }
 }

  #5.3.6 Rate of current smokers among diabetes patients

if (varepi$smok_stat==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)
```

```
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if (dim(episode[is.na(episode$smok_stat)==FALSE,])[1]>0) {

  last_smoke<-BIRO_aggregate(data=episode[is.na(episode$smok_stat)==FALSE,],
                                  request=c("patient_id"),
                                  fun="max",  #the greatest (1) value is taken
                                  vard=c("epi_date"),
                                  isdate=TRUE) #Select the last valid measurement

  names(last_smoke)<-c("patient_id","epi_date")

  data5_3_6<-merge(last_smoke,episode[,c("patient_id","epi_date","smok_stat")],by=c(
"patient_id","epi_date"),all.x=TRUE,all.y=FALSE)
  rm(last_smoke)

  data5_3_6$current_smoke<-ifelse(data5_3_6$smok_stat==1,1,2)

  data5_3_6<-merge(data5_3_6,patient[,c("patient_id","age_c","age2_c","type_dm","sex"
)],by=c("patient_id"),all.x=FALSE,all.y=TRUE)

  data5_3_6<-data5_3_6[,c("patient_id","current_smoke","age_c","age2_c","sex",
"type_dm")]

  numlevelcat<-na.omit(unique(data5_3_6$current_smoke))

  data5_3_6$current_smoke<-ifelse(is
.na(data5_3_6$current_smoke),99,data5_3_6$current_smoke)
  lev_var_cat=c("current smoker","no current smoker")[sort(numlevelcat)]


       if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_3_6<-merge(data5_3_6,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_3_6)[pmatch("sub_ds_id",names(data5_3_6))]<-"centre_id"

    data5_3_6$agesex<-ifelse(data5_3_6$sex=="1",data5_3_6$age_c,0)

  data5_3_6<-data5_3_6[data5_3_6$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_3_6",monitor=TRUE,filelog="")

      BIRO_report(ind="5_3_6",
                  title="Rate of current smokers among diabetes patients",
                  var=NULL,
                  var_cat="current_smoke",
                  class=c("sex","age2_c","type_dm"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dmst),
                  lab_var_cat="Current Smokers",
                  lab_class=c("Gender","Age","Type of Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  box=0,
                  bar=1,
```

```
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="current_smoke",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                  sucvalue=1,
                  per=100,
                  caption="Rate of current smokers among diabetes patients")

  rm(data5_3_6)
  rm(patient)
  rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.3.6 Rate of current smokers among diabetes patients")
  }
   }
} else {
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.3.6 Rate of current smokers among diabetes patients")
  }
 }

  #5.3.7 Former or current foot ulceration

if (varepi$ulcer==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  if (dim(episode[is.na(episode$ulcer)==FALSE,])[1]>0) {

  last_ulcer<-BIRO_aggregate(data=episode[is.na(episode$ulcer)==FALSE,],
                                request=c("patient_id"),
                                fun="max",  #the greatest (1) value is taken
                                vard=c("epi_date"),
                                isdate=TRUE) #Select the last valid measurement

  names(last_ulcer)<-c("patient_id","epi_date")

  data5_3_7<-merge(last_ulcer,episode[,c("patient_id","epi_date","ulcer")],by=c(
"patient_id","epi_date"),all.x=TRUE,all.y=FALSE)
  rm(last_ulcer)
  data5_3_7$ulcer<-ifelse(data5_3_7$ulcer==0,2,data5_3_7$ulcer)

  data5_3_7<-merge(data5_3_7,patient[,c("patient_id","age_c","age2_c","type_dm","sex"
)],by=c("patient_id"),all.x=TRUE,all.y=TRUE)

  data5_3_7<-data5_3_7[,c("patient_id","ulcer","age_c","age2_c","sex","type_dm")]

  numlevelcat<-na.omit(unique(data5_3_7$ulcer))

  data5_3_7$ulcer<-ifelse(is.na(data5_3_7$ulcer),99,data5_3_7$ulcer)
  lev_var_cat=names(levulcer)[sort(numlevelcat)]

      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_3_7<-merge(data5_3_7,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=FALSE)
  names(data5_3_7)[pmatch("sub_ds_id",names(data5_3_7))]<-"centre_id"

    data5_3_7$agesex<-ifelse(data5_3_7$sex=="1",data5_3_7$age_c,0)

  data5_3_7<-data5_3_7[data5_3_7$type_dm %in% c("1","2"),]

  BIRO_dframe(dataname="data5_3_7",monitor=TRUE,filelog="")
```

```
      BIRO_report(ind="5_3_7",
                  title="Former or current foot ulceration",
                  var=NULL,
                  var_cat="ulcer",
                  class=c("sex","age2_c","type_dm"),
                  lev_var_cat=lev_var_cat,
                  lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dmst),
                  lab_var_cat="foot ulceration",
                  lab_class=c("Gender","Age","Type of Diabetes"),
                  tab=1,
                  tabside="H",
                  tabperc=TRUE,
                  tabwidth=0.95,
                  box=0,
                  bar=1,
                  barbeside=TRUE,
                  barnumber=TRUE,
                  barperc=TRUE,
                  lines=0,
                  trellis=0,
                  pie=0,
                  texfile=texfile,
                  dircsv=dirdataout,
                  dirgraph=dirgraph,
                  dirgraphlatex=dirgraphlatex,
                  cex=cex,
                  stand=1,
                  catcol=c(2,3,4),
                  outcome="ulcer",
                  cov=c("sex","age_c","agesex","type_dm"),
                  chisq=NULL,
                  factcov=c(TRUE,TRUE,TRUE,TRUE),
                  strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                  sucvalue=1,
                  per=100,
                  caption="Former or current foot ulceration")

  rm(data5_3_7)
  rm(patient)
  rm(episode)

  } else {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.3.7 Former or current foot ulceration")
  }
   }
} else {

  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.3.7 Former or current foot ulceration")
  }
 }

  #5.4 Outcome quality-terminal outcomes
```

```
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)
   assign("htmlfile",paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""
),envir=.GlobalEnv)

   new_section("5.4. Outcome quality - terminal outcomes",texfile=texfile)
   new_section_html("5.4. Outcome quality - terminal outcomes",htmlfile=htmlfile)
  }
#5.4.1 Annual incidence of dialysis and/or transplantation (renal
#      replacement therapy in patients with diabetes


if (varepi$dialysis==TRUE | varepi$transplant==TRUE & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

   for (thisyear in anayears) {

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

  diatrans<-as.data.frame(unique(episode[(episode$dialysis==1 |
episode$transplant==1),"patient_id"]))
  names(diatrans)<-"patient_id"

    if (dim(diatrans)[1]>0) {
```

```
     diatrans$diatrans<-1
    }
     diatrans2<-as.data.frame(unique(episode[(episode$dialysis==0 &
episode$transplant==0),"patient_id"]))
     names(diatrans2)<-"patient_id"
     if (dim(diatrans2)[1]>0) {
      diatrans2$diatrans<-2
     }

    diatrans<-merge(diatrans,diatrans2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("diatrans.x" %in% names(diatrans)) {
     diatrans$diatrans<-ifelse(is
.na(diatrans$diatrans.x),diatrans$diatrans.y,diatrans$diatrans.x)
    }


    data5_4_1<-merge(diatrans,patient,
                     by="patient_id",all.y=TRUE)

    data5_4_1$age=(checkdate-data5_4_1$dob)/365.25
    data5_4_1$age_c=varclass(data5_4_1$age,agethst)
    data5_4_1<-data5_4_1[,c("patient_id","type_dm","sex","age_c","age2_c","diatrans")]


    numlevelcat<-na.omit(unique(data5_4_1$diatrans))

    data5_4_1$diatrans[is.na(data5_4_1$diatrans)]<-99
    lev_var_cat=names(levdiatrans)[sort(numlevelcat)]


      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_4_1<-merge(data5_4_1,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=TRUE)
  names(data5_4_1)[pmatch("sub_ds_id",names(data5_4_1))]<-"centre_id"

    data5_4_1$agesex<-ifelse(data5_4_1$sex=="1",data5_4_1$age_c,0)

  data5_4_1<-data5_4_1[data5_4_1$type_dm %in% c("1","2"),]
  if ("diatrans.x" %in% names(data5_4_1)) {
   data5_4_1<-data5_4_1[,-pmatch(c("diatrans.x"),names(data5_4_1))]
  }
  if ("diatrans.y" %in% names(data5_4_1)) {
   data5_4_1<-data5_4_1[,-pmatch(c("diatrans.y"),names(data5_4_1))]
  }


       BIRO_report(ind="5_4_1",
                   title="Annual incidence of dialysis and/or transplantation (renal
replacement therapy in patients with diabetes",
                   var=NULL,
                   var_cat="diatrans",
                   class=c("sex","age2_c","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dmst),
                   lab_var_cat="dialysis and/or transplantation",
                   lab_class=c("Gender","Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   box=0,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="diatrans",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100000,
                   caption="Annual incidence of dialysis and/or transplantation
(renal replacement therapy in patients with diabetes")



   rm(data5_4_1)
  rm(patient)
  rm(episode)


 }
} else {
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.4.1 Annual incidence of dialysis and/or transplantation
(renal replacement therapy in patients with diabetes")
  }
 }


#5.4.2 ESRF in Persons with Diabetes

 if (varepi$esrf==1 & varpat$dob==TRUE &
    varpat$sex==TRUE &
    varpat$type_dm==TRUE ) {

 for (thisyear in anayears) {

   assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("dirgraph",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)

   assign("dirtables",paste(dirout,"/output/reports/", "#",launchtime,
         "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)

   assign("dirhtml",paste(dirout,"/output/reports/", "#",launchtime,
```

```
          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)

   assign("dirwp",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,"/wp",sep=""),envir=.GlobalEnv)

   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

   assign("texfile",paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""
),envir=.GlobalEnv)

    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")


partname<-paste(thisyear,substr(as.character(refanadate),4,5),substr(as.character(refa
nadate),1,2),sep="")
    ewyear<-paste("episode_wide_",partname,sep="")
    patyear<-paste("profile_wide_",partname,sep="")
    patient<-get(patyear)
    episode<-get(ewyear)

    patient$age2_c<-varclass(patient$age,ageth)

    patient$age_c<-varclass(patient$age,agethst)

  esrf<-as.data.frame(unique(episode[episode$esrf==1  & is.na(episode$esrf)==FALSE,
"patient_id"]))
  names(esrf)<-"patient_id"
    if (dim(esrf)[1]>0) {
     esrf$esrf<-1
    }

     esrf2<-as.data.frame(unique(episode[episode$esrf==0  & is
.na(episode$esrf)==FALSE,"patient_id"]))
     names(esrf2)<-"patient_id"
     if (dim(esrf2)[1]>0) {
      esrf2$esrf<-2
     }

    esrf<-merge(esrf,esrf2,by="patient_id",all.x=TRUE,all.y=TRUE)
    if ("esrf.x" %in% names(esrf)) {
     esrf$esrf<-ifelse(is.na(esrf$esrf.x),esrf$esrf.y,esrf$esrf.x)
    }

  data5_4_2<-merge(esrf,patient,
                   by="patient_id",all.y=TRUE)

  data5_4_2$age=(checkdate-data5_4_2$dob)/365.25
  data5_4_2$age_c=varclass(data5_4_2$age,agethst)
  data5_4_2<-data5_4_2[,c("patient_id","type_dm","sex","age_c","age2_c","esrf")]


  numlevelcat<-na.omit(unique(data5_4_2$esrf))

  data5_4_2$esrf[is.na(data5_4_2$esrf)]<-99
  lev_var_cat=names(levesrf)[sort(numlevelcat)]


      if (is.na(pmatch("sub_ds_id",names(patient)))==FALSE)
data5_4_2<-merge(data5_4_2,patient[,c("patient_id","sub_ds_id")],by="patient_id"
,all.x=TRUE,all.y=TRUE)
   names(data5_4_2)[pmatch("sub_ds_id",names(data5_4_2))]<-"centre_id"

    data5_4_2$agesex<-ifelse(data5_4_2$sex=="1",data5_4_2$age_c,0)

  data5_4_2<-data5_4_2[data5_4_2$type_dm %in% c("1","2"),]
  if ("diatrans.x" %in% names(data5_4_2)) {
   data5_4_2<-data5_4_2[,-pmatch(c("esrf.x","esrf.y"),names(data5_4_2))]
  }
       BIRO_report(ind="5_4_2",
                   title="ESRF in Persons with Diabetes",
                   var=NULL,
                   var_cat="esrf",
                   class=c("sex","age2_c","type_dm"),
                   lev_var_cat=lev_var_cat,
                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dmst),
                   lab_var_cat="esrf",
                   lab_class=c("Gender","Age","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=0.95,
                   box=0,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="esrf",
                   cov=c("sex","age_c","agesex","type_dm"),
                   chisq=NULL,
                   factcov=c(TRUE,TRUE,TRUE,TRUE),
                   strata="centre_id",

levstd=list(a=levsex,b=classlabellist(agethst),c=levagesex,d=levtype_dmst),
                   sucvalue=1,
                   per=100000,
                   caption="ESRD in Persons with Diabetes")



  rm(data5_4_2)
  rm(patient)
  rm(episode)

 }
} else {
  for (thisyear in anayears) {
   assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,
          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

    BIRO_report_toc(title="5.4.2 ESRF in Persons with Diabetes")
  }
}
```

```
#if (dim(spider_data)[1]>1) {
#
#   new_section("Radar Plots",texfile=texfile)
#   new_section_html("Radar Plots",htmlfile=htmlfile)
#
#
# row.names(spider_data)<-spider_data$Strata
# spider_data<-spider_data[,-pmatch("Strata",names(spider_data))]
#
# spider_done<-0
# step=6
# while  (spider_done<dim(spider_data)[2]) {
#  last<-min(dim(spider_data)[2],(spider_done+step))

#  file = paste(dirgraph,"/web_risk_adj.svg",sep="")
#  CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
#  stars(spider_data[,(spider_done+1):last],key.loc = c(6,
2),full=TRUE,cex=cex,draw.segments = TRUE,
#        frame.plot=FALSE,len = 0.6)

#
legend("center",legend=names(spider_data[,(spider_done+1):last]),cex=.7,col=rainbow(le
ngth(names(spider_data[,(spider_done+1):last]))),
#         pch=1)
#  dev.off()

#  file = paste(dirgraph,"/web_risk_adj.jpeg",sep="")
#  jpeg(file, width = 11.33, height =4.84 , units="in",pointsize = 9,
#      bg = "transparent",res=9)
#  stars(spider_data[,(spider_done+1):last],key.loc = c(6,
2),full=TRUE,cex=cex,draw.segments = TRUE,
#        frame.plot=FALSE,len = 0.6)

#
legend("center",legend=names(spider_data[,(spider_done+1):last]),cex=.7,col=rainbow(le
ngth(names(spider_data[,(spider_done+1):last]))),
#         pch=1)
#  dev.off()

#  file = paste(dirgraph,"/web_risk_adj.pdf",sep="")
#  Cairopdf(file, width= 813.6, height= 348.48, pointsize=8, bg = "transparent")
#  stars(spider_data[,(spider_done+1):last],key.loc = c(6,
2),full=TRUE,cex=cex,draw.segments = TRUE,
#        frame.plot=FALSE,len = 0.6)

#
legend("center",legend=names(spider_data[,(spider_done+1):last]),cex=cex,col=rainbow(l
ength(names(spider_data[,(spider_done+1):last]))),
#         pch=1)
#  dev.off()

#  file = paste(dirgraph,"/web_risk_adj.pdf",sep="")
#  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
#  stars(spider_data[,(spider_done+1):last],key.loc = c(6,
2),full=TRUE,cex=cex,draw.segments = TRUE,
#        frame.plot=FALSE,len = 0.6)

#
legend("center",legend=names(spider_data[,(spider_done+1):last]),cex=cex,col=rainbow(l
ength(names(spider_data[,(spider_done+1):last]))),
#         pch=1)
#  dev.off()
#  spider_done<-spider_done+step

#    import_pdf(namegraph=paste("web_risk_adj.pdf",sep=""),
#               caption=paste("Webplot: Risk Adjusted Indicators",sep=""),
#               width=0.95,
#               dirgraph=dirgraphlatex,
#               texfile=texfile)

#    import_pdf_html(image=paste("../graphs/web_risk_adj.pdf",sep=""),
#                    htmlfile=paste(dirhtml,"/webplots.html",sep=""),
#                    caption=paste("Webplot: Risk Adjusted Indicators",sep=""))

# }

#}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_boxplots.r                                                        #
#                                                                              #
#
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# biro_location.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_boxplot
#   BIRO_plotbox
#
# ##############################################################################

# ##############################################################################
#  BIRO_boxplot                                                                #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_boxplot draws boxplot for a conditionated distribution
#
# DEPENDENCIES
#
#  BIRO_out     (BIRO_dispersion.r)
#  BIRO_plotbox
#
# ##############################################################################
# PARAMETERS(!=required)
#
# data     !               = dataset containing data
# var      !               = the name of the target variable
# lev      !               = a list containing the vactor with che levels of
#                            the categorical variable (list)
# stratum                  = the name of the stratum variables
# out     ["out"]          = the name of the outliers csv files
# notch    !               = logical. if notch is TRUE, a notch is drawn in each
#                            side of the boxes. If the notches of two plots do
#                            not overlap this is ï¿½strong evidenceï¿½ that the two
#                            means differ (Chambers et al., 1983, p. 62).
#                            See boxplot.stats for the calculations used.
# ext     [csv]            = the extension of the output files
# date    [System date]    = date of analysis
# dirout     [current]     = the path of the output text files
# namegraph !              = the coded name of the output files
# dirgraph   [current]     = the path of the output (jpg,eps,pdf files)
# ##############################################################################

BIRO_boxplot<-function(data,
                       var,
                       lev=as.list(),
                       stratum=NULL,
                       out="out",
                       notch,
                       dirout=getwd(),
                       ext="csv",
                       date=Sys.Date(),
                       namegraph,
                       dirgraph,
                       nameso) {


 depend=""
 if (length(stratum)>0) {
  depend="~"
 }

# write("Preparing to do boxp",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do boxp",file=logfile,append=TRUE)
# }
 boxp<-boxplot(as.formula(paste(var,depend,paste(stratum,collapse="+"))),
               notch = notch,
               boxfill= rainbow(5),
               frame= FALSE,
               outl= TRUE,
               pch=4,
               outline=TRUE,
               plot=FALSE,
               data=data)
```

```
#write("boxp done",file="")
#if (nchar(logfile)>0) {
# write("boxp done",file=logfile,append=TRUE)
#}

assign("boxp",boxp,envir = .GlobalEnv)
longnames<-unlist(strsplit(as.character(boxp$names[boxp$n>=1]), ".", fixed = TRUE))
stats<-t(as.data.frame((boxp$stats[,boxp$n>=1])))
conf<-t(as.data.frame((boxp$conf[,boxp$n>=1])))
oldn<-boxp$n
n<-as.data.frame((boxp$n[boxp$n>=1]))
longnames<-as.data.frame(matrix(longnames,ncol=length(stratum),byrow=TRUE))
names(longnames)<-stratum
if (dim(stats)[1]>dim(longnames)[1]) {
 stats<-na.omit(stats)
}
 else if (dim(stats)[1]==dim(longnames)[1]) {
 boxp<-cbind(na.omit(stats),na.omit(longnames))
}
 else if (dim(stats)[1]<dim(longnames)[1]) {
 for (i in 1:dim(stats)[1]) {
  eq=0
  for (j in 2:dim(stats)[2]) {
  first<-stats[i,1]
  if (first==stats[i,j]) eq+1
  }
  if (eq==dim(stats)[1]-1) {stats<-stats[-i]}
 }
 boxp<-cbind(na.omit(stats),na.omit(longnames))
}
else boxp<-cbind(stats,longnames)

names(boxp)<-c(c("l_wisk","pcl_25x","median","pcl_75x","u_wisk"),stratum)
assign("conf",conf,envir = .GlobalEnv)
assign("oldn",oldn,envir = .GlobalEnv)
assign("longnames",longnames,envir = .GlobalEnv)

if (dim(conf)[1]!=dim(longnames)[1]) {
 conf<-cbind(na.omit(conf[oldn>=1,]),na.omit(longnames))
} else conf<-cbind(conf,longnames)
names(conf)<-c(c("l_notch","u_notch"),stratum)
if (dim(n)[1]!=dim(longnames)[1]) {
 n<-cbind(na.omit(n[n>0]),na.omit(longnames))
} else n<-cbind(n,longnames)
names(n)<-c("n",stratum)

if (dim(n[n>0,][1])==dim(longnames)[1]) {
 outliers<-BIRO_out(data=data,
                    var=var,
                    stratum=stratum,
                    out="temp",
                    dirout=getwd(),
                    ext="csv",
                    date=Sys.Date())
} else outliers<-NULL
if (is.null(outliers)==FALSE) {
# file.remove(paste(getwd(),"//","out.csv",sep=""))
 names(outliers)[pmatch("value",names(outliers))]<-"outlie_x"
 write("Boxplot statistics calculated",file="")
 if (nchar(logfile)>0) {
  write("Boxplot statistics calculated",file=logfile,append=TRUE)
 }
}
INDEX<-list()
for (i in 1:length(stratum)) {
 INDEX[[i]]<-data[,pmatch(stratum[i],names(data))]
}
means<-aggregate(data[,pmatch(var,names(data))],INDEX,FUN="median")
names(means)<-c(stratum,"mean")
if (is.null(outliers)==FALSE) {
 bxptab<-merge(means,outliers,by=stratum,all.x=TRUE)
} else {
 bxptab<-means
 bxptab$outlie_x<-NA #these are no outliers so we puts NAs instead of nothing
 bxptab$variable<-var
 bxptab$Freq<-0
}
assign("bxptab", bxptab,envir = .GlobalEnv)
assign("boxp", boxp,envir = .GlobalEnv)
bxptab<-merge(bxptab,boxp,by=stratum,all.x=TRUE)
bxptab<-merge(bxptab,conf,by=stratum,all.x=TRUE)
bxptab<-merge(bxptab,n,by=stratum,all.x=TRUE)
bxp<-as.data.frame(bxptab)
bxptab$variable=var
bxptab$codist=centre_id
bxptab$start=oneyearbf
bxptab$end=checkdate
bxptab$dbname=dbname
write.csv(bxptab,paste(dirdataout,"/i",tolower(nameso),"d4_3a.csv",sep=""),
          row.names = FALSE)

write(paste("Boxplot data saved into: [OUTDATA]","/i",tolower(nameso),"d4_3a.csv"
,sep=""),file="")
if (nchar(logfile)>0) {
 write(paste("Boxplot data saved into: [OUTDATA]","/i",tolower(nameso),"d4_3a.csv"
,sep=""),file=logfile,append=TRUE)
}

BIRO_plotbox(bxpdata=bxptab,
             lev=lev,
             notch=notch,
             namegraph=namegraph,
             dirgraph=dirgraph,
             out=out,
             n=n,
             variable=variable,
             stratum=stratum)
}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_dispersion.r                                                      #
#                                                                              #
#
# ##############################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_dispersion.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_range
#   BIRO_out
#   BIRO_variance
#
##############################################################################


# ##############################################################################
#  BIRO_range                                                                  #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_range calculates range for Conditionated Frequency Distribution
#
# DEPENDENCIES
#
#  R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# idcentro    !             = the id of the regional repository
# date        [System date] = date of analysis
# data        !             = the dataset processed containing data
# var         !             = the name of the stratum variable
# stratum     !             = the name of the stratum variable
# codtab      !             =
# dircsv      !             = the path of the output (csv files)
# ##############################################################################

BIRO_range<-function(idcentro,
                     date=Sys.Date(),
                     data,
                     var,
                     stratum=NULL,
                     codtab,
                     dircsv) {

 onlythese<-c(var,stratum)
 data<-na.omit(data[,pmatch(onlythese,names(data))])

 retdata<-NULL
 for (k in 1:length(var)) {
  spltdata<-
  split(data[,pmatch(var,names(data))],data[,pmatch(stratum,names(data))])

  newdata<-NULL
  for (i in 1:(length(names(spltdata)))) {
   temp<-as.data.frame(range(as.vector(spltdata[[i]])))
   temp$stat=c("min","max")
   names(temp)<-"value"
   temp$variable=var
   temp$stratum<-names(spltdata)[i]
   names(temp)[pmatch("stratum",names(temp))]<-stratum
   temp$idcentro=idcentro
   temp$date=date

   newdata<-rbind(newdata,temp)
  }

 retdata<-rbind(retdata,newdata)
 }
 write.csv(as.data.frame(retdata),paste(dircsv,"/i",codtab,"d2_3a.csv",sep=""),
           row.names = FALSE)
 return(retdata)
}

# ##############################################################################
#  BIRO_out                                                                    #
```

```r
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_out calculates outliers for Conditionated Frequency Distribution
#
# DEPENDENCIES
#
#  R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# data      !              = input dataframe
# var       !              = one or more target variable corresponding to frame
#                            column names
# stratum   !              = one or more strata variable corresponding to frame
#                            column names (including centre id)
#                            all variables preceding n are strata variables
# dirout    [getwd()]      = output location
# out       [out]          = name of output csv dataset
# ext       [csv]          = output file format
# date      [Sys.Date()]   = date variable or value
# ##############################################################################

BIRO_out<-function(data,
                   var,
                   stratum=NULL,
                   out="out",
                   dirout=getwd(),
                   ext="csv",
                   date=Sys.Date()) {

 # initializes the return data.frame
 retdata<-NULL

 # for each variables (this con do more then a variable each time)
 #
 # IMPORTANT: they must be analyzed with the same stratum
 for (k in 1:length(var)) {
  # for each variables (this con do more then a variable each time)

  # is stratum are spedified
  if (is.null(stratum)==FALSE) {
   boxp<-
   boxplot(as.formula(paste(var[k],"~",paste(stratum,collapse="+"))),
           plot=FALSE,data=data)
   longnames<-unlist(strsplit(as.character(boxp$names), ".", fixed = TRUE))
   group<-1:length(boxp$names)
   #if there aren't outliers
   if (length(boxp$out)>0 && length(boxp$out)>0) {
    boxp<-as.data.frame(table(as.vector(boxp$out),as.vector(boxp$group)))
   }
   #if there aren't outliers
   if (length(boxp$out)==0) {
    boxp<-as.data.frame(cbind(NA,NA,NA))
    names(boxp)<-c("Var1","Var2","Freq")
    boxp<-na.omit(boxp)
   }
   st<-NULL
   for (i in 1:length(stratum)) {
    st[[i]]<-longnames[seq(from=i,to=length(longnames),by=length(stratum))]
   }
   ns<-as.data.frame(st)
   names(ns)<-stratum
   nsd<-cbind(na.omit(group),na.omit(ns))
  }
  # if stratum are not spedified
  # do simply the boxplot of the variable
  if (is.null(stratum)==TRUE) {
   boxp<-boxplot(as.vector(data[,pmatch(var[k],names(data))]))
   boxp<-as.data.frame(table(as.vector(boxp$out)))
  }
  # if there aren't outliers
  if (dim(boxp)[1]>0) {
   boxp<-boxp[boxp$Freq>0,]
  }
  # if stratum are specified
  # prepare data.frame for the merge in this way
  if (is.null(stratum)==FALSE) {
   names(boxp)<-c("value","group","Freq")
   boxp<-merge(boxp,nsd,by="group")
   boxp<-boxp[,pmatch(c("value","Freq",stratum),names(boxp))]
  }
  # if stratum are not specified
  # prepare data.frame for the merge in this way
  if (is.null(stratum)==TRUE) {
   names(boxp)<-c("value","Freq")
  }
  # if there aren't outliers
  if (dim(boxp)[1]>0) {
   boxp$variable<-var[k]
  }
  #append the data.frame to the return data.frame
  retdata<-rbind(retdata,as.data.frame(boxp))
 }
 # if there are outliers for some variables
 if (dim(retdata)[1]>0) {
  retdata$date=date
  write.csv(as.data.frame(retdata),paste(dirout,"/i",out,"d1_2a.",ext,sep=""),
            row.names = FALSE)
  return(retdata)
 }

 #if there aren't outliers for noone variables
 if (dim(retdata)[1]==0) {
   return(NULL)
 }

}

# ##############################################################################
#  BIRO_out                                                                    #
# ##############################################################################
# ------------------------
```

```
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_out calculates outliers for Conditionated Frequency Distribution
#
# DEPENDENCIES
#
#  R package: base
#
# ###############################################################################
# PARAMETERS(!=required)
#
# data     !               = input dataframe
# var      !               = one or more target variable corresponding to frame
#                            column names
#                            stats for more variables are appended, a variable
#                            name column is automatically produced
# stratum  !               = one or more strata variable corresponding to frame
#                            column names (including centre id)
#                            all variables preceding n are strata variables
# dirout    [getwd()]      = output location
# out       [var]         <= name of output csv dataset
# ext       [csv]          = output file format
# date      [Sys.Date()]   = date variable or value
# ###############################################################################

BIRO_variance<-function(data,
                        var,
                        stratum=NULL,
                        out="out",
                        dirout=getwd(),
                        ext="csv",
                        date=Sys.Date()) {

 myvar=var
 rm(var)
 retdata<-NULL
 for (k in 1:length(myvar)) {
  data<-data[is.na(data[,pmatch(c(myvar[k],stratum),names(data))])==FALSE,]
  tab<-NULL
  if (is.null(stratum)==FALSE) {
   data<-na.omit(data[,pmatch(c(myvar,stratum),names(data))])
   INDEX<-list()
   for (i in 1:length(stratum)) {
    INDEX[[i]]<-data[,pmatch(stratum[i],names(data))]
   }
   tabn<-
     as.data.frame(aggregate(data[,pmatch(myvar[k],names(data))],INDEX,length))
   tabsum<-
     as.data.frame(aggregate(data[,pmatch(myvar[k],names(data))],INDEX,sum))
   tabvar<-
     as.data.frame(aggregate(data[,pmatch(myvar[k],names(data))],INDEX,var))
   for (j in 1:length(stratum)) {
    names(tabsum)[j]<-stratum[j]
    names(tabn)[j]<-stratum[j]
    names(tabvar)[j]<-stratum[j]
   }
   names(tabsum)[length(stratum)+1]<-"sum"
   names(tabn)[length(stratum)+1]<-"n"
   names(tabvar)[length(stratum)+1]<-"var"
   tab<-merge(tabn,tabsum,by=stratum)
   tab<-merge(tab,tabvar,by=stratum)
  }
  if (is.null(stratum)==TRUE) {
   tab$mean<-mean(na.omit(data[,pmatch(myvar,names(data))]))
   tab$n<-length(na.omit(data[,pmatch(myvar,names(data))]))
   tab$var<-var(na.omit(data[,pmatch(myvar,names(data))]))
  }
  tab$variable=myvar[k]
  retdata<-rbind(retdata,as.data.frame(tab))
 }
 retdata$date=date
 write("Variance Data Frame Calculated",file="",sep="\n")
 write.csv(as.data.frame(retdata),paste(dirout,"/i",out,".3_1a",ext,sep=""),
           row.names = FALSE)
 write("Variance Data Frame Stored",file="",sep="\n")
 return(retdata)
}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: biro_histograms.r                                                      #
#                                                                              #
#
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# biro_histograms.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_barplot
#   BIRO_drawbars
#   BIRO_pie
#
#
# ##############################################################################

# ##############################################################################
#  BIRO_barplot                                                                #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_barplot prepares data for BIRO_drawbars
#
# DEPENDENCIES
#
#  R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# data      !              = dataset containing data
# var       !              = two or more vector with the values of the target
#                            variables, passed as a list (from the second there
#                            are stratum variables)
# labelvar  !              = the labels of the variable
# lev       !              = a list containing the vactor with che levels of the
#                            categorical variable (list)
# beside     [TRUE]        = a logical value. If FALSE, the columns of height
#                            are portrayed as stacked bars,
#                            and if TRUE the columns are portrayed as juxtaposed
#                            bars.
# number     [FALSE]       = a logical value. If TRUE values are printed over
#                            each bar, FALSE otherwise.
# perc       [TRUE]        = a logical value. If TRUE percentages are calculated.
# what       ["patients"]  = the subject of the representation (this is for the
#                            label)
# dirgraph  !              = the path of the output (jpg,eps,pdf files)
# namegraph !              = the coded name of the output files
# cex       !              = A numerical value giving the amount by which
#                            plotting text and symbols should be magnified
#                            relative to the default. Note that some graphics
#                            functions such as plot.default have an argument of
#                            this name which multiplies this graphical parameter,
#                            and some functions such as points accept a vector
#                            of values which are recycled. Other uses will take
#                            just the first value if a vector of length greater
#                            than one is supplied.
# ##############################################################################


BIRO_barplot<-function(data,
                       var,
                       labelvar,
                       lev=list(),
                       beside=TRUE,
                       number=FALSE,
                       perc=TRUE,
                       what="patients",
                       namegraph,
                       dirgraph,
                       cex,
                       nameso) {


# calculates the number of each group of bars (conbination of stratum variable)

 count=0
 nrows=1
 varlist<-list()
```

```
 for (i in 1:length(var)) {
  count=count+1
  varlist[[count]]<-as.factor(data[,pmatch(var[count],names(data))])
  levels(varlist[[count]])<-lev[[count]]
   if (count>1) nrows<-nrows*nlevels(varlist[[count]])
 }

 # calculates the contingency table (for each combination)

 tab<-table(varlist)
 tab<-as.data.frame(tab)

 names(tab)<-c(var,"Freq")

 # create the splitvar (a variable that contains a different
 # value for each group of bars)

 splitvar<-rep("",dim(tab)[1])
 for (i in 1:(dim(tab)[1])) {
  for (j in 2:(dim(tab)[2]-1)) {
   splitvar[i]<-paste(splitvar[i],as.character(tab[i,j]),sep=":")
  }
  splitvar[i]<-substr(splitvar[i],2,nchar(splitvar[i]))
 }

 # Calculates sums of frequency for each group of bars (useful for calculating
 # percentages)
 sums<-aggregate(tab$Freq,
                 list(splitvar = splitvar),
                 sum)

 names(sums)[2]<-"sum"
 sums$splitvar<-as.character(sums$splitvar)
 tab<-cbind(tab,splitvar)

 #puts the data in the same data.frame
 tab<-merge(tab,sums,by.x = "splitvar", by.y = "splitvar")
 tab<-as.data.frame(tab)
 tab$id=centre_id

 tab<-tab[,2:dim(tab)[2]]
 names(tab)[pmatch("Freq",names(tab))]<-"n"
 tab$start=oneyearbf
 tab$end=checkdate
 tab$dbname=dbname

 #writecsv
 write.csv(tab,paste(dirdataout,"/i",tolower(nameso),"d4_1a.csv",sep=""),
           row.names = FALSE)

 tab<-tab[,-pmatch(c("start","end","dbname"),names(tab))]
 write(paste("Barplot data saved into: [OUTDATA]","/i",tolower(nameso),"d4_1a.csv"
,sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("Barplot data saved into: [OUTDATA]","/i",tolower(nameso),"d4_1a.csv"
,sep=""),file=logfile,append=TRUE)
 }

 #Draw graph
 #print(tab)
 BIRO_drawbars(tab=tab,
               labelvar=labelvar,
               lev=lev,
               beside=beside,
               number=number,
               perc=perc,
               what=what,
               namegraph,
               dirgraph,
               cex=cex)

}

# ##############################################################################
# BIRO_pie
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_pie draws Pie Plots
#
# DEPENDENCIES
#
#  R packages: grDevice, graphics, Cairo
#
# ##############################################################################
# PARAMETERS(!=required)
#
# formula        !          = a formula with <contitionated variable> ~
#                             <contitionating variables> (separated by +)
# data           !          = a data frame containing data
# labelvar       !          = the label of the variable
# labelstratum   !          = a vector containing string values (the labels
#                             of the conditionating variables)
# lev_var_cat   [NULL]      = a list containing levels of the conditionated
#                             variable (list)
# levclass       !          = a list containing the levels of the contitionating
#                             variables (list)
# dirgraph       !          = the path of the graphical output (jpg,eps,pdf files)
# namegraph      !          = the coded name of the output files
# dirdataout     !          = the path of the output csv file
# cex            !          = A numerical value giving the amount by which
#                             plotting text and symbols should be magnified
#                             relative to the default. Note that some graphics
#                             functions such as plot.default have an argument of
#                             this name which multiplies this graphical
#                             parameter, and some functions such as points accept
#                             a vector of values which are recycled.
#                             Other uses will take just the first value if a
#                             vector of length greater than one is supplied.
# ##############################################################################

BIRO_pie<-function(formula=as.formula(),
                   data,
                   labelvar,
```

```
                  labelstratum,
                  lev_var_cat=NULL,
                  lev_class,
                  dirgraph,
                  namegraph,
                  dirdataout=dirdataout,
                  cex) {

 temp<-data[,pmatch(all.vars(formula),names(data))]
 names(temp)<-all.vars(formula)

 for (i in 1:(length(all.vars(formula))-1)) {
  Freqs<-as.data.frame(table(temp[,1],temp[,1+i]))
  nlevels<-nlevels(as.factor(Freqs[,2]))
  levels(Freqs[,2])<-lev_class
#  write("Preparing to do SVG file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do SVG file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/",namegraph,".svg",sep="")
  CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
  par("mfrow"=c(1,nlevels))
  par(cex=cex)

  split.data<-split(Freqs,Freqs[2])
  for (i in (1:length(split.data))) {
   pie(split.data[i][[1]]$Freq,col=rainbow(length(lev_var_cat)),
       cex=cex,labels="")
   mtext(names(lev_class[i]),cex=cex)
  }
  legend("topright",col=rainbow(length(lev_var_cat)),pch=1,cex=cex,
         title=labelvar,legend=lev_var_cat)
  dev.off()

  write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

#  write("Preparing to do JPEG file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do JPEG file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/",namegraph,".jpeg",sep="")
  jpeg(file, width= 813.6, height= 348.48, units="px",pointsize = 9,
       bg = "transparent")

  par("mfrow"=c(1,nlevels))
  par(cex=cex)

  split.data<-split(Freqs,Freqs[2])

  for (i in (1:length(split.data))) {
   pie(split.data[i][[1]]$Freq,col=rainbow(length(lev_var_cat)),
       cex=cex,labels="")
   mtext(names(lev_class[i]),cex=cex)

  }
  legend("topright",col=rainbow(length(lev_var_cat)),pch=1,cex=cex,
         title=labelvar,legend=lev_var_cat)
  dev.off()

  write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

 # write("Preparing to do pdf file",file="")
 # if (nchar(logfile)>0) {
 #  write("Preparing to do pdf file",file=logfile,append=TRUE)
 # }

  file = paste(dirgraph,"/",namegraph,".png",sep="")
  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
  par("mfrow"=c(1,nlevels))
  par(cex=cex)

  split.data<-split(Freqs,Freqs[2])

  for (i in (1:length(split.data))) {
   pie(split.data[i][[1]]$Freq,col=rainbow(length(lev_var_cat)),
       cex=cex,labels="")
   mtext(names(lev_class[i]),cex=cex)

  }

  legend("topright",col=rainbow(length(lev_var_cat)),pch=1,cex=cex,
         title=labelvar,legend=lev_var_cat)

  dev.off()

  write(paste("pdf file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("pdf file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

 # write("Preparing to do PDF file",file="")
 # if (nchar(logfile)>0) {
 #  write("Preparing to do PDF file",file=logfile,append=TRUE)
 # }

  file = paste(dirgraph,"/",namegraph,".pdf",sep="")

  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
  par("mfrow"=c(1,nlevels))
  par(cex=cex)

  split.data<-split(Freqs,Freqs[2])

  for (i in (1:length(split.data))) {
   pie(split.data[i][[1]]$Freq,col=rainbow(length(lev_var_cat)),
       cex=cex,labels="")
   mtext(names(lev_class[i]),cex=cex)

  }

  legend("topright",col=rainbow(length(lev_var_cat)),pch=1,cex=cex,
         title=labelvar,legend=lev_var_cat)
```

```
  dev.off()

  write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

 }


 names(split.data)<-c(all.vars(formula))
 write.csv(split.data,paste(dirdataout,"/",namegraph,".csv",sep=""),
           row.names = FALSE)

 write(paste("Pieplot data saved into: ",dirdataout,"/",namegraph,".csv",sep=""),file=
"")
 if (nchar(logfile)>0) {
  write(paste("Pieplot data saved into: ", dirdataout,"/",namegraph,".csv",sep=""
),file=logfile,append=TRUE)
 }

 if (struc==TRUE) {
  cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
  cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
  cat("  <tr>",file=sthtml,fill=TRUE,append=TRUE)
  cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/"
,namegraph,".pdf\">","</td>",sep=""),file=sthtml,fill=TRUE,append=TRUE)
  cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
  cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
  cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
 }

 return(split.data)

}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_location.r                                                        #
#                                                                              #
#
# ##############################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_location.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_mean
#
# ##############################################################################

# ##############################################################################
#  BIRO_mean                                                                   #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_mean calculates conditionated mean
#
# DEPENDENCIES
#
#  R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# data    !               = input dataframe
# var     !               = one or more target variable corresponding to frame
#                           column names
#                           stats for more variables are appended, a variable
#                           name column is automatically produced
# stratum   [NULL]        = one or more strata variable corresponding to frame
#                           column names (including centre id)
#                           all variables preceding n are strata variables
# dirout    [getwd()]     = output location
# out       [mean]        = name of output csv dataset
# ext       [csv]         = output file format
# date      [Sys.Date()]  = date variable or value
# ##############################################################################

BIRO_mean<-function(data,
                    var,
                    stratum=NULL,
                    dirout=getwd(),
                    out="2_1a",
                    ext="csv",
                    date=Sys.Date()) {

 retdata<-NULL

 for (k in 1:length(var)) {
  tab<-NULL
  if (is.null(stratum)==FALSE) {
   data<-na.omit(data[,pmatch(c(var,stratum),names(data))])
   INDEX<-list()
   for (i in 1:length(stratum)) {
    INDEX[[i]]<-data[,pmatch(stratum[i],names(data))]
   }
   tabn<-as.data.frame(aggregate(data[,pmatch(var[k],names(data))],INDEX,length))
   tabsum<-as.data.frame(aggregate(data[,pmatch(var[k],names(data))],INDEX,sum))
   for (j in 1:length(stratum)) {
    names(tabsum)[j]<-stratum[j]
    names(tabn)[j]<-stratum[j]
   }
   names(tabsum)[length(stratum)+1]<-"sum"
   names(tabn)[length(stratum)+1]<-"n"
   tab<-merge(tabn,tabsum,by=stratum)
  }
  if (is.null(stratum)==TRUE) {
   tab$sum<-sum(na.omit(data[,pmatch(var[k],names(data))]))
   tab$n<-length(na.omit(data[,pmatch(var[k],names(data))]))
  }
  tab$variable=var[k]
  retdata<-rbind(retdata,as.data.frame(tab))
 }
 retdata$date=date
 write("Mean Data Frame Calculated",file="",sep="\n")
```

```
 retdata$start=oneyearbf
 retdata$end=checkdate

 write.csv(as.data.frame(retdata),paste(dirout,"/",out,".",ext,sep=""),
           row.names = FALSE)
 write("Mean Data Frame Stored",file="",sep="\n")
 return(as.data.frame(retdata))
}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: biro_se_report.r                                                       #
#                                                                              #
#
# ##############################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# biro_se_report.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_report
#   open_tex
#   new_chapter
#   new_section
#   new_sub_section
#   import_pdf
#   close_tex
#
# ##############################################################################


# ##############################################################################
#  BIRO_report                                                                 #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_report manages the graphical outputs and tables for each indicator
#
# DEPENDENCIES
#
#  R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# ind    !                = string. the index of the indicator
# var                     = string. the name of the countinous conditionated
#                           variable
# var_cat     !           = string. the name of the categorical conditionated
#                           variable
# class                   = a vector containing the names of the conditionating
#                           variables
# lev_var_cat             = levels of the categorical conditionated variable
# lev_class               = levels of the categorical conditionating variable
# lab_var                 = a list for the labels of the categorical conditionated
#                           variable
# lab_var_cat             = a list for the labels of the categorical conditionating
#                           variable
# tab                     = a boolean (0,1) 1 if output table is required
# chisq                   = a vector containing the name of variable to calculate
#                           Cochran-Mantel-Haenszel Chi-Squared Test for Count
#                           Data or Pearson Chi-Squared Test
# tabside                 = "H" for orizontal, "V" for vertical
# tabperc                 = logical. TRUE if you want percentages in the tables
# tabwidth                = the width of table in the PDF document
# bar                     = a boolean (0,1) 1 if output barplot is required
# barbeside     [TRUE]    = a logical value. If FALSE, the columns of height
#                           are portrayed as stacked bars,
#                           and if TRUE the columns are portrayed as juxtaposed
#                           bars.
# barnumber     [FALSE]   = a logical value. If TRUE values are printed over
#                           each bar, FALSE otherwise.
#
# barperc       [TRUE]    = a logical value. If TRUE percentages are calculated
#                           in the barplot
# box                     = a boolean (0,1) 1 if output boxplot is required
# notch      !            = logical. if notch is TRUE, a notch is drawn in
#                           each side of the boxes. If the notches of two
#                           plots do not overlap this is ï¿½strong evidenceï¿½ that
#                           the two means differ (Chambers et al., 1983, p. 62).
#                           See boxplot.stats for the calculations used.
# lines                   = a boolean (0,1) 1 if output trend is required
# printlines              = logical. TRUE if you want to print
# varlines               = the name of
# timelines
# pie                     = a boolean (0,1) 1 if output pieplot is required
# cex                     = expansion factor for words and number printed
# struct                  = if you want to create the structure of HTML pages
# catcol                  = the number of categorical variables in the data.frame
```

```
#                            that you want to print
# stand                    = if you want to call BIRO_standardize function
# outcome                  = the name of outcome variable (for BIRO_standardize)
# cov                      = a vector containing the names of covariates
# factcov
# levstd                   = a list contaning levels of depvar and indepvar
# strata                   = the name of the variable of stratification
# factindep                = a vector containig logistic values TRUE if the i-th
independent
#                            variable is a factor, FALSE otherwise
# per                      = the moltiplication factor of rates
# pop                      = the data.frame containing the population data
# caption                  =  caption to print after the table (BIRO_standardize)
# printrate                = logical. TRUE if you want to print rate
# patmap                   = a boolean (0,1) 1 if output map is required
# shapefile                = the name of the SpatialPolygonsDataFrame
# mapvar                   = the variable that must be represented
# maplev                   = the levels of mapvar
# maplab                   = the label of mapvar
# mapstat                  = only 2 values "freq" for Absolute Frequencies
# shapearea_id             = the name of area_id variable in the
SpatialPolygonsDataFrame
# valuesarea_id            = the name of area_id variable in the data data.frame
# texfile                  = the path of the texfile
# dirtables                = the path of HTML tables directory
# dircsv                   = the path of output .csv directory
# dirgraph                 = the path of output graphs directory
# dirgraphlatex            = the relative path of output graphs directory (for Latex)

# ##############################################################################

BIRO_report<-function(ind,
                      title="title",
                      var=NULL,
                      var_cat=NULL,
                      class=NULL,
                      lev_var_cat=NULL,
                      lev_class=NULL,
                      lab_var=NULL,
                      lab_var_cat=NULL,
                      lab_class=NULL,
                      tab=1,
                      chisq=NULL,
                      tabside="H",
                      tabperc=TRUE,
                      tabwidth=0,
                      barbeside=TRUE,
                      barnumber=TRUE,
                      barperc=TRUE,
                      box=0,
                      bar=0,
                      what="patients",
                      lines=0,
                      printlines=1,
                      trellis=0,
                      pie=0,
                      varlines=0,
                      timelines=0,
                      notch=FALSE,
                      texfile=texfile,
                      dircsv=dirdataout,
                      dirgraph=dirgraph,
                      dirgraphlatex=dirgraphlatex,
                      cex=1,
                      struc=TRUE,
                      catcol=1,
                      stand=0,
                      outcome=NULL,
                      cov=NULL,
                      factcov=NULL,
                      levstd,
                      strata=NULL,
                      sucvalue=1,
                      per=100,
                      pop=NULL,
                      caption="",
                      printrate=TRUE,
                      patmap=0,
                      shapefile,
                      mapvar,
                      maplev,
                      maplab,
                      mapstat="freq",
                      shapearea_id,
                      valuesarea_id) {


tabjump<-length(na.omit(class))
if (length(class)==3) {
 if (is.na(class[3])==FALSE) tabjump=tabjump-1
 assign("tabjump",tabjump,envir = .GlobalEnv)
}
indprint<-gsub("_",".",as.character(ind))
assign("ind",ind,envir = .GlobalEnv)
assign("indprint",indprint,envir = .GlobalEnv)

captionstd<-title
title<-paste(indprint, title)

write(paste("Processing BIRO INDICATOR:",title,sep=" "),
      file="")

titlelatex<-gsub("\\%","\\\\%",as.character(title))
titlelatex<-gsub("<","$<$",as.character(titlelatex))
titlelatex<-gsub(">","$>$",as.character(titlelatex))


if (is.null(chisq)==FALSE){
 chisq<-na.omit(class[1:2])
}



#print title in .tex file
new_sub_section(title=title,texfile=texfile)

#create html indicator
file.copy(paste(dirse,"/source/html/","layout_open.html",sep=""),
          paste(dirhtml,"/",ind,".html",sep=""),
          overwrite=TRUE)

cat("<body>",
    file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
cat("<table class=\"reportheader\" border=\"0\">",
```

```
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("<tbody>",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <tr>",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("  <td><img src=\"../images/biro-logo01.jpg\"></td>",
      file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" </tr>",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("</tbody>",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("</table>",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)

 cat("<table class=\"reportheader\" border=\"0\">",file=paste(dirhtml,"/",ind,".html"
,sep=""),fill = TRUE,append=TRUE)
 cat("<tbody>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <td class=\"reportheader\"><a href=\"",paste("../",dbname,"_",thisyear,".html"
,sep="") ,"\"> BIRO Report </a> site: ",centre_id,", ",
     format(as.Date(launchtime,format="%d%m%y%H%M%S"),format="%d %B %Y"),
     "</td>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("  <td class=\"reportheader\">Reference date: ",format(checkdate,format=
"%d/%m/%y"),"</td>",file=paste(dirhtml,"/",ind,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("  <td class=\"reportheader\">Parameter: ",title,"</td>",file=paste(dirhtml,"/"
,ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("</tbody>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("</table>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)


 # insert link in report.html
 cat("<a href=\" html/",ind,".html\">",title,"</a><br>",sep="",
     file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill =
TRUE,append=TRUE)
 if (struc==TRUE) {
  dirstruc<-paste(dirout,"/output/reports/structure",sep="")
  dir.create(dirstruc,showWarnings = FALSE)
  sthtml<-paste(dirstruc,"/",ind,".html",sep="")

  cat("",file=sthtml,fill = FALSE,append=FALSE)

  assign("struc",struc,envir=.GlobalEnv)
  assign("sthtml",sthtml,envir=.GlobalEnv)
 }

 # Load Coorte data

 namedata<-paste("data",ind,sep="")

 if (lines==1) {
  #data for line plot
  namedatal<-paste("data",ind,"l",sep="")
  datal<-get(namedatal)
 }

 data<-get(namedata)
 progtab<-0
 progbar<-0
 progfor<-0


#if (stand==0) {
 if (is.null(var_cat)==FALSE) {
  if (tab==1) {
   #cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
   #cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)


#  write(paste("Preparing To Do Frequency Table...Data Frame: ",ind,sep=""),
#        file="",sep="\n")
  # creating table
  # creating input parameters
  firstvar<-substr(var_cat,1,nchar(var_cat)-2)
  tabvars<-c(firstvar,class)
  lev<- vector("list",(length(class)+1))
  for (i in 1:(length(class)+1)) {
   if (i==1) {
    lev[[1]]<-lev_var_cat
   }
   if (i>1) {
    lev[[i]]<-lev_class[[i-1]]
   }
  }
  names(lev)<-letters[1:(length(class)+1)]

  if (length(class)<3) {
    tabcaption=paste(" - ",lab_var_cat," * ",paste(na.omit(lab_class[1:2]),collapse="
* "),
                     sep="")
  } else {
    tabcaption=paste(" - ",lab_var_cat," * "
,paste(na.omit(lab_class[-length(lab_class)]),collapse=" * "),
                     sep="")
  }
  if (tabside=="H") side=0
  if (tabside=="V") side=1

  nclass<-length(na.omit(class))
  #if (nclass==3) nclass=2

  data$noNA<-ifelse(data[,pmatch(var_cat,names(data))]==99,2,1)
  levNA<-c("Valid Value","NV/NA")
  levmiss<-lev
  levmiss[[1]]<-levNA

  missclass<-rep(NA,length(na.omit(class)))
  for (w in 1:length(na.omit(class))) {
   if (is.na(class[w])) w=w+1

    noNAvar<-as.data.frame(ifelse(is.na(data[,pmatch(class[w],names(data))]),2,1))
    names(noNAvar)<-paste("noNA",class[w],sep="")
    missclass[w]<-paste("noNA",class[w],sep="")
    data<-cbind(data,noNAvar)
    levmiss[[w+1]]<-c("Valid Value","NV/NA")


  }
```

```
  if (length(class)==3) {
   missclass[3]<-"type_dm"
   levmiss[[4]]<-lev[[4]]
  }


  if (length(class)==3) {
   for (k in c(3,1:(nclass-1))) {


   if (is.na(class[k])) k=k+1
  progtab=progtab+1
  assign("progtab",progtab,envir = .GlobalEnv)

  levf2f<-list()
  levf2f[[1]]<-lev_var_cat
  levf2f[[2]]<-lev[[k+1]]
  if (is.na(class[k])) levf2f[[2]]<-lev[[k+2]]

  levmissf2f<-list()
  levmissf2f[[1]]<-levmiss[[1]]
  levmissf2f[[2]]<-levmiss[[k+1]]
  if (is.na(class[k])) levmissf2f[[2]]<-levmiss[[k+2]]
  if (missclass[k]=="type_dm") {
     missclass[k]<-"noNAtype_dm"
     levmissf2f[[2]]<-levmiss[[2]]
  }
  if (is.na(lab_class[k])==FALSE) {

  tabcaption2=paste(" - ",lab_var_cat," * ",paste(na.omit(lab_class)[k],collapse=" * "
),
                    sep="")

  tabcaption2m=paste(" - Missing Data: ",lab_var_cat," * "
,paste(na.omit(lab_class)[k],collapse=" * "),
                    sep="")
  } else {
  tabcaption2=paste(" - ",lab_var_cat," * ",paste(na.omit(lab_class)[k-1],collapse="
* "),
                    sep="")

  tabcaption2m=paste(" - Missing Data: ",lab_var_cat," * "
,paste(na.omit(lab_class)[k-1],collapse=" * "),
                    sep="")
  }

#  if (k<3) {

  #progtab type_dm=1

  thislab_class<-lab_class[k]
  if (is.na(thislab_class)) thislab_class<-lab_class[k-1]

  BIRO_table_compute(data=data,
                     var=c("noNA",missclass[k]),
                     FirstLab=lab_var_cat,
                     lev=levmissf2f,
                     labclass=thislab_class,
                     texfile=texfile,
                     codtab=paste(ind,"_",missclass[k],"_",sep=""),
                     dirtables=dirtables,
                     dircsv=dirdataout,
                     caption=tabcaption2m,
                     side=side,
                     width=tabwidth,
                     perc=tabperc,
                     ind=ind,
                     chisq=NULL,
                     progtab=progtab,
                     codtex=paste(ind,"__",letters[k],sep=""))


  progtab=progtab+1
  assign("progtab",progtab,envir = .GlobalEnv)

  if (length(class)<3) {
    tabcaption=paste(" - ",lab_var_cat," * ",paste(na.omit(lab_class[1:2]),collapse="
* "),
                     sep="")
  } else {
    tabcaption=paste(" - ",lab_var_cat," * "
,paste(na.omit(lab_class[-length(lab_class)]),collapse=" * "),
                     sep="")
  }


BIRO_table_compute(data=data[data[,pmatch(var_cat,names(data))]!=99,pmatch(c(var_cat,
class[k]),names(data))],
                     var=c(var_cat,class[k]),
                     FirstLab=lab_var_cat,
                     lev=levf2f,
                     labclass=thislab_class,
                     texfile=texfile,
                     codtab=paste(ind,"_",class[k],"_",sep=""),
                     dirtables=dirtables,
                     dircsv=dirdataout,
                     caption=tabcaption2,
                     side=side,
                     width=tabwidth,
                     perc=tabperc,
                     ind=ind,
                     chisq=class[k],
                     progtab=progtab,
                     codtex=paste(ind,"__",letters[k],sep=""))

#    }
   }
  }

  if (length(class)>1) {
  jumpprog=0
  for (k in 1:nclass) {


    if (is.na(class[k])) k=k+1

  levf2f<-list()
  levf2f[[1]]<-lev_var_cat
  levf2f[[2]]<-lev[[k+1]]

  levmissf2f<-list()
  levmissf2f[[1]]<-levmiss[[1]]
  levmissf2f[[2]]<-levmiss[[k+1]]
```

```
  if (length(class)<3) {


  if (is.na(lab_class[k])==FALSE) {
  tabcaption2=paste(" - ",lab_var_cat," * ",paste(na.omit(lab_class)[k],collapse=" * "
),
                    sep="")

  tabcaption2m=paste(" - Missing Data: ",lab_var_cat," * "
,paste(na.omit(lab_class)[k],collapse=" * "),
                    sep="")
  } else {
  tabcaption2=paste(" - ",lab_var_cat," * ",paste(na.omit(lab_class)[k-1],collapse="
* "),
                    sep="")

  tabcaption2m=paste(" - Missing Data: ",lab_var_cat," * "
,paste(na.omit(lab_class)[k-1],collapse=" * "),
                    sep="")
  }


  progtab=progtab+1
  assign("progtab",progtab,envir = .GlobalEnv)

  BIRO_table_compute(data=data,
                     var=c("noNA",missclass[k]),
                     FirstLab=lab_var_cat,
                     lev=levmissf2f,
                     labclass=lab_class[k],
                     texfile=texfile,
                     codtab=paste(ind,"_",missclass[k],"_",sep=""),
                     dirtables=dirtables,
                     dircsv=dirdataout,
                     caption=tabcaption2m,
                     side=side,
                     width=tabwidth,
                     perc=tabperc,
                     ind=ind,
                     chisq=NULL,
                     progtab=progtab,
                     codtex=paste(ind,"__",letters[k],sep=""))



  progtab=progtab+1
  assign("progtab",progtab,envir = .GlobalEnv)


BIRO_table_compute(data=data[data[,pmatch(var_cat,names(data))]!=99,pmatch(c(var_cat,
class[k]),names(data))],
                     var=c(var_cat,class[k]),
                     FirstLab=lab_var_cat,
                     lev=levf2f,
                     labclass=lab_class[k],
                     texfile=texfile,
                     codtab=paste(ind,"_",class[k],"_",sep=""),
                     dirtables=dirtables,
                     dircsv=dirdataout,
                     caption=tabcaption2,
                     side=side,
                     width=tabwidth,
                     perc=tabperc,
                     ind=ind,
                     chisq=class[k],
                     progtab=progtab,
                     codtex=paste(ind,"__",letters[k],sep=""))


  }

  if (length(class)==3) {

  levf2fc<-levf2f
  levf2fc[[3]]<-NA
  levf2fc[[4]]<-lev[[4]]

  levmissf2fc<-levmissf2f
  levmissf2fc[[3]]<-NA
  levmissf2fc[[4]]<-levmiss[[4]]

  if (length(na.omit(class))==3 & k<3) {
  progtab=progtab+1
  assign("progtab",progtab,envir = .GlobalEnv)
  assign("jumpprog",jumpprog,envir = .GlobalEnv)

  tabcaption3=paste(" - ",lab_var_cat," * ",paste(lab_class[k],collapse=","),
                    sep="")

  tabcaption3m=paste(" - Missing Data: ",lab_var_cat," * "
,paste(lab_class[k],collapse=","),
                    sep="")

  values<-na.omit(unique(data[,pmatch(class[3],names(data))]))
     if (class[3]=="type_dm") {

      allvalues=c("Type 1", "Type 2" ,"Other Type")
      values<-allvalues[sort(values)]
     }
  for (x in 1:length(values)) {

     valueprint<-paste(unlist(strsplit(as.character(values[x])," ")),collapse="_")
     dir.create(paste(dirreport,"/pdf/temp",sep=""), showWarnings = FALSE, recursive
= TRUE)
     thistexfile<-paste(dirreport,"/pdf/temp/table",ind,"_",valueprint,".tex",sep="")

     if (k > 1) {
      cat("\\newpage",file=thistexfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=thistexfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"\\\\ \\bf ",lab_class[length(lab_class)],"="
,values[x],"}",file=thistexfile,fill = TRUE,append=TRUE)
     }
  }

  BIRO_table_compute(data=data[,pmatch(c("noNA",missclass[k],class[3]),names(data))],
                     var=c("noNA",missclass[k],NA,class[3]),
                     FirstLab=lab_var_cat,
                     lev=levmissf2fc,
                     labclass=lab_class[c(k,3)],
                     texfile=texfile,
                     codtab=paste(ind,"_",missclass[k],"_",class[3],"_",sep=""),
                     dirtables=dirtables,
                     dircsv=dirdataout,
```

```
                    caption=tabcaption3m,
                    side=side,
                    width=tabwidth,
                    perc=tabperc,
                    ind=ind,
                    chisq=NULL,
                    progtab=progtab,
                    codtex=paste(ind,"__",letters[k],letters[3],sep=""))


  progtab=progtab+1
  assign("progtab",progtab,envir = .GlobalEnv)
  if (length(class)<3) {
    tabcaption=paste(" - ",lab_var_cat," * ",paste(na.omit(lab_class[1:2]),collapse="
* "),
                    sep="")
  } else {
    tabcaption=paste(" - ",lab_var_cat," * "
,paste(na.omit(lab_class[-length(lab_class)]),collapse=" * "),
                    sep="")
  }


BIRO_table_compute(data=data[data[,pmatch(var_cat,names(data))]!=99,pmatch(c(var_cat,
class[k],class[3]),names(data))],
                    var=c(var_cat,class[k],NA,class[3]),
                    FirstLab=lab_var_cat,
                    labclass=lab_class[c(k,3)],
                    lev=levf2fc,
                    texfile=texfile,
                    codtab=paste(ind,"_",class[k],"_",class[3],"_",sep=""),
                    dirtables=dirtables,
                    dircsv=dirdataout,
                    caption=tabcaption3,
                    side=side,
                    width=tabwidth,
                    perc=tabperc,
                    ind=ind,
                    chisq=c(class[k]),
                    progtab=progtab,
                    codtex=paste(ind,"__",letters[k],letters[3],sep=""))


  }


  #progtab=progtab+1



  }

  }

  }
  progtab=progtab+1
  assign("progtab",progtab,envir = .GlobalEnv)

  if (length(class)<3) {
    tabcaption=paste(" - ",lab_var_cat," * ",paste(na.omit(lab_class[1:2]),collapse="
* "),
                    sep="")
```

```
    tabcaptionm=paste(" - Missing Data: ",lab_var_cat," * "
,paste(na.omit(lab_class[1:2]),collapse=" * "),
                      sep="")

  } else {
    tabcaption=paste(" - ",lab_var_cat," * "
,paste(na.omit(lab_class[-length(lab_class)]),collapse=" * "),
                    sep="")

    tabcaptionm=paste(" - Missing Data: ",lab_var_cat," * "
,paste(na.omit(lab_class[-length(lab_class)]),collapse=" * "),
                      sep="")

  }

  if (length(class)==3) {

  values<-na.omit(unique(data[,pmatch(class[3],names(data))]))
     if (class[3]=="type_dm") {
      allvalues=c("Type 1", "Type 2" ,"Other Type")
      values<-allvalues[sort(values)]
      missclass[3]<-"type_dm"

     }

  for (x in 1:length(unique(data[,pmatch(class[3],names(data))]))) {

     valueprint<-paste(unlist(strsplit(as.character(values[x])," ")),collapse="_")

     dir.create(paste(dirreport,"/pdf/temp",sep=""), showWarnings = FALSE, recursive
= TRUE)
     thistexfile<-paste(dirreport,"/pdf/temp/table",ind,"_",valueprint,".tex",sep="")

      cat("\\newpage",file=thistexfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=thistexfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=thistexfile,fill = TRUE,append=TRUE)
      cat("\\headsep 0.50in",file=thistexfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"\\\\ \\bf",lab_class[length(lab_class)],"="
,values[x],"}",file=thistexfile,fill = TRUE,append=TRUE)
  }
  }


  BIRO_table_compute(data=data,
                     var=c("noNA",missclass),
                     FirstLab=lab_var_cat,
                     lev=levmiss,
                     labclass=lab_class,
                     texfile=texfile,
                     codtab=paste(ind,"m","_",sep=""),
                     dirtables=dirtables,
                     dircsv=dirdataout,
                     caption=tabcaptionm,
                     side=side,
                     width=tabwidth,
                     perc=tabperc,
                     ind=ind,
                     chisq=NULL,
                     progtab=progtab,
                     codtex=ind)
```

```
  progtab=progtab+1
  assign("progtab",progtab,envir = .GlobalEnv)
  if (length(class)<3) {
    tabcaption=paste(" - ",lab_var_cat," * ",paste(na.omit(lab_class[1:2]),collapse="
* "),
                     sep="")
  } else {
    tabcaption=paste(" - ",lab_var_cat," * "
,paste(na.omit(lab_class[-length(lab_class)]),collapse=" * "),
                     sep="")
  }


  BIRO_table_compute(data=data[data[,pmatch(var_cat,names(data))]!=99,],
                     var=c(var_cat,class),
                     FirstLab=lab_var_cat,
                     lev=lev,
                     labclass=lab_class,
                     texfile=texfile,
                     codtab=paste(ind,"_",sep=""),
                     dirtables=dirtables,
                     dircsv=dirdataout,
                     caption=tabcaption,
                     side=side,
                     width=tabwidth,
                     perc=tabperc,
                     ind=ind,
                     chisq=na.omit(class),
                     progtab=progtab,
                     title=title,
                     codtex=ind)

  progtab=progtab+8
  if ("centre_id" %in% names(data)) {
   progtab=progtab+4
  }
  assign("progtab",progtab,envir = .GlobalEnv)



  if (length(class)>1) {

  if (length(class)<3) {
  for (k in 1:nclass) {
      if (k>1) {
       cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
       cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
       cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
       cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)
      }
   file.append(texfile,
               paste(dirreport,"/pdf/temp/table",ind,"__",letters[k],".tex",sep=""))

   if (file.exists(paste(dirreport,"/html/temp/table",ind,"__",letters[k],".html",sep=
""))) {
    file.append(paste(dirhtml,"/",ind,".html",sep=""),
                paste(dirreport,"/html/temp/table",ind,"__",letters[k],".html",sep=""
))
   }

  }
  }
```

```
  if (length(class)==3) {


      #cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      #cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      #cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)



    file.append(texfile,
                paste(dirreport,"/pdf/temp/table",ind,"__",letters[3],".tex",sep=""))

    if (file.exists(paste(dirreport,"/html/temp/table",ind,"__",letters[3],".html"
,sep=""))) {
     file.append(paste(dirhtml,"/",ind,".html",sep=""),
                 paste(dirreport,"/html/temp/table",ind,"__",letters[3],".html",sep=""
))
    }

   for (k in 1:(nclass-1)) {

      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)



    file.append(texfile,
                paste(dirreport,"/pdf/temp/table",ind,"__",letters[k],".tex",sep=""))

    if (file.exists(paste(dirreport,"/html/temp/table",ind,"__",letters[k],".html"
,sep=""))) {
     file.append(paste(dirhtml,"/",ind,".html",sep=""),
                 paste(dirreport,"/html/temp/table",ind,"__",letters[k],".html",sep=""
))
    }
   }



    values<-na.omit(unique(data[,pmatch(class[3],names(data))]))
    if (class[3]=="type_dm") {
     allvalues=c("Type 1", "Type 2" ,"Other Type")
     values<-allvalues[sort(values)]
    }

   for (k in 1:length(unique(na.omit(data[,pmatch(class[3],names(data))])))) {
    #if (is.na(class[k])) k=k+1


    valueprint<-paste(unlist(strsplit(as.character(values[k])," ")),collapse="_")

    cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
    cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
    cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
    cat("\\headsep 0.50in",file=texfile,fill = TRUE,append=TRUE)
    cat("\\chead{\\small  ",titlelatex," \\\\ \\bf ",lab_class[length(lab_class)]," =
",values[k],"}",file=texfile,fill = TRUE,append=TRUE)

    file.append(texfile,
```

```
              paste(dirreport,"/pdf/temp/table",ind,"_",valueprint,".tex",sep=""))

    if (file.exists(paste(dirreport,"/html/temp/table",ind,"_",valueprint,".html",sep=
""))) {
     file.append(paste(dirhtml,"/",ind,".html",sep=""),
                 paste(dirreport,"/html/temp/table",ind,"_",valueprint,".html",sep=""
))
    }
     }
   }



  # cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
  # cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
  # cat(" \\small BIRO REPORT - Section ",title,file=texfile,fill = TRUE,append=TRUE)
  # cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)









 }

   file.append(texfile,
               paste(dirreport,"/pdf/temp/table",ind,"m.tex",sep=""))

   file.append(texfile,
               paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""))

   if (file.exists(paste(dirreport,"/html/temp/table",ind,"m.html",sep=""))) {
    file.append(paste(dirreport,"/html/",ind,".html",sep=""),
                paste(dirreport,"/html/temp/table",ind,"m.html",sep=""))
   }
   if (file.exists(paste(dirreport,"/html/temp/table",ind,".html",sep=""))) {
    file.append(paste(dirreport,"/html/",ind,".html",sep=""),
                paste(dirreport,"/html/temp/table",ind,".html",sep=""))
   }

  if (length(na.omit(class))==1) {
      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)
  }

  }
 if (typeof(lev_class)!="list") {
  levlist<-list(lev_class)
 }
 if (typeof(lev_class)=="list") {
  levlist<-lev_class
 }

 if (bar==1) {
  # barplots
  progbar=0
  assign("progbar",progbar,envir = .GlobalEnv)

 # write(paste("Preparing To Do Barplots...Data Frame: ",ind,sep=""),
```

```
#       file="",sep="\n")
  noNAclass<-na.omit(class)
  noNAlev_class<-list()
  j=0
  for (i in 1:length(lev_class)) {
   if (is.na(lev_class[[i]])==FALSE) {
    j=j+1
    noNAlev_class[[j]]<-lev_class[[i]]
   }
  }

   if (length(class)==3) {
   if (is.na(class[3])==FALSE) {

   missclass[3]<-class[3]

   if (length(class)>1) {

   orderclass<-1:length(noNAclass)
   orderclass<-orderclass[is.na(class)==FALSE]

   if (length(class)==3) {
    orderclass<-c(orderclass[length(orderclass)],1:(length(orderclass)-1))
   }
   for (i in orderclass) {

      if (is.na(class[i])==FALSE) {

    #if (is.na(lab_class[i])) lab_class[i]<-lab_class[i-1]
    progbar=progbar+1
    assign("progbar",progbar,envir = .GlobalEnv)
      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)

    data$noNA<-ifelse(data[,pmatch(var_cat,names(data))]==99,2,1)
    levNA<-c("Valid Value","NV/NA")
    levmiss<-lev
    levmiss[[1]]<-levNA

    missclass[3]<-paste("noNA",class[3],sep="")

    BIRO_barplot(data=data,
                 var=c("noNA",na.omit(missclass)[i]),
                 labelvar=c(lab_var_cat,lab_class[i]),
                 lev=list(a=levNA,b=levNA),
                 beside=barbeside,
                 what=what,
                 number=TRUE,
                 perc=TRUE,
                 namegraph=paste(ind,"_",progbar,sep=""),
                 dirgraph=dirgraph,
                 cex=cex,
                 nameso=paste(ind,"_m_",class[i],"_",sep=""))

   import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a.pdf",sep=""),
              caption=paste("Barplots: ",indprint,".",progbar," - Missing Data: "
,lab_var_cat,
              " * ",paste(lab_class[i],collapse=","),
              sep=""),
              width=0.90,
```

```
              dirgraph=dirgraphlatex,
              texfile=texfile)

   import_pdf_html(image=paste("../graphs/","i",ind,"_",progbar,"g4_1a.png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Barplots: ",indprint,".",progbar," - Missing Data: "
,lab_var_cat,
                   " * ",paste(lab_class[i],collapse=","),
                   sep=""))



    progbar=progbar+1
    assign("progbar",progbar,envir = .GlobalEnv)


BIRO_barplot(data=data[data[,pmatch(var_cat,names(data))]!=99,pmatch(c(var_cat,na.omit
(noNAclass[i])),names(data))],
                 var=c(var_cat,na.omit(class)[i]),
                 labelvar=c(lab_var_cat,lab_class[i]),
                 lev=list(a=as.vector(lev_var_cat),b=noNAlev_class[[i]]),
                 beside=barbeside,
                 what=what,
                 number=TRUE,
                 perc=TRUE,
                 namegraph=paste(ind,"_",progbar,sep=""),
                 dirgraph=dirgraph,
                 cex=cex,
                 nameso=paste(ind,"_",class[i],"_",sep=""))

   import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a.pdf",sep=""),
              caption=paste("Barplots: ",indprint,".",progbar," - ",lab_var_cat,
              " * ",paste(lab_class[i],collapse=","),
              sep=""),
              width=0.90,
              dirgraph=dirgraphlatex,
              texfile=texfile)

   import_pdf_html(image=paste("../graphs/","i",ind,"_",progbar,"g4_1a.png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Barplots: ",indprint,".",progbar," - ",lab_var_cat,
                   " * ",paste(lab_class[i],collapse=","),
                   sep=""))



    }
    }
    }
    for (t in 1:length(unique(na.omit(data[,pmatch(class[3],names(data))])))) {
     cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
     cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
     cat("\\cfoot{\\thepage}",file=thistexfile,fill = TRUE,append=TRUE)
     cat("\\headsep 0.50in",file=thistexfile,fill = TRUE,append=TRUE)
     cat("\\chead{\\small  ",titlelatex," \\\\ \\bf ",lab_class[length(lab_class)],"
= ",names(lev[[4]])[t],"}",file=texfile,fill = TRUE,append=TRUE)

    for (i in 1:length(noNAclass)) {
      if (i>1 & i<length(noNAclass)) {
       cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
       cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
       cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
       cat("\\headsep 0.50in",file=thistexfile,fill = TRUE,append=TRUE)
       cat("\\chead{\\small  ",titlelatex," \\\\ \\bf ",lab_class[length(lab_class)],
" = ",names(lev[[4]])[t],"}",file=texfile,fill = TRUE,append=TRUE)
       }
     if (length(noNAclass)>1) {


#     if (t>1 & ((i<3 & length(na.omit(class))==3) | (i<2 &
length(na.omit(class))==2))) {
#      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
#      cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
#      cat(" \\small BIRO REPORT - Section ",title,file=texfile,fill =
TRUE,append=TRUE)
#      cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)

#    }

  #  if ("centre_id" %in% names(data)) {
  #   cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
  #  }

    # cat("{\\bf Barplots ",na.omit(lab_class)[i],"}",file=texfile,fill =
TRUE,append=TRUE)
     target<-unique(na.omit(data[,pmatch(class[3],names(data))]))[t]
     datatype<-data[data[,pmatch(class[3],names(data))]==target,]

targetprint<-paste(unlist(strsplit(names(lev[[4]])[as.numeric(as.character(target))],
" ")),collapse="_")

     if (i<3 & noNAclass[i]!=class[3]) {

  datatype$noNA<-ifelse(datatype[,pmatch(var_cat,names(datatype))]==99,2,1)
  levNA<-c("Valid Value","NV/NA")
  levmiss<-lev
  levmiss[[1]]<-levNA

    progbar=progbar+1
    assign("progbar",progbar,envir = .GlobalEnv)


    BIRO_barplot(data=datatype,
                 var=c("noNA",na.omit(missclass[i])),
                 labelvar=c(lab_var_cat,lab_class[i]),
                 lev=list(a=levNA,b=levNA),
                 beside=barbeside,
                 what=what,
                 number=TRUE,
                 perc=TRUE,
                 namegraph=paste(ind,"_",progbar,sep=""),
                 dirgraph=dirgraph,
                 cex=cex,
                 nameso=paste(ind,"_m_",class[i],"_",targetprint,"_",sep=""))

   import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a.pdf",sep=""),
              caption=paste("Barplots: ",indprint,".",progbar," - Missing Data: "
,lab_var_cat,
                   " * ",paste(lab_class[i],collapse=" * "),") ("
,lab_class[length(lab_class)]," = ",names(lev[[4]])[t],")",
              sep=""),
              width=0.90,
              dirgraph=dirgraphlatex,
```

```
              texfile=texfile)

   import_pdf_html(image=paste("../graphs/","i",ind,"_",progbar,"g4_1a.png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Barplots: ",indprint,".",progbar," - Missing Data: "
,lab_var_cat,
                   " * ",paste(lab_class[i],collapse=" * ")," ("
,lab_class[length(lab_class)]," = ",names(lev[[4]])[t],")",
                   sep=""))


    progbar=progbar+1
    assign("progbar",progbar,envir = .GlobalEnv)

    BIRO_barplot(data=datatype[datatype[,pmatch(var_cat,names(datatype))]!=99,],
                 var=c(var_cat,na.omit(noNAclass[i])),
                 labelvar=c(lab_var_cat,lab_class[i]),
                 lev=list(a=as.vector(lev_var_cat),b=noNAlev_class[[i]]),
                 beside=barbeside,
                 what=what,
                 number=TRUE,
                 perc=TRUE,
                 namegraph=paste(ind,"_",progbar,sep=""),
                 dirgraph=dirgraph,
                 cex=cex,
                 nameso=paste(ind,"_",class[i],"_",targetprint,"_",sep=""))


   import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a.pdf",sep=""),
              caption=paste("Barplots: ",indprint,".",progbar," - ",lab_var_cat,
              " * ",paste(lab_class[i],collapse=" * ")," ("
,lab_class[length(lab_class)]," = ",names(lev[[4]])[t],")",
              sep=""),
              width=0.90,
              dirgraph=dirgraphlatex,
              texfile=texfile)

   import_pdf_html(image=paste("../graphs/","i",ind,"_",progbar,"g4_1a.png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Barplots: ",indprint,".",progbar," - ",lab_var_cat,
                   " * ",paste(lab_class[i],collapse=" * ")," ("
,lab_class[length(lab_class)]," = ",names(lev[[4]])[t],")",
                   sep=""))


    if ("centre_id" %in% names(data)) {
     #for (j in 1:length(noNAlev_class)) {

      #cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      #cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      #cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      #cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)
       for (k in 1:length(noNAlev_class[[i]])) {

       cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
       cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
       cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
       cat("\\chead{\\small  ",titlelatex,"\\\\ \\bf",lab_class[length(lab_class)],"
= ",names(lev[[4]])[t],"}",file=texfile,fill = TRUE,append=TRUE)

      if ((k %% 4)==0) {
       cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
       cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
       cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
       cat("\\chead{\\small  ",titlelatex,"\\\\ \\bf",lab_class[length(lab_class)],"
= ",names(lev[[4]])[t],"}",file=texfile,fill = TRUE,append=TRUE)

      }

datacenter<-datatype[datatype[,pmatch(noNAclass[i],names(datatype))]==noNAlev_class[[i
]][k] & is.na(data[pmatch(noNAclass[i],names(datatype))])==FALSE,]
         datacenter2<-datacenter
         if (dim(datacenter)[1]>0) {
         datacenter2$centre_id<-"Overall"
         datacenter<-rbind(datacenter,datacenter2)


         progbar=progbar+1
         assign("progbar",progbar,envir = .GlobalEnv)


BIRO_barplot(data=datacenter[datacenter[,pmatch(var_cat,names(datacenter))]!=99,],
                      var=c(var_cat,"centre_id"),
                      labelvar=c(lab_var_cat,"center"),

lev=list(a=as.vector(lev_var_cat),b=levels(datacenter$centre_id)),
                      beside=barbeside,
                      what=what,
                      number=FALSE,
                      perc=TRUE,
                      namegraph=paste(ind,"_",progbar,sep=""),
                      dirgraph=dirgraph,
                      cex=cex,
                      nameso=paste(ind,"_","source","_",class[i],"_"
,names(lev_class[[i]])[k],"_",class[length(class)],"_",names(lev[[4]])[t],"_",sep=""))

         import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a.pdf",sep=""),
              caption=paste("Barplots: ",indprint,".",progbar," - ",lab_var_cat,
              " by ","data source (",lab_class[i]," = "
,names(lev_class[[i]])[k],collapse=",",", ",lab_class[length(lab_class)]," = "
,names(lev[[4]])[t],")",
              sep=""),
              width=0.90,
              dirgraph=dirgraphlatex,
              texfile=texfile)

         import_pdf_html(image=paste("../graphs/","i",ind,"_",progbar,"g4_1a.png",sep=
""),
                         htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                         caption=paste("Barplots: ",indprint,".",progbar," - "
,lab_var_cat,
              " by ","data source (",lab_class[i]," = "
,names(lev_class[[i]])[k],collapse=",",", ",lab_class[length(lab_class)]," = "
,names(lev[[4]])[t],")",
              sep=""))
      }
     }
    }

    }
   }
   }
   }
```

```
    }
    if (length(class)>1 & i<3) {
    # cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
    # cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
    # cat(" \\small BIRO REPORT - Section ",title,file=texfile,fill =
TRUE,append=TRUE)
    # cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)

    }

  }
  #write("Barplots Done",file="",sep="\n")
 } # i in class

  for (i in 1:length(noNAclass)) {

 #   cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
 #   cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
 #   cat(" \\small BIRO REPORT - Section ",title,file=texfile,fill = TRUE,append=TRUE)
 #   cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)
    #cat("{\\bf Barplots ",na.omit(lab_class)[i],"}",file=texfile,fill =
TRUE,append=TRUE)
   if (length(class)==1) {
    if ("centre_id" %in% names(data)) {
      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)

    }

  data$noNA<-ifelse(data[,pmatch(var_cat,names(data))]==99,2,1)
  levNA<-c("Valid Value","NV/NA")
  levmiss<-lev
  levmiss[[1]]<-levNA

  progbar=progbar+1
  assign("progbar",progbar,envir = .GlobalEnv)

    BIRO_barplot(data=data,
                 var=c("noNA",na.omit(missclass)),
                 labelvar=c(lab_var_cat,lab_class),
                 lev=list(a=levNA,b=levNA),
                 beside=barbeside,
                 number=TRUE,
                 what=what,
                 perc=TRUE,
                 namegraph=paste(ind,"_",progbar,sep=""),
                 dirgraph=dirgraph,
                 cex=cex,
                 nameso=paste(ind,"_m_",class,"_",sep=""))

   import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a.pdf",sep=""),
              caption=paste("Barplots: ",indprint,".",progbar," - Missing Data: "
,lab_var_cat,
              " * ",paste(lab_class[i],collapse=" * "),
              sep=""),
              width=0.90,
              dirgraph=dirgraphlatex,
              texfile=texfile)

   import_pdf_html(image=paste("../graphs/","i",ind,"_",progbar,"g4_1a.pdf",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Barplots: ",indprint,".",progbar," - Missing Data: "
,lab_var_cat,
                   " * ",paste(lab_class[i],collapse=" * "),
                   sep=""))


    progbar=progbar+1
    assign("progbar",progbar,envir = .GlobalEnv)

    namegraph<-paste(ind,"_",progbar,sep="")
    BIRO_barplot(data=data[data[,pmatch(var_cat,names(data))]!=99,],
                 var=c(var_cat,na.omit(noNAclass)),
                 labelvar=c(lab_var_cat,lab_class),
                 lev=list(a=as.vector(lev_var_cat),b=noNAlev_class[[1]]),
                 beside=barbeside,
                 number=TRUE,
                 what=what,
                 perc=TRUE,
                 namegraph=paste(ind,"_",progbar,sep=""),
                 dirgraph=dirgraph,
                 cex=cex,
                 nameso=paste(ind,"_",class,"_",sep=""))

   import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a.pdf",sep=""),
              caption=paste("Barplots: ",indprint,".",progbar," - ",lab_var_cat,
              " * ",paste(lab_class[i],collapse=" * "),
              sep=""),
              width=0.90,
              dirgraph=dirgraphlatex,
              texfile=texfile)

   import_pdf_html(image=paste("../graphs/","i",ind,"_",progbar,"g4_1a.png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Barplots: ",indprint,".",progbar," - ",lab_var_cat,
                   " * ",paste(lab_class[i],collapse=" * "),
                   sep=""))




  if ("centre_id" %in% names(data)) {

     cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
     cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
     cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
     cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)

   for (j in 1:length(noNAlev_class[[1]])) {

     if ((j %% 4)==0) {

     cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
     cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
     cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
     cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)

     }

    datacenter<-data[data[pmatch(class,names(data))]==lev_class[[1]][j] & is
```

```
.na(data[pmatch(class,names(data))])==FALSE,]
    if (dim(datacenter)[1]>0) {
     datacenter2<-datacenter
     datacenter2$centre_id<-"Overall"
     datacenter<-rbind(datacenter,datacenter2)

     progbar=progbar+1
     assign("progbar",progbar,envir = .GlobalEnv)


BIRO_barplot(data=datacenter[datacenter[,pmatch(var_cat,names(datacenter))]!=99,],
                  var=c(var_cat,"centre_id"),
                  labelvar=c(lab_var_cat,"center"),
                  lev=list(a=as.vector(lev_var_cat),b=levels(datacenter$centre_id)),
                  beside=barbeside,
                  what=what,
                  number=FALSE,
                  perc=TRUE,
                  namegraph=paste(ind,"_",progbar,sep=""),
                  dirgraph=dirgraph,
                  cex=cex,
                  nameso=paste(ind,"_","source","_",class[i],"_"
,names(lev_class[[1]])[j],"_",sep=""))

    import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a.pdf",sep=""),
               caption=paste("Barplots: ",indprint,".",progbar,lab_var_cat,
               " by ","data source (",lab_class[i]," = ",names(lev_class[[1]])[j],")"
,collapse=",",
               sep=""),
               width=0.90,
               dirgraph=dirgraphlatex,
               texfile=texfile)

    import_pdf_html(image=paste("../graphs/","i",ind,"_",progbar,"g4_1a.png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                    caption=paste("Barplots: ",indprint,".",progbar,lab_var_cat,
               " by ","data source (",lab_class[i]," = ",names(lev_class[[1]])[j],")"
,collapse=",",
               sep=""))

       }
      }
     }
    }
  }


 # pie charts
 noNAlev_class<-list()
 j=0
 for (i in 1:length(lev_class)) {
  if (is.na(lev_class[[i]])==FALSE) {
   j=j+1
   noNAlev_class[[j]]<-lev_class[[i]]
  }
 }
 if (pie==1) {
 # write(paste("Preparing To Do Pieplots...Data Frame: ",ind,sep=""),
 #       file="",sep="\n")
  noNAclass<-na.omit(class)
  for (i in 1:length(noNAclass)) {
   formula<-as.formula(paste(var_cat,"~",noNAclass[i],sep=""))
   namegraph<-paste("pie",ind,letters[i],sep="")
   BIRO_pie(formula=formula,
            data=data,
            labelvar=lab_var_cat,
            lev_var_cat=lev_var_cat,
            lev_class=noNAlev_class[[i]],
            labelstratum=lab_class[i],
            dirdataout=dirdataout,
            namegraph=namegraph,
            dirgraph=dirgraph,
            cex=cex)
   import_pdf(namegraph=paste(namegraph,".pdf",sep=""),
              caption=paste("Pieplot: ",lab_var_cat,
              " * ",paste(lab_class[i],collapse=","),
              sep=""),
              width=0.90,
              dirgraph=dirgraphlatex,
              texfile=texfile)
   import_pdf_html(image=paste("../graphs/",namegraph,".png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Pieplot: ",lab_var_cat,
                   " * ",paste(lab_class[i],collapse=","),
                   sep=""))
   }
   write("Pieplots Done",file="",sep="\n")
  }
  if (patmap==1) {
   if (maploaded!=shapefile) {
    maploaded=shapefile
    assign(maploaded,shapefile,envir=.GlobalEnv)
    mapfile<-readShapePoly(paste(dirmap,"/",shapefile,".shp",sep=""))
    mapfile<-mapfile[mapfile$ID %in% unique(patient$are),]
    assign("mapfile",mapfile,envir=.GlobalEnv)
   }
   BIRO_patmap(ind=ind,
               shape=mapfile,
               data=data,
               var=mapvar,
               lev=lev_class,
               stat=mapstat,
               labvar=maplab,
               legendpos="topright",
               dirgraph=dirgraph,
               namegraph=ind,
               shapearea_id=shapearea_id,
               valuesarea_id=valuesarea_id)
  }
 }
 # continuous var
 if (is.null(var)==FALSE) {
  if (lines==1) {
 #  write(paste("Preparing To Do Lines plots...Data Frame: ",ind,sep=""),
 #        file="",sep="\n")
   for (i in 1:length(class)) {
    namegraph<-paste("i",ind,letters[i],sep="")
    myvar=list(y=as.numeric(as.character(
               datal[,pmatch(varlines,names(datal))])),
               times=as.numeric(as.character(
               datal[,pmatch(timelines,names(datal))])),
               strata=as.factor(datal[,pmatch(class[i],names(datal))]))
    BIRO_lines(list(y=as.numeric(as.character(
```

```
                datal[,pmatch(varlines,names(datal))])),
            times=as.numeric(as.character(
                datal[,pmatch(timelines,names(datal))])),
            strata=as.numeric(as.character(
                datal[,pmatch(class[i],names(datal))]))),
            descvar=lab_class[i],
            lev=names(lev_class[[i]]),
            namegraph=paste(ind,letters[i],sep=""),
            dirgraph=dirgraph,
            printlines=printlines)
   printlines=0
   if (printlines==1) {
    import_pdf(namegraph=paste("i",ind,letters[i],"g4_5a",sep=""),
               caption=paste("Linesplot: ",
               lab_var_cat,
               " * ",paste(lab_class[i],collapse=","),sep=""),
               width=0.90,
               dirgraph=dirgraphlatex,
               texfile=texfile)

    import_pdf_html(image=paste("../graphs/i",ind,letters[i],"g4_5a.png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                    caption=paste("Linesplot: ",
                    lab_var_cat,
                    " * ",paste(lab_class[i],collapse=","),
                    sep=""))

   }
  }
  write("Lines plots Done",file="",sep="\n")

 }
 if (trellis==1) {
  progdens=0
  assign("progdens",progdens,envir = .GlobalEnv)

  BIRO_fd(date=Sys.Date(),
          data=data,
          var=var[1],
          stratum=class,
          typeround=c("integer"),
          codtab=paste(ind,sep=""),
          dircsv=dircsv)
 # write(paste("Preparing To Do Histogram Trellis...Data Frame: ",ind,sep=""),
 #       file="",sep="\n")

  BIRO_histtrellis(data=data,
                   var=var,
                   strata=class,
                   labvar=lab_var,
                   labclass=lab_class,
                   lev_class=lev_class,
                   dirout=dircsv,
                   ext="csv",
                   date=Sys.Date(),
                   namegraph=ind,
                   dirgraph=dirgraph)

if (length(class)==3) {
 for (k in 1:length(unique(data[,pmatch(class[3],names(data))]))) {

    progdens=progdens+1
```

```
    values<-na.omit(unique(data[,pmatch(class[3],names(data))]))


    if (class[3]=="type_dm") {
     allvalues=c("Type 1", "Type 2" ,"Other Type")
     values<-allvalues[sort(as.vector(values))]
    }

  if (file.exists(paste(dirgraph,"/","i",ind,"_",progdens,"g4_2a.pdf",sep=""))) {

      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\headsep 0.50in",file=thistexfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex," \\\\ \\bf ",lab_class[length(lab_class)],"
= ",names(lev[[4]])[k],"}",file=texfile,fill = TRUE,append=TRUE)


   import_large_pdf(namegraph=paste("i",ind,"_",progdens,"g4_2a.pdf",sep=""),
                    caption=paste("Trellis density plot: ",indprint,".",progdens," - "
,lab_var_cat,
                                  " * ",paste(na.omit(lab_class[1:2]),collapse=" * "),
                                  " (",lab_class[length(lab_class)]," = "
,names(lev[[4]])[k],")",sep=""),
                    dirgraph=dirgraphlatex,
                    texfile=texfile)
   import_pdf_html(image=paste("../graphs/i",ind,"_",progdens,"g4_2a.png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Trellis density plot: ",indprint,".",progdens," - "
,lab_var_cat,
                                 " * ",paste(na.omit(lab_class[1:2]),collapse=" * "),
                                 " (",lab_class[length(lab_class)]," = ",names(lev[[4]])[k],
")",sep=""))

   }
  }
} else {

  if (file.exists(paste(dirgraph,"/","i",ind,"_",progdens,"g4_2a.pdf",sep=""))) {

      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)

   import_large_pdf(namegraph=paste("i",ind,"_",progdens,"g4_2a.pdf",sep=""),
                    caption=paste("Trellis density plot: ",indprint,".",progdens," - "
,lab_var_cat,
                                  " * ",paste(lab_class,collapse=" * "),"("
,lab_class[length(lab_class)]," = ",names(lev[[4]])[t],")",sep=""),
                    dirgraph=dirgraphlatex,
                    texfile=texfile)
   import_pdf_html(image=paste("../graphs/i",ind,"_",progdens,"g4_2a.png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Trellis density plot: ",indprint,".",progdens," - "
,lab_var_cat,
                                 " * ",paste(lab_class,collapse=" * "),sep=""))
  }
}
 #  write("Histogram Trellis Done",file="",sep="\n")
 #  write(paste("Preparing To Do Boxplot Trellis...Data Frame: ",ind,sep=""),
 #        file="",sep="\n")
```

```
 }
 # boxplots
 if (box==1) {
  progbox=0
  assign("progbox",progbox,envir = .GlobalEnv)

#  write(paste("Preparing To Do Boxplots...Data Frame: ",ind,sep=""),
#        file="",sep="\n")


  orderclass<-1:length(class)

  if (length(na.omit(class))==3) {
   orderclass<-c(3,1:(length(class[-3])))
  }
  for (i in orderclass) {

     if (is.na(class[i])) i=i+1

     cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
     cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
     cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
     cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)

   progbox=progbox+1
   assign("progbox",progbox,envir = .GlobalEnv)

   namegraph<-paste(ind,"_",progbox,sep="")

   BIRO_fd(date=Sys.Date(),
           data=data,
           var=var[1],
           stratum=class[i],
           typeround=c("integer"),
           codtab=paste(ind,"_",class[i],"_",sep=""),
           dircsv=dircsv)

   BIRO_boxplot(data=data,
                var=var,
                lev=levlist[[i]],
                notch=notch,
                stratum=class[i],
                namegraph=namegraph,
                dirgraph=dirgraph,
                nameso=paste(ind,"_",class[i],"_",sep=""))

   import_pdf(namegraph=paste("i",ind,"_",progbox,"g4_3a.pdf",sep=""),
              caption=paste("Borplots: ",indprint,".",progbox," - ",
                     lab_var_cat,
                     " * ",paste(lab_class[i],collapse=" * "),sep=""),
              width=0.90,
              dirgraph=dirgraphlatex,
              texfile=texfile)

   import_pdf_html(image=paste("../graphs/i",ind,"_",progbox,"g4_3a.png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Borplots: ",indprint,".",progbox," - ",
                          lab_var_cat,
                          " * ",paste(lab_class[i],collapse=" * "),
                          sep=""))
```

```
    if ("centre_id" %in% names(data)) {

      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)

      for (k in 1:length(lev_class[[i]])) {

      if ((k %% 4)==0) {
      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)
      }

         datacenter<-data[data[,pmatch(class[i],names(data))]==lev_class[[i]][k] & is
.na(data[pmatch(class[i],names(data))])==FALSE & is
.na(data[pmatch(var,names(data))])==FALSE,]
         datacenter2<-datacenter
         if (dim(datacenter2)[1]>0) {

         datacenter2$centre_id<-"Overall"
         datacenter<-rbind(datacenter,datacenter2)

         progbox=progbox+1
         assign("progbox",progbox,envir = .GlobalEnv)

    BIRO_fd(date=Sys.Date(),
            data=datacenter,
            var=c(var),
            stratum="centre_id",
            typeround=c("integer"),
            codtab=paste(ind,"_","source","_",class[i],"_"
,names(lev_class[[i]])[k],sep=""),
            dircsv=dircsv)

         BIRO_boxplot(data=datacenter,
                      var=c(var),
                      lev=levels(datacenter$centre_id),
                      notch=notch,
                      stratum="centre_id",
                      namegraph=paste(ind,"_",progbox,sep=""),
                      dirgraph=dirgraph,
                      nameso=paste(ind,"_","source","_",class[i],"_"
,names(lev_class[[i]])[k],sep=""))

    #cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
    #cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
    #cat(" \\small BIRO REPORT - Section ",title,file=texfile,fill = TRUE,append=TRUE)
    #cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)

    import_pdf(namegraph=paste("i",ind,"_",progbox,"g4_3a.pdf",sep=""),
               caption=paste("Boxplots: ",indprint," ",progbox," - ",lab_var_cat,
              " by ","data source (",lab_class[i]," = ",names(lev_class[[i]])[k],")"
,collapse=",",
              sep=""),
               width=0.90,
               dirgraph=dirgraphlatex,
               texfile=texfile)
    import_pdf_html(image=paste("../graphs/i",ind,"_",progbox,"g4_3a.png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
```

```
                    caption=paste("Boxplots: ",indprint," ",progbox, " - "
,lab_var_cat,
                  " by ","data source (",lab_class[i]," = ",names(lev_class[[i]])[k],")"
,collapse=",",
                  sep=""))

       }
      }
     }

   } #ciclo for class i

    # cat("{\\bf Boxplots ",na.omit(lab_class)[i],"}",file=texfile,fill =
TRUE,append=TRUE)


    for (t in 1:length(unique(na.omit(data[,pmatch("type_dm",names(data))])))) {

       cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
       cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
       cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
       cat("\\headsep 0.50in",file=thistexfile,fill = TRUE,append=TRUE)
       cat("\\chead{\\small  ",titlelatex," \\\\ \\bf ",lab_class[length(lab_class)],
" = ",names(lev[[4]])[t],"}",file=texfile,fill = TRUE,append=TRUE)


     for (i in 1:length(class)) {
     if (i<3) {
     target<-unique(na.omit(data[,pmatch("type_dm",names(data))]))[t]
     datatype<-data[data[,pmatch("type_dm",names(data))]==target & is
.na(data[,pmatch(var,names(data))])==FALSE,]

targetprint<-paste(unlist(strsplit(names(lev[[4]])[as.numeric(as.character(target))],
" ")),collapse="_")

     if(dim(datatype)[1]>0) {



    progbox=progbox+1
    assign("progbox",progbox,envir = .GlobalEnv)

    namegraph<-paste(ind,"_",progbox,sep="")

    BIRO_fd(date=Sys.Date(),
            data=datatype,
            var=var[1],
            stratum=class[i],
            typeround=c("integer"),
            codtab=paste(ind,"_",class[i],"_",targetprint,"_",sep=""),
            dircsv=dircsv)

    BIRO_boxplot(data=datatype,
                 var=var,
                 lev=levlist[[i]],
                 notch=notch,
                 stratum=class[[i]],
                 namegraph=namegraph,
                 dirgraph=dirgraph,
                 nameso=paste(ind,"_",class[[i]],"_",targetprint,"_",sep=""))

    import_pdf(namegraph=paste("i",ind,"_",progbox,"g4_3a.pdf",sep=""),
                 caption=paste("Borplots: ",indprint,".",progbox," - ",
                         lab_var_cat,
                         " * ",paste(lab_class[i],collapse=" * "),"  ("
,lab_class[length(lab_class)]," = ",names(lev[[4]])[t],")",sep=""),
                 width=0.90,
                 dirgraph=dirgraphlatex,
                 texfile=texfile)
    import_pdf_html(image=paste("../graphs/i",ind,"_",progbox,"g4_3a.png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                    caption=paste("Borplots: ",indprint,".",progbox," - ",
                            lab_var_cat,
                            " * ",paste(lab_class[i],collapse=" * "),"  ("
,lab_class[length(lab_class)]," = ",names(lev[[4]])[t],")",sep=""))


      }
     }
    } #type diabete

    } #i<3

   #write("Boxplots Done",file="",sep="\n")
  }
  if (trellis==1) {

   assign("progbox",progbox,envir = .GlobalEnv)

   BIRO_boxtrellis(data=data,
                   var=var,
                   strata=class,
                   labvar=lab_var,
                   labclass=lab_class,
                   lev_class=lev_class,
                   dirout=dircsv,
                   ext="csv",
                   date=Sys.Date(),
                   namegraph=ind,
                   dirgraph=dirgraph)

if (length(class)==3) {
 for (k in 1:length(unique(data[,pmatch(class[3],names(data))]))) {

   progbox=progbox+1

   if (file.exists(paste(dirgraph,"/","i",ind,"_",progbox,"g4_3a.pdf",sep=""))) {
      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\headsep 0.50in",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex," \\\\ \\bf ",lab_class[length(lab_class)],"
= ",names(lev[[4]])[k],"}",file=texfile,fill = TRUE,append=TRUE)

   import_large_pdf(namegraph=paste("i",ind,"_",progbox,"g4_3a.pdf",sep=""),
                    caption=paste("Boxplots: ",indprint,".",progbox," - ",lab_var_cat,
" * ",
                                  paste(na.omit(lab_class[1:2]),collapse=" * "),
                                  "  (",lab_class[length(lab_class)]," = "
,names(lev[[4]])[k],")",sep=""),
                    dirgraph=dirgraphlatex,
                    texfile=texfile)
    import_pdf_html(image=paste("../graphs/i",ind,"_",progbox,"g4_3a.png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                    caption=paste("Boxplots: ",indprint,".",progbox," - ",lab_var_cat,
```

```
" * ",
                                paste(na.omit(lab_class[1:2]),collapse=" * "),
                                " (",lab_class[length(lab_class)],")",sep=""))
,names(lev[[4]])[k],"=",sep=""))
   }
 }
 } else {

   if (file.exists(paste(dirgraph,"/","i",ind,"_",progbox,"g4_3a.pdf",sep=""))) {

      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)

   import_large_pdf(namegraph=paste("i",ind,"_",progbox,"g4_3a.pdf",sep=""),
                    caption=paste("Boxplots: ",indprint,".",progbox," - ",lab_var_cat,
" * ",
                                paste(lab_class,collapse=" * "),sep=""),
                    dirgraph=dirgraphlatex,
                    texfile=texfile)
    import_pdf_html(image=paste("../graphs/i",ind,"_",progbox,"g4_3a.png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                    caption=paste("Boxplots: ",indprint,".",progbox," - ",lab_var_cat,
" * ",
                                paste(lab_class,collapse=" * "),sep=""))
   }

 }
   write("Boxplot Trellis Done",file="",sep="\n")
#    namegraph<-paste("dt",ind,sep="")
#
#    write(paste("Preparing To Do Density Trellis...Data Frame: ",ind,sep=""),
#          file="",sep="\n")
#
#    BIRO_densitytrellis(data=data,
#                        var=var,
#                        strata=class,
#                        labvar=lab_var,
#                        labclass=lab_class,
#                        lev_class=lev_class,
#                        dirout=dircsv,
#                        ext="csv",
#                        date=Sys.Date(),
#                        namegraph=namegraph,
#                        dirgraph=dirgraph)
#
#
#    import_large_pdf(namegraph=paste(namegraph,".pdf",sep=""),
#                     caption=paste("Trellis density plot: ",lab_var_cat," (by ",
#                                   paste(lab_class,collapse=","),")",sep=""),
#                     dirgraph=dirgraphlatex,
#                     texfile=texfile)
#
#    import_pdf_html(image=paste("../graphs/",namegraph,".png",sep=""),
#                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
#                     caption=paste("Trellis density plot: ",lab_var_cat," (by ",
#                                   paste(lab_class,collapse=","),")",sep=""))
#
#
#    write("Density Trellis Done",file="",sep="\n")
```

```
  }
 }

 if (is.null(var)==TRUE & is.null(var_cat)==TRUE & stand==0) {
  BIRO_df2html(data=data,
               htmlfile=paste(dirtables,"/i",ind,"1_3a.html",sep=""),
               catcol=catcol)
  file.append(paste(dirhtml,"/",ind,".html",sep=""),
              paste(dirtables,"/i",ind,"1_3a.html",sep=""))

  for (i in 1:length(names(data))) {
   names(data)[i]<-paste(unlist(strsplit(names(data)[i],"_")),collapse="\\_")
  }

  if (is.na(pmatch("centre\\_id",names(data)))==FALSE) {
   for (i in 1:dim(data)[1]) {
    indexcentre<-pmatch("centre\\_id",names(data))
    data[i,indexcentre]<-paste(unlist(strsplit(as.character(data[i,indexcentre]),"_"
)),collapse="\\_")
   }
  }
  row=data
  if (is.na(pmatch("type\\_dm",names(data)))==TRUE) {
   if ((ind=="4_1_1") | (ind=="4_1_3")) {
     latex(row,file=texfile,append=TRUE,col.just=c("l","r","r","r","r"
),table.env=FALSE,rowname="",title="")
   } else latex(row,file=texfile,append=TRUE,table.env=FALSE,rowname="",title="")
  }

  #split table in pdf by type_dm
  if ("type\\_dm" %in% names(data)) {
   types<-unique(data$'type\\_dm')
   for (i in 1:length(unique(data$'type\\_dm'))) {
   row<-data[data$'type\\_dm'==types[i],]
   if ((ind=="4_1_1") | (ind=="4_1_3")) {
     latex(row,file=texfile,append=TRUE,col.just=c("l","r","r","r"
),table.env=FALSE,rowname="",title="")
   } else latex(row,file=texfile,append=TRUE,table.env=FALSE,rowname="",title="")
   }
  }

 }

  if (stand==1) {
  if (is.null(pop)==TRUE) {
  if ("type_dm" %in% cov) {

   for (i in 1:length(unique(na.omit(data[,pmatch("type_dm",names(data))])))) {

    #if ("sub_ds_id" %in% names(patient)) {
    # if (i>1) {
    #   cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
    #   cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
    #   cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
    #   cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)
    # }
    #}

    target<-unique(na.omit(data[,pmatch("type_dm",names(data))]))[i]
    datatype<-data[data[,pmatch("type_dm",names(data))]==target,]

    #save total frequencies for standardization
```

```
mycalc<-as.data.frame(cbind(centre_id,dim(datatype[datatype[,pmatch(outcome,names(data
type))]==sucvalue,])[1],dim(datatype)[1],target))
    names(mycalc)<-c("Strata","Obs","N","Type_dm")

    if (file.exists(paste(dirdataout,"/i",ind,"d6_1c.csv",sep=""))) {
     cnames=FALSE
    } else {
     cnames=TRUE
    }
    write.table(mycalc,file=paste(dirdataout,"/i",ind,"d6_1c.csv",sep=""
),append=TRUE,row.names=FALSE,col.names=cnames,sep=",")

    if (dim(datatype)[1]>0) {
     if
((dim(datatype[datatype[,pmatch(outcome,names(datatype))]==sucvalue,])[1])==dim(dataty
pe)[1]) {
      strata<-NULL
      #standardization impossible
     }
     newcov<-cov[-pmatch("type_dm",cov)]
     newfactcov<-factcov[-pmatch("type_dm",cov)]
     newchisq<-chisq[-pmatch("type_dm",chisq)]
     if ("sub_ds_id" %in% names(patient)) {
    values<-na.omit(unique(data[,pmatch(class[3],names(data))]))


    if (class[3]=="type_dm") {
     allvalues=c("Type 1", "Type 2")
     values<-allvalues[sort(as.vector(values))]
    }

      cat("\\newpage",file=thistexfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=thistexfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"\\\\ \\bf ",lab_class[length(lab_class)],"="
,values[i],"}",file=thistexfile,fill = TRUE,append=TRUE)

     }
     cat("  ",file=texfile,fill = TRUE,append=TRUE)

     #last control  if data are totaly missing in this type jump
     if
((dim(datatype[datatype[,pmatch(outcome,names(datatype))]==99,])[1])<dim(datatype)[1])
 {

     progtab=progtab+1
     assign("progtab",progtab,envir = .GlobalEnv)
     progbar=progbar+1
     assign("progfor",progfor,envir = .GlobalEnv)
     progfor=progfor+1
     assign("progfor",progfor,envir = .GlobalEnv)


datastd<-BIRO_standardize(data=datatype[datatype[,pmatch(outcome,names(datatype))]!=99
,],
                               target=target,
                               outcome=outcome,
                               cov=newcov,
                               title=captionstd,
                               factcov=newfactcov,
                               strata=strata,
                               lev=levstd,
                               sucvalue=sucvalue,
                               per=per,
                               pop=pop,
                               csvindex=ind,
                               dirdataout=dircsv,
                               filename=ind,
                               dirgraph=dirgraph,
                               dirgraphlatex=dirgraphlatex,
                               texfile=texfile,
                               caption=caption,
                               chisq=chisq,
                               progtab=progtab,
                               progfor=progfor,
                               progbar=progbar)
                               #by=paste("(",lab_class[3]," =
",names(lev[[4]])[i],")",sep=""))

   if (printrate==FALSE) {
   #datastd<-datastd[,-pmatch("Denom",names(datastd))]
    datastd<-datastd[-dim(datastd)[2]]
   }
   mycov<-NULL
   for (i in 1:length(cov)) {
    mycov[i]<-paste(unlist(strsplit(cov[i],"_")),collapse="")
   }

   catcol<-pmatch(mycov,names(datastd))
#  BIRO_df2html(data=datastd,
#               htmlfile=paste(dirtables,"/i",ind,"d1_3a.html",sep=""),
#               catcol=na.omit(catcol))

   file.append(paste(dirhtml,"/",ind,".html",sep=""),
               paste(dirtables,"/i",ind,"d1_3a.html",sep=""))

   file.append(paste(dirhtml,"/",ind,".html",sep=""),
               paste(dirtables,"/i",ind,"d1_3b.html",sep=""))


   for (i in 1:length(names(datastd))) {
    names(datastd)[i]<-paste(unlist(strsplit(names(datastd)[i],"_")),collapse="")
   }
#  if (is.null(chisq)==TRUE) {
#   latex(datastd,file=texfile,append=TRUE,table.env=FALSE,rowname="",title="")
#  }
     }
    }
   }
  }
 }
  if (is.null(pop)==FALSE) {

     progtab=progtab+1
     assign("progtab",progtab,envir = .GlobalEnv)
     progbar=progbar+1
     assign("progfor",progfor,envir = .GlobalEnv)
     progfor=progfor+1
     assign("progfor",progfor,envir = .GlobalEnv)

   datastd<-BIRO_standardize(data=data[data[,pmatch(outcome,names(data))]!=99,],
                             target="",
                             outcome=outcome,
```

```
                          cov=cov,
                          title=captionstd,
                          factcov=factcov,
                          strata=strata,
                          lev=levstd,
                          sucvalue=sucvalue,
                          per=per,
                          pop=pop,
                          csvindex=ind,
                          dirdataout=dircsv,
                          filename=ind,
                          dirgraph=dirgraph,
                          dirgraphlatex=dirgraphlatex,
                          texfile=texfile,
                          caption=caption,
                          chisq=chisq,
                          progtab=progtab,
                          progfor=progfor,
                          progbar=progbar)

  if (printrate==FALSE) {
   #datastd<-datastd[,-pmatch("Denom",names(datastd))]
   datastd<-datastd[-dim(datastd)[2]]
  }
  mycov<-NULL
  for (i in 1:length(cov)) {
   mycov[i]<-paste(unlist(strsplit(cov[i],"_")),collapse="")
  }

  catcol<-pmatch(mycov,names(datastd))
#  BIRO_df2html(data=datastd,
#               htmlfile=paste(dirtables,"/i",ind,"d1_3a.html",sep=""),
#               catcol=na.omit(catcol))

  file.append(paste(dirhtml,"/",ind,".html",sep=""),
              paste(dirtables,"/i",ind,"d1_3a.html",sep=""))

  file.append(paste(dirhtml,"/",ind,".html",sep=""),
              paste(dirtables,"/i",ind,"d1_3b.html",sep=""))


  for (i in 1:length(names(datastd))) {
   names(datastd)[i]<-paste(unlist(strsplit(names(datastd)[i],"_")),collapse="")
  }
#  if (is.null(chisq)==TRUE) {
#   latex(datastd,file=texfile,append=TRUE,table.env=FALSE)
#  }
   }


}

 write(paste("Indicator:",title,"Done"),file="")
 if (nchar(logfile)>0) {
  write(paste("Indicator:",title,"Done"),file=logfile,append=TRUE)
 }

# printing into pdf and html indicator page


 file.append(paste(dirhtml,"/",ind,".html",sep=""),
             paste(dirse,"/source/html/","layout_close.html",sep=""))
```

```
#}

# Missing table in PDF
# if (is.null(data)==FALSE & tab==1) {
#
#  missdata<-as.data.frame(ifelse(is.na(data[,var_cat]),"Missing","No missing"))
#  names(missdata)<-lab_var_cat
#
#  for (w in 1:length(na.omit(class))) {
#   missvect<-ifelse(is.na(data[,na.omit(class)[w]]),"Missing","No Missing")
#   missdata<-cbind(missdata,missvect)
#   names(missdata)<-c(names(missdata)[-dim(missdata)[2]],na.omit(lab_class)[w])
#  }
#  misstab<-as.data.frame(table(missdata))
#  levmiss<-list("No Missing"=0,"Missing"=1)
#
#  for (w in 1:(dim(misstab)[2]-1)) {
#   misstab[,w]<-factor(misstab[,w],levels=levmiss)
#  }
#
#  for (w in 1:length(names(misstab))) {
#   tempname<-unlist(strsplit(names(misstab)[w],''))
#   tempname[tempname=="."]<-" "
#   names(misstab)[w]<-paste(tempname,sep="")
#   rm(tempname)
#  }
#
#  latex(misstab,file=texfile,append=TRUE,table.env=FALSE,center="none")
#
# }
}

##############################################################################
#                                                                            #
##############################################################################

BIRO_report_toc<-function(title="") {
 cat(title,"<br>",sep="",
```

```
 file=paste(dirreport,"/",dbname,"_",thisyear,".html",sep=""),fill = TRUE,append=TRUE)
 write(paste("Indicator:",title," Not Done"),file="")
 if (nchar(logfile)>0) {
  write(paste("Indicator:",title,"Not Done"),file=logfile,append=TRUE)
 }
}


################################################################################
#  open_tex                                                                    #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# open_tex manages the creation and style of the tex file
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#title     [""] = a string containing the title of the report
#authors   [""] = a vector of strings containing the authors of the report
#
#footnote  [""] = a string containing the footnote of the report
#
#time      [""] = a string containing the date of the report
#
#logo      [""] = a string containing the path of the logo (pdf format)
#layout     !   = a string containing the name of the layout .tex file
#final      !   = a string containing the name of the report .tex file
################################################################################

open_tex<-function(title="",
                   authors="",
                   footnote="",
                   time="",
                   logo="",
                   layout,
                   final) {

 x<-FALSE
 if (is.null(layout)) {
  print(layout)
  stop("layout not recognized")
 }

 if (is.null(final)) {
  print(final)
  stop("final not recognized")
 }
 if (file.exists(layout)==FALSE) {
  print(layout)
  stop("layout not exists")
 }

 if (file.exists(layout)==TRUE) {
  x<-  file.copy(layout,final,overwrite=TRUE)
 }
 invisible(x)
}

################################################################################
#  new_chapter                                                                 #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# new_chapter creates a new chapter in the report PDF file
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#title        !   = a string containing the title of the chapter
#texfile      !   = a string containing the name of the report .tex file
################################################################################

new_chapter<-function(title,
                      texfile) {

 title<-paste(strsplit(title,"<"),collapse="$<$")
 title<-paste(strsplit(title,"%"),collapse="\\\\%")
 title<-paste(strsplit(title,">"),collapse="$>$")
 cat("\\chapter{",title,"}",file=texfile,fill = TRUE,append=TRUE)

}

################################################################################
#  new_section                                                                 #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.1, Fedora Core 13
# ------------------------
```

```
# DESCRIPTION
#
# new_section creates a new section in the report PDF file
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#title        !   = a string containing the title of the section
#texfile      !   = a string containing the name of the report .tex file
################################################################################

new_section<-function(title,
                      texfile) {

 title<-gsub("%","\\\\%",as.character(title))
 title<-gsub("<","$<$",as.character(title))
 title<-gsub(">","$>$",as.character(title))

 cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
 cat("\\pagestyle{plain}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\section{",title,"}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)

}

################################################################################
#  new_sub_section                                                             #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# new_sub_section creates a new sub section in the report PDF file
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#title        !   = a string containing the title of the sub section
#texfile      !   = a string containing the name of the report .tex file
################################################################################

new_sub_section<-function(title,texfile) {

 title<-gsub("%","\\\\%",as.character(title))
 title<-gsub("<","$<$",as.character(title))
 title<-gsub(">","$>$",as.character(title))

 cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
 cat("\\pagestyle{plain}",file=texfile,fill = TRUE,append=TRUE)
 cat(paste("\\subsection{",title,"}",sep=""),file=texfile,fill = TRUE,append=TRUE)
 #cat("Reference Date:
",format(checkdate,format="%d/%m/%Y"),"\\par",file=texfile,fill = TRUE,append=TRUE)

}

################################################################################
#  include_tex                                                                 #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# include_tex include a tex file into another
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#incfile      !   = a string containing the name of the report .tex file
#texfile      !   = a string containing the name of the report .tex file
#caption      !   = a string containing the caption
################################################################################

include_tex<-function(incfile,
                      texfile,
                      caption) {

 cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\parbox[2]{",1,"\\textwidth}{",file=texfile,fill = TRUE,append=TRUE)
 cat("\\include{",incfile,"}",file=texfile,fill = TRUE,append=TRUE)
 cat("}",file=texfile,fill = TRUE,append=TRUE)

}

################################################################################
#  import_pdf                                                                  #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
```

```
# ------------------------
# DESCRIPTION
#
# import_pdf include a pdf file into a te
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#namegraph    !   = a string containing the name of the pdf file
#caption      !   = a string containing the caption
#width        !   = a numeric value between 0 and 1 indicating the width in
#                   the report
#dirgraph     !   = the directory where is the PGN file (latex format)
#texfile      !   = a string containing the name of the report .tex file
################################################################################

import_pdf<-function(namegraph,
                     caption,
                     width,
                     dirgraph,
                     texfile) {


 for (i in 1:nchar(dirgraph)) {
  if (substr(dirgraph,i,i)=="\\") {
   substr(dirgraph,i,i)<-"/"
  }
 }

 caption<-gsub("%","\\%",as.character(caption))
 caption<-gsub("<","$<$",as.character(caption))
 caption<-gsub(">","$>$",as.character(caption))
 caption<-gsub("\\%","\\\\%",as.character(caption))



 cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\parbox[2]{",width,"\\textwidth}{",file=texfile,fill = TRUE,append=TRUE)
 cat("\\fbox{",file=texfile,fill = TRUE,append=TRUE)
 cat("\\includegraphics[width=",width,"\\textwidth]{",file=texfile,fill =
FALSE,append=TRUE,sep="")
 cat(dirgraph,"/",namegraph,"}",file=texfile,fill = TRUE,append=TRUE,sep="")
 cat("}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\newline",file=texfile,fill = TRUE,append=TRUE)
 cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
 cat(paste(caption,sep=""),file=texfile,fill = TRUE,append=TRUE)
 cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)
 cat("}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)

}

################################################################################
#  import_large_pdf                                                            #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# import_pdf include a pdf file into a te, layout landscape
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#namegraph    !   = a string containing the name of the pdf file
#caption      !   = a string containing the caption
#dirgraph     !   = the directory where is the PGN file (latex format)
#texfile      !   = a string containing the name of the report .tex file
################################################################################

import_large_pdf<-function(namegraph,
                           caption,
                           dirgraph,
                           texfile) {


 for (i in 1:nchar(dirgraph)) {
  if (substr(dirgraph,i,i)=="\\") {
   substr(dirgraph,i,i)<-"/"
  }
 }

 caption<-gsub("%","\\\\%",as.character(caption))
 caption<-gsub("<","$<$",as.character(caption))
 caption<-gsub(">","$>$",as.character(caption))



 cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\fbox{",file=texfile,fill = TRUE,append=TRUE)
 cat("\\includegraphics[width=0.90\\textwidth]",
     file=texfile,fill = TRUE,append=TRUE,sep="")
 cat("{",paste(dirgraph,"/",namegraph,sep=""),"}",file=texfile,fill = TRUE,
     append=TRUE,sep="")
 cat("}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\",file=texfile,fill = TRUE,append=TRUE)
 #cat("\\begin{rotate}{90}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\parbox[2]{1\\textwidth}{",file=texfile,fill = TRUE,append=TRUE)
 #cat("\\scriptsize",file=texfile,fill = TRUE,append=TRUE)
 cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
 cat(paste(caption,sep=""),file=texfile,fill = TRUE,append=TRUE)
 cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)
 cat("}",file=texfile,fill = TRUE,append=TRUE)
 #cat("\\end{rotate}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)
}

################################################################################
```

```
#  close_tex                                                                  #
##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
## ------------------------
# DESCRIPTION
#
# BIRO_df2html writes the last rows of a .tex file (closes comands like
#  \document,...)
#
# DEPENDENCIES
#
# R package: base
#
##############################################################################
# PARAMETERS(!=required)
#
#file       !               =  string containig the name of the file that must
#                              be closed
#
##############################################################################

close_tex<-function(file) {

 x<-FALSE
 if (is.null(file)) {
  print(file)
  stop("file not recognized")
 }

 if (file.exists(file)==TRUE) {
  cat("\\end{document}",file=file,fill = TRUE,append=TRUE)
 }
 invisible(x)
}

# ############################################################################
#  import_pdf_html                                                           #
# ############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# import_pdf include a pdf file into an HTML file
#
# DEPENDENCIES
#
#  R package: base
#
# ############################################################################
# PARAMETERS(!=required)
#
# image        !   = a string containing the path of the pdf file
# caption      !   = a string containing the caption
# htmlfile     !   = a string containing the name of an HTML file
# ############################################################################

import_pdf_html<-function(image,
                          htmlfile,caption) {

 cat("<BR><BR><BR><BR><BR><BR>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<table style=\"width: 100%;\" border=\"1\">",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<tbody>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<tr>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<td class=\"graph\">",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat(paste("<img src=\"",image,"\">",sep=""),
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</td>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</tr>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</tbody>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat(caption, file=htmlfile,fill = TRUE,append=TRUE)
 cat("</table>",
    file=htmlfile,fill=TRUE,append=TRUE)
}

# ############################################################################
#  new_section_html                                                          #
# ############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# new_section_html create a new section into a HTML file
#
# DEPENDENCIES
#
#  R package: base
#
# ############################################################################
# PARAMETERS(!=required)
#
# title        !   = a string containing the title of the sub section
# htmlfile      !   = a string containing the path of the HTML file
```

```
# ##############################################################################

new_section_html<-function(title,htmlfile) {

 title<-paste(strsplit(title,"<"),collapse="&lt;")
 title<-paste(strsplit(title,"%"),collapse="\\%")
 title<-paste(strsplit(title,">"),collapse="&gt;")

 cat("<table >",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<tbody>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<tr>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<td class=\"section\">",title,
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</td>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</tr>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</tbody>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</table>",
     file=htmlfile,fill=TRUE,append=TRUE)
}
# ##############################################################################
#  new_chapter_html                                                            #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# new_chapter_html create a new chapter into a HTML file
#
# DEPENDENCIES
#
#  R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# title        !   = a string containing the title of the sub section
# htmlfile      !   = a string containing the path of the HTML file
# ##############################################################################

new_chapter_html<-function(title,
                           htmlfile) {

 title<-paste(strsplit(title,"<"),collapse="&lt;")
 title<-paste(strsplit(title,"%"),collapse="\\%")
 title<-paste(strsplit(title,">"),collapse="&gt;")

 cat("<table >",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<tbody>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<tr>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<td class=\"chapter\">",title,
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</td>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</tr>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</tbody>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</table>",
     file=htmlfile,fill=TRUE,append=TRUE)
}
# ##############################################################################
#  new_sub_section_html                                                        #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# new_sub_section_html create a new section into a HTML file
#
# DEPENDENCIES
#
#  R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# title        !   = a string containing the title of the sub section
# htmlfile      !   = a string containing the path of the HTML file
# ##############################################################################

new_sub_section_html<-function(title,
                               htmlfile) {

 title<-paste(strsplit(title,"<"),collapse="&lt;")
 title<-paste(strsplit(title,"%"),collapse="\\%")
 title<-paste(strsplit(title,">"),collapse="&gt;")
 cat("<table >",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<tbody>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<tr>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("<td class=\"subsection\">",title,
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</td>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</tr>",
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</tbody>",
```

```
     file=htmlfile,fill=TRUE,append=TRUE)
 cat("</table>",
     file=htmlfile,fill=TRUE,append=TRUE)
 }

##############################################################################
#   BIRO_dframe_latex
##############################################################################
#
##############################################################################

BIRO_dframe_latex<-function(data,
                            texfile) {

 cat("\\begin{tabular}{|c",rep("|r",dim(data)[2]),"|}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\hline",file=texfile,fill = TRUE,append=TRUE)

 for (i in 1:dim(data)[1]) {
  for (j in 1:dim(data)[2]) {
   cat(data[i,j], " & ",file=texfile,fill=FALSE,append=TRUE)
  }
 }

}



BIRO_simpletab_latex<-function(data,
                           freq="n",
                           perc="perc",
                           texfile,
                           caption) {

 cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\parbox[2]{",0.50,"\\textwidth}{",file=texfile,fill = TRUE,append=TRUE)
 cat("\\resizebox{",0.50,"\\textwidth}{!}{",file=texfile,fill = TRUE,
     append=TRUE)

 cat("\\begin{tabular}{|c|r|}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\hline",file=texfile,fill = TRUE,append=TRUE)
 cat(names(data)[1]," & n (\\%)","  \\\\ ",file=texfile,fill=FALSE,append=TRUE)
 cat("\\hline",file=texfile,fill = TRUE,append=TRUE)


 for (i in 1:dim(data)[1]) {
   cat(data[i,1],"&",sum(data[i,pmatch(freq,names(data))]), "("
,data[i,pmatch(perc,names(data))],")","  \\\\ ",
   file=texfile,fill=FALSE,append=TRUE)

 }
   cat("\\hline",file=texfile,fill = TRUE,append=TRUE)
   cat("TOTAL & ",sum(data[,pmatch(freq,names(data))]), "(100\\%)","  \\\\ ",
   file=texfile,fill=FALSE,append=TRUE)
   cat("\\hline",file=texfile,fill = TRUE,append=TRUE)

cat("\\end{tabular}",
     file=texfile,fill = TRUE,append=TRUE)
cat("}",file=texfile,fill = TRUE,append=TRUE)
#cat("\\scriptsize",file=texfile,fill = TRUE,append=TRUE)
cat(caption,file=texfile,fill = TRUE,append=TRUE)
cat("\\vspace{5mm}",file=texfile,fill = TRUE,append=TRUE)
cat("}",file=texfile,fill = TRUE,append=TRUE)
cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)

}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: biro_se_tables.r                                                       #
#                                                                              #
#
# ##############################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2008-10-23
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# biro_se_tables.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_fd
#   BIRO_table_format
#   BIRO_table_compute
#   BIRO_table_print
#   BIRO_df2html
#
# ##############################################################################

# ##############################################################################
#  BIRO_fd                                                                     #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
## ------------------------
# DESCRIPTION
#
# BIRO_fd for Conditional Frequency Distribution
#
# DEPENDENCIES
#
# R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# date       [System date] = date of analysis
# data                     = a data.frame containing the data
# var                      = the name of the target variable
# stratum                  = the name of the stratum variable
# typeround                = the type of rounding:
#                            option values:
#                             - "integer" , for integer rounding
#                             - a vector containig the thresholds
# codtab                   = the code table according the BIRO indicators list
# dircsv                   = the path of the output (csv files)
#
# ##############################################################################

BIRO_fd<-function(date,data,var,stratum,typeround,codtab,dircsv) {

 for (i in 1:length(var)) {
   temp<-as.vector(as.numeric(as.character(data[,pmatch(var[i],names(data))])))

 if ((typeround[[i]][1])=="integer") {
  temp<-as.data.frame(round(temp))
  names(temp)<-paste(var[i],"_i",sep="")
 }


 if (typeof(typeround[[i]][1])=="double") {
   temp<-as.data.frame(varclass(temp,typeround[[i]]))
   names(temp)<-paste(var[i],"_c",sep="")

 }

 if (i==1) {
  newdata=temp
 }

 if (i>1) {
 newdata<-as.data.frame(cbind(newdata,temp))
 }

 }

 strata<-as.data.frame(data[,pmatch(stratum,names(data))])
 names(strata)<-stratum

 newdata<-cbind(newdata,strata)


 tab<-as.data.frame(table(newdata))
```

```
 tab$id=centre_id
 tab$date=date
 #Delete 0-Frequency cell
 tab<-tab[tab$Freq>0,]
 names(tab)[pmatch("Freq",names(tab))]<-"n"
 tab$start=oneyearbf
 tab$end=checkdate
 tab$dbname=dbname
 write.csv(tab,paste(dircsv,"/i",tolower(codtab),"d1_1a.csv",sep=""),row.names =
FALSE)
 #return(tab)
}


# ############################################################################
#  BIRO_table_format                                                         #
# ############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_table_format manages the condition: table with 4 variables, according
# the BIRO layout
#
# DEPENDENCIES
#
# R package: base
#
# ############################################################################
# PARAMETERS(!=required)
#
# var       !              = two or more vector with the values of the target
#                            variables, Passed as a list
# FirstLab  !              = the name of the first variable, the first variable
#                            of var is the vertical variable in the table
# lev       !              = a list containing the vactor with che levels of
#                            the categorical variable (list)
# texfile   !              = the path of the output texfile
# codtab    !              = the coded name of the output csv file
# dirtables !              = directory for the output html table file
# dircsv    !              = directory for the output csv file
# caption   !              = string for the caption
# side      [0]            = if side is egual to 1 to table in the pdf file is
#                            printed in portrait layout
# width     [1]            = width for the pdf (see latex for details)
# perc      [TRUE]         = TRUE for percentage
# dirout    [getwd()]      = output location
# out       [out]          = name of output csv dataset
# ext       [csv]          = output file format
# date      [Sys.Date()]   = date variable or value
# ############################################################################

BIRO_table_format<-function(data,
                            var,
                            FirstLab,
                            lev=as.list(),
                            texfile,
                            codtab,
                            dirtables,
                            dircsv,
                            caption,
                            side=0,
                            width=1,
                            perc=TRUE) {


temp<-na.omit(data[,pmatch(var,names(data))])
if (dim(temp)[2]==4) {
 for (k in 1:length(sort(unique(temp[,4])))) {
  funcdata<-rep(NA,dim(temp)[2])
  for (h in 1:length(var)) {
   funcdata<-temp[temp[,4]==sort(unique(temp[4]))[k],]
  }
  if (length(var) > 0 & length(lev)>0) {
   funcdata[,pmatch(var[1],names(funcdata))]<-
   as.factor(funcdata[,pmatch(var[1],names(funcdata))])
   levels(funcdata[,pmatch(var[1],names(funcdata))])<-lev[[1]]
  }
  if (length(var) > 0 & length(lev)==0) {
   levels(funcdata[,pmatch(var[1],names(funcdata))])<-
   levels(as.factor(funcdata[,pmatch(var[1],names(funcdata))]))
  }
  if (length(var) > 1 & length(lev)>1) {
   funcdata[,pmatch(var[2],names(funcdata))]<-
   as.factor(funcdata[,pmatch(var[2],names(funcdata))])
   levels(funcdata[,pmatch(var[2],names(funcdata))])<-lev[[2]]
  }
  if (length(var) > 1 & length(lev)==1) {
   levels(funcdata[,pmatch(var[2],names(funcdata))])<-
   levels(as.factor(funcdata[,pmatch(var[2],names(funcdata))]))
  }
  if (length(var) > 2 & length(lev)>2)  {
   funcdata[,pmatch(var[3],names(funcdata))]<-
   as.factor(funcdata[,pmatch(var[3],names(funcdata))])
   levels(funcdata[,pmatch(var[3],names(funcdata))])<-lev[[3]]
  }
  if (length(var) > 2 & length(lev)==2)  {
   levels(funcdata[,pmatch(var[3],names(funcdata))])<-
   levels(as.factor(funcdata[,pmatch(var[3],names(funcdata))]))
  }
  if (length(var) > 3 & length(lev)>3)   {
   funcdata[,pmatch(var[4],names(funcdata))]<-
   factor(funcdata[,pmatch(var[4],names(funcdata))],levels=lev[[4]][k])
  }
  if (length(var) > 3 & length(lev)==3)   {
   funcdata[,pmatch(var[4],names(funcdata))]<-
   factor(funcdata[,pmatch(var[4],names(funcdata))],
   levels=sort(levels(as.factor(funcdata[,pmatch(var[4],
                                        names(funcdata))])))[k])
  }

  newlev<-list()
  for (i in 1:length(var)) {
   newlev[[i]]<-levels(var[[i]])
  }
```

```
   codtabv4<-paste(codtab,letters[k],sep="")
   BIRO_table_compute(data=funcdata,
                      var=var,
                      FirstLab=FirstLab,
                      lev=newlev,
                      texfile=texfile,
                      codtab=codtabv4,
                      dirtables=dirtables,
                      dircsv=dircsv,
                      caption=newcaption,
                      side=side,
                      width=width,
                      perc=perc,
                      ind=ind,
                      chisq=chisq[1:2],
                      progtab=ptorgtab)
  }
 }
 if (length(var)<4) {
  BIRO_table_compute(data=funcdata,
                     var=var,
                     FirstLab=FirstLab,
                     lev=lev,
                     texfile=texfile,
                     codtab=codtab,
                     dirtables=dirtables,
                     dircsv=dircsv,
                     caption=caption,
                     side=side,
                     width=width,
                     perc=perc,
                     ind=ind,
                     chisq=chisq,
                     progtab=ptorgtab)
 }
}



# ##############################################################################
#  BIRO_table_compute                                                          #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_table_compute calculates contingency table
#
# DEPENDENCIES
#
# R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# var         !            = two or more vector with the values of the target
#                            variables, Passed as a list
# FirstLab    !            = the name of the first variable, the first variable
#                            of var is the vertical variable in the table
# lev         !            = a list containing the vactor with che levels of
#                            the categorical variable (list)
# texfile     !            = the path of the output texfile
# codtab      !            = the coded name of the output csv file
# dirtables   !            = directory for the output html table file
# dircsv      !            = directory for the output csv file
# caption     !            = string for the caption
# side      [0]            = if side is egual to 1 to table in the pdf file is
#                            printed in portrait layout
# width     [1]            = width for the pdf (see latex for details)
# perc      [TRUE]         = TRUE for percentage
# dirout    [getwd()]      = output location
# out       [out]          = name of output csv dataset
# ext       [csv]          = output file format
# date      [Sys.Date()]   = date variable or value
# ind       !              = a string containing the index of indicator
# ##############################################################################

BIRO_table_compute<-function(data,
                             var,
                             FirstLab,
                             lev=list(),
                             labclass,
                             texfile,
                             codtab,
                             dirtables,
                             dircsv,
                             caption,
                             side=0,
                             width=1,
                             perc=TRUE,
                             ind=ind,
                             chisq=chisq,
                             progtab,
                             last=0,
                             title="",
                             codtex) {

 # Creates association between levels of a categorical variable and its values

 funcdata<-data[,pmatch(na.omit(var),names(data))]
 if (length(var) > 0 & length(lev)>0) {
  if (is.na(var[1])==FALSE) {
   funcdata[,pmatch(var[1],names(funcdata))]<-
   as.factor(funcdata[,pmatch(var[1],names(funcdata))])
   levels(funcdata[,pmatch(var[1],names(funcdata))])<-lev[[1]]
  }
 }

 if (length(var) > 0 & length(lev)==0) {
  if (is.na(var[1])==FALSE) {
   levels(funcdata[,pmatch(var[1],names(funcdata))])<-
   levels(as.factor(funcdata[,pmatch(var[1],names(funcdata))]))
  }
 }

 if (length(var) > 1 & length(lev)>1) {
```

```
 if (is.na(var[2])==FALSE) {
  funcdata[,pmatch(var[2],names(funcdata))]<-
  as.factor(funcdata[,pmatch(var[2],names(funcdata))])
  levels(funcdata[,pmatch(var[2],names(funcdata))])<-lev[[2]]
 }
}

if (length(var) > 1 & length(lev)==1) {
 if (is.na(var[2])==FALSE) {
  levels(funcdata[,pmatch(var[2],names(funcdata))])<-
  levels(as.factor(funcdata[,pmatch(var[2],names(funcdata))]))
 }
}

if (length(var) > 2 & length(lev)>2)  {
 if (is.na(var[3])==FALSE) {
  funcdata[,pmatch(var[3],names(funcdata))]<-
  as.factor(funcdata[,pmatch(var[3],names(funcdata))])
  levels(funcdata[,pmatch(var[3],names(funcdata))])<-lev[[3]]
 }
}

if (length(var) > 2 & length(lev)==2)  {
 if (is.na(var[3])==FALSE) {
  levels(funcdata[,pmatch(var[3],names(funcdata))])<-
  levels(as.factor(funcdata[,pmatch(var[3],names(funcdata))]))
 }
}

if (length(var) > 3 & length(lev)>3)   {
 if (is.na(var[4])==FALSE) {
  funcdata[,pmatch(var[4],names(funcdata))]<-
  factor(funcdata[,pmatch(var[4],names(funcdata))],levels=lev[[4]])
  levels(funcdata[,pmatch(var[4],names(funcdata))])<-lev[[4]]
 }
}

if (length(var) > 3 & length(lev)==3)   {
 if (is.na(var[4])==FALSE) {
  funcdata[,pmatch(var[4],names(funcdata))]<-
  factor(funcdata[,pmatch(var[4],names(funcdata))],
         levels=sort(levels(as.factor(funcdata[,pmatch(var[4],
         names(funcdata))]))))
 }
}

for (i in 1:length(var)) {
 if (typeof(lev[[i]])=="list") {
  lev[[i]]<-names(unlist(lev[[i]]))
 }
}

# creates a vector in which each element is then number of
# the levels of the corrisponding variable
nvar=length(var)
n<-rep(NA,4)
for (i in 1:nvar) {
 if (is.na(var[i])==FALSE) {
  n[i]<-nlevels(funcdata[,pmatch(var[i],names(funcdata))])
 }
}
```

```
NAn<-is.na(n)
n[is.na(n)]<-1
varlist<-as.list(funcdata)
# calculates contingency table
tab<-as.data.frame(table(varlist))

# creates a schema of the output data.frame
names(lev)<-names(varlist)
schema<-NULL
if (length(names(lev))>0) {
 times1<-rep(prod(n[2:nvar]),length(lev[[1]]))
 v1<-rep(lev[[1]],times1)
 schema<-cbind(schema,v1)
}

if (length(names(lev))>1) {
 times2<-rep(prod(n[3:nvar]),length(lev[[2]]))
 if (nvar==2) {times2<-1}
 v2<-rep(lev[[2]],times2)
 schema<-cbind(schema,v2)
}

if (length(names(lev))>2) {
 times3<-rep(n[4],length(lev[[3]]))
 if (nvar==3) {times3<-1}
 v3<-rep(lev[[3]],times3)
 schema<-cbind(schema,v3)
}

if (length(names(lev))==4) {
 v4<-rep(lev[[4]],prod(n)/length(lev[[4]]))
 schema<-cbind(schema,v4)
}
schema<-as.data.frame(schema)
names(schema)<-var

# merges the contingency table with the schema and puts 0
# where there are not observations

tab<-merge(tab,schema,by=na.omit(var))
tab$Freq[is.na(tab$Freq)]<-0
tab$id=centre_id
# calls BIRO_table_print for printing table in tex and html formats

# deleted 2010-08-05
# htmlfile=paste(dirtables,"/",codtab,".html",sep="")
# paste(htmlfile)
# cat("<table border=\"1\">",file=htmlfile,fill = TRUE,append=TRUE)
# cat("",file=htmlfile,fill = TRUE,append=TRUE)
# cat(" <tbody>",file=htmlfile,fill = TRUE,append=TRUE)
# cat("",file=htmlfile,fill = TRUE,append=TRUE)
# cat("  <tr>",file=htmlfile,fill = TRUE,append=TRUE)
# cat("",file=htmlfile,fill = TRUE,append=TRUE)
# cat("   <td>",file=htmlfile,fill = TRUE,append=TRUE)
# if (struc==TRUE) {
#  paste(sthtml)
#  cat("",file=sthtml,fill = TRUE,append=TRUE)
#  cat("<table border=\"1\">",file=sthtml,fill = TRUE,append=TRUE)
#  cat("",file=sthtml,fill = TRUE,append=TRUE)
#  cat(" <tbody>",file=sthtml,fill = TRUE,append=TRUE)
#  cat("",file=sthtml,fill = TRUE,append=TRUE)
#  cat("  <tr>",file=sthtml,fill = FALSE,append=TRUE)
```

```
#  cat("",file=sthtml,fill = TRUE,append=TRUE)
#  cat("   <td>",file=sthtml,fill = TRUE,append=TRUE)
# }

 if (length(var)==4) {
  for (k in 1:length(unique(tab[,pmatch(var[4],names(tab))]))) {
   # values of the 4th categorical variable
   values<-unique(tab[,pmatch(var[4],names(tab))])

   if (var[4]=="type_dm") {
     allvalues=c("Type 1", "Type 2" ,"Other Type")
     #values=c("Other Type","Type 2","Type 1")
     values<-allvalues[sort(na.omit(pmatch(allvalues,values)))]
   }

  if (var[4]=="noNAtype_dm") {
     var[4]<-"type_dm"
     allvalues=c("Type 1", "Type 2" ,"Other Type")
     values=c("Other Type","Type 2","Type 1")
     values<-allvalues[sort(na.omit(pmatch(allvalues,values)))]

   }

   temptab<-tab[tab[,pmatch(var[4],names(tab))]==values[k],]
   newlev<-list()
   for (i in 1:3) {
    newlev[[i]]<-lev[[i]]
   }
   newlev[[4]]<-as.character(values[k])
   n[4]<-1
   noNAnewlev<-list()
   j=0
   for (i in 1:4) {
    if (is.na(newlev[[i]])==FALSE) {
     j=j+1
     noNAnewlev[[j]]<-newlev[[i]]
    }
   }

noNAnewlev[[length(na.omit(var))]]<-paste(na.omit(labclass[1:length(labclass)-1]),coll
apse="*")

   names(noNAnewlev)<-na.omit(var)
#   if (k==length(unique(tab[,pmatch(var[4],names(tab))]))) {
#    newcaption=caption
#   }
#   if (k<length(unique(tab[,pmatch(var[4],names(tab))]))) {
     if (k>1) {
     if (tabjump==2) progtab=progtab+6
     if (tabjump==1) progtab=progtab+2
     #assign("progtab",progtab,envir = .GlobalEnv)

     if (k==length(unique(tab[,pmatch(var[4],names(tab))])))
     assign("progtab",progtab,envir = .GlobalEnv)
    }
    newcaption=paste("Table ", indprint,".",progtab, caption," ("
,labclass[length(labclass)], " = ",newlev[[4]][1],")",sep="")
#   }
   nnew<-c(n[NAn==FALSE],rep(1,length(n[NAn==TRUE])))
   temptab<-temptab[,pmatch(c(na.omit(var),"Freq","id"),names(temptab))]
   newcodtab<-paste(codtab,letters[k],sep="")
   valueprint<-paste(unlist(strsplit(as.character(values[k])," ")),collapse="_")

   newcodtab<-paste(ind,"_",paste(na.omit(var[2:(length(var)-1)]),collapse="_"),"_"
,valueprint,sep="")



   BIRO_table_print(tab=temptab,
                    var=na.omit(var),
                    n=nnew,
                    FirstLab=FirstLab,
                    lev=noNAnewlev,
                    texfile=texfile,
                    codtab=newcodtab,
                    dirtables=dirtables,
                    dircsv=dircsv,
                    caption=newcaption,
                    side=side,
                    width=width,
                    perc=perc,
                    ind=ind,
                    chisq=na.omit(chisq[1:2]),
                    title=title,
                    codtex=paste(ind,"_",valueprint,sep=""))


  # cat("<BR>",file=htmlfile,fill = TRUE,append=TRUE)
  # cat("<BR>",file=sthtml,fill = TRUE,append=TRUE)
  }
 }
 if (length(var)<4) {
 newcaption<-paste("Table ", indprint,".",progtab, caption,sep="")

 newlev<-lev
 newlev[[length(lev)]]<-paste(na.omit(labclass[1]),collapse="*")

 if (length(na.omit(var))==3) {
  newlev[[1]]<-lev[[1]]
  newlev[[2]]<-lev[[2]]
  newlev[[3]]<-lev[[3]]
  newlev[[4]]<-paste(na.omit(labclass[1:2]),collapse="*")
  var[4]<-var[2]
  n[4]<-1
 }

 if (length(na.omit(var))==2) {
  newlev[[1]]<-lev[[1]]
  newlev[[2]]<-lev[[2]]
  newlev[[3]]<-labclass[1]
  newlev[[4]]<-labclass[1]
  var[3]<-NA
  var[4]<-var[2]
  n[3]<-1
  n[4]<-1
 }

 #chisq<-na.omit(unique(var[-1]))

  BIRO_table_print(tab=tab,
                   var=na.omit(var),
                   n=na.omit(n),
                   FirstLab=FirstLab,
```

```
                  lev=newlev,
                  texfile=texfile,
                  codtab=codtab,
                  dirtables=dirtables,
                  dircsv=dircsv,
                  caption=newcaption,
                  side=side,
                  width=width,
                  perc=perc,
                  ind=ind,
                  chisq=chisq,
                  codtex=codtex)

 }

# cat("",file=htmlfile,fill = TRUE,append=TRUE)
# cat("   </td>",file=htmlfile,fill = FALSE,append=TRUE)
# cat("",file=htmlfile,fill = TRUE,append=TRUE)
# cat("  </tr>",file=htmlfile,fill = FALSE,append=TRUE)
# cat("",file=htmlfile,fill = TRUE,append=TRUE)
# cat(" </tbody>",file=htmlfile,fill = FALSE,append=TRUE)
# cat("",file=htmlfile,fill = TRUE,append=TRUE)
# cat("</table>",file=htmlfile,fill = FALSE,append=TRUE)
# cat("",file=htmlfile,fill = TRUE,append=TRUE)

# if (struc==TRUE) {
#  cat("",file=sthtml,fill = TRUE,append=TRUE)
#  cat("   </td>",file=sthtml,fill = FALSE,append=TRUE)
#  cat("",file=sthtml,fill = TRUE,append=TRUE)
#  cat("  </tr>",file=sthtml,fill = TRUE,append=TRUE)
#  cat("",file=sthtml,fill = TRUE,append=TRUE)
#  cat(" </tbody>",file=sthtml,fill = TRUE,append=TRUE)
#  cat("",file=sthtml,fill = TRUE,append=TRUE)
#  cat("</table>",file=sthtml,fill = TRUE,append=TRUE)
#  cat("",file=sthtml,fill = TRUE,append=TRUE)
# }
}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: biro_se_trend.r                                                        #
#                                                                              #
#
# ##############################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# biro_se_trend.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_lines
#   BIRO_plotlines
#
# ##############################################################################

# ##############################################################################
#  BIRO_lines                                                                  #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_lines calculates mean of patient means for a particular and prepare data
# for BIRO_plotlines
#
# DEPENDENCIES
#
# BIRO_plotlines
#
# ##############################################################################
# PARAMETERS(!=required)
#
# var        !            = two or more vector with the values of the target
#                           variables, passed as a list (from the second there
#                           are stratum variables)
# descvar    !            = the labels of the first variable
# lev        !            = a list containing the vactor with che levels of the
#                           categorical variable (list)
# namegraph  !            = the coded name of the output files
# dirgraph   !            = the path of the output (jpg,eps,pdf files)
# stat       ["mean"]     = statistic applied to patient data
#
# ##############################################################################

BIRO_lines<-function(var=list(),
                     descvar,
                     lev,
                     namegraph,
                     dirgraph,
                     stat="mean",
                     printlines) {

 if (is.null(lev)==FALSE) {
  levels(var$strata)=lev
 }


 if (tolower(stat)=="mean") {
  new_data<-as.data.frame(var)
  new_data$y=as.numeric(new_data$y)
  new_data=na.omit(new_data)
  new_data$strata=as.numeric(as.character(new_data$strata))
  new_data<-new_data[is.na(new_data$y)==FALSE,]
  splitvar<-interaction(new_data$times,new_data$strata,sep = ":")
  spltdata<-split(new_data, as.factor(splitvar), drop = FALSE)
  for (i in 1:length(spltdata)) {
   spltdata[i]<-na.omit(spltdata[i])
  }
  means<-as.data.frame(t(mapply(mean,spltdata)))
  for (i in 1:length(spltdata)) {
   if (dim(spltdata[[i]])[1]==0) spltdata[[i]]<-
       as.data.frame(matrix(rep(999,6),2))
  }
  sds<-t(lapply(spltdata,sd))
  sds<-(unlist(sds[1,]))
  sds<-as.vector(sds[seq(1,length(sds),3)])
  sds[is.na(sds)]<-0
  num<-aggregate(new_data$y,list(splitvar),"length")
  names(num)<-c("byvar","n")
 }
 means2<-as.data.frame(cbind(means[1],means$times,means$strata))
```

```
 means2$byvar=row.names(means2)
 names(means2)<-c("y","times","strata","byvar")
 means2<-merge(means2,num,by="byvar",all.x=TRUE)
 means2<-means2[,-1]
 names(means2)<-c("y","times","strata","n")
 means2$stat=stat
 means2$sd<-sds
 means2$codist=centre_id
 means2<-na.omit(means2)
 means2$start=oneyearbf
 means2$end=checkdate

 write.csv(means2,
           paste(dirdataout,"/i",namegraph,"d4_5a.csv",sep=""),
            row.names = FALSE)

 write("Lines Data Stored",file="")
 if (nchar(logfile)>0) {
  write("Lines Data Stored",file=logfile,append=TRUE)
 }

if (printlines==1 & dim(means2)[1]>1) {
 BIRO_plotlines(data=means2,
                lev=lev,
                descvar=descvar,
                namegraph=namegraph,
                dirgraph=dirgraph)
}

}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_se_recode.r                                                         #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_se_recode.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
################################################################################

  #Thresholds

  ageth<-c(18,35,55,75)
  ageth5<-seq(5,85,5)
  durdiabth<-c(10,20)
  weightth<-c(50,70,90,110,130)
  bmith<-c(25,27,30,40)
  sbpth<-c(130,160)
  dbpth<-c(70,100,120)
  cholth<-c(2.59,5.18,7.77)
  hdlth<-c(1.295)
  ldlth<-c(2.59,5.18)
  creatth<-c(50,100,150)
  hba1cth<-c(7,8)
  agethst<-c(15,25,35,45,55,65,75,85)

  #Limits

  bmilimit<-c(4,44)
  weightlimit<-c(25,200)
  heightlimit<-c(60,250)
  chollimit<-c(70,400)
  hdllimit<-c(10,100)
  ldllimit<-c(30,300)
  creatlimit<-c(0.1,10)
  hba1climit<-c(2.5,20)
  ma_testlimit<-c(0,300)
  dbplimit<-c(30,150)
  sbplimit<-c(60,250)
  triglimit<-c(10,1000)
  trigl12limit<-c(10,1000)

  #Levels of Categorical Variables

  levhba1c9<-list("Yes"=1,"No"=2)
  levhba1c75<-list("Yes"=1,"No"=2)
  levtype_dm<-list("Type 1"="1","Type 2"="2","Other Type"="3")
  levtype_dmst<-list("Type 1"="1","Type 2"="2")
  levsex<-list("Male"=1,"Female"=2)
  levagesex<-list("Male"=0,"0 - 14"=1,"15 - 24"=2,"25 - 34"=3,"35 - 44"=4,"45 - 54"=5,
"55 - 64"=6,"65 - 74"=7,"75 - 84"=8,"85 +"=9)
  levma_test<-list("No MA Test Recorded"=0,"MA Test Normal"=1,"MA Test Abnormal"=2)
  levsmok_stat<-list("Current smoker"=1,"Non-Smoker"=2,"Ex-Smoker"=3)
  levretin<-list("No Retinopathy"=1,"Background Retinopathy"=2,"Referable Retinopathy"
=3)
  levmacul<-list("No Maculopathy"=1,"Maculopathy"=2)
  levfoot_exam<-list("No"=0,"Yes"=1)
  levesrf<-list("Yes"=1,"No"=2)
  levulcer<-list("Yes"=1,"No"=2)
  levamput<-list("Yes"=1,"No"=2)
  levstroke<-list("Yes"=1,"No"=2)
  levmi<-list("Yes"=1,"No"=2)
  levhypertension<-list("Yes"=1,"No"=2)
  leveye_exam<-list("No"=0,"Yes"=1)
  levbp<-list("Yes"=1,"No"=2)
  levma_testdone<-list("Recorded"=1,"No Recorded"=2)
  levhba1c_done=list("Yes"=1,"No"=2)
  levma_done=list("Yes"=1,"No"=2)
  levfoot_done=list("Yes"=1,"No"=2)
  leveye_done=list("Yes"=1,"No"=2)
  levsmoke_doc=list("Documented"=1,"Not Documented"=2)
  levcreat_done=list("Yes"=1,"No"=2)
  levlipids=list("Yes"=1,"No"=2)
  levhypert_med=list("Yes"=1,"No"=2)
  levpump<-list("Yes"=1,"No"=2)
  levdiet<-list("Yes"=1,"No"=2)
  levtablet<-list("Yes"=1,"No"=2)
  levinsulin<-list("Yes"=1,"No"=2)
  levinstab<-list("Yes"=1,"No"=2)
  levpump_med<-list("Yes"=1,"No"=2)
  levbp12<-list("Yes"=1,"No"=2)
  levoral_th12<-list("Sulphonylureas"=1,"Biguanides"=2,"Glucosidase Inhib."=3,
"Glitazones"=4,"Glinides"=5)
  levinsulin12<-list("Treated"=1,"Not Treated"=2)
  levinsOAD<-list("Treated"=1,"Not Treated"=2)
  levinspump<-list("Yes"=1,"No"=2)
  levhypert12<-list("Yes"=1,"No"=2)
```

```
  levhba1c_c9<-list("Yes"=1,"No"=2)
  levcurrentsmoker<-list("Yes"=1,"No"=2)
  levlipid_med<-list("Yes"=1,"No"=2)
  levasa_med<-list("Yes"=1,"No"=2)
  levself_mon<-list("Urine"=1,"Blood Glucose"=2,"Both"=3)
  levvisit_freq<-list("Only one"=1,"More than one"=2)
  levhypert<-list("Yes"=1,"No"=2)
  levbp_done<-list("Yes"=1,"No"=2)
  levdiatrans<-list("Yes"=1,"No"=2)

# tests

  levhba1c_test<-list("at least one test"=1,"no test"=2)
  levma_test<-list("at least one test"=1,"no test"=2)
```

```
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-02-12
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_aggregate.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#  BIRO_aggregate
#
#
################################################################################

################################################################################
#  BIRO_aggregate                                                              #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-02-12
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
#
# Splits the data into subsets, computes summary statistics for each,
# and returns the result in a convenient form.
# This is also helpful, if you want to apply this linear function to date format
# variables, and create date format result variable.
#
# DEPENDENCIES
#
#  R package: stats
#
################################################################################
# Parameters(!=required)
#
# data       !       = the dataset processed
# request    !       = the variablesin which data are splitted by
# fun        !       = a scalar function to compute the summary statistics
#                      which can be applied to all data subsets
# vard       !       = the target variable
# isdate     [FALSE] = logical. TRUE if vard is in date format, FALSE otherwise
# year_epi   !       = variable containing the yaer of the episodes
# create_year[FALSE] = logical. TRUE if you need to create the year of episodes
#                      inside the function and this is not in the dataset, FALSE
#                      otherwise
#
################################################################################
#
# Example
#
#patient_id<-as.factor(rbinom(50,4,0.5))
#episode_date<-format(as.Date("1900-01-01") +36500+rnorm(50,2000,500),
#                     format="%Y-%m-%d")
#episode_field_value=rnorm(50,2000,500)
#year_episode<-as.numeric(format(as.Date(as.character(episode_date),
#                                format="%Y-%m-%d"),format="%Y"))
#
#data<-as.data.frame(cbind(patient_id,episode_date,
#                          episode_field_value,year_episode))
#Checks to do before
#
#data$patient_id=as.numeric(as.character(as.factor(data$patient_id)))
#data$episode_date=as.Date(as.character(data$episode_date))
#data$year_episode=as.numeric(as.character(data$year_episode))
#data$episode_field_value=as.numeric(as.character(data$episode_field_value))
#
#BIRO_aggregate(data=data,
#               request=c("patient_id"),
#               fun="mean",
#               vard="episode_field_value",
#               isdate=TRUE)
#
################################################################################

BIRO_aggregate<-function(data,
                         request,
                         fun,
                         vard,
                         isdate=FALSE,
                         year_epi,
                         create_year=FALSE) {

 # write("BIRO_aggregate called",file="")
 # if (nchar(logfile)>0) {
 #  write("BIRO_aggregate called",file=logfile,append=TRUE)
 # }

data<-data[is.na(data[,pmatch(vard,names(data))])==FALSE,]

if (create_year==TRUE) {
  data$year_epi=as.numeric(format(as.Date(as.character(data[,pmatch("epi_date",
```

```
                        names(data))]),
                        format="%Y-%m-%d"),format="%Y"))

}


if (fun!="mean") {

 if (is.na(pmatch("year_epi",request))==FALSE) {
  year="year_"
 } else {
  year=""
 }


 temp<-data[,c(pmatch(request,names(data)),pmatch(vard,names(data)))]
 if (fun=="max") {

 names(temp)[pmatch(vard,names(temp))]<-paste("last_",year,vard,sep="")

 vard<-paste("last_",year,vard,sep="")

 }
 if (fun=="min") {
 names(temp)[pmatch(vard,names(temp))]<-paste("first_",year,vard,sep="")

 vard<-paste("first_",year,vard,sep="")

 }
}

if (fun=="mean") {
 temp<-data[,c(pmatch(request,names(data)),pmatch(vard,names(data)))]
 names(temp)[pmatch(vard,names(temp))]<-paste("mean_",vard,sep="")
 vard<-paste("mean_",vard,sep="")
}



request2<-list()
for (i in 1:length(request)) {
request2[[i]]<-as.character(temp[,pmatch(request[i],names(temp))])

}

names(request2)<-request


temp<-aggregate(temp,request2,fun)

temp<-as.data.frame(cbind(temp[,1:length(request2)],
                          temp[,pmatch(vard,names(temp))]))
names(temp)<-c(names(request2),vard)

if (isdate) {

 temp[,pmatch(vard,names(temp))]<-as.Date("1970-01-01"
)+as.numeric(as.character(temp[,pmatch(vard,names(temp))]))
}

  #write("BIRO_aggregate finished",file="")
  #if (nchar(logfile)>0) {
  # write("BIRO_aggregate finished",file=logfile,append=TRUE)
  #}


return(temp)


}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_demographic.r                                                     #
#                                                                              #
#
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-02-14
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_demographic.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#  BIRO_demographic
#
#
################################################################################


################################################################################
#  BIRO_demographic                                                            #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-02-14
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# BIRO_demographic calculates the value of target variables, at a specified
# date, for each patient
#
# DEPENDENCIES
#
#  BIRO_aggregate
#
################################################################################
# PARAMETERS(!=required)
#
# ! patient   : the dataset processed containing patients data
# ! episode   : the dataset processed containing episodes data
# ! var       : the target variables names, which
#               can be applied to all data subsets
# ! date      : date of measurement
#               valid date or vector (for separate date for each patient)
#                 in the format DD/MM/YYYY [e.g. c("31/12/2006","31/12/2007")]
#               "last episode date"
#               "analysis date"
#               "last episode date for year"
#               "first episode date"
# ! startvar  : the initial values of a specific time intervals
#               for example:dob for age)
# ! th        : thresholds for a numeric variable
# ! hold      : logical. TRUE if you want to keep in the return dataset
#               var and date
# ! varname   : the name of the variable created by this function
#
################################################################################
#
# EXAMPLE
#
# This is the cohort for the BIRO indicator 1.1
#
# BIRO_demographic(patient=patient,episode=episode,
# var=c("sex","age_c"),date=refdate,startvar="dob",
# th=ageth,hold=FALSE,varname="age")
################################################################################


BIRO_demographic<-function(patient,episode,var,date,startvar,th,hold,varname) {


  #write("BIRO_demographic called",file="")
  #if (nchar(logfile)>0) {
  # write("BIRO_demographic called",file=logfile,append=TRUE)
  #}

no_last_epi=0
if ((as.character(date)=="analysis date") || (nchar(date)<=10)) {

 if (nchar(date)<=10) {
   date<-as.Date(date,"%d/%m/%Y")
 }

 if (as.character(date)=="analysis date") {
  date<-as.Date(Sys.Date())
 }
```

```
patient$myvar<-as.numeric(difftime(date,as.Date(as.character(patient[,pmatch(startvar,
names(patient))])),units="days"))/365.25
 patient$myvar_c<-as.factor(varclass(patient$myvar,th))

 no_last_epi=1

}

if (as.character(date)=="episode date") {

 patient<-merge(patient,episode,by="patient_id",all=FALSE)

patient$myvar<-as.numeric(difftime(as.Date(as.character(patient$epi_date)),as.Date(as.
character(patient[,pmatch(startvar,names(patient))])),units="days"))/365.25
 patient$myvar_c<-as.factor(varclass(patient$myvar,th))

  no_last_epi=1

}


if (as.character(date)=="last episode date") {

 last_epi<-BIRO_aggregate(data=episode,request=c("patient_id"),fun="max",vard=
"epi_date",isdate=TRUE)

 patient<-merge(patient,last_epi,by="patient_id",all=FALSE)

patient$myvar<-as.numeric(difftime(as.Date(as.character(patient$last_epi_date)),as.Dat
e(as.character(patient[,pmatch(startvar,names(patient))])),units="days"))/365.25
 patient$myvar_c<-as.factor(varclass(patient$myvar,th))
}

if (as.character(date)=="first episode date") {

 first_epi<-BIRO_aggregate(data=episode,request=c("patient_id"),fun="min",vard=
"epi_date",isdate=TRUE)
 patient<-merge(patient,first_epi,by="patient_id",all=FALSE)

patient$myvar<-as.numeric(difftime(as.Date(as.character(patient$first_epi_date)),as.Da
te(as.character(patient[,pmatch(startvar,names(patient))])),units="days"))/365.25
 patient$myvar_c<-varclass(patient$myvar,th)

}


if (as.character(date)=="last episode date for year") {



 episode$year_epi=as.numeric(format(as.Date(as.character(episode[,pmatch("epi_date"
,names(episode))]),format="%Y-%m-%d"),format="%Y"))
 last_epi<-BIRO_aggregate(data=episode,request=c("patient_id","year_epi"),fun="max"
,vard="epi_date",isdate=TRUE,create_year=TRUE)

 patient<-merge(patient,last_epi,by="patient_id",all=FALSE)

patient$myvar<-as.numeric(difftime(as.Date(as.character(patient$last_year_epi_date)),a
s.Date(as.character(patient[,pmatch(startvar,names(patient))])),units="days"))/365.25
 patient$myvar_c<-as.factor(varclass(patient$myvar,th))
}
 names(patient)[pmatch("myvar",names(patient))]<-varname
 names(patient)[pmatch("myvar_c",names(patient))]<-paste(varname,"_c",sep="")

#Select target variables


if (hold==TRUE) {
 if (no_last_epi==0) {
  var<-c(var,names(last_epi))
 }
}
 #write("BIRO_demographic finished",file="")
 # if (nchar(logfile)>0) {
 #  write("BIRO_demographic finished",file=logfile,append=TRUE)
 # }


return(patient[,pmatch(var,names(patient))])

}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_explife.r                                                         #
#                                                                              #
#
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-08-30
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_explife.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#  BIRO_explife
#
#
# ##############################################################################


# ##############################################################################
#  BIRO_explife                                                                #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-08-30
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# BIRO_explife calculates the life expectancy (age in classes)
#
# DEPENDENCIES
#
# R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# pop       !     = the dataset processed containing population data
# mor       !     = the dataset processed containing mortality data
# agewidth  !     = the width of age classes
# sex       !     = the last part of sex variable indicating if Female or Male
# morname   !     = the initial part of sex variable containing mortality data
# popname   !     = the initial part of sex variable containing population data
# labsex    !     = a string value for the sex label
#
#
# ##############################################################################
#
# EXAMPLE
#
#    exm<-BIRO_explife(pop=pop,
#                      mor=pop,
#                      agewidth=5,
#                      sex="M",
#                      morname="mor",
#                      popname="pop",
#                      labsex='Males')
#
#    data.frame(pop):
#
#     age_band    tot   popM   popF  morM  morF
#            1  37578  19356  18289     8    14
#            2  36865  18623  17240     3     0
#            3  43135  18641  17562     5     1
#            .      .      .      .     .     .
#            .      .      .      .     .     .
#            .      .      .      .     .     .
#
#
# ##############################################################################

BIRO_explife<-function(pop,mor,agewidth,ageth,sex,morname,popname,labsex) {

 #write("BIRO_explife called",file="")
 #if (nchar(logfile)>0) {
 # write("BIRO_explife called",file=logfile,append=TRUE)
 #}

 mx<-rep(NA,sum(agewidth))
 j=1
 for (i in 1:sum(agewidth)) {
  if (i%%sum(agewidth[1:j])==0) j<-j+1
   mx[i]<-mor[j,pmatch(paste(morname,sex,sep=""),names(mor))]/agewidth[j]
 }


mx<-100000*(mx/sum(mx))
```

```
 lx<-rep(NA,sum(agewidth))
 j=1
 for (i in 1:sum(agewidth)) {
  if (i%%sum(agewidth[1:j])==0) j<-j+1
  if (i==1) lx[1]<-100000
  if (i>1)  lx[i]= lx[i-1]-mx[i]
  #-(mor[j,pmatch(paste(morname,sex,sep=""),names(mor))]/agewidth[j])
 }

# lx<-rep(NA,length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))]))
# for (i in 1:(length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])[1])) {
#  if (i==1) lx[1]<-100000
#  if (i>1)  lx[i]= lx[i-1]-mor[,pmatch(paste(morname,sex,sep=""),names(mor))][i-1]
# }
 qx<-rep(NA,sum(agewidth))
 for (i in 1:sum(agewidth)) {
  qx[i]= mx[i]/lx[i]
 }
 ndx<-rep(NA,sum(agewidth))
 for (i in 1:sum(agewidth)) {
  ndx[i]<-lx[i]-lx[sum(agewidth)]
 }
 npx<-rep(NA,sum(agewidth))
 for (i in 1:sum(agewidth)) {
  npx[i]<-lx[i+1]/lx[i]
 }
# Lx<-rep(NA,length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))]))
# for (i in 1:length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])) {
#  if (i<length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))]))
Lx[i]=(lx[i]+lx[i+1])/2
#  if (i==length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])) Lx[i]<-NA
# }
# print(Lx)
# Tx<-rep(NA,length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))]))
# for (i in 1:length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])) {
#
Tx[i]=sum(Lx[i:(length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])-1)],na.rm=T
RUE)
# }
# print(Tx)

# ex<-rep(length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))]))
# for (i in 1:length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])) {
#  ex[i]=(Tx[i]/lx[i])*agewidth[i]
# }
# print(ex)

 ex<-rep(NA,sum(agewidth))
 for (i in 1:sum(agewidth)) {
  ex[i]<-sum(npx[i:length(ex)],na.rm=TRUE)
 }

 agelab<-classlabel(ageth)

 ind_ex<-rep(NA,length(agewidth))
 for (i in 1:length(agewidth)) {
  ind_ex[i]=sum(agewidth[1:i]+1)
 }

 retdata<-as.data.frame(cbind(agelab,round(ex[ind_ex])))
 names(retdata)<-c('Age',labsex)
 return(retdata)

 #write("BIRO_explife finished",file="")
 #if (nchar(logfile)>0) {
 # write("BIRO_explife finished",file=logfile,append=TRUE)
 #}

}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_forest.r                                                          #
#                                                                              #
#
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-00-00
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_forest.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#  BIRO_forest
#
#
# ##############################################################################


# ##############################################################################
#  BIRO_forest                                                                 #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2002-00-00
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# Draw forest plot

# DEPENDENCIES
#
#  R packages: grDevice, rmeta, Cairo
#
# ##############################################################################
# PARAMETERS(!=required)
# tabletext            !  = a table with labels
# values               !  = a vector with central values
# lbound               !  = a vector with lower values
# ubound               !  = a vector with upper values
# limits      [c(0,200)]  = a vector with axis limits
# xlab                 !  = a string containing the X label
# col                  !  = See meta.colors
# zero                 !  = x-axis coordinate for zero line
# csvindex             !  = the coded name fo the csv file
#
# ##############################################################################

BIRO_forest<-function(tabletext,
                      values,
                      lbound,
                      ubound,
                      limits=c(0,200),
                      xlab,
                      col,
                      zero,
                      csvindex,
                      dirgraph,
                      cex,
                      is.summary,
                      graphwidth) {

# write("Preparing to do SVG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do SVG file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/for",csvindex,".svg",sep="")
# CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")

forestplot(labeltext=as.data.frame(tabletext),mean=values,lower=lbound,upper=ubound,cl
ip=limits,
           xlab=xlab,col=col,zero=zero,cex=cex,is.summary=is.summary)
# dev.off()

 write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }
# write("Preparing to do JPEG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do JPEG file",file=logfile,append=TRUE)
# }
 file = paste(dirgraph,"/for",csvindex,".jpeg",sep="")
 jpeg(file, width= 813.6, height= 348.48, units="px",pointsize = 9,
      bg = "transparent")
```

```
 forestplot(as.data.frame(tabletext),values,lbound,ubound,clip=limits,
           xlab=xlab,col=col,zero=zero,cex=cex,is.summary=is
.summary,graphwidth=graphwidth)
 dev.off()

 write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }
# write("Preparing to do PNG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do PNG file",file=logfile,append=TRUE)
# }


 file = paste(dirgraph,"/for",csvindex,".png",sep="")
 CairoPNG(file, width= 813.6, height= 348.48, pointsize=8, bg = "transparent")
 forestplot(as.data.frame(tabletext),values,lbound,ubound,clip=limits,
           xlab=xlab,col=col,zero=zero,cex=cex,is.summary=is
.summary,graphwidth=graphwidth)
 dev.off()

 write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }
# write("Preparing to do PDF file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do PDF file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/for",csvindex,".pdf",sep="")
 pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
 forestplot(as.data.frame(tabletext),values,lbound,ubound,clip=limits,
           xlab=xlab,col=col,zero=zero,cex=cex,is.summary=is
.summary,graphwidth=graphwidth)
 dev.off()
 write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }
 data<-cbind(tabletext,values,lbound,ubound)
 write.csv(data,
           paste(dirdataout,"/for",csvindex,".csv",sep=""),
           row.names = FALSE)
 write(paste("Forest plot data saved into: ",dirdataout,"/for",csvindex,".csv",sep=""
),file="")
 if (nchar(logfile)>0) {
  write(paste("Forest plot data saved into: ",dirdataout,"/for",csvindex,".csv",sep=""
),file=logfile,append=TRUE)
 }
 if (struc==TRUE) {
  cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
  cat("<tbody>",file=sthtml,fill=TRUE,append=TRUE)
  cat("<tr>",file=sthtml,fill=FALSE,append=TRUE)
  cat(paste("<td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/for"
,namegraph,".png\">","</td>",sep=""),file=sthtml,fill=FALSE,append=TRUE)
  cat("</tr>",file=sthtml,fill=TRUE,append=TRUE)
  cat("</tbody>",file=sthtml,fill=TRUE,append=TRUE)
  cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
 }

}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_maps.r                                                            #
#                                                                              #
#
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-08-10
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_maps.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#  BIRO_map
#  BIRO_patmap
#
################################################################################


################################################################################
#  BIRO_map                                                                    #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-08-10
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
#
# DESCRIPTION
#
# To plot a map with desidered values
#
# DEPENDENCIES
#
#  R packages: maptools,sp,lattice,Cairo
#
################################################################################
# PARAMETERS(!=required)
#
# shape            ! = the name the the S4 object (shape file loaded)
# datavalues       ! = the data.frame containing the value that you want to represent
# frqvar           ! = the variable the variable that you want to represent
# shapearea_id     ! = the id in the S4 of the areas
# valuesarea_id    ! = the id in the values data.frame of the areas
# labvar           ! = label written in the legend
# levels           ! = a vector containing the threshold for the map
# legendpos        ! = the position of the legend
# dirgraph               !  = a string containing the path of the output graph
# namegraph              !  = a string containing the name of the graph
#
################################################################################

BIRO_map<-function(shape,
                   datavalues,
                   frqvar=region$POP2001,
                   shapearea_id,
                   valuesarea_id,
                   labvar="Population",
                   breaks=c(5000,10000),
                   legendpos="topright",
                   dirgraph,
                   namegraph) {

 onlyarea<-unique(datavalues[,pmatch(valuesarea_id,names(datavalues))])
 myshapefile=shape[shape$ID%in%onlyarea,]
 shape$myid=1:length(shape$ID)
 shape$ID<-as.character(shape$ID)
 assign("datavalues",datavalues,envir=.GlobalEnv)
 assign("myshapefile",myshapefile,envir=.GlobalEnv)


 shapedata<-merge(shape,datavalues,by.x="ID"
,by.y=valuesarea_id,all.x=FALSE,all.y=FALSE)
 na.omit(shapedata)
 shapedata$myarea<-varclass(shapedata[,pmatch(frqvar,names(shapedata))],breaks)
 cols<-rainbow(length(breaks)+1,start = 0.4)

 datacol<-as.data.frame(cols)
 names(datacol)<-"col"
 datacol$col<-as.character(datacol$col)
 datacol$lab<-classlabel(breaks)
 datacol$idc<-1:(length(breaks)+1)
 datacol$idcolor<-1:(length(breaks)+1)

 datacol<-merge(shapedata,datacol,by.x="myarea",by.y="idc",all.x=TRUE,all.y=FALSE)
 datacol<-sort.data.frame(datacol,by=~myid)
 file = paste(dirgraph,"/i",namegraph,"g4_8a.png",sep="")

 CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
```

```
 plot(myshapefile,col=datacol$col)
 datacol<-sort.data.frame(datacol,by=~idcolor)
 legend(legendpos,legend=classlabel(breaks),cex=cex,fill=cols,title=labvar)

 dev.off()

 file = paste(dirgraph,"/i",namegraph,"g4_8a.pdf",sep="")
 pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
 plot(myshapefile,col=datacol$col)
 datacol<-sort.data.frame(datacol,by=~idcolor)
 legend(legendpos,legend=classlabel(breaks),cex=cex,fill=cols,title=labvar)
 dev.off()

 file = paste(dirgraph,"/i",namegraph,"g4_8a.svg",sep="")
 CairoSVG(file, width = 11.33, height =4.84, pointsize=9,  bg = "transparent")
 plot(myshapefile,col=datacol$col)
 datacol<-sort.data.frame(datacol,by=~idcolor)
 legend(legendpos,legend=classlabel(breaks),cex=cex,fill=cols,title=labvar)
 dev.off()

 file = paste(dirgraph,"/i",namegraph,"g4_8a.jpeg",sep="")
 jpeg(file, width= 813.6, height= 348.48, pointsize=9, units="px",
      bg = "transparent")
 plot(myshapefile,col=datacol$col)
 datacol<-sort.data.frame(datacol,by=~idcolor)
 legend(legendpos,legend=classlabel(breaks),cex=cex,fill=cols,title=labvar)
 dev.off()



}

################################################################################
#  BIRO_patmap                                                                 #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-08-10
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# Manage the plotting of maps (absolute frequencies or percentages)
#
# DEPENDENCIES
#
#  R packages: maptools,sp,lattice,Cairo
#
################################################################################
# PARAMETERS(!=required)
#
# ind              ! = the code of the BIRO indicator
# shape            ! = the name the the S4 object (shape file loaded)
# data             ! = the data.frame containing the value that you want to represent
# lev              !  = a list contaning levels of variables specified in var
# var              ! = a vector containing the names of variables
# frqvar           ! = the variable the variable that you want to represent
# shapearea_id     ! = the id in the S4 of the areas
# valuesarea_id    ! = the id in the values data.frame of the areas
# labvar           ! = label written in the legend
# stat             ! = a sting containing "freq" or "perc"
# legendpos        ! = the position of the legend
# dirgraph               !  = a string containing the path of the output graph
# namegraph              !  = a string containing the name of the graph
#
################################################################################

BIRO_patmap<-function(ind,
                      shape,
                      data,
                      lev,
                      var,
                      labvar,
                      stat,
                      legendpos="topright",
                      dirgraph,
                      namegraph,
                      frqvar=region$POP2001,
                      shapearea_id,
                      valuesarea_id) {


 myFreqs<-list()
 allFreqs<-as.data.frame(NULL)
 for (i in 1:length(stat)) {
  if (stat=="freq") {
   for (j in 1:length(var)) {

splitdata<-split(data[,pmatch(valuesarea_id,names(data))],list(data[,pmatch(var[i],nam
es(data))]))
    for (k in 1:length(unique(data[,pmatch(var[i],names(data))]))) {
     Freqs<-as.data.frame(table(splitdata[[k]]))
     names(Freqs)[pmatch("Freq",names(Freqs))]<-"n"
     names(Freqs)[pmatch("Var1",names(Freqs))]<-"id_geo"
     Freqs$centre_id=centre_id
     Freqs$sum=sum(Freqs$n)
     Freqs$value=unique(data[,pmatch(var[i],names(data))])
     names(Freqs)[pmatch("value",names(Freqs))]<-var[i]
     Freqs$id=centre_id

     write.csv(Freqs,paste(dirdataout,"/i",ind,letters[j],"d4_8a.csv",sep=""
),row.names=FALSE)

     if (length(unique(Freqs$n))<=4) {
      breaks<-mean(unique(Freqs$n))
     }
     if (length(unique(Freqs$n))>4) {
      step<-(min(unique(Freqs$n))-max(unique(Freqs$n)))/5
      breaks<-round(seq(from
=min(unique(Freqs$n))+step,to=max(unique(Freqs$n))-step,by=step))
     }
     namegraph<-paste(namegraph,letters[k],sep="")
     BIRO_map(shape=shape,
              datavalues=Freqs,
              frqvar="n",
              shapearea_id=shapearea_id,
              valuesarea_id="id_geo",
              labvar=labvar,
              breaks=breaks,
```

```
            legendpos="topright",
            dirgraph=dirgraph,
            namegraph=namegraph)


      import_pdf(namegraph=paste("i",namegraph,"g4_8a.png",sep=""),
                 caption=paste("Map:",labvar[j],lev[[j]][k],sep=""),
                 width=1,
                 dirgraph=dirgraphlatex,
                 texfile=texfile)

      import_pdf_html(image=paste("../graphs/i",namegraph,"g4_8a.png",sep=""),
                      htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                      caption=paste("Map: ",labvar,sep=""))

   }


#   for (k in 2:length(unique(data[,pmatch(var[i],names(data))]))) {
#    allFreqs<-rbind(allFreqs,myFreqs[[k]])
#   }
#   allFreqs<-as.data.frame(allFreqs)
#   assign("allFreqs",allFreqs,envir = .GlobalEnv)


  }
 }
 }
}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_se_trend.r                                                        #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.0
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_se_trend.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_plotlines
#   BIRO_table_print
#   BIRO_plotbox
#
################################################################################

################################################################################
#  BIRO_plotlines                                                              #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
## ------------------------
# DESCRIPTION
#
#BIRO_plotlines drawing lines into a graph
#
# DEPENDENCIES
#
#  R packages: grDevice, graphics, Cairo
#
################################################################################
# PARAMETERS(!=required)
#
#data       !             = dataset containing data prepared by BIRO_lines
#lev        !             = a list containing the vactor with che levels of the
#                          categorical variable (list)
#descvar    !             = string for the x axe
#dirgraph   !             = the path of the output (jpg,eps,png files)
#namegraph  !             = the coded name of the output files
#
################################################################################


BIRO_plotlines<-function(data,
                         lev,
                         descvar,
                         namegraph,
                         dirgraph) {


#X11(width = 21, height =9 , pointsize=9)

#Confidence Intervals
  data$Lower=as.numeric(as.character(data$y))-
             1.65*as.numeric(as.character(data$sd))
  data$Upper=as.numeric(as.character(data$y))+
             1.65*as.numeric(as.character(data$sd))

#  write("Preparing to do SVG file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do SVG file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/","/i",namegraph,"g4_5a.svg",sep="")

  CairoSVG(file, width = 11.33, height =4.84, pointsize=9,  bg = "transparent")
#First line
  mins<-min(data$Lower,na.rm=TRUE)-sd(data$y,na.rm=TRUE)
  maxs<-max(data$Upper,na.rm=TRUE)+3*nlevels(as.factor(data$strata))

  if (is.na(mins)) mins<-0
  if (is.na(maxs)) maxs<-100
  plot(data$times[data$strata==levels(as.factor(data$strata))[1]],
       data$y[data$strata==levels(as.factor(data$strata))[1]],
       type="l",
       col=rainbow(nlevels(as.factor(data$strata)),start=.2,end=.8,gamma=1)[1],
       ylim=c(mins,
              maxs),
       xlab=descvar[2],
       ylab="",
       lty=2,
       cex=cex)
```

```
#Confidence boundary for the first line
  for (j in 1:dim(data)[1]) {

   if (data$strata[j]==levels(as.factor(data$strata))[1]) {
     text(data$time[j],
          data[j,pmatch("Lower",names(data))],"|-----|",
          col=rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1)[1],
          lty=2,
          lwd=2)
     text(data$time[j],
          data[j,pmatch("Upper",names(data))],"|-----|",
          col=rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1)[1],
          lty=2,
          lwd=2)
   }

}


#Further lines
  for (i in 2:nlevels(as.factor(data$strata))) {
   lines(data$time[data$strata==levels(as.factor(data$strata))[i]],
         data$y[data$strata==levels(as.factor(data$strata))[i]],
         col=rainbow(length(lev),,start=0.2,end=.8,gamma=1)[i],
         lty=2,
         lwd=2,
         cex=cex)

   for (j in 1:dim(data)[1]) {
    if (data$strata[j]%in%levels(as.factor(data$strata))[i]) {
         text(data$time[j],
              data[j,pmatch("Lower",names(data))],"|-----|",
              col=rainbow(length(lev),,start=0.2,end=.8,gamma=1)[i],
              lty=2,
              lwd=2)
         text(data$time[j],
              data[j,pmatch("Upper",names(data))],"|-----|",
              col=rainbow(length(lev),start=0.2,end=.8,gamma=1)[i],
              lty=2,
              lwd=2)
    }

   }


  }

  axis(1,levels(data$times))
  leg<-rep(NA,nlevels(as.factor(data$strata)))
  for (i in 1:nlevels(as.factor(data$strata))) {
   leg[i]<-lev[i]
  }
  x<-legend("topright", leg,
            col = rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1),
            lty=2, lwd=2,pt.bg="white",cex=cex,title=descvar)

  dev.off()

  write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

#  write("Preparing to do JPEG file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do JPEG file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/","/i",namegraph,"g4_5a.jpeg",sep="")
  jpeg(file, width= 813.6, height= 348.48, pointsize=9,
       units="px", bg = "transparent")
#First line
  plot(data$times[data$strata==levels(as.factor(data$strata))[1]],
       data$y[data$strata==levels(as.factor(data$strata))[1]],
       type="l",
       col=rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1)[1],
       ylim=c(mins,
              maxs),
       xlab=descvar[2],
       ylab="",
       lty=2,
      cex=cex)

  #Confidence boundary for the first line
  for (j in 1:dim(data)[1]) {

   if (data$strata[j]==levels(as.factor(data$strata))[1]) {
     text(data$time[j],
          data[j,pmatch("Lower",names(data))],"|-----|",
          col=rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1)[1],
          lty=2,
          lwd=2)
     text(data$time[j],
          data[j,pmatch("Upper",names(data))],"|-----|",
          col=rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1)[1],
          lty=2,
          lwd=2)
   }

  }


#Further lines
  for (i in 2:nlevels(as.factor(data$strata))) {
   lines(data$time[data$strata==levels(as.factor(data$strata))[i]],
       data$y[data$strata==levels(as.factor(data$strata))[i]],
       col=rainbow(length(lev),start=0.2,end=.8,gamma=1)[i],
       lty=2,
       lwd=2,
       cex=cex)

   for (j in 1:dim(data)[1]) {
    if (data$strata[j]%in%levels(as.factor(data$strata))[i]) {
     text(data$time[j],
          data[j,pmatch("Lower",names(data))],"|-----|",
          col=rainbow(length(lev),start=0.2,end=.8,gamma=1)[i],
          lty=2,
          lwd=2)
     text(data$time[j],
          data[j,pmatch("Upper",names(data))],"|-----|",
          col=rainbow(length(lev),start=0.2,end=.8,gamma=1)[i],
          lty=2,
```

```
          lwd=2)
     }
    }


  }
  axis(1,levels(data$times))
  leg<-rep(NA,nlevels(as.factor(data$strata)))
  for (i in 1:nlevels(as.factor(data$strata))) {
   leg[i]<-lev[i]
  }
  x<-legend("topright", leg, col =
rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1),
            lty=2, lwd=2,pt.bg="white",cex=cex,title=descvar)
  dev.off()

  write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

#  write("Preparing to do PNG file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do PNG file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/","/i",namegraph,"g4_5a.png",sep="")
  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
#First line
  plot(data$times[data$strata==levels(as.factor(data$strata))[1]],
       data$y[data$strata==levels(as.factor(data$strata))[1]],
       type="l",
       col=rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1)[1],
       ylim=c(mins,
              maxs),
       xlab=descvar[2],
       ylab="",
       lty=2,
       cex=cex)

#Confidence boundary for the first line
  for (j in 1:dim(data)[1]) {

   if (data$strata[j]==levels(as.factor(data$strata))[1]) {
     text(data$time[j],
          data[j,pmatch("Lower",names(data))],"|-----|",
          col=rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1)[1],
          lty=2,
          lwd=2)
     text(data$time[j],
          data[j,pmatch("Upper",names(data))],"|-----|",
          col=rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1)[1],
          lty=2,
          lwd=2)
   }

  }

#Further lines
  for (i in 2:nlevels(as.factor(data$strata))) {
   lines(data$time[data$strata==levels(as.factor(data$strata))[i]],
         data$y[data$strata==levels(as.factor(data$strata))[i]],
         col=rainbow(length(lev),start=0.2,end=.8,gamma=1)[i],
         lty=2,
         lwd=2,
         cex=cex)

   for (j in 1:dim(data)[1]) {
    if (data$strata[j]%in%levels(as.factor(data$strata))[i]) {
     text(data$time[j],
          data[j,pmatch("Lower",names(data))],"|-----|",
          col=rainbow(length(lev),start=0.2,end=.8,gamma=1)[i],
          lty=2,
          lwd=2)
     text(data$time[j],
          data[j,pmatch("Upper",names(data))],"|-----|",
          col=rainbow(length(lev),start=0.2,end=.8,gamma=1)[i],
          lty=2,
          lwd=2)
    }

   }


  }
  axis(1,levels(data$times))
  leg<-rep(NA,nlevels(as.factor(data$strata)))
  for (i in 1:nlevels(as.factor(data$strata))) {
   leg[i]<-lev[i]
  }
  x<-legend("topright", leg, col =
rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1),
            lty=2, lwd=2,pt.bg="white",cex=cex,title=descvar)
  dev.off()

  write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

#  write("Preparing to do PDF file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do PDF file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/","/i",namegraph,"g4_5a.pdf",sep="")
  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
  #First line
  plot(data$times[data$strata==levels(as.factor(data$strata))[1]],
       data$y[data$strata==levels(as.factor(data$strata))[1]],
       type="l",
       col=rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1)[1],
       ylim=c(mins,
              maxs),
       xlab=descvar[2],
       ylab="",
       lty=2,
       cex=cex)

#Confidence boundary for the first line
  for (j in 1:dim(data)[1]) {
   if (data$strata[j]==levels(as.factor(data$strata))[1]) {
```

```
    text(data$time[j],
         data[j,pmatch("Lower",names(data))],"|-----|",
         col=rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1)[1],
         lty=2,
         lwd=2)
    text(data$time[j],
         data[j,pmatch("Upper",names(data))],"|-----|",
         col=rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1)[1],
         lty=2,
         lwd=2)
   }

  }


#Further lines
  for (i in 2:nlevels(as.factor(data$strata))) {
   lines(data$time[data$strata==levels(as.factor(data$strata))[i]],
         data$y[data$strata==levels(as.factor(data$strata))[i]],
         col=rainbow(length(lev),start=0.2,end=.8,gamma=1)[i],
         lty=2,
         lwd=2,
         cex=cex)

   for (j in 1:dim(data)[1]) {
    if (data$strata[j]%in%levels(as.factor(data$strata))[i]) {
      text(data$time[j],
           data[j,pmatch("Lower",names(data))],"|-----|",
           col=rainbow(length(lev),start=0.2,end=.8,gamma=1)[i],
           lty=2,
           lwd=2)
      text(data$time[j],
           data[j,pmatch("Upper",names(data))],"|-----|",
           col=rainbow(length(lev),start=0.2,end=.8,gamma=1)[i],
           lty=2,
           lwd=2)
    }


   }


  }

  axis(1,levels(data$times))
  leg<-rep(NA,nlevels(as.factor(data$strata)))
  for (i in 1:nlevels(as.factor(data$strata))) {
   leg[i]<-lev[i]
  }
  x<-legend("topright", leg,
            col =
rainbow(nlevels(as.factor(data$strata)),start=0.2,end=.8,gamma=1),lty=2, lwd=2,
            pt.bg="white",cex=cex,title=descvar)
  dev.off()

  write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

  if (struc==TRUE) {
    cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
    cat("<tbody>",file=sthtml,fill=TRUE,append=TRUE)
    cat("<tr>",file=sthtml,fill=FALSE,append=TRUE)
    cat(paste("<td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/lines"
,namegraph,".png\">","</td>",sep=""),file=sthtml,fill=FALSE,append=TRUE)
    cat("</tr>",file=sthtml,fill=TRUE,append=TRUE)
    cat("</tbody>",file=sthtml,fill=TRUE,append=TRUE)
    cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
  }
  #print("BIRO Lines done")

}


################################################################################
#  BIRO_table_print                                                            #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# BIRO_table_print print table in html/latex2e format
#
# DEPENDENCIES
#
# R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#tab       !             = dataset as the type of the result dataset producted
#                          by the BIRO_tab
#var       !             = two or more vector with the values of the target
#                          variables, Passed as a list
#n         !             = a vector which each element is the number of levels
#                          of each categorical variables
#FirstLab  !             = the name of the first variable, the first variable
#                          of var is the vertical variable in the table
#lev       !             = a list containing the vactor with che levels of the
#                          categorical variable (list)
#texfile   !             = the path of the output texfile
#codtab    !             = the coded name of the output csv file
#dirtables !             = directory for the output html table file
#dircsv    !             = directory for the output csv file
#caption   !             = string for the caption
#side      [0]           = if side is egual to 1 to table in the pdf file is
#                          printed in portrait layout
#width     [1]           = width for the pdf (see latex for details)
# perc     [TRUE]        = TRUE for percentage
#dirout    [getwd()]     = output location
#out       [out]         = name of output csv dataset
#ext       [csv]         = output file format
#date      [Sys.Date()]  = date variable or value
#ind       !             = a string containing the index of indicator
```

```
#chisq                    = a vector containing the name of variables to
#                           calculate the Cochran-Mantel-Haenszel Chi-Squared
#                           Test for Count Data
##############################################################################


BIRO_table_print<-function(tab,
                           var,
                           n,
                           FirstLab,
                           lev=list(),
                           texfile,
                           codtab,
                           dirtables,
                           dircsv,
                           caption,
                           side=0,
                           width=1,
                           perc=TRUE,
                           ind=ind,
                           chisq=chisq,
                           title="",
                           codtex=NULL) {

FirstLablatex<-gsub("<","$<$",as.character(FirstLab))
FirstLablatex<-gsub(">","$>$",as.character(FirstLablatex))
FirstLablatex<-gsub("%","\\\\%",as.character(FirstLablatex))

captiontex<-gsub("<","$<$",as.character(caption))
captiontex<-gsub(">","$>$",as.character(captiontex))
captiontex<-gsub("%","\\\\%",as.character(captiontex))

if (is.null(codtex)) codtex<-codtab
dir.create(paste(dirreport,"/pdf/temp",sep=""), showWarnings = FALSE, recursive =
TRUE)
texfile<-paste(dirreport,"/pdf/temp/table",codtex,".tex",sep="")

dir.create(paste(dirreport,"/html/temp",sep=""), showWarnings = FALSE, recursive =
TRUE)
htmlclass<-paste(dirreport,"/html/temp/table",codtex,".html",sep="")
if (var[1]=="noNA") htmlfile<-paste(dirreport,"/html/temp/table",codtex,"m.html",sep=
"")

if (any(na.omit(tab$Freq)>0)) {


 sperc=""
 hsperc=""
 #tab<-data.frame(tab)
 nvar=length(var)

# INDEX<-list()
# for (i in 1:(nvar-1)) {
#  if (i==1) {j=i-1}
#   if (i==2) {
#    j=3
#   } else {j=j+1}
#  INDEX[[i]]<-tab[,j]
# }

 INDEX<-list()
 for (i in 1:(nvar-1)) {
  j=i+1
  INDEX[[i]]<-tab[,pmatch(var[j],names(tab))]

 }
 sums<-as.data.frame(aggregate(tab$Freq,INDEX,FUN="sum"))
 x<-1:nvar
 names(sums)[-nvar]<-var[-1]
 names(sums)[nvar]<-"sum"
 tab<-merge(sums,tab,by=names(sums)[-length(names(sums))])
 tab$perc<-round((tab$Freq/tab$sum)*100,1)
 sperc=" \\% "
 hsperc="(%)"
 #tab<-data.frame(tab)
 tab$Freq[is.na(tab$Freq)]<-0
 tab$perc[is.nan(tab$perc)]<-0
 tab$sum[is.na(tab$sum)]<-0
 #latex
  tab<-sort(tab, by = as.formula(paste("~",paste(var[c(1,nvar:2)],collapse="+"))))

# if (engine=="bu") {
#  indexa<-as.data.frame(rep(lev[[1]],each=length(lev[[1]])))
#  names(indexa)<-"a"
#  tempa<-rep(NA,dim(tab)[1])
#  for (r in dim(tab)[1]) {
#   tempa[r]<-substr(as.character(tab$a[r]),2,nchar(as.character(tab$a[r]))-1)
#  }
#  tab<-tab[-pmatch("a",names(tab))]
#  tab$a<-tempa
#  tab<-merge(indexa,tab,by="a",all.x=TRUE,all.y=FALSE)
#  tab$Freq[is.na(tab$Freq)]<-0
#  tab$perc[is.nan(tab$perc)]<-0
#  tab$sum[is.na(tab$sum)]<-0
#
# }
 l<-list()
 p<-list()
 s<-list()
 k=0
 for (i in 1:(dim(tab)[1]/n[1])) {
  l[[i]]<-tab$Freq[seq(i,dim(tab)[1],(dim(tab)[1]/n[1]))]
  p[[i]]<-tab$perc[seq(i,dim(tab)[1],(dim(tab)[1]/n[1]))]
  s[[i]]<-tab$sum[seq(i,dim(tab)[1],(dim(tab)[1]/n[1]))]
 }
 t<-unlist(l)
 t<-matrix(t,n[1],length(l))
 rowsum<-apply(t,1, sum)

 colsum<-apply(t,2, sum)
 for (i in 1:4) {
  if (is.na(n[i]==TRUE)) n[i]<-1
 }

# if (length(na.omit(var))==4 & nchar(title)>0) {
#  print(texfile)
#  print(var)
#    cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
#    print("luca1")
#     cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
#    print("luca2")
#     cat(" \\small BIRO REPORT - Section ",title,file=texfile,fill =
TRUE,append=TRUE)
#    print("luca3")
```

```
#     cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)
#    print("luca4")

# }

 cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
 if (side==1) {
  cat("\\begin{sidewaystable}",file=texfile,fill = TRUE,append=TRUE)
 }
 cat("\\parbox[2]{",width,"\\textwidth}{",file=texfile,fill = TRUE,append=TRUE)
 cat("\\resizebox{",width,"\\textwidth}{!}{",file=texfile,fill = TRUE,
     append=TRUE)
 cat("\\begin{tabular}{|c",rep("|r",prod(n[2:4])+(prod(n[2:4])/n[2])+1),"|}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\hline",file=texfile,fill = TRUE,append=TRUE)

 cat("\\multirow{",length(n[n>1])-1,"}{*}{",paste(FirstLablatex,sep=""),"}  "
,file=texfile,fill = FALSE,
     append=TRUE)

 if (n[4]>=1 & nvar==4) {
  cat("& \\multicolumn{",prod(n[2:3])+(prod(n[2:4])/n[2])+1,"}{|c|}{",
      lev[[4]][1],"} \\\\",file=texfile,fill = TRUE,append=TRUE)
  cat("\\cline{2-",prod(n[2:4])+(prod(n[2:4])/n[2])+2,"}",file=texfile,
      fill = TRUE,append=TRUE)
 }

#Terza variabile

if (nvar>=3) {
thislab3<-rep(NA,n[3])
for (i in n[3])
 if (substr(lev[[3]][i],1,1)=="[") {
  thislab3<-paste("\\[",lev[[3]],sep="")
 } else {
   thislab3<-lev[[3]]
 }


second_head<-rep(thislab3,prod(n[4]))
cat(" & ",file=texfile,fill = FALSE,append=TRUE)
if (n[3]>1) {
 for(i in 1:(length(second_head)-1)) {
  cat("\\multicolumn{",n[2]+1,"}{|c|}{",second_head[i],"} &",file=texfile,
      fill = FALSE,append=TRUE)
 }
}
 cat("\\multicolumn{",n[2]+1,"}{|c|}{",second_head[length(second_head)],
     "}  & \\\\",file=texfile,fill = TRUE,append=TRUE)

 cat("\\cline{2-",prod(n[2:4])+(prod(n[2:4])/n[2])+2,"}",file=texfile,
     fill = TRUE,append=TRUE)
}

if (n[2]>1) {
thislab2<-rep(NA,n[2])
for (i in n[2])
 if (substr(lev[[2]][i],1,1)=="[") {
  thislab2<-paste("\\[",lev[[2]],sep="")
 } else {
   thislab2<-lev[[2]]
 }


third_head<-rep(thislab2,prod(na.omit(n[3:4])))

for(i in 1:(length(third_head)-1)) {
cat(" & ",third_head[i],"(",sperc,")",file=texfile,fill = FALSE,append=TRUE)
if (i %in% seq(0,prod(n[2:nvar])-1,by=n[2])) {
cat("&  ",file=texfile,fill = FALSE,append=TRUE)
 }
}
cat(" & ",third_head[length(third_head)],"(",sperc,") &  & N ( \\% ) \\\\"
,file=texfile,
    fill = TRUE,append=TRUE)

}
cat("\\hline",file=texfile,fill = TRUE,append=TRUE)


for (i in 1:n[1]) {  #rows

#if (engine=="se") {
 if (substr(lev[[1]][i],1,1)=="[") {
  thislab<-paste("\\[",lev[[1]][i],sep="")
 } else {
   thislab<-lev[[1]][i]
 }
 cat(thislab,"  &",file=texfile,fill=FALSE,append=TRUE)
#}

#if (engine=="ce") {
# cat(tab$a[prod(n[2:4])*(i-1)+1],"  &",file=texfile,fill=FALSE,append=TRUE)
#}
for(j in 1:(length(l)-1)) {


cat(l[[j]][i]," (",format(p[[j]][i],nsmall=1,width=5),") & "
,file=texfile,fill=FALSE,append=TRUE,sep="")

if (j %in% seq(0,prod(n[2:nvar])-1,by=n[2])) {
 #cat(s[[j]][i],  " & ",file=texfile,fill=FALSE,append=TRUE)
 cat(" & ",file=texfile,fill=FALSE,append=TRUE)

}

}
#cat(l[[length(l)]][i],"(",p[[length(l)]][i],") & ",s[[length(l)]][i],  " & ",
#    rowsum[i],"  \\\\",file=texfile,fill=TRUE,append=TRUE)
 if (is.na(rowsum[i])==FALSE) {
  cat(l[[length(l)]][i],"(",format(p[[length(l)]][i],nsmall=1,width=5),") &  & ",
      rowsum[i]," (",format(round(rowsum[i]/sum(rowsum)*100,1),nsmall=1,width=5),")
\\\\",file=texfile,fill=TRUE,append=TRUE,sep="")
 }
 if (is.na(rowsum[i])==TRUE) {
  cat(l[[length(l)]][i],"(",format(round(p[[length(l)]][i],1),nsmall=1,width=5),") &
& ",
      0," ( 0.0 ) \\\\",file=texfile,fill=TRUE,append=TRUE,sep="")
 }


}
```

```
cat("\\hline",file=texfile,fill = TRUE,append=TRUE)
cat(" TOTAL & ",file=texfile,fill = FALSE,append=TRUE)

for (i in 1:length(colsum)) {  #totale in fondo

if (sum(colsum)>0) {
 cat(colsum[i],"(",format(round(colsum[i]/sum(colsum)*100,1),nsmall=1,width=5),")","&"
,file=texfile,fill = FALSE,append=TRUE,sep="")
}
if (sum(colsum)==0) {
 cat(colsum[i],"(  0.0)","&",file=texfile,fill = FALSE,append=TRUE)
}

  if (i %in% seq(0,prod(n[2:nvar])-1,by=n[2])) {

 cat(" & ",file=texfile,fill=FALSE,append=TRUE)

}

}
if (round(sum(colsum))==0) {
 cat("&",round(sum(colsum)),"  (  0.0) \\\\",file=texfile,fill = TRUE,append=TRUE)
}
if (round(sum(colsum))>0) {
 cat("&",round(sum(colsum)),"  (100.0) \\\\",file=texfile,fill = TRUE,append=TRUE)
}

cat("\\hline",file=texfile,fill = TRUE,append=TRUE)
cat("\\end{tabular}",file=texfile,fill = TRUE,append=TRUE)
cat("}",file=texfile,fill = TRUE,append=TRUE)
#cat("\\scriptsize",file=texfile,fill = TRUE,append=TRUE)
cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
cat(captiontex,file=texfile,fill = TRUE,append=TRUE)
cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)
cat("\\vspace{5mm}",file=texfile,fill = TRUE,append=TRUE)
cat("}",file=texfile,fill = TRUE,append=TRUE)
if (side==1) {
cat("\\end{sidewaystable}",file=texfile,fill = TRUE,append=TRUE)
}
cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)

# HTML table


#if (engine=="ce") {
# htmlfile=paste(dirtables,"/",codtab,".html",sep="")
#}
#if (engine=="se") {
 htmlfile=paste(dirtables,"/",codtab,"1_3a.html",sep="")
#}


#if (engine=="ce") {
# chicod<-paste(substr(codtab,1,nchar(codtab)-1),"b",sep="")
# htmlfilep=paste(dirtables,"/",chicod,".html",sep="")
#}

#if (engine=="se") {
 htmlfilep=paste(dirtables,"/",codtab,"1_3b.html",sep="")
#}
```

```
cat("",file=htmlfile,fill = TRUE,append=TRUE)
cat("     <table border=\"1\">",file=htmlfile,fill = TRUE,append=TRUE)
cat("",file=htmlfile,fill = TRUE,append=TRUE)
cat("      <tbody>",file=htmlfile,fill = TRUE,append=TRUE)
cat("",file=htmlfile,fill = TRUE,append=TRUE)
cat("       <thead>",file=htmlfile,fill = TRUE,append=TRUE)
cat("",file=htmlfile,fill = TRUE,append=TRUE)

rowspan<-length(na.omit(var))-1

cat("        <tr>",file=htmlfile,fill = TRUE,append=TRUE)
cat("         <th rowspan=\"",rowspan,"\">",FirstLab,"</th>",file=htmlfile,
    fill = TRUE,append=TRUE)

if (n[4]>1) {

for(i in 1:(n[4]-1)) {

cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][i],"</th>",file=htmlfile,
    fill = TRUE,append=TRUE)

}

cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][n[4]]," </th>",
    file=htmlfile,fill = TRUE,append=TRUE)
cat("         <th></th>",file=htmlfile,fill = TRUE,append=TRUE)
}

if ((length(lev)==4) & n[4]==1 & length(na.omit(unique(var)))>3) {
cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][1],"</th>",
    file=htmlfile,fill = TRUE,append=TRUE)
cat("         <th></th>",file=htmlfile,fill = TRUE,append=TRUE)
cat("         </tr>",file=htmlfile,fill = TRUE,append=TRUE)
#cat("         <tr>",file=htmlfile,fill = TRUE,append=TRUE)

}

if (nvar>=3) {
second_head<-rep(lev[[3]],prod(n[4]))
if (nvar==4) {
 cat("         <tr>",file=htmlfile,fill = TRUE,append=TRUE)
}
if (n[3]>1) {
 for(i in 1:(length(second_head)-1)) {
 cat("         <th colspan=\"",n[2],"\">",second_head[i],"</th>",file=htmlfile,
     fill = TRUE,append=TRUE)
 }
}
 cat("         <th colspan=\"",n[2],"\">",second_head[length(second_head)],
     "</th>",file=htmlfile,fill = TRUE,append=TRUE)
cat("         <th></th>",file=htmlfile,fill = TRUE,append=TRUE)
cat("         </tr>",file=htmlfile,fill = TRUE,append=TRUE)
}

if (n[2]>1) {
third_head<-rep(lev[[2]],prod(n[3:4]))
if (nvar==3) {

 cat("",file=htmlfile,fill = TRUE,append=TRUE)
 cat("        <tr>",file=htmlfile,fill = FALSE,append=TRUE)
}
if (n[3]>=1) {
```

```
 for(i in 1:(length(third_head)-1)) {
  cat("          <th>",third_head[i]," (%)</th>",file=htmlfile,fill =
TRUE,append=TRUE)
 }
}
cat("          <th>",third_head[length(third_head)]," (%)</th>",
    file=htmlfile,fill = TRUE,append=TRUE)
cat("          <th></th>",file=htmlfile,fill = TRUE,append=TRUE)
cat("         </tr>",file=htmlfile,fill = TRUE,append=TRUE)

}
cat("       </thead>",file=htmlfile,fill = TRUE,append=TRUE)

for (i in 1:n[1]) {  #rows

cat("",file=htmlfile,fill = TRUE,append=TRUE)
cat("        <tr>",file=htmlfile,fill=TRUE,append=TRUE)
#if (engine=="se") {
cat("         <td class=\"catcol\">",lev[[1]][i],"</td>"
,file=htmlfile,fill=TRUE,append=TRUE)
#}
#if (engine=="ce") {
# cat("          <td
class=\"catcol\">",tab$a[prod(n[2:4])*(i-1)+1],"</td>",file=htmlfile,fill=TRUE,append=
TRUE)
#}
cat("         <td>",file=htmlfile,fill=FALSE,append=TRUE)

for(j in 1:(length(l)-1)) {


cat(l[[j]][i]," (",format(round(p[[j]][i],1),nsmall=1,width=5),") </td> "
,file=htmlfile,fill=TRUE,append=TRUE,sep="")
cat("         <td>",file=htmlfile,fill=FALSE,append=TRUE)
}
if (sum(colsum)>0) {
 cat(l[[length(l)]][i]," (",format(round(p[[length(l)]][i],1),nsmall=1,width=5),")
<td class=\"rowsn\" > ",rowsum[i],"("
,format(round(100*rowsum[i]/round(sum(colsum)),digits=1),nsmall=1,width=5),")  </td>",
     file=htmlfile,fill=TRUE,append=TRUE,sep="")
}
if (sum(colsum)==0) {
 cat(l[[length(l)]][i]," (",format(round(p[[length(l)]][i],1),nsmall=1,width=5),")
<td class=\"rowsn\" > ",rowsum[i],"(",0,")  </td>",
     file=htmlfile,fill=TRUE,append=TRUE,sep="")
}

cat("        </tr>",file=htmlfile,fill=TRUE,append=TRUE)
}

cat("",file=htmlfile,fill = TRUE,append=TRUE)
cat("        <tr>",file=htmlfile,fill = TRUE,append=TRUE)
cat("         <td></td>",file=htmlfile,fill = TRUE,append=TRUE)
cat("         <td class=\"colsn\">",file=htmlfile,fill = FALSE,append=TRUE)
for (i in 1:length(colsum)) {  #total last row
 if (sum(colsum)>0) {
  cat(colsum[i]," ("
,format(round(100*(colsum[i]/round(sum(colsum))),1),nsmall=1,width=5),") </td>"
,file=htmlfile,fill = TRUE,append=TRUE)
 }
 if (sum(colsum)==0) {
  cat(0," (0.0) </td>",file=htmlfile,fill = TRUE,append=TRUE)
 }

 if (i<length(colsum)) {
  cat("         <td class=\"colsn\">",file=htmlfile,fill = FALSE,append=TRUE)
 }
 if (i==length(colsum)) {
  cat("         <td class=\"grandn\">",file=htmlfile,fill = FALSE,append=TRUE)
 }

}
cat(round(sum(colsum))," (100.0)</td>",file=htmlfile,fill = TRUE,append=TRUE)
cat("        </tr>",file=htmlfile,fill = TRUE,append=TRUE)
cat("",file=htmlfile,fill = TRUE,append=TRUE)
cat("        </tbody>",file=htmlfile,fill = TRUE,append=TRUE)
cat("",file=htmlfile,fill = TRUE,append=TRUE)
cat("       </table>",file=htmlfile,fill = TRUE,append=TRUE)
cat("       <caption>",caption,"</caption>",file=htmlfile,fill = TRUE,append=TRUE)
cat("        <br>",file=htmlfile,fill = TRUE,append=TRUE)
cat("        <br>",file=htmlfile,fill = TRUE,append=TRUE)

write(paste("HTML Table: [OUTDIR]/tables/",basename(htmlfile),sep=""),file="")


#Drupal (web portal)

wpfile=paste(dirwp,"/",ind,".html",sep="")

cat("",file=wpfile,fill = TRUE,append=TRUE)
cat("     <table border=\"1\">",file=wpfile,fill = TRUE,append=TRUE)
cat("",file=wpfile,fill = TRUE,append=TRUE)

rowspan<-length(na.omit(var))-1

cat("        <tr>",file=wpfile,fill = TRUE,append=TRUE)
cat("         <th rowspan=\"",rowspan,"\">",FirstLab,"</th>",file=wpfile,
    fill = TRUE,append=TRUE)

if (n[4]>1) {

for(i in 1:(n[4]-1)) {

cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][i],"</th>",file=wpfile,
    fill = TRUE,append=TRUE)

}

cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][n[4]]," </th>",
    file=wpfile,fill = TRUE,append=TRUE)
cat("         <th></th>",file=wpfile,fill = TRUE,append=TRUE)
}

if ((length(lev)==4) & n[4]==1 & length(na.omit(unique(var)))>3) {
cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][1],"</th>",
    file=wpfile,fill = TRUE,append=TRUE)
cat("         <th></th>",file=wpfile,fill = TRUE,append=TRUE)
cat("         </tr>",file=wpfile,fill = TRUE,append=TRUE)
#cat("         <tr>",file=wpfile,fill = TRUE,append=TRUE)

}

if (nvar>=3) {
second_head<-rep(lev[[3]],prod(n[4]))
```

```
if (nvar==4) {
 cat("         <tr>",file=wpfile,fill = TRUE,append=TRUE)
}
if (n[3]>1) {
 for(i in 1:(length(second_head)-1)) {
 cat("         <th colspan=\"",n[2],"\">",second_head[i],"</th>",file=wpfile,
     fill = TRUE,append=TRUE)
 }
}
 cat("         <th colspan=\"",n[2],"\">",second_head[length(second_head)],
     "</th>",file=wpfile,fill = TRUE,append=TRUE)
cat("         <th></th>",file=wpfile,fill = TRUE,append=TRUE)
cat("         </tr>",file=wpfile,fill = TRUE,append=TRUE)
}

if (n[2]>1) {
third_head<-rep(lev[[2]],prod(n[3:4]))
if (nvar==3) {

 cat("",file=wpfile,fill = TRUE,append=TRUE)
 cat("         <tr>",file=wpfile,fill = FALSE,append=TRUE)
}
if (n[3]>=1) {
 for(i in 1:(length(third_head)-1)) {
  cat("          <th>",third_head[i]," (%)</th>",file=wpfile,fill = TRUE,append=TRUE)
 }
}
cat("          <th>",third_head[length(third_head)]," (%)</th>",
    file=wpfile,fill = TRUE,append=TRUE)
cat("          <th></th>",file=wpfile,fill = TRUE,append=TRUE)
cat("         </tr>",file=wpfile,fill = TRUE,append=TRUE)

}

for (i in 1:n[1]) {  #rows

cat("",file=wpfile,fill = TRUE,append=TRUE)
cat("        <tr>",file=wpfile,fill=TRUE,append=TRUE)
#if (engine=="se") {
cat("         <td class=\"catcol\">",lev[[1]][i],"</td>"
,file=wpfile,fill=TRUE,append=TRUE)
#}
#if (engine=="ce") {
# cat("          <td
class=\"catcol\">",tab$a[prod(n[2:4])*(i-1)+1],"</td>",file=wpfile,fill=TRUE,append=TR
UE)
#}
cat("         <td>",file=wpfile,fill=FALSE,append=TRUE)

for(j in 1:(length(l)-1)) {


cat(l[[j]][i]," (",format(round(p[[j]][i],1),nsmall=1,width=5),") </td> "
,file=wpfile,fill=TRUE,append=TRUE,sep="")
cat("         <td>",file=wpfile,fill=FALSE,append=TRUE)
}
if (sum(colsum)>0) {
 cat(l[[length(l)]][i]," (",format(round(p[[length(l)]][i],1),nsmall=1,width=5),")
<td class=\"rowsn\" > ",rowsum[i],"("
,format(round(100*rowsum[i]/round(sum(colsum)),digits=1),nsmall=1,width=5),")  </td>",
     file=wpfile,fill=TRUE,append=TRUE,sep="")
}
if (sum(colsum)==0) {
 cat(l[[length(l)]][i]," (",format(round(p[[length(l)]][i],1),nsmall=1,width=5),")
<td class=\"rowsn\" > ",rowsum[i],"(",0,")  </td>",
     file=wpfile,fill=TRUE,append=TRUE,sep="")
}

cat("        </tr>",file=wpfile,fill=TRUE,append=TRUE)
}

cat("",file=wpfile,fill = TRUE,append=TRUE)
cat("        <tr>",file=wpfile,fill = TRUE,append=TRUE)
cat("         <td></td>",file=wpfile,fill = TRUE,append=TRUE)
cat("         <td class=\"colsn\">",file=wpfile,fill = FALSE,append=TRUE)
for (i in 1:length(colsum)) {  #total last row
 if (sum(colsum)>0) {
  cat(colsum[i]," ("
,format(round(100*(colsum[i]/round(sum(colsum))),1),nsmall=1,width=5),") </td>"
,file=wpfile,fill = TRUE,append=TRUE)
 }
 if (sum(colsum)==0) {
  cat(0," (0.0) </td>",file=wpfile,fill = TRUE,append=TRUE)
 }

 if (i<length(colsum)) {
  cat("         <td class=\"colsn\">",file=wpfile,fill = FALSE,append=TRUE)
 }
 if (i==length(colsum)) {
  cat("         <td class=\"grandn\">",file=wpfile,fill = FALSE,append=TRUE)
 }

}
cat(round(sum(colsum))," (100.0)</td>",file=wpfile,fill = TRUE,append=TRUE)
cat("        </tr>",file=wpfile,fill = TRUE,append=TRUE)
cat("",file=wpfile,fill = TRUE,append=TRUE)
cat("",file=wpfile,fill = TRUE,append=TRUE)
cat("       </table>",file=wpfile,fill = TRUE,append=TRUE)
cat("       <caption>",caption,"</caption>",file=wpfile,fill = TRUE,append=TRUE)
cat("        <br>",file=wpfile,fill = TRUE,append=TRUE)
cat("        <br>",file=wpfile,fill = TRUE,append=TRUE)

write(paste("WebPortal Table: [OUTDIR]/wp/",basename(htmlfile),sep=""),file="")

 if (struc==TRUE) {

cat("",file=sthtml,fill = TRUE,append=TRUE)
cat("     <table border=\"1\">",file=sthtml,fill = TRUE,append=TRUE)
cat("",file=sthtml,fill = TRUE,append=TRUE)
cat("      <tbody>",file=sthtml,fill = TRUE,append=TRUE)
cat("",file=sthtml,fill = TRUE,append=TRUE)
cat("       <thead>",file=sthtml,fill = TRUE,append=TRUE)
cat("",file=sthtml,fill = TRUE,append=TRUE)

rowspan<-length(na.omit(var))-1

#if (length(na.omit(var))==4) {
# rowspan<-length(na.omit(unique(var)))
#}

cat("        <tr>",file=sthtml,fill = TRUE,append=TRUE)
cat("         <th rowspan=\"",rowspan,"\">",FirstLab,"</th>",file=sthtml,
    fill = TRUE,append=TRUE)
```

```
if (n[4]>1) {

for(i in 1:(n[4]-1)) {

cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][i],"</th>",file=sthtml,
    fill = TRUE,append=TRUE)

}

cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][n[4]]," </th>",
    file=sthtml,fill = TRUE,append=TRUE)
cat("         <th></th>",file=sthtml,fill = TRUE,append=TRUE)
}

if ((length(lev)==4) & n[4]==1 & length(na.omit(unique(var)))>3) {
cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][1],"</th>",
    file=sthtml,fill = TRUE,append=TRUE)
cat("         <th></th>",file=sthtml,fill = TRUE,append=TRUE)
cat("         </tr>",file=sthtml,fill = TRUE,append=TRUE)
#cat("         <tr>",file=sthtml,fill = TRUE,append=TRUE)

}

if (nvar>=3) {
second_head<-rep(lev[[3]],prod(n[4]))
if (nvar==4) {
 cat("         <tr>",file=sthtml,fill = TRUE,append=TRUE)
}
if (n[3]>1) {
 for(i in 1:(length(second_head)-1)) {
 cat("         <th colspan=\"",n[2],"\">",second_head[i],"</th>",file=sthtml,
     fill = TRUE,append=TRUE)
 }
}
 cat("         <th colspan=\"",n[2],"\">",second_head[length(second_head)],
     "</th>",file=sthtml,fill = TRUE,append=TRUE)
cat("         <th></th>",file=sthtml,fill = TRUE,append=TRUE)
cat("         </tr>",file=sthtml,fill = TRUE,append=TRUE)
}

if (n[2]>1) {
third_head<-rep(lev[[2]],prod(n[3:4]))
if (nvar==3) {

 cat("",file=sthtml,fill = TRUE,append=TRUE)
 cat("         <tr>",file=sthtml,fill = FALSE,append=TRUE)
}
if (n[3]>=1) {
 for(i in 1:(length(third_head)-1)) {
  cat("          <th>",third_head[i]," (%)</th>",file=sthtml,fill = TRUE,append=TRUE)
 }
}
cat("          <th>",third_head[length(third_head)]," (%)</th>",
    file=sthtml,fill = TRUE,append=TRUE)
cat("          <th></th>",file=sthtml,fill = TRUE,append=TRUE)
cat("         </tr>",file=sthtml,fill = TRUE,append=TRUE)

}
cat("       </thead>",file=sthtml,fill = TRUE,append=TRUE)
for (i in 1:n[1]) {  #rows

cat("",file=sthtml,fill = TRUE,append=TRUE)
cat("         <tr>",file=sthtml,fill=TRUE,append=TRUE)
#if (engine=="se") {
cat("          <td class=\"catcol\">",lev[[1]][i],"</td>"
,file=sthtml,fill=TRUE,append=TRUE)
#}
#if (engine=="ce") {
# cat("          <td
class=\"catcol\">",tab$a[prod(n[2:4])*(i-1)+1],"</td>",file=sthtml,fill=TRUE,append=TR
UE)
#}
cat("          <td>",file=sthtml,fill=FALSE,append=TRUE)

for(j in 1:(length(l)-1)) {


cat("n"," (","p",") </td> ",file=sthtml,fill=TRUE,append=TRUE,sep="")
cat("          <td>",file=sthtml,fill=FALSE,append=TRUE)
}
if (sum(colsum)>0) {
 cat("n"," (","p",") <td class=\"rowsn\" > ","N","(","p",")  </td>",
     file=sthtml,fill=TRUE,append=TRUE,sep="")
}
if (sum(colsum)==0) {
 cat("n"," (","p",") <td class=\"rowsn\" > ",rowsum[i],"(",0,")  </td>",
     file=sthtml,fill=TRUE,append=TRUE,sep="")
}

cat("         </tr>",file=sthtml,fill=TRUE,append=TRUE)
}

cat("",file=sthtml,fill = TRUE,append=TRUE)
cat("         <tr>",file=sthtml,fill = TRUE,append=TRUE)
cat("          <td></td>",file=sthtml,fill = TRUE,append=TRUE)
cat("          <td class=\"colsn\">",file=sthtml,fill = FALSE,append=TRUE)
for (i in 1:length(colsum)) {  #total last row
 if (sum(colsum)>0) {
  cat("N"," (","p",") </td>",file=sthtml,fill = TRUE,append=TRUE)
 }
 if (sum(colsum)==0) {
  cat("N"," (p) </td>",file=sthtml,fill = TRUE,append=TRUE)
 }

 if (i<length(colsum)) {
  cat("          <td class=\"colsn\">",file=sthtml,fill = FALSE,append=TRUE)
 }
 if (i==length(colsum)) {
  cat("          <td class=\"grandn\">",file=sthtml,fill = FALSE,append=TRUE)
 }

}
cat("N"," (p)</td>",file=sthtml,fill = TRUE,append=TRUE)
cat("         </tr>",file=sthtml,fill = TRUE,append=TRUE)
cat("",file=sthtml,fill = TRUE,append=TRUE)
cat("         </tbody>",file=sthtml,fill = TRUE,append=TRUE)
cat("",file=sthtml,fill = TRUE,append=TRUE)
cat("       </table>",file=sthtml,fill = TRUE,append=TRUE)
cat("       <caption>",caption,"</caption>",file=sthtml,fill = TRUE,append=TRUE)
cat("         <br>",file=sthtml,fill = TRUE,append=TRUE)
cat("         <br>",file=sthtml,fill = TRUE,append=TRUE)

write(paste("Structure Table: [OUTDIR]/structure/",basename(htmlfile),sep=""),file="")
```

```
 }
#write the csv file

 names(tab)[pmatch("Freq",names(tab))]<-"n"

 duplvar<-paste(var,".1",sep="")
 for (w in 1:length(duplvar))
 if (duplvar[w] %in% names(tab)) tab<-tab[,-pmatch(duplvar[w],names(tab))]

 tab$start=oneyearbf
 tab$end=checkdate
 tab$dbname=dbname
 write.csv(tab,paste(dircsv,"/i",tolower(codtab),"d1_3a.csv",sep=""),row.names =
FALSE)


# file.append(paste(dirhtml,"/",ind,".html",sep=""),
#             htmlfile)


if (is.null(chisq)==FALSE) {

if (length(na.omit(chisq))==3) {
 if (all(tab$n>0)) {
 formula<-as.formula(paste("n~",paste(c(var[1],na.omit(chisq[1:3])),collapse="+"),sep=
""))
    mytable<-xtabs(formula=formula,data=tab)

    #Cochran-Mantel-Haenszel Chi-Squared Test for Count Data


    mh<-chisq.test(mytable)


    chitab<-NULL

chitab<-as.data.frame(cbind(round(mh$statistic,4),round(mh$p.value,4),mh$parameter))
    names(chitab)<-c("CMH Chi-Square","p.value","df")

    chitabcsv<-chitab
    chitabcsv$start=oneyearbf
    chitabcsv$end=checkdate
    chitabcsv$dbname=dbname
    write.csv(chitabcsv,paste(dircsv,"/i",tolower(codtab),"1_3b.csv",sep=""
),row.names = FALSE)

#    BIRO_df2html(data=chitab,
#                htmlfile=htmlfilep,
#                catcol=NULL)

# file.append(paste(dirhtml,"/",ind,".html",sep=""),
#             htmlfilep)

   for (j in 1:length(names(chitab))) {
    names(chitab)[j]<-paste(unlist(strsplit(names(chitab)[j],"_")),collapse="\\_")
   }
   row.names(chitab)="Value"
   latex(chitab,title="",file=texfile,append=TRUE,table.env=FALSE)

   BIRO_df2html(data=chitab,
               htmlfile=htmlfilep,
               catcol=NULL)
  } else {
   chitab<-NULL
   chitab<-as.data.frame("One or more cells have 0 obs")
   names(chitab)<-"CMH Chi-Square"
   row.names(chitab)="Value"
   latex(chitab,title="",file=texfile,append=TRUE,table.env=FALSE)

   BIRO_df2html(data=chitab,
               htmlfile=htmlfilep,
               catcol=NULL)

   }
}


if (length(na.omit(chisq))==2) {
 if (all(tab$n>0)) {
 formula<-as.formula(paste("n~",paste(c(var[1],na.omit(chisq[1:2])),collapse="+"),sep=
""))
    mytable<-xtabs(formula=formula,data=tab)

    #Cochran-Mantel-Haenszel Chi-Squared Test for Count Data


    mh<-chisq.test(mytable)


    chitab<-NULL

chitab<-as.data.frame(cbind(round(mh$statistic,4),round(mh$p.value,4),mh$parameter))
    names(chitab)<-c("CMH Chi-Square","p.value","df")

    chitabcsv=chitab
    chitabcsv$start=oneyearbf
    chitabcsv$end=checkdate
    chitabcsv$dbname=dbname
    write.csv(chitabcsv,paste(dircsv,"/i",tolower(codtab),"1_3b.csv",sep=""
),row.names = FALSE)

    BIRO_df2html(data=chitab,
                htmlfile=htmlfilep,
                catcol=NULL)

# file.append(paste(dirhtml,"/",ind,".html",sep=""),
#             htmlfilep)

   for (j in 1:length(names(chitab))) {
    names(chitab)[j]<-paste(unlist(strsplit(names(chitab)[j],"_")),collapse="\\_")
   }
   row.names(chitab)="Value"
   latex(chitab,title="",file=texfile,append=TRUE,table.env=FALSE)

  } else {
   chitab<-NULL
   chitab<-as.data.frame("One or more cells have 0 obs")
   names(chitab)<-"CMH Chi-Square"
   row.names(chitab)="Value"
   latex(chitab,title="",file=texfile,append=TRUE,table.env=FALSE)
   BIRO_df2html(data=chitab,
               htmlfile=htmlfilep,
               catcol=NULL)
```

```
   }
}

if (length(na.omit(chisq))==1) {
 if (all(tab$n>0)) {
  formula<-as.formula(paste("n~",paste(c(var[1],na.omit(chisq)),collapse="+"),sep=""))
  mytable<-xtabs(formula=formula,data=tab)
  #Pearson Chi-Squared Test for Count Data
  mh<-chisq.test(mytable)
  chitab<-as.data.frame(cbind(round(mh$statistic,4),round(mh$p.value,4),mh$parameter))
  names(chitab)<-c("CMH Chi-Square","p.value","df")
  if (nvar==4) {
   row.names(chitab)="Value"
  }

   BIRO_df2html(data=chitab,
                htmlfile=htmlfilep,
                catcol=NULL)

  chitabcsv=chitab
  chitabcsv$start=oneyearbf
  chitabcsv$end=checkdate
  chitabcsv$dbname=dbname
  write.csv(chitabcsv,paste(dircsv,"/i",tolower(codtab),"d1_3b.csv",sep=""),row.names
= FALSE)
  for (j in 1:length(names(chitab))) {
   names(chitab)[j]<-paste(unlist(strsplit(names(chitab)[j],"_")),collapse="\\_")
  }
   row.names(chitab)="Value"
   latex(chitab,title="",file=texfile,append=TRUE,table.env=FALSE)

 } else {
   chitab<-as.data.frame("One or more cells have 0 obs")
   names(chitab)<-"CMH Chi-Square"
   row.names(chitab)<-"Value"
   latex(chitab,title="",file=texfile,append=TRUE,table.env=FALSE)

   BIRO_df2html(data=chitab,
                htmlfile=htmlfilep,
                catcol=NULL)

  }
 }
 }

#   file.append(paste(dirhtml,"/",ind,".html",sep=""),
#               htmlfile)

   file.append(htmlclass,
               htmlfile)

#   if (var[1]!="noNA") file.append(paste(dirhtml,"/",ind,".html",sep=""),
#                                   htmlfilep)

   file.append(htmlclass,
               htmlfilep)

}
 #print("Table printed")
}
################################################################################
```

```
#  BIRO_df2html                                                                #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
## ------------------------
# DESCRIPTION
#
# BIRO_df2html writes a data.frame into an html table
#
# DEPENDENCIES
#
# R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#data       !              =  a string containing the name of a t data.frame
#
#htmlfile   !              =  a string containing the path and the name of the
#                             html file
#catcol     !              =  the position (into the data.frame) of the
#                             category variables
#
################################################################################


BIRO_df2html<-function(data,htmlfile,catcol=1) {
 #print("BIRO_df2html called")
 cat("       <table border=\"1\">",file=htmlfile,fill = TRUE,append=TRUE)
 cat("        <tbody>",file=htmlfile,fill = TRUE,append=TRUE)
 cat("         <thead>",file=htmlfile,fill = TRUE,append=TRUE)
 headerpos<-1:length(names(data))
 if (is.null(catcol)==FALSE) {
  headerpos<-headerpos[-catcol]
 }
 cat("          <tr>",file=htmlfile,fill = TRUE,append=TRUE)
 for (j in catcol) {
  cat("          <th>",file=htmlfile,fill = FALSE,append=TRUE)
  cat(names(data)[j],file=htmlfile,fill = FALSE,append=TRUE)
  cat("          </th>",file=htmlfile,fill = TRUE,append=TRUE)
 }
 for (i in headerpos) {
  cat("          <th>",file=htmlfile,fill = FALSE,append=TRUE)
  cat(names(data)[i],file=htmlfile,fill = FALSE,append=TRUE)
  cat("          </th>",file=htmlfile,fill = TRUE,append=TRUE)
 }
 cat("          </tr>",file=htmlfile,fill = TRUE,append=TRUE)
 cat("         </thead>",file=htmlfile,fill = TRUE,append=TRUE)
 for (k in 1:dim(data)[1]) {
 cat("         <tr>",file=htmlfile,fill = TRUE,append=TRUE)
 for (j in catcol) {
  cat("          <td class=\"catcol\">",file=htmlfile,fill = FALSE,append=TRUE)
  cat(as.character(data[k,j]),file=htmlfile,fill = FALSE,append=TRUE)
  cat("          </td>",file=htmlfile,fill = TRUE,append=TRUE)
 }
```

```
for (i in headerpos) {
  cat("         <td>",file=htmlfile,fill = FALSE,append=TRUE)
  cat(as.character(data[k,i]),file=htmlfile,fill = FALSE,append=TRUE)
  cat("         </td>",file=htmlfile,fill = TRUE,append=TRUE)
 }
cat("         </tr>",file=htmlfile,fill = TRUE,append=TRUE)
}
cat("        </tbody>",file=htmlfile,fill = TRUE,append=TRUE)
cat("       </table>",file=htmlfile,fill = TRUE,append=TRUE)
cat("       <br>",file=htmlfile,fill = TRUE,append=TRUE)
cat("       <br>",file=htmlfile,fill = TRUE,append=TRUE)
cat("       <br>",file=htmlfile,fill = TRUE,append=TRUE)
cat("       <br>",file=htmlfile,fill = TRUE,append=TRUE)

# Table for WP

wpfile<-paste(dirwp,"/",ind,".html",sep="")
cat("       <table border=\"1\">",file=wpfile,fill = TRUE,append=TRUE)
headerpos<-1:length(names(data))
if (is.null(catcol)==FALSE) {
 headerpos<-headerpos[-catcol]
}
cat("          <tr>",file=wpfile,fill = TRUE,append=TRUE)
for (j in catcol) {
 cat("           <th>",file=wpfile,fill = FALSE,append=TRUE)
 cat(names(data)[j],file=wpfile,fill = FALSE,append=TRUE)
 cat("           </th>",file=wpfile,fill = TRUE,append=TRUE)
}
for (i in headerpos) {
 cat("           <th>",file=wpfile,fill = FALSE,append=TRUE)
 cat(names(data)[i],file=wpfile,fill = FALSE,append=TRUE)
 cat("           </th>",file=wpfile,fill = TRUE,append=TRUE)
}
cat("          </tr>",file=wpfile,fill = TRUE,append=TRUE)
for (k in 1:dim(data)[1]) {
cat("          <tr>",file=wpfile,fill = TRUE,append=TRUE)
for (j in catcol) {
 cat("           <td class=\"catcol\">",file=wpfile,fill = FALSE,append=TRUE)
 cat(as.character(data[k,j]),file=wpfile,fill = FALSE,append=TRUE)
 cat("           </td>",file=wpfile,fill = TRUE,append=TRUE)
}
for (i in headerpos) {
  cat("         <td>",file=wpfile,fill = FALSE,append=TRUE)
  cat(as.character(data[k,i]),file=wpfile,fill = FALSE,append=TRUE)
  cat("         </td>",file=wpfile,fill = TRUE,append=TRUE)
 }
cat("         </tr>",file=wpfile,fill = TRUE,append=TRUE)
}
cat("       </table>",file=wpfile,fill = TRUE,append=TRUE)


if (struc==TRUE) {

 cat("       <table border=\"1\">",file=sthtml,fill = TRUE,append=TRUE)
 cat("        <tbody>",file=sthtml,fill = TRUE,append=TRUE)
 cat("         <thead>",file=sthtml,fill = TRUE,append=TRUE)
 headerpos<-1:length(names(data))
 if (is.null(catcol)==FALSE) {
  headerpos<-headerpos[-catcol]
 }
 cat("          <tr>",file=sthtml,fill = TRUE,append=TRUE)
 for (j in catcol) {
  cat("           <th>",file=sthtml,fill = FALSE,append=TRUE)
  cat(names(data)[j],file=sthtml,fill = FALSE,append=TRUE)
  cat("           </th>",file=sthtml,fill = TRUE,append=TRUE)
 }
 for (i in headerpos) {
  cat("           <th>",file=sthtml,fill = FALSE,append=TRUE)
  cat(names(data)[i],file=sthtml,fill = FALSE,append=TRUE)
  cat("           </th>",file=sthtml,fill = TRUE,append=TRUE)
 }
 cat("          </tr>",file=sthtml,fill = TRUE,append=TRUE)
 cat("         </thead>",file=sthtml,fill = TRUE,append=TRUE)
 for (k in 1:dim(data)[1]) {
 cat("          <tr>",file=sthtml,fill = TRUE,append=TRUE)
 for (j in catcol) {
  cat("           <td class=\"catcol\">",file=sthtml,fill = FALSE,append=TRUE)
  cat(as.character(data[k,j]),file=sthtml,fill = FALSE,append=TRUE)
  cat("           </td>",file=sthtml,fill = TRUE,append=TRUE)
 }
 for (i in headerpos) {
  cat("           <td>",file=sthtml,fill = FALSE,append=TRUE)
  cat("n",file=sthtml,fill = FALSE,append=TRUE)
  cat("           </td>",file=sthtml,fill = TRUE,append=TRUE)
 }
 cat("          </tr>",file=sthtml,fill = TRUE,append=TRUE)
 }
 cat("        </tbody>",file=sthtml,fill = TRUE,append=TRUE)
 cat("       </table>",file=sthtml,fill = TRUE,append=TRUE)
 cat("       <br>",file=sthtml,fill = TRUE,append=TRUE)
 cat("       <br>",file=sthtml,fill = TRUE,append=TRUE)
}


################################################################################
# BIRO_drawbars
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# BIRO_barplot prepares data for BIRO_drawbars
#
# DEPENDENCIES
#
#  R packages: grDevice, graphics, Cairo
#
################################################################################
# PARAMETERS(!=required)
#
#tab        !            = a data.frame from BIRO_barplot
#labelvar   !            = the label of the variable
#lev        !            = a list containing the vactor with che levels of the
#                          categorical variable (list)
#beside     [TRUE]       = a logical value. If FALSE, the columns of height
```

```
#                           are portrayed as stacked bars,
#                           and if TRUE the columns are portrayed as juxtaposed
#                           bars.
#number     [FALSE]       = a logical value. If TRUE values are printed over
#                           each bar, FALSE otherwise.
#perc       [TRUE]        = a logical value. If TRUE percentages are calculated.
#what       ["patients"]  = the subject of the representation (this is for the
#                           label)
#dirgraph   !             = the path of the output (jpg,eps,png files)
#namegraph  !             = the coded name of the output files
##############################################################################

 BIRO_drawbars<-function(tab,
                         labelvar,
                         lev=list(),
                         beside=TRUE,
                         number=FALSE,
                         perc=TRUE,
                         what="patients",
                         namegraph,
                         dirgraph,
                         cex=cex) {

  #correction to user's mistake
  if (beside==FALSE) {
   number=TRUE
  }

  tab$Perc<-round((tab$n/tab$sum)*100,2)

  l<-list()
  p<-list()
  for (i in 1:(dim(tab)[1]/nlevels(as.factor(tab[,1])))) {
  start<-(i-1)*nlevels(as.factor(tab[,1]))+1
  l[[i]]<-tab$n[start:(start+nlevels(as.factor(tab[,1]))-1)]
  p[[i]]<-tab$Perc[start:(start+nlevels(as.factor(tab[,1]))-1)]
  }


   #Create the summary dataset
  sumdata<-NULL

  columnnames<-as.data.frame(as.matrix(unique(tab[,1:(dim(tab)[2]-3)])))
  if (perc==FALSE) {
  for (i in 1:1:(dim(tab)[1]/nlevels(as.factor(tab[,1])))) {
   sumdata<-cbind(sumdata,as.vector(l[[i]]))
  }
  }

  if (perc==TRUE) {
  for (i in 1:1:(dim(tab)[1]/nlevels(as.factor(tab[,1])))) {
   sumdata<-cbind(sumdata,as.vector(p[[i]]))
  }
  }

  nlevd<-1

  nvar<-dim(tab)[2]-4

  for (i in 2:nvar) {
   nlevd<-nlevd*nlevels(as.factor(tab[,1]))
  }


  equal=""
  if (length(labelvar)>1 ) {
    if (labelvar[2]>"")
      equal="= " }
  columnnames2<-rep(NA,nlevd)
  for (i in 1:dim(columnnames)[2]) {
   for (k in 1:(length(var)-1)) {
    if (is.na(columnnames2[i])) columnnames2[i]<-
                                  paste(labelvar[k+1],equal,columnnames[i,k],
                                        sep="")
    else columnnames2[i]<-
         paste(columnnames2[i],", ",labelvar[k+1],equal,columnnames[i,k],
         sep="")
   }
  }

  columnnames<-as.data.frame(columnnames2)
  rm(columnnames2)
  colnames(sumdata)<-as.vector(unique(tab[,2]))
  row.names(sumdata)<-as.vector(unique(tab[,1]))
  sumdata<-as.data.frame(sumdata)
  sumdata$lev<-as.vector(unique(tab[,1]))
  namevar<-
  names(tab)[-c(dim(tab)[2]-3,dim(tab)[2]-2,dim(tab)[2]-1,dim(tab)[2])]
  nrows=1
  for (i in 1:nvar) {

    tab[,pmatch(namevar[i],names(tab))]<-
    as.factor(tab[,pmatch(namevar[i],names(tab))])
    levels(tab[,pmatch(namevar[i],names(tab))])<-
    lev[[i]]
    if (i>1) nrows<-
             nrows*nlevels(as.factor(tab[,pmatch(namevar[i],names(tab))]))
  }

  if ((beside==TRUE)) {uplim<-max(tab$n)+sd(tab$n)}

   if (beside==FALSE) {
    sums<-apply(sumdata[,1:dim(sumdata)[2]-1], 2, sum)
    uplim<-max(as.numeric(as.character(sums)))
   }

  if (perc==TRUE) if (uplim>100) uplim=150
  dwlim<-(-(nrows*4)-4)
   #Drawing

  if (perc==TRUE) {

    ylab=paste("% of", what)
    }
  if (perc==FALSE) {
    ylab=paste("Frequencies (",what,")",sep="")
    }

    height<-sumdata[,-dim(sumdata)[2]]

  height<-as.matrix(height)
  par(cex=1.5*cex)
#  write("Preparing to do SVG file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do SVG file",file=logfile,append=TRUE)
```

```
#  }
  file = paste(dirgraph,"/","i",namegraph,"g4_1a.svg",sep="")
  CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
  ylab=""
  r <-
barplot(height,beside=beside,col=rainbow(length(sumdata$lev),start=0.2,end=.8,gamma=1)
,
              ylim= c(dwlim, uplim),ylab=ylab,angle=90,axisnames=FALSE,
              axes=FALSE,font=1,cex.names=1.5*cex,cex.axis=1.5*cex)
  legend("topright"
,legend=lev[[1]],cex=1.5*cex,col=rainbow(length(sumdata$lev),start=0.2,end=.8,gamma=1)
,
        pch=1,title=labelvar[1])

  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  ataxis=apply(r,MARGIN=2,FUN="median")

axis(side=1,at=ataxis,colnames(sumdata)[-length(colnames(sumdata))],cex.axis=1.5*cex,c
ex=1.5*cex,lty=0)

  if (beside==TRUE) {x<-r[(round(nrow(r)/2))+1,]}

  if (beside==FALSE) {x<-r}

  for(i in 1:ncol(sumdata)) {
   s<-strsplit(colnames(sumdata)[i],", ")
   s<-as.vector(s[[1]])
   for (j in 1:length(s)) {
    ##text(x[i]-0.3,-par("cxy")[2]*j,s[j],font=1,cex=cex)
   }
  }
    if ((beside==TRUE)  & (perc==FALSE)){uplim<-max(height)}

    if (perc==TRUE) {uplim=120}

  par(cex=1.5*cex)
  axis(side=2,at=c(0,25,50,75,100),cex.axis=1.5*cex)


  #write upper or lower to the upper line of a bar
  halfheight<-uplim/2

  signs<-rep(NA,length(height))

  signs[height>halfheight]<--1
  signs[height<halfheight]<-1

  #write numbers

  if (beside==TRUE & number==TRUE) {
  text(r,height+signs*par("cxy")[2],round(height),font=1,cex=1.5*cex)
  }

  dev.off()



  write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

#  write("Preparing to do JPEG file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do JPEG file",file=logfile,append=TRUE)
#  }
   file = paste(dirgraph,"/","i",namegraph,"g4_1a.jpeg",sep="")
  jpeg(file, width= 813.6, height= 348.48, units="px", bg = "transparent")
  ylab=""
  r <-
barplot(height,beside=beside,col=rainbow(length(sumdata$lev),start=0.2,end=.8,gamma=1)
,
              ylim= c(dwlim, uplim),ylab=ylab,angle=90,axisnames=FALSE,
              axes=FALSE,font=1,cex.names=1.5*cex,cex.axis=1.5*cex)
  legend("topright"
,legend=lev[[1]],cex=1.5*cex,col=rainbow(length(sumdata$lev),start=0.2,end=.8,gamma=1)
,
        pch=1,title=labelvar[1])

  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  ataxis=apply(r,MARGIN=2,FUN="median")

axis(side=1,at=ataxis,colnames(sumdata)[-length(colnames(sumdata))],cex.axis=1.5*cex,c
ex=1.5*cex,lty=0)

  if (beside==TRUE) {x<-r[(round(nrow(r)/2))+1,]}

  if (beside==FALSE) {x<-r}

  for(i in 1:ncol(sumdata)) {
   s<-strsplit(colnames(sumdata)[i],", ")
   s<-as.vector(s[[1]])
   for (j in 1:length(s)) {
    #text(x[i]-0.3,-par("cxy")[2]*j,s[j],font=1,cex=cex)
   }
  }
    if ((beside==TRUE)  & (perc==FALSE)){uplim<-max(height)}

    if (perc==TRUE) {uplim=120}
  par(cex=1.5*cex)
  axis(side=2,at=c(0,25,50,75,100),cex.axis=1.5*cex)


  #write upper or lower to the upper line of a bar
  halfheight<-uplim/2

  signs<-rep(NA,length(height))

  signs[height>=halfheight]<--1
  signs[height<halfheight]<-1

  #write numbers

  if (beside==TRUE & number==TRUE) {
  text(r,height+signs*par("cxy")[2],round(height),font=1,cex=1.5*cex)
  }

  dev.off()

  write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
```

```
   write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

#  write("Preparing to do PNG file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do PNG file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/","i",namegraph,"g4_1a.png",sep="")
  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
  ylab=""
  r <-
barplot(height,beside=beside,col=rainbow(length(sumdata$lev),start=0.2,end=.8,gamma=1)
,
               ylim= c(dwlim, uplim),ylab=ylab,angle=90,axisnames=FALSE,
               axes=FALSE,font=1,cex.names=1.5*cex,cex.axis=1.5*cex)
  legend("topright"
,legend=lev[[1]],cex=1.5*cex,col=rainbow(length(sumdata$lev),start=0.2,end=.8,gamma=1)
,
         pch=1,title=labelvar[1])

  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  ataxis=apply(r,MARGIN=2,FUN="median")

axis(side=1,at=ataxis,colnames(sumdata)[-length(colnames(sumdata))],cex.axis=1.5*cex,c
ex=1.5*cex,lty=0)

  if (beside==TRUE) {x<-r[(round(nrow(r)/2))+1,]}

  if (beside==FALSE) {x<-r}

  for(i in 1:ncol(sumdata)) {
   s<-strsplit(colnames(sumdata)[i],", ")
   s<-as.vector(s[[1]])
   for (j in 1:length(s)) {
    #text(x[i]-0.3,-par("cxy")[2]*j,s[j],font=1,cex=cex)
   }
  }
    if ((beside==TRUE)  & (perc==FALSE)){uplim<-max(height)}

    if (perc==TRUE) {uplim=120}
  par(cex=1.5*cex)
  axis(side=2,at=c(0,25,50,75,100),cex.axis=1.5*cex)


  #write upper or lower to the upper line of a bar
  halfheight<-uplim/2

  signs<-rep(NA,length(height))

  signs[height>halfheight]<--1
  signs[height<halfheight]<-1

  #write numbers

  if (beside==TRUE & number==TRUE) {
  text(r,height+signs*par("cxy")[2],round(height),font=1,cex=1.5*cex)
  }

  dev.off()
```

```
  write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

#  write("Preparing to do PDF file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do PDF file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/","i",namegraph,"g4_1a.pdf",sep="")
  pdf(file, width = 11.33, height =4.84 , pointsize = 7, bg = "transparent")
  ylab=""
  r <-
barplot(height,beside=beside,col=rainbow(length(sumdata$lev),start=0.2,end=.8,gamma=1)
,
               ylim= c(dwlim, uplim),ylab=ylab,angle=90,axisnames=FALSE,
               axes=FALSE,font=1,cex.names=1.5*cex,cex.axis=1.5*cex)
  legend("topright"
,legend=lev[[1]],cex=1.5*cex,col=rainbow(length(sumdata$lev),start=0.2,end=.8,gamma=1)
,
         pch=1,title=labelvar[1])

  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  ataxis=apply(r,MARGIN=2,FUN="median")

axis(side=1,at=ataxis,colnames(sumdata)[-length(colnames(sumdata))],cex.axis=1.5*cex,c
ex=1.5*cex,lty=0)
  if (beside==TRUE) {x<-r[(round(nrow(r)/2))+1,]}

  if (beside==FALSE) {x<-r}

  for(i in 1:ncol(sumdata)) {
   s<-strsplit(colnames(sumdata)[i],", ")
   s<-as.vector(s[[1]])
   for (j in 1:length(s)) {
    #text(x[i]-0.3,-par("cxy")[2]*j,s[j],font=1,cex=cex)
   }
  }
    if ((beside==TRUE)  & (perc==FALSE)){uplim<-max(height)}

    if (perc==TRUE) {uplim=120}
  par(cex=1.5*cex)
  axis(side=2,at=c(0,25,50,75,100),cex.axis=1.5*cex)


  #write upper or lower to the upper line of a bar
  halfheight<-uplim/2

  signs<-rep(NA,length(height))

  signs[height>halfheight]<--1
  signs[height<halfheight]<-1

  #write numbers

  if (beside==TRUE & number==TRUE) {
  text(r,height+signs*par("cxy")[2],round(height),font=1,cex=1.5*cex)
  }
```

```
  dev.off()

  write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

  if (struc==TRUE) {
    cat("<table border=\"1\">",file=sthtml,fill=TRUE,append=TRUE)
    cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
    cat("  <tr>",file=sthtml,fill=FALSE,append=TRUE)
    cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/","i"
,namegraph,"g4_1a.png\">","</td>",sep=""),file=sthtml,fill=FALSE,append=TRUE)
    cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
    cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
    cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
  }

}



################################################################################
#  BIRO_plotbox                                                                #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# BIRO_plotbox draws boxplot for a conditionated distribution
#
# DEPENDENCIES
#
#  R packages: grDevice, graphics, Cairo
#
################################################################################
# PARAMETERS(!=required)
#
# bxpdata    !            = dataset containing data
#
# lev        !            = a list containing the vactor with che levels of
#                           the categorical variable (list)
# notch      !            = logical. if notch is TRUE, a notch is drawn in
#                           each side of the boxes. If the notches of two
#                           plots do not overlap this is ï¿½strong evidenceï¿½ that
#                           the two means differ (Chambers et al., 1983, p. 62).
#                           See boxplot.stats for the calculations used.
# namegraph  !            = the coded name of the output files
# dirgraph     [current]  = the path of the output (jpg,eps,png files)
# out     [out"]          = the name of the outliers csv files
# n      [n]              = a vector containint he number of levels of each
#                           variable
#variable    !            = the name of the target variable
# stratum    !            = the names of the stratum variables
################################################################################

BIRO_plotbox<-function(bxpdata,
                       lev,
                       notch=TRUE,
                       namegraph,
                       dirgraph,
                       out=out,
                       n=n,
                       variable=variable,
                       stratum=stratum) {


 bxpdata<-sort.data.frame(bxpdata,
                          by=as.formula(paste("~",paste(stratum,collapse="+"))))

  #write("Boxplot statistics sorted",file="")
  #if (nchar(logfile)>0) {
  # write("Boxplot statistics sorted",file=logfile,append=TRUE)
  #}

  for (j in 1:length(stratum)) {

  bxpdata[,pmatch(stratum[j],names(bxpdata))]<-
  as.factor(as.character(bxpdata[,pmatch(stratum[j],names(bxpdata))]))

  levels(bxpdata[,pmatch(stratum[j],names(bxpdata))])<-lev

  bxpdata$newvar<-as.character(bxpdata[,pmatch(stratum[j],names(bxpdata))])
  names(bxpdata)[pmatch("newvar",names(bxpdata))]<-paste("names",j,sep="")
 }

  for (i in 1:dim(bxpdata)[1]) {
  bxpdata$names[i]=
  paste(bxpdata[i,pmatch(paste(rep("names",length(stratum)),
                               1:length(stratum),sep=""),
        names(bxpdata))],collapse=".")
 }


 statdata<-
 bxpdata[pmatch(c("mean","l_wisk","pcl_25x","median","pcl_75x",
                  "u_wisk","names",stratum),names(bxpdata))]




 stats<-aggregate(statdata,by=list(statdata$names),FUN="unique")
 stats$group<-1:dim(stats)[1]

 groupdata<-stats[,c("names","group")]


bxpdata<-merge(bxpdata,groupdata,by="names")



bxpdata<-sort.data.frame(bxpdata,~group)
stats<-sort.data.frame(stats,~group)
```

```
#matrix with summary statistics
statsm<-as.matrix(t(stats[c("l_wisk","pcl_25x","median","pcl_75x","u_wisk")]))


n<-as.vector(as.numeric(as.character(bxpdata$n)))



conf<-as.matrix(t(unique(bxpdata[,c("l_notch","u_notch")])))

out<-as.vector(rep(bxpdata$outlie_x[is.na(bxpdata$outlie_x)==FALSE],
                   bxpdata$Freq[is.na(bxpdata$outlie_x)==FALSE]))

groupout<-as.vector(rep(bxpdata$group[is.na(bxpdata$outlie_x)==FALSE],
                        bxpdata$Freq[is.na(bxpdata$outlie_x)==FALSE]))

boxp<-list(stats=statsm,
           n=n,
           conf=conf,
           out=as.numeric(out),
           group=as.numeric(groupout),
           names=as.vector(unique(bxpdata$names)))

#X11(width = 21, height =9 , pointsize=9)

  #colours
  start=0.2
  end=0.7
  if (stratum=="sex") {
    start=0.03
    end=0.6
  }
  if (stratum=="age_c") {
    start=0.2
    end=0.7
  }
  if (stratum=="type_dm") {
    start=0.15
    end=0.7
  }


#  write("Preparing to do SVG file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do SVG file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/","/i",namegraph,"g4_3a.svg",sep="")
  CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")

bxp(boxp,notch=notch,boxfill=rainbow(dim(groupdata)[1],start=start,end=end,gamma=1),ce
x.axis=1.5*cex)
  dev.off()

  write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

#  write("Preparing to do JPEG file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do JPEG file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/","/i",namegraph,"g4_3a.jpeg",sep="")
  jpeg(file, width= 813.6, height= 348.48, pointsize=9, units="px",
       bg = "transparent")

bxp(boxp,notch=notch,boxfill=rainbow(dim(groupdata)[1],start=start,end=end,gamma=1),ce
x.axis=1.5*cex)
  dev.off()

  write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

#  write("Preparing to do PNG file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do PNG file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/","/i",namegraph,"g4_3a.png",sep="")
  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")

bxp(boxp,notch=notch,boxfill=rainbow(dim(groupdata)[1],start=start,end=end,gamma=1),ce
x.axis=1.5*cex)
  dev.off()

  write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

#  write("Preparing to do PDF file",file="")
#  if (nchar(logfile)>0) {
#   write("Preparing to do PDF file",file=logfile,append=TRUE)
#  }

  file = paste(dirgraph,"/","/i",namegraph,"g4_3a.pdf",sep="")
  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")

bxp(boxp,notch=notch,boxfill=rainbow(dim(groupdata)[1],start=start,end=end,gamma=1),ce
x.axis=1.5*cex)
  dev.off()

  write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }
  if (struc==TRUE) {
    cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
    cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
    cat("  <tr>",file=sthtml,fill=TRUE,append=TRUE)
    cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/"
,namegraph,".png\">","</td>",sep=""),file=sthtml,fill=TRUE,append=TRUE)
    cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
```

```
    cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
    cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
  }



}



BIRO_df2htmlwp<-function(data,htmlfile,catcol=1) {
 cat("       <table border=\"1\">",file=htmlfile,fill = TRUE,append=TRUE)
 headerpos<-1:length(names(data))
 if (is.null(catcol)==FALSE) {
  headerpos<-headerpos[-catcol]
 }
 cat("          <tr>",file=htmlfile,fill = TRUE,append=TRUE)
 for (j in catcol) {
  cat("           <th>",file=htmlfile,fill = FALSE,append=TRUE)
  cat(names(data)[j],file=htmlfile,fill = FALSE,append=TRUE)
  cat("           </th>",file=htmlfile,fill = TRUE,append=TRUE)
 }
 for (i in headerpos) {
  cat("           <th>",file=htmlfile,fill = FALSE,append=TRUE)
  cat(names(data)[i],file=htmlfile,fill = FALSE,append=TRUE)
  cat("           </th>",file=htmlfile,fill = TRUE,append=TRUE)
 }
 cat("          </tr>",file=htmlfile,fill = TRUE,append=TRUE)
 for (k in 1:dim(data)[1]) {
 cat("         <tr>",file=htmlfile,fill = TRUE,append=TRUE)
 for (j in catcol) {
  cat("           <td class=\"catcol\">",file=htmlfile,fill = FALSE,append=TRUE)
  cat(as.character(data[k,j]),file=htmlfile,fill = FALSE,append=TRUE)
  cat("           </td>",file=htmlfile,fill = TRUE,append=TRUE)
 }
 for (i in headerpos) {
   cat("          <td>",file=htmlfile,fill = FALSE,append=TRUE)
   cat(as.character(data[k,i]),file=htmlfile,fill = FALSE,append=TRUE)
   cat("          </td>",file=htmlfile,fill = TRUE,append=TRUE)
  }
 cat("         </tr>",file=htmlfile,fill = TRUE,append=TRUE)
 }
 cat("       </table>",file=htmlfile,fill = TRUE,append=TRUE)

}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_regression.r                                                      #
#                                                                              #
#
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-08-30
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
## ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_regression.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#  BIRO_regression
#
#
################################################################################


################################################################################
#  BIRO_regression                                                             #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-08-30
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
## ------------------------
# DESCRIPTION
#
# BIRO_regression is used to fit generalized linear models (linear models, logistic
models and poisson models)
#
# DEPENDENCIES
#
#  R package: stats
#
################################################################################
# PARAMETERS(!=required)
#
# data      ! = a data frame containing data
# depvar    ! = the name of dependent variable
# indepvar  ! = a vector containig the names of independent variables
# lev       ! = a list contaning levels of depvar and indepvar
# factindep ! = a vector containig logistic values TRUE if the i-th independent
#               variable is a factor, FALSE otherwise
# type      ! = possible values: "linear","logistic","poisson"
# sucvalue  ! = the value of depvar indicating a success
# csvindex  ! = the string indicating the name of the output csv file
# dirdataout! = the path of the output directory
################################################################################

regression<-function(data,
                     depvar,
                     indepvar,
                     lev=list(),
                     factindep,
                     type,
                     sucvalue,
                     csvindex,
                     dirdataout) {

if (length(indepvar)>=0) {

  if (tipe!="poisson") {
   funcdata<-data[,c(pmatch(depvar,names(data)),pmatch(indepvar,names(data)))]

  if (length(depvar) > 0 & length(lev)>0) {

funcdata[,pmatch(depvar,names(funcdata))]<-as.factor(funcdata[,pmatch(depvar,names(fun
cdata))])
    levels(funcdata[,pmatch(depvar,names(funcdata))])<-lev[[1]]
     }

  if (length(depvar) > 0 & length(lev)==0) {

levels(funcdata[,pmatch(depvar,names(funcdata))])<-levels(as.factor(funcdata[,pmatch(d
epvar,names(funcdata))]))
    }
  }
  if (tipe=="poisson") {
   funcdata<-data[,pmatch(indepvar,names(data))]
  }


  for (i in 1:length(indepvar)) {
   if (length(var) > i & length(lev)>i) {

funcdata[,pmatch(indepvar[i],names(funcdata))]<-as.factor(funcdata[,pmatch(indepvar[i]
```

```
,names(funcdata))])
     levels(funcdata[,pmatch(indepvar[i],names(funcdata))])<-lev[[i+1]]
      }

   if (length(var) > i & length(lev)==i) {

levels(funcdata[,pmatch(indepvar[i],names(funcdata))])<-levels(as.factor(funcdata[,pma
tch(indepvar[i],names(funcdata))]))
      }

  }

  for (i in 1:length(var)) {
   if (typeof(lev[[i]])=="list") {
   lev[[i]]<-names(unlist(lev[[i]]))
   }
  }

  varlist<-as.list(funcdata)

  #calculates contingency table (total frequency of non-missing values)
  tab<-as.data.frame(table(varlist))


  write.csv(tab,paste(dirdataout,"\\log",csvindex,".csv",sep=""),row.names = FALSE)
  }

  #create the formula part with the covariates
  covformula<-""
  for (i in 1:length(indepvar)) {
   if (factindep[i]==TRUE) {
    covformula<-paste(covformula,"+as.factor(",indepvar[i],")",sep="")
   }
   if (factindep[i]==FALSE) {
    covformula<-paste(covformula,"+",indepvar[i],sep="")
   }
  }


  if (type=="logistic") {
  #calculates contingency table of successes
  tabs<-tab[tab[,1]==sucvalue,]
  names(tabs)[dim(tabs)[2]]<-"Successes"
  tabs<-tabs[,-1]


  #calculates contingency table of failures
  tabf<-tab[tab[,1]!=sucvalue,]
  names(tabf)[dim(tabf)[2]]<-"Insuccesses"
  tabf<-tabf[,-1]


  tab<-merge(tabs,tabf,by=var[2:length(var)],all=TRUE)
  tab$out=var[1]

  #create formula

  formula<-paste("cbind(Successes,Insuccesses)",covformula,sep="~")

  model<-glm(formula=as.formula(formula),data=tab,family=binomial())
  }

  if (tipe=="linear") {
     formula<-paste(depvar,paste(indepvar,collapse="+"),sep="~")
  }

  if (tipe=="poisson") {
     formula<-paste("Freq",paste(indepvar,collapse="+"),sep="~")
  }

      est.disp <- FALSE
    df.r <- model$df.residual

        if (model$family$family %in% c("poisson",
            "binomial")) {
            dispersion <- 1
        } else if (df.r > 0) {
            est.disp <- TRUE
            if (any(model$weights == 0)) {
                warning("observations with zero weight not used for calculating
dispersion")
            dispersion <- sum((model$weights * model$residuals^2)[model$weights >
0])/df.r
        }
        } else {
            est.disp <- TRUE
            dispersion <- NaN
            NaN
        }



p <- model$rank

p1 <- 1:p
model$qr$qr
 coef.p <- model$coefficients[model$qr$pivot[p1]]
 covmat.unscaled <- chol2inv(model$qr$qr[p1, p1, drop = FALSE])

 dimnames(covmat.unscaled) <- list(names(coef.p), names(coef.p))
 covmat <- dispersion * covmat.unscaled
 var.cf <- diag(covmat)
 s.err <- sqrt(var.cf)

 confint<-confint(model)
 coeff<- names(coef.p)

 estimates<-as.data.frame(cbind(coeff,coef.p,s.err,confint),row.names="")
 names(estimates)<-c("Coefficient","Value","Std. Err.","2.5 \\%","97.5 \\%")

 row.names(estimates)<- NULL

 write.csv(estimates,paste(dirdataout,"\\est",csvindex,".csv",sep=""),row.names =
FALSE)

 latex(estimates,file="pippo.tex",append=TRUE)

 #Calculate test

 test<-anova(model)
 ResDev=test[1,4]
 Df=test[1,3]
 Chisq=ResDev/Df
 pvalue=1-pchisq(Chisq,Df)
```

```
 testcsv<-as.data.frame(cbind(ResDev,Df,Chisq,pvalue))

 write.csv(test,paste(dirdataout,"\\tst",csvindex,".csv",sep=""),row.names = FALSE)

 invisible(estimates)
}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_standardize.r                                                     #
#                                                                              #
#
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-08-30
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_standardize.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#  BIRO_standardize
#
#
# ##############################################################################


# ##############################################################################
#  BIRO_standardize                                                            #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-08-30
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_standardize is used to fit generalized linear models (linear models, logistic
models and poisson models)
#
# DEPENDENCIES
#
#  R package: stats
#
# ##############################################################################
# PARAMETERS(!=required)
#
# data                        = a data frame containing data
# datafreq                    = a data frame containing frequencies
# target                      = the value of the class variable
# title                       = title to print
# outcome                     = the name of outcome variable
# cov                         = a vector containig the names of covariate variables
# factcov                     = logical. indication if cov are factor or not
# strata                      = the name of the variable of stratification
# strataname                  = strata's label
# std                         = 1 standardize, 0 create only statistical object for
standardization
#                               and print the overall rate
# lev                         = a list contaning levels of depvar and indepvar
# factindep                   = a vector containig logistic values TRUE if the i-th
independent
#                               variable is a factor, FALSE otherwise
# url                         = the url/path of a csv file containing previous
estimates or
#                               estimates providing form the BIRO central analysis
centre
# sucvalue                    = the value of depvar indicating a success
# pop    [NULL]               = the name of the data frame containing population data
(area
#                               level)
# per    [NULL]               = the moltiplication factor of rates
# csvindex                    = the string indicating the name of the output csv file
# dirdataout                  = the path of the output directory
# texfile                     = the path of the output texfile
# dirgraph                    = the path of the output graphs directory
# dirgraphlatex               = the relative path of the output graphs directory (for
Latex)
# caption                     = caption for table and graphs
# map    [0]                  = 1 for map, 0 otherwise
# shapefile [eurnuts3]        = shapefile to use for mapping
# mapvar,                     = variable of shapefile used for mapping
# valuesarea_id               = variable of data used for mapping
# shapearea_id                = variable of shapefile used for mapping
# frqvar="Adjusted Rate",     = represented variable in maps
# labvarmap="Adjusted Rate", = label of represented variable in maps
# mapth=NULL,                 = map thresholds
# progtab,                    = table progressive number
# progbar,                    = barplot progressive number
# progfor,                    = forestplot progressive number
# by     [""]                 = variable for aggregated Statistical Object

# ##############################################################################
#
# EXAMPLE
# y<-rbinom(1000,1,0.5)
```

```
# sex<-rbinom(1000,1,0.5)
# sex<-sex+1
# age<-rbinom(1000,3,0.5)
# age<-age+1
# levsex<-list("Female"=2,"Male"=1)
# levage<-list("0-34"=1,"35-54"=2,"55-74"=3,"75+"=4)
# mydata<-as.data.frame(cbind(y,sex,age))
# BIRO_standardize(data=mydata,
#                  outcome="y",
#                  cov=c("sex","age"),
#                  factcov=c(TRUE,TRUE),
#                  strata=NULL,
#                  lev=list(a=levsex,b=levage),
#                  sucvalue=1,
#                  per=100,
#                  csvindex="provastd",
#                  dirdataout=getwd())
#
# strata<-rep(c("101","102","103","104"),250)
# mydata2<-cbind(mydata,strata)
#
# BIRO_standardize(data=mydata2,
#                  outcome="y",
#                  cov=c("sex","age"),
#                  factcov=c(TRUE,TRUE),
#                  strata="strata",
#                  lev=list(a=levsex,b=levage),
#                  sucvalue=1,
#                  per=100000,
#                  csvindex="provastd",
#                  dirdataout=getwd(),
#                  strataname="Centres")
# ##############################################################################

BIRO_standardize<-function(data=NULL,
                           datafreq=NULL,
                           target=as.character(),
                           title="",
                           outcome,
                           cov,
                           factcov,
                           strata=NULL,
                           std=0,
                           lev=list(),
                           sucvalue,
                           url=NULL,
                           pop=NULL,
                           per,
                           csvindex=ind,
                           dirdataout,
                           strataname="Centres",
                           filename,
                           dirgraph,
                           dirgraphlatex,
                           texfile,
                           caption,
                           chisq,
                           map=0,
                           shapefile="eurnuts3",
                           mapvar,
                           valuesarea_id="area_id",
                           shapearea_id="ID",
                           frqvar="Adjusted Rate",
                           labvarmap="Adjusted Rate",
                           mapth=NULL,
                           progtab,
                           progbar,
                           progfor,
                           by="") {



#To preserve total frequencies (by stratum)
 if (is.null(data)==FALSE) {
  if (is.null(strata)==FALSE) {
   if (strata %in% names(data)) {
    DenomTot<-table(data[,pmatch(strata,names(data))])

ObsTot<-table(data[data[,pmatch(outcome,names(data))]==sucvalue,pmatch(strata,names(da
ta))])
   }
  }
 }
#Print Total RawRate

 if (is.null(data)==FALSE) {
  if (is.null(pop)==TRUE) {
    tottab<-as.data.frame(dim(data)[1])
    names(tottab)<-'N'
    tottab$Obs<-dim(data[data[,pmatch(outcome,names(data))]==sucvalue,])[1]
    tottab$Rate<-round((tottab$Obs/tottab$N)*per,1)
    if (per==100) {
     names(tottab)[pmatch('Rate',names(tottab))]<-'Percentage'
    }
    if (per!=100) {
     names(tottab)[pmatch('Rate',names(tottab))]<-paste('Rate x ',per,sep="")
    }
    if (per==100000) {
     names(tottab)[pmatch('Rate',names(tottab))]<-'Rate x 100000'
    }

#    latex(tottab,file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE,rowlabel="")
#    cat(paste("Type of Diabetes",target),file=texfile,fill = TRUE,append=TRUE)
#    if (target=="2") mylet<-"a" else mylet<-"b"
#    BIRO_df2html(data=tottab,
#                 htmlfile=paste(dirtables,"/i",csvindex,mylet,"d1_3c.html",sep=""),
#                 catcol=NULL)
#
#     file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
#                 paste(dirtables,"/i",csvindex,mylet,"d1_3c.html",sep=""))
#
#    cat(paste("Type of
Diabetes",target),file=paste(dirhtml,"/",csvindex,".html",sep=""),fill =
TRUE,append=TRUE)

  }
 }

 if (is.null(data)==FALSE) {
  if (is.null(strata)==FALSE) {
   if (strata %in% names(data)) {
    if (length(unique(data[,pmatch(strata,names(data))]))==1) {
     strata=NULL
    # cov=NULL
```

```
    }
   }
  } else {
    strata=NULL
   # cov=NULL
  }
 }
 if (is.null(data)==FALSE) {
  if (is.null(strata)==FALSE) {
   if ((strata %in% names(data))==FALSE) {
     strata=NULL
   }
  }
 }

 if (is.null(datafreq)==FALSE) {
  datafreq<-datafreq[datafreq$Insuccesses!="NA",]
 }

 pval=0
 #write("BIRO_standardize called",file="")
 #if (nchar(logfile)>0) {
 # write("BIRO_standardize called",file=logfile,append=TRUE)
 #}

  if (is.null(data)==FALSE) {
   data[data[,pmatch(outcome,names(data))]==99,outcome]<-NA
  }

 if (length(cov)>=0) {
  if (std==0) {
 #99=missing in outcome


   funcdata<-data[,c(pmatch(outcome,names(data)),pmatch(cov,names(data)))]
   names(funcdata)<-c(outcome,cov)

   if (is.null(strata)==FALSE)  {
    funcdata<-cbind(funcdata,data[,pmatch(strata,names(data))])
    names(funcdata)<-c(outcome,cov,strata)
   }

funcdata[,pmatch(outcome,names(funcdata))]<-as.factor(funcdata[,pmatch(outcome,names(f
uncdata))])
   if (length(lev)==0) {
    levels(funcdata[,pmatch(outcome,names(funcdata))])<-lev[[1]]
   }

   if (length(lev)==0) {

levels(funcdata[,pmatch(outcome,names(funcdata))])<-levels(as.factor(funcdata[,pmatch(
outcome,names(funcdata))]))
   }

   for (i in 1:length(cov)) {
     if (length(var) > i & length(lev)>i) {

funcdata[,pmatch(cov[i],names(funcdata))]<-as.factor(funcdata[,pmatch(cov[i],names(fun
cdata))])
      if (is.null(lev[[i+1]])==FALSE) {
       levels(funcdata[,pmatch(cov[i],names(funcdata))])<-lev[[i+1]]
      }
     }
   }
   # calculates contingency table (total frequency of non-missing values)
   tab<-as.data.frame(table(na.omit(funcdata)))

   if (is.null(pop)==FALSE) {
    wtab<-tab
    wtab$start=oneyearbf
    wtab$end=checkdate
    wtab$centre_id=centre_id
    wtab$dbname=dbname
    #write.table(wtab,paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""),row.names =
FALSE,append=TRUE)
    rm(wtab)
   }

   # calculates contingency table of successes

   tabs<-tab[tab[,1]==sucvalue,]
   names(tabs)[dim(tabs)[2]]<-"Successes"
   tabs<-tabs[,-1]
   assign("tabs",tabs,envir = .GlobalEnv)

   if (is.null(pop)==TRUE) {
    # calculates contingency table of failures
    tabf<-tab[tab[,1]!=sucvalue,]
    names(tabf)[dim(tabf)[2]]<-"Insuccesses"
    tabf<-tabf[,-1]
    if (dim(tabf)[1]>0 & dim(tabs)[1]>0) {
     tab<-merge(tabs,tabf,by=cov,all=TRUE)
    }
    if (dim(tabf)[1]==0) {
     tab<-tabs
     assign("tab",tab,envir = .GlobalEnv)
     tab$Insuccesses<-0
    }
    if (dim(tabs)[1]==0) {
     tab<-tabf
     tab$Successes<-0
   }

   if (is.null(strata)==FALSE & is.null(pop)==TRUE) {
    tab<-merge(tabs,tabf,by=c(cov,strata),all=TRUE)
   }

   tab$out=outcome
   wtab<-tab
   wtab$centre_id=centre_id
   wtab$dbname=dbname
   wtab$start=oneyearbf
   wtab$end=checkdate
   if (file.exists(paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""))) {
    cnames=FALSE
   } else {
    cnames=TRUE
   }
   wtab$target=target
   wtab<-wtab[wtab$Successes>0 | wtab$Insuccesses>0,]
   write.table(wtab,paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""),row.names =
FALSE,append=TRUE,col.names=cnames,sep=",")
   rm(wtab)
  }
```

```
  if (is.null(pop)==FALSE & is.null(strata)==FALSE) {
   tab<-tabs
   tab<-merge(pop,tab,by=c(cov,strata),all.x=TRUE,all.y=FALSE)
   tab$Insuccesses<-(tab$pop)-tab$Successes
   tab$out=outcome
   tab$Successes[is.na(tab$Successes)]<-0
   tab$Insuccesses<-ifelse(is.na(tab$Insuccesses),tab$pop,tab$Insuccesses)
   tab<-tab[,-pmatch("pop",names(tab))]
  }
  if (is.null(pop)==FALSE & is.null(strata)==TRUE) {
   tab<-tabs
   tab<-merge(pop,tab,by=c(cov),all.x=TRUE,all.y=TRUE)
   tab$Insuccesses<-(tab$pop)-tab$Successes
   tab$Successes[is.na(tab$Successes)]<-0
   tab$Insuccesses<-ifelse(is.na(tab$Insuccesses),tab$pop,tab$Insuccesses)
   tab<-tab[,-pmatch("pop",names(tab))]
   tab$out=outcome
   wtab<-tab
   wtab$centre_id=centre_id
   wtab$dbname=dbname
   wtab$start=oneyearbf
   wtab$end=checkdate
   if (file.exists(paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""))) {
    cnames=FALSE
   } else {
    cnames=TRUE
   }
   wtab$target=target
   wtab<-wtab[wtab$Successes>0 | wtab$Insuccesses>0,]
   write.table(wtab,paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""),row.names =
FALSE,append=TRUE,col.names=cnames,sep=",")
   rm(wtab)
  }
  datafreq<-tab
 }
 # calculates p-values
 tab<-datafreq
 ptabs<-tab[,c(cov,"Successes")]
 names(ptabs)<-c(cov,"Freq")
 ptabs$myout=1
 ptabi<-tab[,c(cov,"Insuccesses")]
 names(ptabi)<-c(cov,"Freq")
 ptabi$myout=0
 ptab<-rbind(ptabs,ptabi)
 nvar<-length(cov)+1
 var<-c("myout",cov)
 if (is.null(chisq)==FALSE) {
  if (length(cov)==3) {
   chilist<-list()

   for (k in 1:nlevels(factor(ptab[,pmatch(var[4],names(ptab))]))) {
     formula<-as.formula(paste("Freq~",paste(c("myout",chisq[1:2]),collapse="+"),sep=
""))


subptab<-ptab[ptab[,pmatch(var[4],names(ptab))]==sort(unique(ptab[,pmatch(var[4],names
(ptab))]))[k],]
     for (d in 1:length(cov)) {
       subptab[,pmatch(cov[d],names(subptab))]
       for (r in 1:dim(subptab)[1]) {

subptab[r,pmatch(cov[d],names(subptab))]<-as.numeric(substr(subptab[r,pmatch(cov[d],na
mes(subptab))],2,2))
       }
     }



     #subptab$Freq<-as.numeric(as.character(subptab$Freq))



    if (all(na.omit(subptab$Freq)>0,na.rm=TRUE)) {
     if (all(na.omit(subptab$Freq)>1,na.rm=TRUE)) {

      # Cochran-Mantel-Haenszel Chi-Squared Test for Count Data
      mytable<-xtabs(formula=formula,data=subptab)

      if (all(dim(mytable)>=2)) {
       mh<-mantelhaen.test(mytable)

chitab<-as.data.frame(cbind(round(mh$statistic,4),format(round(mh$p.value,4),nsmall=4)
,mh$parameter))
       names(chitab)<-c("Chi-Squared","p.value","df")
       lab4<-paste(unlist(strsplit(var[4],"_")),collapse=" ")
       row.names(chitab)=paste("C-M-H Chi-Squared Test"
,lab4,sort(unique(ptab[,pmatch(var[4],names(ptab))]))[k])
       chitab$centre_id=centre_id
       chitab$dbname=dbname
       chitab$start=oneyearbf
       chitab$end=checkdate
       write.table(chitab,paste(dirdataout,"/i",csvindex,"d1_3b.csv",sep=""
),row.names = FALSE,append=TRUE,col.names=cnames,sep=",")

      # BIRO_df2html(data=chitab,
      #
htmlfile=paste(dirtables,"/",csvindex,letters[k],"d1_3b.html",sep=""),
      #              catcol=NULL)


   #   file.append(paste(dirhtml,"/i",csvindex,".html",sep=""),
   #               paste(dirtables,"/i",csvindex,"d1_3b.html",sep=""))

       for (j in 1:length(names(chitab))) {
         names(chitab)[j]<-paste(unlist(strsplit(names(chitab)[j],"_")),collapse=" ")
       }
        chilist[[k]]<-chitab
      } else {
       chitab<-as.data.frame("One or more cells have less than 2 obs")
       names(chitab)<-"C-M-H Chi-Squared Test"
       chilist[[k]]<-chitab
      }
     }
    }
   }

  if (length(cov)==2) {

   formula<-as.formula(paste("Freq~",paste(c("myout",chisq),collapse="+"),sep=""))

   ptab$Freq<-as.numeric(as.character(ptab$Freq))
   if (all(na.omit(ptab$Freq)>0,na.rm=TRUE)) {
    if (all(na.omit(ptab$Freq)>1)) {
```

```
     mytable<-xtabs(formula=formula,data=ptab)
    # Cochran-Mantel-Haenszel Chi-Squared Test for Count Data

     if (all(dim(mytable)>=2)) {
      mh<-mantelhaen.test(mytable)

chitab<-as.data.frame(cbind(round(mh$statistic,4),round(format(mh$p.value,nsmall=4),4)
,mh$parameter))
      names(chitab)<-c("Chi-Squared","p.value","df")
      row.names(chitab)="C-M-H Chi-Squared Test"
      chitab$centre_id=centre_id
      chitab$dbname=dbname
      write.table(chitab,paste(dirdataout,"/i",csvindex,"d1_3b.csv",sep=""),row.names
= FALSE,append=TRUE,col.names=cnames,sep=",")

    # BIRO_df2html(data=chitab,
    #              htmlfile=paste(dirtables,"/",csvindex,"d1_3b.html",sep=""),
    #              catcol=NULL)


    #file.append(paste(dirhtml,"/i",csvindex,".html",sep=""),
    #            paste(dirtables,"/i",csvindex,"1_3b.html",sep=""))

      for (j in 1:length(names(chitab))) {
        names(chitab)[j]<-paste(unlist(strsplit(names(chitab)[j],"_")),collapse=" ")
      }
     } else {
      chitab<-as.data.frame("One or more cells have less than 2 obs")
      names(chitab)<-"C-M-H Chi-Squared Test"
     }
    }
   }
  }

  if (length(na.omit(cov))==1) {
   if (all(na.omit(ptab$Freqs))>1) {

    formula<-as.formula(paste("Freq~",paste(c(var[1],cov),collapse="+"),sep=""))
    mytable<-xtabs(formula=formula,data=ptab)
    # Cochran-Mantel-Haenszel Chi-Squared Test for Count Data

    if (all(dim(mytable)>=2)) {
     mh<-chisq.test(mytable)

chitab<-as.data.frame(cbind(round(mh$statistic,4),round(mh$p.value,4),mh$parameter))
     names(chitab)<-c("Chi-Squared","p.value","df")
     row.names(chitab)="C-M-H Chi-Squared Test"
     chitab$centre_id=centre_id
     chitab$dbname=dbname
     write.table(chitab,paste(dirdataout,"/i",csvindex,"d1_3b.csv",sep=""),row.names
= FALSE,append=TRUE,col.names=cnames,sep=",")

    # BIRO_df2html(data=chitab,
    #              htmlfile=paste(dirtables,"/",csvindex,"d1_3b.html",sep=""),
    #              catcol=NULL)

    #file.append(paste(dirhtml,"/i",csvindex,".html",sep=""),
    #            paste(dirtables,"/i",csvindex,"1_3b.html",sep=""))

     for (j in 1:length(names(chitab))) {
       names(chitab)[j]<-paste(unlist(strsplit(names(chitab)[j],"_")),collapse=" ")
     }
    }
   }
  }
 }
 # read coefficients from the central repository
 if (is.null(url)==FALSE) {
  coeff=read.csv(url)
  # create dummy in the table (covariates)
  tabdummy<-tab
  for (i in 1:length(cov)) {
    if (factcov[i]==TRUE) {
     for (j in 2:nlevels(factor(tabdummy[,pmatch(cov[i],names(tabdummy))]))) {

tabdummy[,dim(tabdummy)[2]+1]<-ifelse(tabdummy[,pmatch(cov[i],names(tabdummy))]==uniqu
e(tabdummy[,pmatch(cov[i],names(tabdummy))])[order(unique(tabdummy[,pmatch(cov[i],name
s(tabdummy))]))][j],1,0)

names(tabdummy)<-c(names(tabdummy)[1:dim(tabdummy)[2]-1],paste(cov[i],unique(tabdummy[
,pmatch(cov[i],names(tabdummy))])[order(unique(tabdummy[,pmatch(cov[i],names(tabdummy)
))]))][j],sep=""))
     }
    }
  }
  oldcoeff<-as.data.frame(t(coeff[,c(2)]))
  # fitted values
  ncoeff<-dim(oldcoeff)[2]-1

  for (i in 1:dim(tabdummy)[1]) {
    sum=oldcoeff$Intercept
    for (j in 2:ncoeff) {
      newvar<-names(oldcoeff)[j]
      sum=sum+tabdummy[i,pmatch(newvar,names(tabdummy))]*oldcoeff[1,j]
    }
    model$fitted.values[i]=exp(sum[i])/(1+exp(sum[i]))
  }
 }
 if (is.null(url)==TRUE & is.null(strata)==FALSE) {
  # logistic regression
  if (is.null(datafreq)==TRUE) {
   tab<-as.data.frame(table(funcdata))
   if (is.null(pop)==FALSE) {
    wtab<-tab
    wtab$centre_id=centre_id
    wtab$dbname=dbname
    if (file.exists(paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""))) {
     cnames=FALSE
    } else {
     cnames=TRUE
    }
    wtab$target=target
    wtab<-wtab[wtab$Successes>0 | wtab$Insuccesses>0,]
    write.table(wtab,paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""),row.names =
FALSE,append=TRUE,col.names=cnames,sep=",")
    rm(wtab)
   }
   # calculates contingency table of successes
   tabs<-tab[tab[,1]==sucvalue,]
   names(tabs)[dim(tabs)[2]]<-"Successes"
   tabs<-tabs[,-1]

   if (is.null(pop)==TRUE) {
```

```
    # calculates contingency table of failures
    tabf<-tab[tab[,1]!=sucvalue,]
    names(tabf)[dim(tabf)[2]]<-"Insuccesses"
    tabf<-tabf[,-1]
    if (dim(tabf)[1]>0 & dim(tabs)[1]>0) {
     tab<-merge(tabs,tabf,by=cov,all=TRUE)
    }
    if (dim(tabf)[1]==0) {
     tab<-tabs
     tab$Insuccesses<-0
    }
    if (dim(tabs)[1]==0) {
     tab<-tabf
     tab$Successes<-0
   }

   if (is.null(strata)==FALSE & is.null(pop)==TRUE) {
    tab<-merge(tabs,tabf,by=c(cov,strata),all=TRUE)
   }

   tab$out=outcome
   wtab<-tab
   wtab$centre_id=centre_id
   wtab$dbname=dbname
   if (file.exists(paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""))) {
    cnames=FALSE
   } else {
    cnames=TRUE
   }
   wtab$target=target
   wtab<-wtab[wtab$Successes>0 | wtab$Insuccesses>0,]
   write.table(wtab,paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""),row.names =
FALSE,append=TRUE,col.names=cnames,sep=",")
   rm(wtab)
  }
  if (is.null(pop)==FALSE & is.null(strata)==FALSE) {
   tab<-tabs
   tab<-merge(pop,tab,by=c(cov,strata),all.x=TRUE,all.y=FALSE)
   tab$Insuccesses<-(tab$pop)-tab$Successes
   tab$out=outcome
   tab$Successes[is.na(tab$Successes)]<-0
   tab$Insuccesses<-ifelse(is.na(tab$Insuccesses),tab$pop,tab$Insuccesses)
   tab<-tab[,-pmatch("pop",names(tab))]
  }
  if (is.null(pop)==FALSE & is.null(strata)==TRUE) {
   tab<-tabs
   tab<-merge(pop,tab,by=c(cov),all.x=TRUE,all.y=TRUE)
   tab$Insuccesses<-(tab$pop)-tab$Successes
   tab$Successes[is.na(tab$Successes)]<-0
   tab$Insuccesses<-ifelse(is.na(tab$Insuccesses),tab$pop,tab$Insuccesses)
   tab<-tab[,-pmatch("pop",names(tab))]
   tab$out=outcome
   wtab<-tab
   wtab$centre_id=centre_id
   wtab$dbname=dbname
   if (file.exists(paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""))) {
    cnames=FALSE
   } else {
    cnames=TRUE
   }
   wtab$target=target
   wtab<-wtab[wtab$Successes>0 | wtab$Insuccesses>0,]
   write.table(wtab,paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""),row.names =
FALSE,append=TRUE,col.names=cnames,sep=",")
   rm(wtab)
  }
  datafreq<-tab[,1:7]
 }

  if (is.null(data)==TRUE) {
   tab<-datafreq
  }
#  if (cov[1]=="type_dm") {
#
tab[,pmatch("type_dm",names(tab))]<-factor(tab[,pmatch("type_dm",names(tab))],levels=l
evtype_dmst)
#
tab[,pmatch("type_dm",names(tab))]<-relevel(tab[,pmatch("type_dm",names(tab))],ref="1"
)
#  }
  for (q in 1:length(cov)) {
   if (cov[q]=="sex") {
    tab[,pmatch("sex",names(tab))]<-relevel(tab[,pmatch("sex",names(tab))],ref=1)
   }
  }
  for (q in 1:length(cov)) {
   if (cov[q]=="age_c") {
    if (length(unique(tab[,pmatch("age_c",names(tab))]))>=5 & "5" %in%
as.character(unique(tab[,pmatch("age_c",names(tab))]))) {
     tab[,pmatch("age_c",names(tab))]<-relevel(tab[,pmatch("age_c",names(tab))],ref=5)
    }
   }
  }
  for (q in 1:length(cov)) {
   if (cov[q]=="agesex") {
    if (length(unique(tab[,pmatch("agesex",names(tab))]))>=5 & "5" %in%
as.character(unique(tab[,pmatch("agesex",names(tab))]))) {
     tab[,pmatch("agesex",names(tab))]<-relevel(tab[,pmatch("agesex"
,names(tab))],ref=5)
    }
   }
  }
  model<-glm(as.formula(paste("cbind(Successes,Insuccesses)~",paste(cov,collapse="+"
),sep="")),data=tab,family=binomial())
  est.disp <- FALSE
  df.r <- model$df.residual
  if (model$family$family %in% c("poisson","binomial")) {
   dispersion <- 1
  } else if (df.r > 0) {
   est.disp <- TRUE
   if (any(model$weights == 0)) {
    warning("observations with zero weight not used for calculating dispersion")
    dispersion <- sum((model$weights * model$residuals^2)[model$weights > 0])/df.r
   }
  } else {
   est.disp <- TRUE
   dispersion <- NaN
  }


  p <- model$rank
  p1 <- 1:p
  model$qr$qr
  coef.p <- model$coefficients[model$qr$pivot[p1]]
```

```
  covmat.unscaled <- chol2inv(model$qr$qr[p1, p1, drop = FALSE])

  dimnames(covmat.unscaled) <- list(names(coef.p), names(coef.p))
  covmat <- dispersion * covmat.unscaled
  var.cf <- diag(covmat)
  s.err <- sqrt(var.cf)
  confint<-confint.default(model)
  coeff<- names(coef.p)
  estimates<-as.data.frame(cbind(coeff,coef.p,s.err,confint),row.names="")
  names(estimates)<-c("Coefficient","Value","Std. Err.","2.5 \\%","97.5 \\%")
  row.names(estimates)<- NULL
  estimates$codist=centre_id
  estimates$dbname=dbname
   if (file.exists(paste(dirdataout,"/i",csvindex,"d5_1a.csv",sep=""))) {
    cnames=FALSE
   } else {
    cnames=TRUE
   }
  estimates<-as.data.frame(estimates)
  estimates$target=target
  #print(estimates)
  write.table(estimates,paste(dirdataout,"/i",csvindex,"d5_1a.csv",sep=""),row.names
= FALSE,append=TRUE,col.names=cnames,sep=",")

  #write("Estimates written in csv file",file="")
  #if (nchar(logfile)>0) {
  # write("Estimates written in csv file",file=logfile,append=TRUE)
  #}
 }
 if (is.null(strata)==TRUE) {
  tab$Denom <- as.numeric(as.character(tab$Successes+tab$Insuccesses))
  tab$Obs <- as.numeric(as.character(tab$Successes))
  tot_Obs <- sum(tab$Obs,na.rm=TRUE)
  tot_Denom <- sum(tab$Denom,na.rm=TRUE)
  tab<-tab[,pmatch(c("Obs","Denom",cov),names(tab))]
  tab$Obs<-as.numeric(as.character(tab$Obs))
  tab$Denom<-as.numeric(as.character(tab$Denom))
  tab$RawRate<-(tab$Obs/tab$Denom)*per
  # Total rate
  tab<-rbind(tab,NA)
  tab$RawRate[dim(tab)[1]]<-(tot_Obs/tot_Denom)*per
  tab$Obs[dim(tab)[1]]<-tot_Obs
  tab$Denom[dim(tab)[1]]<-tot_Denom
  if (per==100000) {
   tab$RawRate<-round(tab$RawRate,0)
  }
  if (per==100) {
   tab$RawRate<-round(tab$RawRate,1)
  }
  if (per==1000) {
   tab$RawRate<-round(tab$RawRate,1)
  }
  for (i in 1:length(cov)) {
    if (factcov[i]==TRUE) {
     tab[,pmatch(cov[i],names(tab))]<-as.factor(tab[,pmatch(cov[i],names(tab))])
     levels(tab[,pmatch(cov[i],names(tab))])<-lev[[i+1]]
    }
  }
 }
 if (per==100) {
  names(tab)[pmatch("RawRate",names(tab))]<-'Percentage'
 }
 if (per==100000) {
  names(tab)[pmatch("RawRate",names(tab))]<-paste('Rate x ',per,sep="")
 }

 if (per==1000) {
  names(tab)[pmatch("RawRate",names(tab))]<-paste('Rate x ',per,sep="")
  tab$RawRate<-round(as.numeric(as.character(tab$RawRate)),1)
 }

 # Expecteds (standardization)
 if (is.null(strata)==FALSE  ) {
  #std==1
  if (is.null(data)==TRUE) {
  tab$Denom <- as.numeric(as.character(tab$Successes+tab$Insuccesses))

  tab$Obs <- as.numeric(as.character(tab$Successes))
  tot_Obs <- sum(tab$Obs,na.rm=TRUE)
  tot_Denom <- sum(tab$Denom,na.rm=TRUE)
  tab<-tab[,pmatch(c("Obs","Denom",cov),names(tab))]
  tab$Obs<-as.numeric(as.character(tab$Obs))
  tab$Denom<-as.numeric(as.character(tab$Denom))
  tab$RawRate<-(tab$Obs/tab$Denom)*per
  # Total rate
  tab<-rbind(tab,NA)
  tab$RawRate[dim(tab)[1]]<-(tot_Obs/tot_Denom)*per
  tab$Obs[dim(tab)[1]]<-tot_Obs
  tab$Denom[dim(tab)[1]]<-tot_Denom
  if (per==100000) {
   tab$RawRate<-round(tab$RawRate,0)
  }
  if (per==100) {
   tab$RawRate<-round(tab$RawRate,1)
  }
  if (per==1000) {
   tab$RawRate<-round(tab$RawRate,1)
  }
  for (i in 1:length(cov)) {
    if (factcov[i]==TRUE) {
     tab[,pmatch(cov[i],names(tab))]<-as.factor(tab[,pmatch(cov[i],names(tab))])
     levels(tab[,pmatch(cov[i],names(tab))])<-lev[[i+1]]
    }
  }
  }
  if (is.null(data)==TRUE) {
   tab<-datafreq
  }

  tab$Denom <-
as.numeric(as.character(tab$Successes))+as.numeric(as.character(tab$Insuccesses))
  tab$Exp <- round(tab$Denom*model$fitted.value)
  #print(aggregate(as.data.frame(tab$Denom),by=list(STRATA=tab$Strata),FUN=sum))
  # (p^* (1-p^) fitted)

NP1_P<-as.numeric(as.character(tab$Denom))*((model$fitted.value)*(1-model$fitted.value
))
  #*(1-model$fitted.value)
  NP1_P<-aggregate(NP1_P,list(strata=tab[,pmatch(strata,names(tab))]),FUN=sum)
  NP1_P<-NP1_P$x

  tab$Obs <- as.numeric(as.character(tab$Successes))
  tot_Obs <- sum(tab$Obs)
  tot_Denom <- sum(tab$Denom)
```

```
totDenomExp<-sum(as.numeric(as.character(tab$Successes))+as.numeric(as.character(tab$I
nsuccesses)))

  # Keep only Obs, Exp, Denom, strata
  tab<-tab[,pmatch(c("Obs","Exp","Denom",strata),names(tab))]
  tab$Obs<-as.numeric(as.character(tab$Obs))
  tab$Exp<-as.numeric(as.character(tab$Exp))
  tab$Denom<-as.numeric(as.character(tab$Denom))

  # Aggregate data (sums)
  mystrata=list(Strata=tab[,pmatch(strata,names(tab))])
  tab<-aggregate(tab[,c("Obs","Exp","Denom")],by=mystrata,FUN=sum)

  ObsExp<-as.numeric(as.character(tab$Obs))
  Tot_ObsExp<-sum(as.numeric(as.character(tab$Obs)))
  DenomExp<-as.numeric(as.character(tab$Denom))
  totDenom<-sum(as.numeric(tab$Denom))
  popra_<-sum(as.numeric(as.character(tab$Obs)))/totDenom

  p_obs=(as.numeric(as.character(ObsExp))/as.numeric(as.character(tab$Denom)))
#  p_exp=(as.numeric(as.character(tab$Exp))/as.numeric(as.character(tab$Denom)))

 # popra_<-as.numeric(as.character(tab$Obs))/as.numeric(as.character(tab$Denom))

  if (is.null(data)==FALSE & is.null(pop)==TRUE) {
   tab$Obs <- as.numeric(as.character(ObsTot))
   tab$Denom <- as.numeric(as.character(DenomTot))
  }
 # if (is.null(data)==TRUE & is.null(pop)==TRUE) {
 #  #if (target=="2") myletter<-"a" else myletter<-"b"
 #  nametot<-paste("i",ind,"d6_1c",sep="")
 #  if (dbExistsTable(conn, nametot)) {
 #  totdata<-dbReadTable(conn, nametot)
 #
 #  tottype<-totdata[totdata$Type_dm==target & as.character(totdata$Obs)!="\"0\"",]
 #
tottype$Obs2<-as.numeric(substr(as.character(as.character(tottype$Obs)),2,nchar(as.cha
racter(tottype$Obs))-1))
 #
tottype$N2<-as.numeric(substr(as.character(as.character(tottype$N)),2,nchar(as.charact
er(tottype$N))-1))
 #  tottype<-tottype[,-pmatch(c("Obs","N"),names(tottype))]
 #
 #
 #  tab<-merge(tab,tottype,by="Strata",all.x=TRUE,all.y=FALSE)
 #  tab$Obs<-ifelse(tab$Obs2>tab$Obs,tab$Obs2,tab$Obs)
 #  tab$Denom<-ifelse(tab$N2>tab$N,tab$N2,tab$Denom)
 #  tab<-tab[,-pmatch(c("Obs2","N2"),names(tab))]
 #
 # }
 # }
  if (target=="") {
   tab$Denom<-as.numeric(as.character(tab$Denom))/2
  }

  tab$RawRate<-(tab$Obs/tab$Denom)*per
  tab$AdjRate=(tot_Obs/tot_Denom)*(tab$Obs/tab$Exp)*per

  if (target=="") {
   tab$AdjRate<-as.numeric(as.character(tab$AdjRate))*2
  }


se<-sqrt(((popra_/(as.numeric(as.character(tab$Exp))/as.numeric(as.character(DenomExp)
)))^2) * ((1/as.numeric(as.character(DenomExp)))^2) * NP1_P  );
#

  tab$LowAdj=((tab$AdjRate/per)-1.96*se)*per
  tab$UppAdj=((tab$AdjRate/per)+1.96*se)*per

  # Reference rate
  tab<-rbind(tab,NA)

tab$RawRate[dim(tab)[1]]<-(sum(as.numeric(as.character(tab$Obs)))/sum(as.numeric(as.ch
aracter(tab$Denom))))*per
  tab$Obs[dim(tab)[1]]<-tot_Obs
  tab$Denom[dim(tab)[1]]<-tot_Denom
  if (per==100000) {
   tab$RawRate<-round(tab$RawRate,0)
   tab$LowAdj<-ifelse(tab$AdjRate<0,0,tab$LowAdj)
   tab$UppAdj<-ifelse(tab$AdjRate>100000,100000,tab$UppAdj)
   tab$AdjRate<-round(tab$AdjRate,0)
   tab$AdjRate<-ifelse(tab$AdjRate>100000,100000,tab$AdjRate)
   tab$AdjRate<-ifelse(tab$AdjRate<0,0,tab$AdjRate)
   tab$LowAdj<-round(tab$LowAdj,0)
   tab$UppAdj<-round(tab$UppAdj,0)
  }
  if (per==100) {
   tab$RawRate<-round(tab$RawRate,1)
   tab$AdjRate<-round(tab$AdjRate,1)
   tab$LowAdj<-round(tab$LowAdj,1)
   tab$LowAdj<-ifelse(tab$AdjRate<0,0,tab$LowAdj)
   tab$UppAdj<-round(tab$UppAdj,1)
   tab$UppAdj<-ifelse(tab$AdjRate>100,100,tab$UppAdj)
   tab$AdjRate<-ifelse(tab$AdjRate>100,100,tab$AdjRate)
   tab$AdjRate<-ifelse(tab$AdjRate<0,0,tab$AdjRate)

  }
  if (per==1000) {
   tab$RawRate<-round(tab$RawRate,1)
   tab$AdjRate<-round(tab$AdjRate,1)
   tab$LowAdj<-round(tab$LowAdj,1)
   tab$LowAdj<-ifelse(tab$AdjRate<0,0,tab$LowAdj)
   tab$UppAdj<-round(tab$UppAdj,1)
   tab$UppAdj<-ifelse(tab$AdjRate>1000,1000,tab$UppAdj)
   tab$AdjRate<-ifelse(tab$AdjRate>1000,1000,tab$AdjRate)
   tab$AdjRate<-ifelse(tab$AdjRate<0,0,tab$AdjRate)

  }

  fortab<-tab
  #tab<-tab[,-pmatch("SEAdj",names(tab))]
  tab$strataname=strataname
  tab$codist=as.character(centre_id)
  tab$dbname=dbname
   if (file.exists(paste(dirdataout,"/i",csvindex,"d6_1b.csv",sep=""))) {
    cnames=FALSE
   } else {
    cnames=TRUE
   }
  tab<-as.data.frame(tab)
  tab$target=target
```

```
  write.table(tab,paste(dirdataout,"/i",csvindex,"d6_1b.csv",sep=""),row.names =
FALSE,append=TRUE,col.names=cnames,sep=",")
  tab<-tab[,1:(dim(tab)[2])-1]
  names(tab)<-c("Strata","Observed","Expected","N",'Raw Rate','Adjusted Rate','LCL',
'UCL',"Stratename")

  if (map==1) {
   maptab<-as.data.frame(tab)
  }
  if (per==100) {
   names(tab)[pmatch("RawRate",names(tab))]<-'Percentage'
   names(tab)[pmatch("AdjRate",names(tab))]<-'Adj. Percentage'
  }
  if (per==100000) {
   names(tab)[pmatch("RawRate",names(tab))]<-paste('Rate x ',per,sep="")
   names(tab)[pmatch("AdjRate",names(tab))]<-paste('Adj. Rate x ',per,sep="")
  }
  # Graphs


#  assign("tabletext",tabletext,envir=.GlobalEnv)
#  assign("fortab",fortab,envir=.GlobalEnv)

# forestplot(labeltext=tabletext,
#            mean=c(NA,fortab$AdjRate),
#            lower=c(NA,fortab$LowAdj),
#            upper=c(NA,fortab$UppAdj),
#            col=meta.colors(box="royalblue",line="darkblue",
summary="royalblue",zero="red"),
#            cex=cex,
#            is.summary=rep(FALSE,dim(fortab)[1]))









  if (map==1) {
   if (exists("mapcod")) {
    lastvar<-mcv(mapcod,ord=names(mapcod)[2:dim(mapcod)[2]])
    if (lastvar=="country") {
     shapefile="eurnuts0"
    }
    if (lastvar=="macroarea") {
     shapefile="eurnuts1"
    }
    if (lastvar=="region") {
     shapefile="eurnuts2"
    }
    if (lastvar=="province") {
     shapefile="eurnuts3"
    }
    if (shapefile=="eurnuts3") {
     lastvar="province"
    }
    if (shapefile=="eurnuts2") {
     lastvar="region"
    }
    if (shapefile=="eurnuts1") {
     lastvar="macroarea"
    }

    if (maploaded!=shapefile) {
     maploaded=shapefile

     assign("maploaded",shapefile,envir=.GlobalEnv)

     mapfile<-readShapePoly(paste(dirmap,"/",shapefile,".shp",sep=""))

#    mapfile<-mapfile[mapfile$ID %in% unique(mapcod$centre_id),]

     assign("mapfile",mapfile,envir=.GlobalEnv)

    }


    maptab$Strata<-noquote(as.character(maptab$Strata))
    #assign("mapcod",mapcod,envir=.GlobalEnv)
    assign("maptab",maptab,envir=.GlobalEnv)

    mapcod$centre_id<-noquote(as.character(mapcod$centre_id))
    maptab$Strata<-substr(maptab$Strata,2,nchar(maptab$Strata)-1)
    maptab$Observed<-as.numeric(as.character(maptab$Observed))
    maptab$Expected<-as.numeric(as.character(maptab$Expected))
    maptab$N<-as.numeric(as.character(maptab$N))
    maptab$'Raw Rate'<-as.numeric(as.character(maptab$'Raw Rate'))
    maptab$'Adjusted Rate'<-as.numeric(as.character(maptab$'Adjusted Rate'))
    maptab$'LCL'<-as.numeric(as.character(maptab$'LCL'))
    maptab$'UCL'<-as.numeric(as.character(maptab$'UCL'))
    myvalues<-merge(x=mapcod,y=maptab,by.x="centre_id",by.y="Strata"
,all.x=FALSE,all.y=FALSE)
    mapth<-c(min(myvalues$'Adjusted Rate')+sd(myvalues$'Adjusted Rate'),mean(myvalues$
'Adjusted Rate'),max(myvalues$'Adjusted Rate')-sd(myvalues$'Adjusted Rate'))

#    BIRO_map(shape=mapfile,
#             datavalues=myvalues,
#             valuesarea_id=lastvar,
#             shapearea_id="ID",
#             frqvar="Adjusted Rate",
#             labvar=labvarmap,
#             breaks=round(mapth,2),
#             legendpos="topright",
#             dirgraph=dirgraph,
#             namegraph=csvindex)

#    import_pdf(namegraph=paste("i",ind,"g4_8a.png",sep=""),
#               caption=paste("Map:", title,
#                             sep=""),
#               width=1,
#               dirgraph=dirgraphlatex,
#               texfile=texfile)


   }
  }
  for (i in 1:dim(tab)[2]) {
   names(tab)[i]<-paste(unlist(strsplit(names(tab)[i],"_")),collapse=" ")
  }
  }
  #write("BIRO_standardize finished",file="")
  #if (nchar(logfile)>0) {
  # write("BIRO_standardize finished",file=logfile,append=TRUE)
```

```
 #}

 if (is.null(chisq)==FALSE) {
  if (length(cov)<3) {

   tab1<-tab[,1:(round(dim(tab)[2]/2))]
#   tab1[dim(tab1)[1]+1,"Obs"]<-sum(tab1$Obs,na.rm=TRUE)
#   if (is.null(pop)==FALSE) {
#    tab1[dim(tab1)[1]+1,"Denom"]<-sum(tab1$Denom,na.rm=TRUE)
#   }
#   if (is.null(pop)==TRUE) {
#    tab1[dim(tab1)[1]+1,"Population"]<-sum(tab1$Population,na.rm=TRUE)
#   }


   tab2<-tab[,c(1,(round(dim(tab)[2]/2)+1):dim(tab)[2])]
#   tab2[dim(tab2)[1],"Obs"]<-sum(tab2$Obs,na.rm=TRUE)
#   if (is.null(pop)==FALSE) {
#    tab2[dim(tab2)[1],"Denom"]<-sum(tab2$Denom,na.rm=TRUE)
#   }
#   if (is.null(pop)==TRUE) {
#    tab2[dim(tab2)[1],"Population"]<-sum(tab2$Population,na.rm=TRUE)
#   }

   if (dim(tab1)[2]>2) {
    tab1<-na.omit(tab1)

     #1latex(tab1,file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE)

  if (engine=="se") {
    #BIRO_df2html(data=tab1,
    #             htmlfile=paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""),
    #             catcol=NULL)
  }
   }
   if (dim(tab2)[2]>2) {
    tab2<-na.omit(tab2)

    #2latex(tab2,file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE)

    if (engine=="se") {
     # BIRO_df2html(data=tab2,
     #              htmlfile=paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""),
     #              catcol=NULL)
     }
    }

    #latex(chitab,file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE)

    if (engine=="se") {
     #BIRO_df2html(data=chitab,
     #             htmlfile=paste(dirtables,"/",csvindex,"d1_3b.html",sep=""),
     #             catcol=NULL)
    }
  }

  if ((length(cov)==3) & is.null(datafreq)) {
   mycov<-paste(unlist(strsplit(cov[3],"_")),collapse=" ")
   #nlevels(factor(tab[,pmatch(mycov,names(tab))])
   mylevels<-levels(factor(tab[,pmatch(mycov,names(tab))]))
   for (k in 1:length(mylevels)) {
     thislev<-mylevels[k]
      thisvar<-tab[,pmatch(mycov,names(tab))]


#    if (is.null(pop)==FALSE) {
#     printtab[dim(printtab)[1],"Denom"]<-sum(printtab$Denom,na.rm=TRUE)
#    }
#    if (is.null(pop)==TRUE) {
#     printtab[dim(printtab)[1],"Population"]<-sum(printtab$Population,na.rm=TRUE)
#    }

     if (is.null(pop)==FALSE) {
  #lucanames(tab)[pmatch("Denom",names(tab))]<-'Population'
      tab$RawRate<-round(tab$RawRate,3)
      tab[dim(tab)[1],is.na(tab[dim(tab)[1],])]<-""
     }
     if (length(names(chilist))>=k) {
     chitab<-chilist[[k]]
     } else {
     chitab<-NULL
     }
     if (dim(printtab)[1]>2) {

      #latex(printtab,file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE)

      if (engine=="se") {
       #BIRO_df2html(data=printtab,
       #             htmlfile=paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""),
       #             catcol=NULL)
      }
     }

     if (exists("chitab")) {

      if (is.null(chitab)==FALSE) {

       for (i in 1:length(names(chitab))) {
        names(chitab)[i]<-paste(unlist(strsplit(names(chitab)[i],"_")),collapse=" ")
       }
       latex(chitab,file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE,rowname=""
,title="")

       BIRO_df2html(data=chitab,
                    htmlfile=paste(dirtables,"/",csvindex,"d1_3b.html",sep=""),
                    catcol=NULL)


     }
    #latex(chilist[[k]],file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE)
     }
    }
   }
  }

  if (is.na(pmatch("Stratanname",names(tab)))==FALSE) tab<-tab[,-pmatch("Stratanname"
,names(tab))]


if (is.null(strata)==FALSE ) {
 #& is.null(pop)==TRUE

   printtab<-tab
```

```
#  if (per==100) {
#   printtab$'Raw Rate'<-paste(printtab$'Raw Rate', "%")
#   printtab$'Adjusted Rate'<-paste(printtab$'Adjusted Rate', "%")
#   printtab$'LCL'<-paste(printtab$'LCL', "%")
#   printtab$'UCL'<-paste(printtab$'UCL', "%")

#  }
 if (per==1000) {
  printtab$'Raw Rate'<-format(round(as.numeric(as.character(printtab$'Raw Rate'
)),1),nsmall=1)
  printtab$'Adjusted Rate'<-format(round(as.numeric(as.character(printtab$'Adjusted
Rate')),1),nsmall=1)
  printtab$'LCL'<-format(round(as.numeric(as.character(printtab$'LCL')),1),nsmall=1)
  printtab$'UCL'<-format(round(as.numeric(as.character(printtab$'UCL')),1),nsmall=1)
 }
  printtab$Strata<-as.character(printtab$Strata)
  printtab$Strata[length(printtab$Strata)]<-""
  printtab$Strata[is.na(printtab$Strata)]<-""
  printtab$Strata[printtab$Strata=="<NA>"]<-""
  printtab$Strata[dim(printtab)[1]]<-""
  printtab$Expected[is.na(printtab$Expected)]<-""
  printtab$'Adjusted Rate'[is.na(printtab$'Adjusted Rate')]<-""
  printtab$'LCL'[is.na(printtab$'LCL')]<-""
  printtab$'UCL'[is.na(printtab$'UCL')]<-""
  printtab$'LCL'[is.infinite(printtab$'LCL')]<-""
  printtab$'UCL'[is.infinite(printtab$'UCL')]<-""
  printtab$'LCL'[printtab$'LCL'=="NA"]<-""
  printtab$'UCL'[printtab$'UCL'=="NA"]<-""
  printtab$'LCL'[printtab$'LCL'=="Inf"]<-""
  printtab$'UCL'[printtab$'UCL'=="Inf"]<-""
  printtab$'LCL'[printtab$'LCL'=="-Inf"]<-""
  printtab$'UCL'[printtab$'UCL'=="-Inf"]<-""
  printtab$'Adjusted Rate'[printtab$'Adjusted Rate'=="NA"]<-""
  printtab$'Adjusted Rate'[printtab$'Adjusted Rate'=="Inf"]<-""
  printtab$'Adjusted Rate'[printtab$'Adjusted Rate'=="-Inf"]<-""
  printtab<-printtab[-dim(printtab)[1],]

  if (per==100000) {
   printtab$'LCL'<-ifelse(printtab$'Adjusted Rate'<0,0,printtab$'LCL')
   printtab$'LCL'<-ifelse(printtab$'LCL'<0,0,printtab$'LCL')
   printtab$'UCL'<-ifelse(as.numeric(as.character(printtab$'Adjusted Rate'))>100000,
"100000",printtab$'UCL')
   printtab$'UCL'<-ifelse(as.numeric(as.character(printtab$'UCL'))>100000,"100000"
,printtab$'UCL')
   printtab$'Adjusted Rate'<-ifelse(as.numeric(as.character(printtab$'Adjusted Rate'
))>100000,"100000",printtab$'Adjusted Rate')
   printtab$'Adjusted Rate'<-ifelse(as.numeric(as.character(printtab$'Adjusted Rate'
))<0,"    0",printtab$'Adjusted Rate')
  }
  if (per==100) {
   printtab$'LCL'<-ifelse(printtab$'Adjusted Rate'<0,0,printtab$'LCL')
   printtab$'LCL'<-ifelse(printtab$'LCL'<0,0,printtab$'LCL')
   printtab$'UCL'<-ifelse(as.numeric(as.character(printtab$'Adjusted Rate'))>100,
"100.0",printtab$'UCL')
   printtab$'UCL'<-ifelse(as.numeric(as.character(printtab$'UCL'))>100,"100.0"
,printtab$'UCL')
    printtab$'Adjusted Rate'<-ifelse(as.numeric(as.character(printtab$'Adjusted Rate'
))>100,"100.0",printtab$'Adjusted Rate')
   printtab$'Adjusted Rate'<-ifelse(as.numeric(as.character(printtab$'Adjusted Rate'
))<0,"  0.0",printtab$'Adjusted Rate')
  }
  if (per==1000) {
   printtab$'LCL'<-ifelse(printtab$'Adjusted Rate'<0,0,printtab$'LCL')
   printtab$'LCL'<-ifelse(printtab$'LCL'<0,0,printtab$'LCL')
   printtab$'UCL'<-ifelse(as.numeric(as.character(printtab$'Adjusted Rate'))>1000,
"1000.0",printtab$'UCL')
   printtab$'UCL'<-ifelse(as.numeric(as.character(printtab$'UCL'))>1000,"1000.0"
,printtab$'UCL')
   printtab$'Adjusted Rate'<-ifelse(as.numeric(as.character(printtab$'Adjusted Rate'
))>1000,"1000.0",printtab$'Adjusted Rate')
   printtab$'Adjusted Rate'<-ifelse(as.numeric(as.character(printtab$'Adjusted Rate'
))<0,"  0.0",printtab$'Adjusted Rate')
  }

  totObs<-sum(as.numeric(printtab$Obs))
  totPercentage<-round((totObs/totDenom)*100,1)
  printtab$'[O-E]/E'
<-format(round((as.numeric(as.character(ObsExp))-as.numeric(as.character(printtab$Expe
cted)))/as.numeric(as.character(printtab$Expected))*100,1),nsmall=1)
  p<-as.numeric(as.character(printtab$'[O-E]/E'))/100
  n<-as.numeric(printtab$N)

  popra_<-sum(as.numeric(as.character(printtab$Obs)))/totDenom

  assign("printtab",printtab,envir=.GlobalEnv)

#  se<-
sqrt(((popra_/(as.numeric(as.character(printtab$Exp))/as.numeric(as.character(printtab
$N))))^2) * ((1/as.numeric(as.character(printtab$N)))^2) * NP1_P);

   p_obs=(as.numeric(as.character(ObsExp))/as.numeric(as.character(printtab$N)))
   p_exp=(as.numeric(as.character(printtab$Exp))/as.numeric(as.character(printtab$N)))

   printtab$'LCL [O-E]/E'
<-format(round(((((p_obs-p_exp)-1.96*se)*as.numeric(as.character(DenomExp)))/as.numeri
c(as.character(printtab$Exp)))*100,1),nsmall=1)
   printtab$'UCL [O-E]/E'
<-format(round(((((p_obs-p_exp)+1.96*se)*as.numeric(as.character(DenomExp)))/as.numeri
c(as.character(printtab$Exp)))*100,1),nsmall=1)

  if (target=="") {
   printtab$'LCL [O-E]/E'<-as.character(as.numeric(as.character(printtab$'LCL [O-E]/E'
))/2)
   printtab$'UCL [O-E]/E'<-as.character(as.numeric(as.character(printtab$'UCL [O-E]/E'
))/2)

  }

   printtab<-printtab[printtab$N>0,]

  printtab$'LCL [O-E]/E'[printtab$'Adjusted Rate'==""]<-""
  printtab$'UCL [O-E]/E'[printtab$'Adjusted Rate'==""]<-""

   printtab2<-printtab
   if (length(unique(printtab$Strata))>1) {

   lastrow<-c("T",totObs,"",totDenom,totPercentage,"","","","","","","","")
   if (engine=="ce") lastrow<-c("T",totObs,"",totDenom,totPercentage,"","","","","","",
"","","")
   printtab<-rbind(na.omit(printtab),lastrow)

   printtab2<-printtab
   }
```

```
  if (length(unique(printtab$Strata))>1) {
   #Create totals for N and Raw Rate
  if (engine=="ce") {
   if (is.null(data)==TRUE & is.null(pop)==TRUE) {
    #if (target=="2") myletter<-"a" else myletter<-"b"
    nametot<-paste("i",ind,"d6_1c",sep="")
    if (dbExistsTable(conn, nametot)) {
     totdata<-dbReadTable(conn, nametot)
     tottype<-totdata[totdata$Type_dm==target & as.character(totdata$Obs)!="\"0\"",]

tottype$Obs2<-as.numeric(substr(as.character(as.character(tottype$Obs)),2,nchar(as.cha
racter(tottype$Obs))-1))

tottype$N2<-as.numeric(substr(as.character(as.character(tottype$N)),2,nchar(as.charact
er(tottype$N))-1))
     tottype<-tottype[,-pmatch(c("Obs","N"),names(tottype))]

     #assign("printtab2",printtab2,envir = .GlobalEnv)

     printtab2<-merge(printtab2,tottype,by="Strata",all.x=TRUE,all.y=FALSE)
     printtab2$Obs<-ifelse(printtab2$Obs2>printtab2$Obs,printtab2$Obs2,printtab2$Obs)
     printtab2$N<-ifelse(printtab2$N2>printtab2$N,printtab2$N2,printtab2$N)
     printtab2<-printtab2[,-pmatch(c("Obs2","N2"),names(printtab2))]

    }
   }
  }



printtab2$N[length(printtab2$N)]<-sum(as.numeric(as.character(printtab2$N[-length(prin
ttab2$N)])))
   if (is.null(pop)==FALSE) {

printtab2$N[length(printtab2$N)]=(as.numeric(as.character(printtab2$N[length(printtab2
$N)]))/(length(printtab2$N)-1))
   }



printtab2[,5]<-round((as.numeric(as.character(printtab2[,2]))/as.numeric(as.character(
printtab2[,4])))*per,1)

   printtab2$'Adjusted Rate'<-format(round(as.numeric(as.character(printtab2$
'Adjusted Rate')),1),nsmall=1)
   printtab2$'[O-E]/E'<-as.character(format(printtab2$'[O-E]/E',nsmall=1))
   printtab2$'LCL [O-E]/E'<-as.character(format(printtab2$'LCL [O-E]/E',nsmall=1))
   printtab2$'UCL [O-E]/E'<-as.character(format(printtab2$'UCL [O-E]/E',nsmall=1))
   printtab2$'Raw Rate'<-as.character(format(printtab2$'Raw Rate',nsmall=1))
   printtab2$'Adjusted Rate'<-as.character(format(printtab2$'Adjusted Rate',nsmall=1))
   printtab2$'LCL'<-as.character(format(printtab2$'LCL',nsmall=1))
   printtab2$'UCL'<-as.character(format(printtab2$'UCL',nsmall=1))

   names(printtab2)[pmatch('[O-E]/E',names(printtab2))]<-'[O-E]/E \\%'
   names(printtab2)[pmatch('LCL [O-E]/E',names(printtab2))]<-'LCL [O-E]/E \\%'
   names(printtab2)[pmatch('UCL [O-E]/E',names(printtab2))]<-'UCL [O-E]/E \\%'

   #ordering descending
   names(printtab2)[pmatch('Adjusted Rate',names(printtab2))]<-'AR'

   orderprinttab2<-printtab2[1:(dim(printtab2)[1]-1),]
   orderprinttab2<-sort.data.frame(orderprinttab2,by=~"\\-AR")

   printtab2<-rbind(orderprinttab2,printtab2[dim(printtab)[1],])
   names(printtab2)[pmatch('AR',names(printtab2))]<-'Adjusted Rate'


   printtab3<-printtab2


   printtab2$'[O-E]/E \\%'<-format(round(as.numeric(as.character(printtab2$'[O-E]/E
\\%')),1),nsmall=1)
   printtab2$'LCL [O-E]/E \\%'<-format(round(as.numeric(as.character(printtab2$'LCL
[O-E]/E \\%')),1),nsmall=1)
   printtab2$'UCL [O-E]/E \\%'<-format(round(as.numeric(as.character(printtab2$'UCL
[O-E]/E \\%')),1),nsmall=1)
   printtab2$'LCL'<-format(round(as.numeric(as.character(printtab2$'LCL'
)),1),nsmall=1,width=5)
   printtab2$'UCL'<-format(round(as.numeric(as.character(printtab2$'UCL'
)),1),nsmall=1,width=5)


   printtab2$'95\\% C.I.'<-paste("(",format(printtab2$'LCL',nsmall=1,width=5),";"
,format(printtab2$'UCL',nsmall=1,width=5),")",sep="")

   printtab2$'95\\% C.I. [O-E]/E'<-paste("(",format(printtab2$'LCL [O-E]/E \\%'
,nsmall=1,width=5),";",format(printtab2$'UCL [O-E]/E \\%',nsmall=1,width=5),")",sep=""
)
   printtab2$'95\\% C.I.'[printtab2$'95\\% C.I.'=="(   ;   )"]<-""
   printtab2$'95\\% C.I.'[printtab2$'95\\% C.I.'=="( ; )"]<-""
   printtab2$'95\\% C.I.'[printtab2$'95\\% C.I.'=="( )"]<-""
   printtab2$'95\\% C.I. [O-E]/E'[printtab2$'95\\% C.I. [O-E]/E'=="(NA;NA)"]<-""
   printtab2$'95\\% C.I. [O-E]/E'[printtab2$'95\\% C.I. [O-E]/E'=="( )"]<-""
   printtab2$'[O-E]/E \\%'<-as.character(printtab2$'[O-E]/E \\%')
   printtab2$'[O-E]/E \\%'[printtab2$'[O-E]/E \\%'=="NA"]<-""
   printtab2$'[O-E]/E \\%'[is.na(printtab2$'[O-E]/E \\%')]<-""


   #last modify for latex
   names(printtab2)[pmatch('Strata',names(printtab2))]<-'s'
   names(printtab2)[pmatch('Observed',names(printtab2))]<-'O'
   names(printtab2)[pmatch('Expected',names(printtab2))]<-'E'
   names(printtab2)[pmatch('Raw Rate',names(printtab2))]<-'CR'
   names(printtab2)[pmatch('Adjusted Rate',names(printtab2))]<-'AR'
   #names(printtab2)[pmatch('LCL [O-E]/E \\%',names(printtab2))]<-'LCL ([O-E]/E)'
   #names(printtab2)[pmatch('UCL [O-E]/E \\%',names(printtab2))]<-'UCL ([O-E]/E)'

   assign("printtab2",printtab2,envir=.GlobalEnv)

   cat("  ",file=texfile,fill = TRUE,append=TRUE)
   prvar<-10:12
   if (engine=="ce" & target!="\"\"") {
     prvar<-c(12:13)
    vtemp<-printtab2[,8]
    printtab2[,8]<-printtab2[,7]
    printtab2[,7]<-vtemp
   }

   #remove Inf and NA from printtab


   printtab2$'LCL ([O-E]/E)'<-as.character(printtab2$'LCL ([O-E]/E)')
   printtab2$'UCL ([O-E]/E)'<-as.character(printtab2$'UCL ([O-E]/E)')
   printtab2$'[O-E]/E \\%'[printtab2$'[O-E]/E \\%'== "Inf" | printtab2$'[O-E]/E \\%'
```

```
== "-Inf" | printtab2$'[O-E]/E \\%'== "NA"]<-""
   printtab2$'AR'[printtab2$'AR'== "Inf" | printtab2$'AR'== "-Inf" | printtab2$'AR'==
"NA"]<-""
   printtab2$'LCL'[printtab2$'LCL'== "Inf" | printtab2$'LCL'== "-Inf" | printtab2$
'LCL'== "NA"]<-""
   printtab2$'UCL'[printtab2$'UCL'== "Inf" | printtab2$'UCL'== "-Inf" | printtab2$
'UCL'== "NA"]<-""
   printtab2$'LCL ([O-E]/E)'[printtab2$'LCL ([O-E]/E)'== "Inf" | printtab2$'LCL
([O-E]/E)'== "-Inf" | printtab2$'LCL ([O-E]/E)'== "NA"]<-""
   printtab2$'UCL ([O-E]/E)'[printtab2$'UCL ([O-E]/E)'== "Inf" | printtab2$'UCL
([O-E]/E)'== "-Inf" | printtab2$'UCL ([O-E]/E)'== "NA"]<-""
   printtab2$'LCL ([O-E]/E)'[is.na(printtab2$'LCL ([O-E]/E)')]=""
   printtab2$'UCL ([O-E]/E)'[is.na(printtab2$'UCL ([O-E]/E)')]=""
   printtab2$'[O-E]/E \\%'[is.na(printtab2$'[O-E]/E \\%')]=""
   printtab2$'[O-E]/E \\%'[printtab2$'[O-E]/E \\%'=="NA"]=""
#   printtab2$'95\\% C.I.'<-paste("(",printtab2$'LCL'," ",printtab2$'UCL',")",sep="")
#   printtab2$'95\\% C.I. [O-E]/E'<-paste("(",printtab2$'LCL [O-E]/E \\%'," 
",printtab2$'UCL [O-E]/E \\%',")",sep="")

    titlelatex<-gsub("%","\\\\%",as.character(title))
    titlelatex<-gsub("<","$<$",as.character(titlelatex))
    titlelatex<-gsub(">","$>$",as.character(titlelatex))

     if (target=="") {
      #cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      #cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      #cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)
     }
     if (target!="") {
      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",indprint," ",titlelatex,"\\\\ \\bf Type of Diabetes = "
,names(lev[[4]])[as.numeric(target)],"}",file=texfile,fill = TRUE,append=TRUE)
     }

   row.names(printtab2)<-c(1:(dim(printtab2)-1)," ")
   assign("printtab2",printtab2,envir=.GlobalEnv)

   #Printtab delete NA

   #printtab2[dim(printtab2)[1],c(6,13,10,14)]<-c("","","","")

   printtab2[dim(printtab2)[1],c(6,14,11,15)]<-c("","","","")
   colprint<-c(1:6,14,11,15)
   #colprint<-c(1:6,13,10,14)
   if (exists("cevar")) {
   if (target=="" & dividedprev==TRUE) {

printtab2$N[dim(printtab2)[1]]<-as.numeric(as.character(printtab2$N[dim(printtab2)[1]]
))/divisornumber

printtab2$CR[dim(printtab2)[1]]<-round((as.numeric(as.character(printtab2$O[dim(printt
ab2)[1]]))/as.numeric(as.character(printtab2$N[dim(printtab2)[1]])))*per,1)

   }
   printtab2[dim(printtab2)[1],c(6,14,11,15)]<-c("","","","")
   colprint<-c(1:6,14,11,15)
   }
   #ordering descending

   orderprinttab2<-printtab2[1:(dim(printtab2)[1]-1),]
   orderprinttab2$AR<-as.numeric(as.character(orderprinttab2$AR))
   orderprinttab2<-orderprinttab2[order(orderprinttab2$AR,decreasing=TRUE),]
   row.names(orderprinttab2)<-1:dim(orderprinttab2)[1]
   printtab2<-rbind(orderprinttab2,printtab2[dim(printtab)[1],])

printtab2$AR<-format(round(as.numeric(as.character(printtab2$AR)),1),nsmall=1,width=5)

#printtab2$E[dim(printtab2)[1]]<-sum(as.numeric(as.character(printtab2$E[-dim(printtab
2)[1]])))
   printtab2[dim(printtab2)[1],c(6,14,11,15)]<-c("","","","")

   latex(printtab2[,colprint],title="",col.just=c("l","r","r","r","r","r","r","r","r"
),file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE,rowlabel="")

   cat(" ",file=texfile,fill = TRUE,append=TRUE)
   cat("  ",file=texfile,fill = TRUE,append=TRUE)


#latex(printtab2[,c(1,prvar)],title="",c("l",rep("r",length(prvar))),file=texfile,appe
nd=TRUE,table.env=FALSE,na.blank=TRUE,rowlabel="")
   #cat(" ",file=texfile,fill = TRUE,append=TRUE)


#      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)


   cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE,sep="")
   cat("Standardized Estimates ",indprint,".",progtab," - ",titlelatex,
by,file=texfile,fill = TRUE,append=TRUE,sep="")
   cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE,sep="")

  # cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)

   names(printtab2)[pmatch('[O-E]/E \\%',names(printtab2))]<-'[O-E]/E %'
   names(printtab2)[pmatch('LCL [O-E]/E \\%',names(printtab2))]<-'LCL [O-E]/E'
   names(printtab2)[pmatch('UCL [O-E]/E \\%',names(printtab2))]<-'UCL [O-E]/E'
   names(printtab2)[pmatch('95\\% C.I.',names(printtab2))]<-'C.I. (95%)'
   names(printtab2)[pmatch('95\\% C.I. [O-E]/E',names(printtab2))]<-'C.I. [O-E]/E
(95%)'

   BIRO_df2html(data=printtab2[,colprint],
                htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                catcol=NULL)
   cat("Standardized Estimates ",indprint,".",progtab," - ",title,
by,file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE,sep="")
   if (target!="") {
    cat("(Type of Diabetes = ",names(lev[[4]])[as.numeric(target)],")"
,file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
   }
    #file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
    #            paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""))

   #modify after smart latex report
   names(printtab2)[pmatch('AR',names(printtab2))]<-'Adjusted Rate'
   names(printtab2)[pmatch('s',names(printtab2))]<-'Strata'
   names(printtab2)[pmatch('O',names(printtab2))]<-'Observed'
   names(printtab2)[pmatch('E',names(printtab2))]<-'Expected'
   names(printtab2)[pmatch('CR',names(printtab2))]<-'Raw Rate'
```

```
  #printtab2<-printtab3
   #library(gplots)
   #Barplot Statistical Engine Standardization

  #prepared for eventual problem in x1000 indicators
  if (per==100) {
   xlim=c(0,100)
  }
  if (per==1000) {
   xlim=c(0,1000)
  }

  halfheight<-xlim[2]*0.85

  signs<-rep(NA,length(as.numeric(printtab2$'Adjusted Rate'[-dim(printtab)[1]])))

  signs[as.numeric(printtab2$'Adjusted Rate'[-dim(printtab)[1]])>halfheight]<--1
  signs[as.numeric(printtab2$'Adjusted Rate'[-dim(printtab)[1]])<halfheight]<-1


  #printtab2<-printtab2[printtab2$N>0 & printtab2$Expected>0,]

   assign("bubu",printtab2,envir=.GlobalEnv)
   orderprinttab2<-printtab2
   orderprinttab2$'Adjusted Rate'<-as.numeric(as.character(orderprinttab2$'Adjusted
Rate'))
   orderprinttab2<-orderprinttab2[order(orderprinttab2$'Adjusted Rate'
,decreasing=FALSE),]
   row.names(orderprinttab2)<-1:dim(orderprinttab2)[1]
   printtab2<-orderprinttab2

  file = paste(dirgraph,"/i",csvindex,"t",target,"g4_1a.svg",sep="")



  CairoSVG(file, width = 11.33, height =11.33, pointsize=9, bg = "transparent")
   par(mar=c(5.1,5.1,3.6,2.1))

   r <- barplot(height=as.numeric(printtab2$'Adjusted Rate'[-dim(printtab)[1]]),
                names.arg=printtab2$'Strata'[-dim(printtab)[1]],
                beside=FALSE,col="yellow",
                #ylim= c(0,max(na.omit(as.numeric(printtab2$'Adjusted
Rate'[-dim(printtab)[1]])))+30*par("cxy")[2]),
                xlim=xlim,
                angle=90,axisnames=TRUE,
                axes=TRUE,font=1,cex.names=1.5*cex,cex.axis=1.5*cex,horiz=TRUE,las=2)

#legend("topright",legend=printtab2$'Strata'[-dim(printtab)[1]],cex=1.5*cex,col=rainbo
w(length(printtab2$'Strata'[-dim(printtab)[1]])),
  #       pch=1,title=strata)

  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  text(as.numeric(printtab2$'Adjusted Rate'[-dim(printtab)[1]])+signs*4*par("cxy"
)[2],r,format(round(as.numeric(printtab2$'Adjusted Rate'
[-dim(printtab)[1]]),1),nsmall=1),font=1,cex=1.5*cex)
  lines(rep(printtab$'Raw Rate'[dim(printtab)[1]],2),c(-5,1000),col="red",lwd=2)
  axis(3,printtab$'Raw Rate'[dim(printtab)[1]],printtab$'Raw Rate'
[dim(printtab)[1]],col="red",lwd=2,cex=1.5*cex,cex.axis=1.5*cex)
  #par(cex.lab=1.5*cex)
  #par(cex=1.5*cex)
  #ataxis=t(r)

#axis(side=1,at=ataxis,as.numeric(printtab2$'Strata'[-dim(printtab)[1]]),cex.axis=1.5*
cex,cex=1.5*cex,lty=0)

  dev.off()

# bartab<-as.data.frame(cbind(printtab2[,pmatch(c("Strata"),names(printtab2))]))
# names(bartab)<-"Strata"
# bartab$'Adjusted Rate'<-factor('Adjusted Rate')
# bartab$n<-as.numeric(as.character(printtab2[,pmatch('Adjusted
Rate',names(printtab2))]))
# bartab$sum=100
# bartab$id<-centre_id
# print(bartab)
# BIRO_drawbars(tab=bartab,
#                          labelvar="data source",
#
lev=list(a=levels(printtab2$Strata),b=levels(factor(bartab$'Adjusted Rate'))),
#                          beside=TRUE,
#                          number=TRUE,
#                          perc=FALSE,
#                          what="Adjusted Rate",
#                          namegraph=csvindex,
#                          dirgraph=dirgraph,
#                          cex=cex)

# tabbar<-printtab2
# tabbar$n<-as.numeric(as.character('Adjusted Rate'))*100
# tabbar$sum=100

# BIRO_drawbars(tab=tab,
#                          labelvar=NULL,
#                          lev=lev,
#                          beside=TRUE,
#                          number=FALSE,
#                          perc=TRUE,
#                          what="patients",
#                          namegraph=csvindex,
#                          dirgraph=dirgraph,
#                          cex=cex)



  file = paste(dirgraph,"/i",csvindex,"t",target,"g4_1a.jpeg",sep="")
  jpeg(file, width= 813.6, height= 813.6, units="px",pointsize = 9,
      bg = "transparent")
   par(mar=c(5.1,5.1,3.6,2.1))

   r <- barplot(height=as.numeric(printtab2$'Adjusted Rate'[-dim(printtab)[1]]),
                names.arg=printtab2$'Strata'[-dim(printtab)[1]],
                beside=FALSE,col="yellow",
                #ylim= c(0,max(na.omit(as.numeric(printtab2$'Adjusted
Rate'[-dim(printtab)[1]])))+30*par("cxy")[2]),
                xlim=xlim,
                angle=90,axisnames=TRUE,
                axes=TRUE,font=1,cex.names=1.5*cex,cex.axis=1.5*cex,horiz=TRUE,las=2)


#legend("topright",legend=printtab2$'Strata'[-dim(printtab)[1]],cex=1.5*cex,col=rainbo
w(length(printtab2$'Strata'[-dim(printtab)[1]])),
  #       pch=1,title=strata)
```

```
  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  text(as.numeric(printtab2$'Adjusted Rate'[-dim(printtab)[1]])+signs*4*par("cxy"
)[1],r,format(round(as.numeric(printtab2$'Adjusted Rate'
[-dim(printtab)[1]]),1),nsmall=1),font=1,cex=1.5*cex)
  lines(rep(printtab$'Raw Rate'[dim(printtab)[1]],2),c(-5,1000),col="red",lwd=2)
  axis(3,printtab$'Raw Rate'[dim(printtab)[1]],printtab$'Raw Rate'
[dim(printtab)[1]],col="red",lwd=2,cex=1.5*cex,cex.axis=1.5*cex)
  #par(cex.lab=1.5*cex)
  #par(cex=1.5*cex)
  #ataxis=t(r)

#axis(side=1,at=ataxis,as.numeric(printtab2$'Strata'[-dim(printtab)[1]]),cex.axis=1.5*
cex,cex=1.5*cex,lty=0)

  dev.off()

  file = paste(dirgraph,"/i",csvindex,"t",target,"g4_1a.png",sep="")
  CairoPNG(file, width= 813.6, height= 813.6, pointsize=8, bg = "transparent")
   par(mar=c(5.1,5.1,3.6,2.1))

   r <- barplot(height=as.numeric(printtab2$'Adjusted Rate'[-dim(printtab)[1]]),
                names.arg=printtab2$'Strata'[-dim(printtab)[1]],
                beside=FALSE,col="yellow",
                #ylim= c(0,max(na.omit(as.numeric(printtab2$'Adjusted
Rate'[-dim(printtab)[1]])))+30*par("cxy")[2]),
                xlim=xlim,
                angle=90,axisnames=TRUE,
                axes=TRUE,font=1,cex.names=1.5*cex,cex.axis=1.5*cex,horiz=TRUE,las=2)


#legend("topright",legend=printtab2$'Strata'[-dim(printtab)[1]],cex=1.5*cex,col=rainbo
w(length(printtab2$'Strata'[-dim(printtab)[1]])),
  #       pch=1,title=strata)

  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  text(as.numeric(printtab2$'Adjusted Rate'[-dim(printtab)[1]])+signs*4*par("cxy"
)[1],r,format(round(as.numeric(printtab2$'Adjusted Rate'
[-dim(printtab)[1]]),1),nsmall=1),font=1,cex=1.5*cex)
  lines(rep(printtab$'Raw Rate'[dim(printtab)[1]],2),c(-5,1000),col="red",lwd=2)
  axis(3,printtab$'Raw Rate'[dim(printtab)[1]],printtab$'Raw Rate'
[dim(printtab)[1]],col="red",lwd=2,cex=1.5*cex,cex.axis=1.5*cex)
  #par(cex.lab=1.5*cex)
  #par(cex=1.5*cex)
  #ataxis=t(r)

#axis(side=1,at=ataxis,as.numeric(printtab2$'Strata'[-dim(printtab)[1]]),cex.axis=1.5*
cex,cex=1.5*cex,lty=0)

  dev.off()

  file = paste(dirgraph,"/i",csvindex,"t",target,"g4_1a.pdf",sep="")
  pdf(file, width = 11.33, height =11.33 , pointsize = 9, bg = "transparent")
   par(mar=c(5.1,5.1,3.6,2.1))

   r <- barplot(height=as.numeric(printtab2$'Adjusted Rate'[-dim(printtab)[1]]),
                names.arg=printtab2$'Strata'[-dim(printtab)[1]],
                beside=FALSE,col="yellow",
                #ylim= c(0,max(na.omit(as.numeric(printtab2$'Adjusted
Rate'[-dim(printtab)[1]])))+30*par("cxy")[2]),
                xlim=xlim,
                angle=90,axisnames=TRUE,
                axes=TRUE,font=1,cex.names=1.5*cex,cex.axis=1.5*cex,horiz=TRUE,las=2)


#legend("topright",legend=printtab2$'Strata'[-dim(printtab)[1]],cex=1.5*cex,col=rainbo
w(length(printtab2$'Strata'[-dim(printtab)[1]])),
  #       pch=1,title=strata)

  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  text(as.numeric(printtab2$'Adjusted Rate'[-dim(printtab)[1]])+signs*4*par("cxy"
)[1],r,format(round(as.numeric(printtab2$'Adjusted Rate'
[-dim(printtab)[1]]),1),nsmall=1),font=1,cex=1.5*cex)
  lines(rep(printtab$'Raw Rate'[dim(printtab)[1]],2),c(-5,1000),col="red",lwd=2)
  axis(3,printtab$'Raw Rate'[dim(printtab)[1]],printtab$'Raw Rate'
[dim(printtab)[1]],col="red",lwd=2,cex=1.5*cex,cex.axis=1.5*cex)
  #par(cex.lab=1.5*cex)
  #par(cex=1.5*cex)
  #ataxis=t(r)

#axis(side=1,at=ataxis,as.numeric(printtab2$'Strata'[-dim(printtab)[1]]),cex.axis=1.5*
cex,cex=1.5*cex,lty=0)

  dev.off()

     if (target=="") {
      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)
     }
     if (target!="") {
      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"\\\\ \\bf Type of Diabetes = "
,names(lev[[4]])[as.numeric(target)],"}",file=texfile,fill = TRUE,append=TRUE)
     }

    import_pdf(namegraph=paste("i",csvindex,"t",target,"g4_1a.pdf",sep=""),
               caption=paste("Barplots: ",indprint,".", progbar, " - ","Adjusted
Rates ",title,sep=""),
               width=0.90,
               dirgraph=dirgraphlatex,
               texfile=texfile)

    import_pdf_html(image=paste("../graphs/i",csvindex,"t",target,"g4_1a.png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                    caption=paste("Barplots: ",indprint,".", progbar, " - ","Adjusted
Rates ",title,sep=""))




  #print("barplot done")

  names(printtab2)[pmatch('[O-E]/E \\%',names(printtab2))]<-'[O-E]/E'
  names(printtab2)[pmatch('LCL [O-E]/E \\%',names(printtab2))]<-'LCL [O-E]/E'
  names(printtab2)[pmatch('UCL [O-E]/E \\%',names(printtab2))]<-'UCL [O-E]/E'
```

```
  printtab2$'[O-E]/E'<-as.numeric(as.character(printtab2$'[O-E]/E'))
  printtab2$'LCL [O-E]/E'<-as.numeric(as.character(printtab2$'LCL [O-E]/E'))
  printtab2$'UCL [O-E]/E'<-as.numeric(as.character(printtab2$'UCL [O-E]/E'))

  #Forest plot Statistical Engine Standardization

  orderprinttab2<-printtab2
  orderprinttab2<-orderprinttab2[order(orderprinttab2$'[O-E]/E',decreasing=FALSE),]
  orderprinttab2<-orderprinttab2[is.na(orderprinttab2$'[O-E]/E')==FALSE,]
  printtab2<-orderprinttab2

  #printfor<-printtab2[printtab2$'[O-E]/E'!= 0 &  printtab2$'LCL [O-E]/E'!=0 &
printtab2$'UCL [O-E]/E'!=0,]
  file = paste(dirgraph,"/i",csvindex,"t",target,"g4_9a.svg",sep="")
  CairoSVG(file, width = 11.33, height =11.33, pointsize=9, bg = "transparent")

  y<-1:length(printtab2$Strata)
  x<-as.numeric(as.character(printtab2$'[O-E]/E'))
  ucl<-as.numeric(as.character(printtab2$'LCL [O-E]/E'))
  lcl<-as.numeric(as.character(printtab2$'UCL [O-E]/E'))
  xmax<-max(abs(c(min(as.numeric(as.character(printtab2$'LCL [O-E]/E')))-20,
                  max(as.numeric(as.character(printtab2$'UCL [O-E]/E')))+20)))
  xlim<-c(-100,100)
  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  par(mar=c(5.1,5.1,3.6,2.1))

  plot(x,y,pch=16, col="red",xlab='[O-E]/E %',ylab="",type="p"
,lwd=15,xlim=xlim,las=1,cex.axis=1.5*cex,axes=FALSE)
  axis(1,cex.axis=1.5*cex)
  axis(2,at=y,labels=printtab2$Strata,las=1,cex.axis=1.5*cex,)
  for (i in 1:length(y)) {
   par(cex.lab=1.5*cex)
   par(cex=1.5*cex)
   lines(c(ucl[i],lcl[i]),c(i,i),type="l")
  }
  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  lines(c(0,0),c(0,length(y)+1),type="l",col="red")

  dev.off()

  file = paste(dirgraph,"/i",csvindex,"t",target,"g4_9a.jpeg",sep="")
  jpeg(file, width= 813.6, height= 813.6, units="px",pointsize = 9,
      bg = "transparent")

  y<-1:length(printtab2$Strata)
  x<-as.numeric(as.character(printtab2$'[O-E]/E'))
  ucl<-as.numeric(as.character(printtab2$'LCL [O-E]/E'))
  lcl<-as.numeric(as.character(printtab2$'UCL [O-E]/E'))
  xmax<-max(abs(c(min(as.numeric(as.character(printtab2$'LCL [O-E]/E')))-20,
                  max(as.numeric(as.character(printtab2$'UCL [O-E]/E')))+20)))
  xlim<-c(-100,100)
  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  par(mar=c(5.1,5.1,3.6,2.1))

  plot(x,y,pch=16, col="red",xlab='[O-E]/E %', ylab="",type="p"
,lwd=15,xlim=xlim,las=1,cex.axis=1.5*cex,axes=FALSE)
  axis(1,cex.axis=1.5*cex)
  axis(2,y,printtab2$Strata,las=1,cex.axis=1.5*cex)
  for (i in 1:length(y)) {
   par(cex.lab=1.5*cex)
   par(cex=1.5*cex)
   lines(c(ucl[i],lcl[i]),c(i,i),type="l")
  }
  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  lines(c(0,0),c(0,length(y)+1),type="l",col="red")

#  forestplot(labeltext=cbind(c(strata,printtab2$Strata[-length(printtab2$Strata)])),
#
mean=as.numeric(c(NA,as.numeric(as.character(printtab2$'[O-E]/E'))[-length(printtab2$S
trata)])),
#              lower=as.numeric(c(NA,as.numeric(as.character(printtab2$'LCL
[O-E]/E'))[-length(printtab2$Strata)])),
#              upper=as.numeric(c(NA,as.numeric(as.character(printtab2$'UCL
[O-E]/E'))[-length(printtab2$Strata)])),
#              clip=c(min(as.numeric(as.character(printtab2$'LCL
[O-E]/E'))[-length(printtab2$Strata)])-20,
#                     max(as.numeric(as.character(printtab2$'UCL
[O-E]/E'))[-length(printtab2$Strata)])+20),
#              col=meta.colors(box="royalblue",line="darkblue",
summary="royalblue",zero="red"),
#              zero=0,
#              xlab="",
#
is.summary=c(TRUE,rep(FALSE,length(printtab2$Strata[-length(printtab2$Strata)]))),
#              cex=1,
#              graphwidth= unit(7,"inches"))
  dev.off()

  file = paste(dirgraph,"/i",csvindex,"t",target,"g4_9a.png",sep="")
  CairoPNG(file, width= 813.6, height= 813.6, pointsize=8, bg = "transparent")

  y<-1:length(printtab2$Strata)
  x<-as.numeric(as.character(printtab2$'[O-E]/E'))
  ucl<-as.numeric(as.character(printtab2$'LCL [O-E]/E'))
  lcl<-as.numeric(as.character(printtab2$'UCL [O-E]/E'))
  xmax<-max(abs(c(min(as.numeric(as.character(printtab2$'LCL [O-E]/E')))-20,
                  max(as.numeric(as.character(printtab2$'UCL [O-E]/E')))+20)))
  xlim<-c(-100,100)
  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  par(mar=c(5.1,5.1,3.6,2.1))

  plot(x,y,pch=16, col="red",xlab='[O-E]/E %', ylab="",type="p"
,lwd=15,xlim=xlim,las=1,cex.axis=1.5*cex,axes=FALSE)
  axis(1,cex.axis=1.5*cex)
  axis(2,y,printtab2$Strata,las=1,cex.axis=1.5*cex)
  for (i in 1:length(y)) {
   par(cex.lab=1.5*cex)
   par(cex=1.5*cex)
   lines(c(ucl[i],lcl[i]),c(i,i),type="l")
  }
  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  lines(c(0,0),c(0,length(y)+1),type="l",col="red")


  dev.off()

  file = paste(dirgraph,"/i",csvindex,"t",target,"g4_9a.pdf",sep="")
```

```
  pdf(file, width = 11.33, height =11.33 , pointsize = 9, bg = "transparent")

  y<-1:length(printtab2$Strata)
  x<-as.numeric(as.character(printtab2$'[O-E]/E'))
  ucl<-as.numeric(as.character(printtab2$'LCL [O-E]/E'))
  lcl<-as.numeric(as.character(printtab2$'UCL [O-E]/E'))
  xmax<-max(abs(c(min(as.numeric(as.character(printtab2$'LCL [O-E]/E')))-20,
                  max(as.numeric(as.character(printtab2$'UCL [O-E]/E')))+20)))
  xlim<-c(-100,100)
  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  par(mar=c(5.1,5.1,3.6,2.1))

  plot(x,y,pch=16, col="red",xlab='[O-E]/E %', ylab="",type="p"
,lwd=15,xlim=xlim,las=1,cex.axis=1.5*cex,axes=FALSE)
  axis(1,cex.axis=1.5*cex)
  axis(2,y,printtab2$Strata,las=1,cex.axis=1.5*cex)
  for (i in 1:length(y)) {
   par(cex.lab=1.5*cex)
   par(cex=1.5*cex)
   lines(c(ucl[i],lcl[i]),c(i,i),type="l")
  }
  par(cex.lab=1.5*cex)
  par(cex=1.5*cex)
  lines(c(0,0),c(0,length(y)+1),type="l",col="red")

  dev.off()

     if (target=="") {
      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)
     }
     if (target!="") {
      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\cfoot{\\thepage}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small  ",titlelatex,"\\\\ \\bf Type of Diabetes = "
,names(lev[[4]])[as.numeric(target)],"}",file=texfile,fill = TRUE,append=TRUE)
     }

    import_pdf(namegraph=paste("i",csvindex,"t",target,"g4_9a.pdf",sep=""),
               caption=paste("Forest plots: ",indprint,".", progfor, " - ",title,sep=
""),
               width=0.90,
               dirgraph=dirgraphlatex,
               texfile=texfile)

    import_pdf_html(image=paste("../graphs/i",csvindex,"t",target,"g4_9a.png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                    caption=paste("Forest plots: ",indprint,".", progfor, " - "
,title,sep=""))
  #print("forest plot done")
  #Map Statistical Engine Standardization


  #Preparing Data for Final Spider Plots Statistical Engine Standardization
  printtab2$'Adjusted Rate'<-as.numeric(as.character(printtab2$'Adjusted Rate'))
   #  spider_data<-merge(spider_data,printtab2[,c('Strata','Adjusted
Rate')],by='Strata')
   #  names(spider_data)[pmatch('Adjusted Rate',names(spider_data))]<-filename
   #  assign("spider_data",spider_data,envir=.GlobalEnv)
  }

  if (length(unique(printtab$Strata))==1) {
   #assign("printtab2",printtab2,envir=.GlobalEnv)

   latex(printtab2[dim(printtab2)[1],],title=""
,file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE,rowname="")

   BIRO_df2html(data=printtab2[dim(printtab2)[1],],
                htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                catcol=NULL)

    #file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
    #            paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""))

  }


  if (is.null(chisq)==FALSE) {
  stab1<-c(rep(1,dim(tab)[1]-1),rep(0,dim(tab)[1]-1))
  stab2<-rep(tab$Strata[1:dim(tab)[1]-1],2)
  stab3<-c(tab$Observed[1:dim(tab)[1]-1],tab$Expected[1:dim(tab)[1]-1])

  stab<-as.data.frame(cbind(stab1,stab2,stab3))
  names(stab)<-c("A","B","Freq")
  mytable<-xtabs(as.formula("Freq~A+B"),data=stab)
  mh<-chisq.test(mytable)
  chitab<-as.data.frame(cbind(round(mh$statistic,4),round(mh$p.value,4),mh$parameter))
  names(chitab)<-c("Chi-Squared","p.value","df")
  row.names(chitab)="C-M-H Chi-Squared Test"
  chitab$centre_id=centre_id
  chitab$dbname=dbname
  write.table(chitab,paste(dirdataout,"/i",csvindex,"d1_3b.csv",sep=""),row.names =
FALSE,col.names=cnames,sep=",")

  if (is.na(pmatch("centre_id",names(chitab)))==FALSE) chitab<-chitab[,-pmatch(
"centre_id",names(chitab))]

   chitab$'Chi-Squared'<-round(chitab$'Chi-Squared',4)
   chitab$'p.value'<-format(round(chitab$'p.value',4),nsmall=4)

    # BIRO_df2html(data=chitab,
    #              htmlfile=paste(dirtables,"/i",csvindex,"d1_3b.html",sep=""),
    #              catcol=NULL)
    #file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
    #            paste(dirtables,"/i",csvindex,"d1_3b.html",sep=""))

  }


   }


 if (is.null(strata)==TRUE & is.null(pop)==TRUE) {

  printtab<-tab
  covspace<-rep(NA,length(na.omit(cov)))
  for (i in 1:length(na.omit(cov))) {
    covspace[i]<-paste(unlist(strsplit(na.omit(cov)[i],"_")),collapse="")
  }
```

```
 for (i in 1:length(na.omit(names(printtab)))) {
   names(printtab)[i]<-paste(unlist(strsplit(names(printtab)[i]," ")),collapse="")
 }

 if (is.na(pmatch("typedm",names(printtab)))==FALSE) {
  printtab<-sort.data.frame(printtab,as.formula(paste("~",paste(covspace,collapse="+"
),sep="")))

  printtab3<-as.data.frame(NULL)
  for (i in 1:length(unique(na.omit(printtab[,pmatch("typedm",names(printtab))])))) {

    target<-unique(na.omit(printtab[,pmatch("typedm",names(printtab))]))[i]
    subprinttab<-na.omit(printtab[printtab[,pmatch("typedm"
,names(printtab))]==target,])
    totObs<-sum(subprinttab$Obs)
    totDenom<-sum(subprinttab$Denom)
    totPercentage<-round((totObs/totDenom)*100,1)
    lastrow<-as.data.frame(cbind(totObs,totDenom,target,totPercentage))
    names(lastrow)<-c("Obs","Denom","typedm","Percentage")

    for (i in 1:length(na.omit(cov))) {

subprinttab[,pmatch(covspace[i],names(subprinttab))]<-as.character(subprinttab[,pmatch
(covspace[i],names(subprinttab))])
    }

    printtab3<-rbind(printtab3,lastrow)

  }
    printtab3<-sort.data.frame(printtab3,as.formula("~typedm"))

    #calculate tot percentage only one center



#latex(printtab3[dim(printtab3)[1],],file=texfile,append=TRUE,table.env=FALSE,na.blank
=TRUE)
    #BIRO_df2html(data=printtab3[dim(printtab3)[1],],
    #             htmlfile=paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""),
    #             catcol=NULL)
    #file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
    #            paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""))


    # BIRO_df2html(data=chitab,
    #              htmlfile=paste(dirtables,"/i",csvindex,"d1_3b.html",sep=""),
    #              catcol=NULL)

  for (i in 1:length(na.omit(cov))) {
    names(printtab)[i]<-paste(unlist(strsplit(names(printtab)[i],"_")),collapse=" ")
  }

 }

 if (is.na(pmatch("typedm",names(printtab)))==TRUE) {

  printtab<-sort.data.frame(printtab,as.formula(paste("~",paste(covspace,collapse="+"
),sep="")))
  printtab<-printtab[-dim(printtab)[1],]
  totObs<-sum(printtab$Obs)
  totDenom<-sum(printtab$Denom)
  totPercentage<-round((totObs/totDenom)*100,1)
  lastrow<-as.data.frame(cbind(totObs,totDenom,totPercentage))
  names(lastrow)<-c("Obs","N","Percentage")
#  printtab<-rbind(printtab,lastrow)
#  printtab<-lastrow
#  printtab<-printtab[dim(printtab)[1],]

mycalc<-as.data.frame(cbind(dim(data[data[,pmatch(outcome,names(data))]==sucvalue,])[1
],dim(data),0))

    names(mycalc)<-c("Obs","N",'Raw Rate')
    mycalc$'Raw Rate'<-round((mycalc$Obs/mycalc$N)*per,1)

   #latex(mycalc,file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE)

  #BIRO_df2html(data=mycalc,
  #             htmlfile=paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""),
  #             catcol=NULL)

  #file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
  #            paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""))

    # BIRO_df2html(data=chitab,
    #              htmlfile=paste(dirtables,"/i",csvindex,"d1_3b.html",sep=""),
    #              catcol=NULL)

  for (i in 1:length(na.omit(cov))) {
    names(printtab)[i]<-paste(unlist(strsplit(names(printtab)[i],"_")),collapse=" ")
  }
 }

  #file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
  #            paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""))

  file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
              paste(dirtables,"/i",csvindex,"d1_3b.html",sep=""))

 }

 if (is.null(strata)==TRUE & is.null(pop)==FALSE) {

  printtab<-tab
  printtab<-printtab[,-pmatch("RawRate",names(printtab))]

  lastrow<-printtab[dim(printtab),]
  covspace<-rep(NA,length(na.omit(cov)))

  for (i in 1:length(na.omit(cov))) {
    covspace[i]<-paste(unlist(strsplit(na.omit(cov)[i],"_")),collapse="")
  }

  for (i in 1:length(na.omit(names(printtab)))) {
    names(printtab)[i]<-paste(unlist(strsplit(names(printtab)[i]," ")),collapse="")
  }

  printtab<-printtab[-dim(printtab)[1],]
  printtab<-na.omit(printtab)
  printtab[,3]<-as.character(printtab[,3])
  totObs<-sum(printtab$Obs)
  totDenom<-round(sum(printtab$Denom)/dim(printtab)[1])
  totPercentage<-round((totObs/totDenom)*per,1)
```

```
  lastrow<-c(totObs,totDenom,rep(" ",length(na.omit(cov))),totPercentage)


  printtab<-rbind(printtab,lastrow)




#  if (per==100) {
#   printtab$'RawRate'<-paste(printtab$'RawRate', "%")
#   printtab$'Adjusted Rate'<-paste(printtab$'Adjusted Rate', "%")
#   printtab$'LCL'<-paste(printtab$'LCL', "%")
#   printtab$'UCL'<-paste(printtab$'UCL', "%")

#  }

#  if (is.na(pmatch("RawRate",names(printtab)))==FALSE)
tab<-tab[,-pmatch("RawRate",names(printtab))]


  names(printtab)[pmatch("Denom",names(printtab))]<-'Population'

 #
printtab<-sort.data.frame(printtab,as.formula(paste("~",paste(covspace,collapse="+"),sep="")))



#
mycalc<-as.data.frame(cbind(dim(data[data[,pmatch(outcome,names(data))]==sucvalue,])[1],dim(data),0))
#    names(mycalc)<-c("Obs","N",'Raw Rate')
#    mycalc$'Raw Rate'<-round((mycalc$Obs/mycalc$N)*per,1)

   latex(printtab,title="",col.just=rep("r",dim(printtab)[2]),file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE,rowname="")

   BIRO_df2html(data=printtab,
                htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                catcol=NULL)

    #file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
    #            paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""))

  for (i in 1:length(na.omit(cov))) {
    names(printtab)[i]<-paste(unlist(strsplit(names(printtab)[i],"_")),collapse=" ")
  }


  if (exists("chitab")) {

   BIRO_df2html(data=chitab,
                htmlfile=paste(dirtables,"/i",csvindex,"d1_3b.html",sep=""),
                catcol=NULL)

  }
#      file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
#                  paste(dirtables,"/i",csvindex,"d1_3a.html",sep=""))

}
```

```
  return(tab)
}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: biro_trellis.r                                                         #
#                                                                              #
#
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-09-07
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# biro_trellis.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#  BIRO_histtrellis
#  BIRO_densitytrellis
#  BIRO_boxtrellis
#
# ##############################


# ##############################################################################
#  BIRO_histtrellis                                                            #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-09-07
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# BIRO_histtrellis draws Histograms possibly conditioned on other variables.
#
# DEPENDENCIES
#
#  R package: lattice
#
# ##############################################################################
# PARAMETERS(!=required)
#
# data               ! =  a data frame containing data
# var                ! =  the name of the conditionated variable
# strata   [NULL]       =  a vector containing the names of the conditionating
#                          variables
# labvar             ! =  string value contining the label of the conditionated
#                          variable
# labclass           ! =  a vector containing string values (labels of the
#                          conditionating variable)
# lev_class          ! =  list containing the levels of strata
# dirout    [getwd()]   =  the path of the output directory
# ext        [csv]      =  the extention of the output file
# date    [Sys.Date()]  =  the date of analysis
# namegraph          ! =  the name of the output file (text, jpg,png, eps)
# dirgraph           ! =  the path of the output graph directory
# ##############################################################################

BIRO_histtrellis<-function(data,
                           var,
                           strata=NULL,
                           labvar,
                           labclass,
                           lev_class,
                           dirout=getwd(),
                           ext="csv",
                           date=Sys.Date(),
                           namegraph,
                           dirgraph) {


 secondlay<-4
 #X11(width = 21, height =9 , pointsize=9)

 if (is.null(strata)==FALSE) {

  for (i in 1:length(strata)) {
   data[,pmatch(strata[i],names(data))]<-
   as.factor(data[,pmatch(strata[i],names(data))])
   levels(data[,pmatch(strata[i],names(data))])=lev_class[[i]]
  }


  formula<-paste(var,"~",paste(strata,collapse="+"),sep="")
  n<-NULL
  for (i in 1:length(lev_class)) {
   n<-c(n,length(lev_class[[i]]))
  }
  firstlay<-prod(na.omit(n))/secondlay
  formula<-paste("~ ",var," | ",paste(strata,collapse="+"),sep="")

#  write("preparing to do trellis function histogram",file="")
```

```
data<-na.omit(data)

for (k in 1:length(unique(data[,pmatch(strata[3],names(data))]))) {
 # values of the 4th categorical variable
 values<-unique(data[,pmatch(strata[3],names(data))])

 if (strata[3]=="type_dm") {
   allvalues=c("Type 1", "Type 2" ,"Other Type")
   values=c("Other Type","Type 2","Type 1")
   values<-allvalues[sort(na.omit(pmatch(allvalues,values)))]

 }
 tempdata<-data[data[,pmatch(strata[3],names(data))]==values[k],]
 targetprint<-paste(unlist(strsplit(values[k]," ")),collapse="_")
 if (dim(tempdata)[1]>0) {
 BIRO_fd(date=Sys.Date(),
         data=tempdata,
         var=var,
         stratum=strata,
         typeround=c("integer"),
         codtab=paste(ind,"_",targetprint,"_",sep=""),
         dircsv=dirout)



trellis<-histogram(as.formula(as.character(formula)),
                   data=tempdata,
                   #layout=c(firstlay,secondlay),
                   xlab=labvar,
                   type = "density",
                   panel = function(x, ...) {
                     panel.histogram(x, ...)
                     panel.mathdensity(dmath = dnorm, col = "black",
                     args = list(mean=mean(x),sd=sd(x)))
                   } )

 #write("function histogram done",file="")

progdens=progdens+1
file = paste(dirgraph,"/i",namegraph,"_",progdens,"g4_2a.svg",sep="")
CairoSVG(file, width = 11.33, height = 4.84, pointsize=9, bg = "transparent")
plot(trellis)
dev.off()

file = paste(dirgraph,"/i",namegraph,"_",progdens,"g4_2a.jpeg",sep="")
jpeg(file, width= 813.6, height= 813.6, pointsize=9, units="px",
     bg = "transparent")
plot(trellis)
dev.off()

file = paste(dirgraph,"/i",namegraph,"_",progdens,"g4_2a.png",sep="")
CairoPNG(file, width = 480, height = 480,
        pointsize = 9, bg = "transparent")
plot(trellis)
dev.off()

file = paste(dirgraph,"/i",namegraph,"_",progdens,"g4_2a.pdf",sep="")
pdf(file, width = 11.33, height =11.33 , pointsize = 9, bg = "transparent")
plot(trellis)
dev.off()

# write("trellis histogram saved",file="")
   }
  }
 }
 if (is.null(strata)==TRUE) {

#  write("preparing to do trellis function histogram",file="")
  histogram(data[,pmatch(var,names(data))],
            xlab=labvar,
            type = "density",
            panel = function(x, ...) {
               panel.histogram(x, ...)
               panel.mathdensity(dmath = dnorm, col = "black",
               args = list(mean=mean(x),sd=sd(x)))
            } )

  #write("function histogram done",file="")
 }
 if (struc==TRUE) {
   cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
   cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
   cat("  <tr>",file=sthtml,fill=TRUE,append=TRUE)
   cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/"
,namegraph,".png\">","</td>",sep=""),file=sthtml,fill=TRUE,append=TRUE)
   cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
   cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
   cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
 }

}


# ##############################################################################
#  BIRO_densitytrellis                                                         #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-09-07
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# BIRO_densitytrellis draws Kernel Density Plots possibly conditioned on other
# variables.
#
# DEPENDENCIES
#
#  R package: lattice
#
# ##############################################################################
# PARAMETERS(!=required)
#
# data               ! =  a data frame containing data
# var                ! =  the name of the conditionated variable
# strata    [NULL]     =  a vector containing the names of the conditionating
#                         variables
# labvar             ! =  string value contining the label of the conditionated
#                         variable
```

```
# labclass            ! =  a vector containing string values (labels of the
#                          conditionating variable)
# lev_class           ! =  list containing the levels of strata
# dirout    [getwd()]   =  the path of the output directory
# ext        [csv]      =  the extention of the output file
# date    [Sys.Date()]  =  the date of analysis
# namegraph           ! =  the name of the output file (text, jpg,png, eps)
# dirgraph            ! =  the path of the output graph directory
# ##############################################################################

BIRO_densitytrellis<-function(data,
                              var,
                              strata=NULL,
                              labvar,
                              labclass,
                              lev_class,
                              dirout=getwd(),
                              ext="csv",
                              date=Sys.Date(),
                              namegraph,
                              dirgraph) {

 secondlay<-2

 #X11(width = 21, height =9 , pointsize=9)


 for (i in 1:length(strata)) {
  data[,pmatch(strata[i],names(data))]<-
  as.factor(data[,pmatch(strata[i],names(data))])
  levels(data[,pmatch(strata[i],names(data))])=lev_class[[i]]

 }


 formula<-paste(var,"~",paste(strata,collapse="+"),sep="")


 if (is.null(strata)==FALSE) {

 n<-NULL
 for (i in 1:length(lev_class)) {
  n<-c(n,length(lev_class[[i]]))

 }
  firstlay<-prod(na.omit(n))/secondlay

  formula<-paste("~ ",var," | ",paste(strata,collapse="+"),sep="")

#   write("Preparing to do trellis function densityplot",file="")
#   if (nchar(logfile)>0) {
#    write("Preparing to do trellis function densityplot",file=logfile,
#          append=TRUE)
#   }
  data<-na.omit(data)
  for (k in 1:length(unique(data[,pmatch(strata[3],names(data))]))) {
   # values of the 4th categorical variable
   values<-unique(data[,pmatch(strata[3],names(data))])

   if (strata[3]=="type_dm") {
     allvalues=c("Type 1", "Type 2" ,"Other Type")
     values=c("Other Type","Type 2","Type 1")
     values<-allvalues[sort(na.omit(pmatch(allvalues,values)))]

   }
   tempdata<-data[data[,pmatch(strata[3],names(data))]==values[k],]
   if (dim(tempdata)[1]>0) {
  trellis<-densityplot(as.formula(as.character(formula)),data=tempdata,
                       layout=c(firstlay,secondlay))


  # write("function densityplot done",file="")
  # if (nchar(logfile)>0) {
  #  write("function densityplot done",file=logfile,append=TRUE)
  # }

# write("Preparing to do SVG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do SVG file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/",namegraph,"_",progdens,".svg",sep="")
 CairoSVG(file, width = 11.33, height =11.33, pointsize=9, bg = "transparent")
 plot(trellis)
 dev.off()

 write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }

# write("Preparing to do JPEG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do JPEG file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/",namegraph,"_",progdens,".jpeg",sep="")
 jpeg(file, width= 813.6, height= 813.6, pointsize=9, units="px",
      bg = "transparent")
 plot(trellis)
 dev.off()

 write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }

# write("Preparing to do PNG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do PNG file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/",namegraph,"_",progdens,".png",sep="")
 CairoPNG(file, width = 480, height = 480,
         pointsize = 9 ,bg = "transparent")
 plot(trellis)
 dev.off()

 write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
```

```
 }

# write("Preparing to do PDF file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do PDF file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/",namegraph,"_",progdens,".pdf",sep="")
 pdf(file, width = 11.33, height =11.33 , pointsize = 9, bg = "transparent")
 plot(trellis)
 dev.off()

 write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }
   }
  }
 }
 if (is.null(strata)==TRUE) {

#   write("Preparing to do trellis function densityplot",file="")
#   if (nchar(logfile)>0) {
#    write("Preparing to do trellis function densityplot",file=logfile,
#          append=TRUE)
#   }
  data<-na.omit(data)
  trellis<-densityplot(data[,pmatch(var,names(data))],xlab=labvar)

  # write("function densityplot done",file="")
  # if (nchar(logfile)>0) {
  #  write("function densityplot done",file=logfile,append=TRUE)
  # }

# write("Preparing to do SVG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do SVG file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/",namegraph,".svg",sep="")
 CairoSVG(file, width = 11.33, height =11.33, pointsize=9, bg = "transparent")
 plot(trellis)
 dev.off()

 write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }


# write("Preparing to do JPEG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do JPEG file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/",namegraph,".jpeg",sep="")
 jpeg(file, width= 813.6, height= 813.6, pointsize=9, units="px",
      bg = "transparent")
 plot(trellis)
 dev.off()

 write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }

# write("Preparing to do PNG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do PNG file",file=logfile,append=TRUE)
# }


 file = paste(dirgraph,"/",namegraph,".png",sep="")
 CairoPNG(file, width = 480, height = 480,
         pointsize = 9,bg = "transparent")
 plot(trellis)
 dev.off()

 write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }

# write("Preparing to do PDF file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do PDF file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/",namegraph,".pdf",sep="")
 pdf(file, width = 11.33, height =11.33 , pointsize = 9, bg = "transparent")
 plot(trellis)
 dev.off()

 write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }

 }
 if (struc==TRUE) {
   cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
   cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
   cat("  <tr>",file=sthtml,fill=TRUE,append=TRUE)
   cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/"
,namegraph,".png\">","</td>",sep=""),file=sthtml,fill=TRUE,append=TRUE)
   cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
   cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
   cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
 }


}


# ##############################################################################
#  BIRO_boxtrellis                                                             #
# ##############################################################################
# ------------------------
# Authors:
```

```
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-09-07
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
## ----------------------
# DESCRIPTION
#
# BIRO_boxtrellis draws Box Plots possibly conditioned on other variables.
#
# DEPENDENCIES
#
#  R packages: grDevice, lattice, Cairo
#
# ###########################################################################
# PARAMETERS(!=required)
#
# data                ! =  a data frame containing data
# var                 ! =  the name of the conditionated variable
# strata    [NULL]      =  a vector containing the names of the conditionating
#                          variables
# labvar              ! =  string value contining the label of the conditionated
#                          variable
# labclass            ! =  a vector containing string values (labels of the
#                          conditionating variable)
# lev_class           ! =  list containing the levels of strata
# dirout    [getwd()]   =  the path of the output directory
# ext        [csv]      =  the extention of the output file
# date    [Sys.Date()]  =  the date of analysis
# namegraph           ! =  the name of the output file (text, jpg,png, eps)
# dirgraph            ! =  the path of the output graph directory
# ###########################################################################

BIRO_boxtrellis<-function(data,
                          var,
                          strata=NULL,
                          labvar,
                          labclass,
                          lev_class,
                          dirout=getwd(),
                          ext="csv",
                          date=Sys.Date(),
                          namegraph,
                          dirgraph) {

 secondlay<-4

 #X11(width = 21, height =9 , pointsize=9)


 for (i in 1:length(strata)) {
  data[,pmatch(strata[i],names(data))]<-
  as.factor(data[,pmatch(strata[i],names(data))])
  levels(data[,pmatch(strata[i],names(data))])=lev_class[[i]]

 }

 formula<-paste(var,"~",paste(strata,collapse="+"),sep="")

 if (length(strata)>0) {

 n<-NULL
 for (i in 1:length(lev_class)) {
  n<-c(n,length(lev_class[[i]]))

 }
  firstlay<-prod(na.omit(n))/secondlay

  nlev<-NULL
  for (i in 1:length(strata)) {
   nlev<-c(nlev,nlevels(factor(strata)))
  }
 }

 if (is.null(strata)==FALSE) {

  if (length(strata)==1) {
   secondpart<-strata[1]
  }

  if (length(strata)>1) {
   secondpart<-paste(strata[1]," | ",paste(strata[2:length(strata)],
                     collapse="+"))
  }

  formula<-paste(var," ~ ",secondpart)
#  write("preparing to do trellis function bwplot",file="")
  data<-na.omit(data)
  for (k in 1:length(unique(data[,pmatch(strata[3],names(data))]))) {
   # values of the 4th categorical variable
   values<-unique(data[,pmatch(strata[3],names(data))])

   if (strata[3]=="type_dm") {
     allvalues=c("Type 1", "Type 2" ,"Other Type")
     values=c("Other Type","Type 2","Type 1")
     values<-allvalues[sort(na.omit(pmatch(allvalues,values)))]

   }
   tempdata<-data[data[,pmatch(strata[3],names(data))]==values[k],]
   targetprint<-paste(unlist(strsplit(values[k]," ")),collapse="_")

  if (dim(tempdata)[1]>0) {
   BIRO_fd(date=Sys.Date(),
           data=tempdata,
           var=var,
           stratum=strata,
           typeround=c("integer"),
           codtab=paste(ind,"_",targetprint,"_",sep=""),
           dircsv=dirout)


  trellis<-bwplot(as.formula(as.character(formula)),data=tempdata)
  #layout=c(firstlay,secondlay)

  #write("function bwplot done",file="")
 progbox=progbox+1
 file = paste(dirgraph,"/i",namegraph,"_",progbox,"g4_3a.svg",sep="")
 CairoSVG(file, width = 11.33, height =4.84 , pointsize=9, bg = "transparent")
 plot(trellis)
 dev.off()

 file = paste(dirgraph,"/i",namegraph,"_",progbox,"g4_3a.jpeg",sep="")
```

```
  jpeg(file, width= 813.6, height= 348.48, pointsize=9, units="px",
       bg = "transparent")
  plot(trellis)
  dev.off()

  file = paste(dirgraph,"/i",namegraph,"_",progbox,"g4_3a.png",sep="")
  CairoPNG(file, width= 813.6, height= 348.48,
          pointsize = 9, bg = "transparent")
  plot(trellis)
  dev.off()

  file = paste(dirgraph,"/i",namegraph,"_",progbox,"g4_3a.pdf",sep="")
  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
  plot(trellis)
  dev.off()
    }
   }
  }
  if (is.null(strata)==TRUE) {

#  write("preparing to do trellis function bwplot",file="")
   trellis<-bwplot(data[,pmatch(var,names(data))],xlab=labvar)
  # write("function bwplot done",file="")

  file = paste(dirgraph,"/i",namegraph,"_",progbox,"g4_3a.svg",sep="")
  CairoSVG(file, width = 11.33, height =4.84 , pointsize=9, bg = "transparent")
  plot(trellis)
  dev.off()

  file = paste(dirgraph,"/i",namegraph,"_",progbox,"g4_3a.jpeg",sep="")
  jpeg(file, width= 813.6, height= 348.48, pointsize=9, units="px",
       bg = "transparent")
  plot(trellis)
  dev.off()

  file = paste(dirgraph,"/i",namegraph,"_",progbox,"g4_3a.png",sep="")
  CairoPNG(file, width= 813.6, height= 348.48,
          pointsize = 9, bg = "transparent")
  plot(trellis)
  dev.off()

  file = paste(dirgraph,"/i",namegraph,"_",progbox,"g4_3a.pdf",sep="")
  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
  plot(trellis)
  dev.off()

  }
  if (struc==TRUE) {
    cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
    cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
    cat("  <tr>",file=sthtml,fill=TRUE,append=TRUE)
    cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/"
,namegraph,".png\">","</td>",sep=""),file=sthtml,fill=TRUE,append=TRUE)
    cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
    cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
    cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
  }


}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: biro_util.r                                                            #
#                                                                              #
#                                                                              #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-08-17
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_trellis.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#  varclass
#  classlabel
#  classlabellist
#  BIRO_dframe
#
# ##############################################################################

# ##############################################################################
#
# including also:
#
# sort.data.frame
# Author: Kevin Wright (kw.statr@gmail.com)
# with some ideas from Andy Liaw
# http://tolstoy.newcastle.edu.au/R/help/04/07/1076.html
#
# ##############################################################################

# ##############################################################################
#  varclass
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-08-17
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# varclass: the output is a vector with the same length of v,
#           with elements the values of v espressed in classes,
#           according the c thresholds vector
#
# DEPENDENCIES
#
#  R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# v        !   = a vector containing data
# c        !   = a vector containing the thresholds
# ##############################################################################

varclass<-function(v,c) {

 x<-rep(NA,length(v))
 n<-length(c)

for(i in 1:n) {
 if (i==1) {
  x<-ifelse(is.na(v)==FALSE & v < c[i],i,x)
 }
 if (i>1) {
  x<-ifelse(is.na(v)==FALSE & v >= c[i-1] & v < c[i],i,x)
 }
  x<-ifelse(is.na(v)==FALSE & v >= c[n],n+1,x)
 }
 return(x)
}

# ##############################################################################
#  classlabel
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-08-17
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
```

```r
# classlabel: create label from a thresholds vector. output is a vector
#
# DEPENDENCIES
#
#  R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# v          !  =        a vector containing the thresholds
# ##############################################################################

classlabel<-function(v) {
 label<-rep(NA,length(v)+1)
 for (i in (1:length(v))) {
  if (i==1) {label[i]<-paste("[0 - ",v[1],")",sep="")}
  if (i>1) {label[i]<-paste("[",v[i-1]," - ",v[i],")",sep="")}
 }
 label[length(v)+1]<-paste("[",v[length(v)],"+)",sep="")
 return(label)
}

# ##############################################################################
#  classlabellist
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-08-17
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# classlabellist: create label from a thresholds vector. output is a list
#
# DEPENDENCIES
#
#  R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# v               ! =     a vector containing the thresholds
# ##############################################################################

classlabellist<-function(v) {
 label<-rep(NA,length(v)+1)
 for (i in (1:length(v))) {
  if (i==1) {label[i]<-paste("[0 - ",v[1],")",sep="")}
  if (i>1) {label[i]<-paste("[",v[i-1]," - ",v[i],")",sep="")}
 }
 label[length(v)+1]<-paste("[",v[length(v)]," +)",sep="")

 labellist<-list()
 for (i in 1:(length(v)+1)) {
 labellist[[i]]=i
 }
 names(labellist)<-label
 return(labellist)
}


# ##############################################################################
# sort.data.frame
# ##############################################################################
# Author: Kevin Wright (kw.statr@gmail.com)
# with some ideas from Andy Liaw
# http://tolstoy.newcastle.edu.au/R/help/04/07/1076.html
#
# ##############################################################################
#
#  DESCRIPTION
#
#  A unified method for sorting data frames with mixed data types
#  is provided by the following function by Kevin Wright.
#
# ##############################################################################

sort.data.frame <- function(x, by){
    # Author: Kevin Wright (kw.statr@gmail.com)
    # with some ideas from Andy Liaw
    # http://tolstoy.newcastle.edu.au/R/help/04/07/1076.html

    # x: A data.frame
    # by: A one-sided formula using + for ascending and - for descending
    #     Sorting is left to right in the formula

    # Useage is:
    # library(nlme);
    # data(Oats)
    # sort(Oats, by= ~nitro-Variety)

    if(by[[1]] != "~")
        stop("Argument 'by' must be a one-sided formula.")

    # Make the formula into character and remove spaces
    formc <- as.character(by[2])
    formc <- gsub(" ", "", formc)
    # If the first character is not + or -, add +
    if(!is.element(substring(formc, 1, 1), c("+", "-")))
        formc <- paste("+", formc, sep = "")

    # Extract the variables from the formula
    vars <- unlist(strsplit(formc, "[\\+\\-]"))
    vars <- vars[vars != ""] # Remove any extra "" terms

    # Build a list of arguments to pass to "order" function
    calllist <- list()
    pos <- 1 # Position of + or -
    for(i in 1:length(vars)){
        varsign <- substring(formc, pos, pos)
        pos <- pos + 1 + nchar(vars[i])
        if(is.factor(x[, vars[i]])){
            if(varsign == "-") {
                calllist[[i]] <- -rank(x[, vars[i]])
            } else {
                calllist[[i]] <- rank(x[, vars[i]])
            }
        } else {
            if(varsign == "-") {
```

```
            calllist[[i]] <- -x[, vars[i]]
        } else {
            calllist[[i]] <- x[,vars[i]]
        }
    }
  }
  return(x[do.call("order", calllist), ])
}


# ##############################################################################
#  BIRO_dframe
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-08-17
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# BIRO_dframe: print on screen or in a file the description of a data frame
#
# DEPENDENCIES
#
#  R package: base
#
# ##############################################################################
# PARAMETERS(!=required)
#
# dataname      !       =  the name (string) of the target data frame
# monitor       [TRUE]  =  logical. True if you want the description printed
#                          on the screen
# filelog       [""]    =  the name and the path of the log file
# ##############################################################################

BIRO_dframe<-function(dataname,pos=-1,monitor=TRUE,filelog="") {

 mydata<-get(dataname,pos = pos, envir = as.environment(pos))

 if (monitor==TRUE) {

   print(
"################################################################################################
#####",quote=FALSE)
   print(paste("#                    DATASET name:",dataname),quote=FALSE)
   print("#",quote=FALSE)
   print(paste("#                  Number of Rows:",dim(mydata)[1]),quote=FALSE)
   print(paste("#               Number of Columns:",dim(mydata)[2]),quote=FALSE)
   print(
"#+-------------------------------------------------------------------------------------------
-----",quote=FALSE)
   print("#                 Not Valid / Not Available Values...",quote=FALSE)
   print("#",quote=FALSE)
   for (i in 1:dim(mydata)[2]) {
    NAnum<-length(mydata[is.na(mydata[,i]),i])
    Infnum<-length(mydata[is.infinite(mydata[,i]),i])
    print(paste("#| ",names(mydata)[i],NAnum+Infnum,sep=" "),quote=FALSE)
   }
   print(
"################################################################################################
#####",quote=FALSE)

  }

  if (nchar(filelog)>0) {

    cat(
"################################################################################################
#####",file=filelog,append=TRUE,fill=TRUE)
    cat(paste("# NOTE                 DATASET name:"
,dataname),file=filelog,append=TRUE,fill=TRUE)
    cat("#",file=filelog,append=TRUE,fill=TRUE)
    cat(paste("#                  Number of Rows:"
,dim(mydata)[1]),file=filelog,append=TRUE,fill=TRUE)
    cat(paste("#               Number of Columns:"
,dim(mydata)[2]),file=filelog,append=TRUE,fill=TRUE)
    cat(
"#+-------------------------------------------------------------------------------------------
-----",file=filelog,append=TRUE,fill=TRUE)
    cat("#                          Not Valid / Not Available Values..."
,file=filelog,append=TRUE,fill=TRUE)
    cat("#",file=filelog,append=TRUE,fill=TRUE)
    for (i in 1:dim(mydata)[2]) {
     NAnum<-length(mydata[is.na(mydata[,i]),i])
     Infnum<-length(mydata[is.infinite(mydata[,i]),i])
     cat(paste("#| ",names(mydata)[i],NAnum+Infnum,sep=" "
),file=filelog,append=TRUE,fill=TRUE)
    }
  cat(
"################################################################################################
#####",file=filelog,append=TRUE,fill=TRUE)
  }


 }



# ##############################################################################
#  BIRO_dframe
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-08-17
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# BIRO_dframe: print on screen or in a file the description of a data frame
#
# DEPENDENCIES
#
#  R package: base
#
```

```
# ##############################################################################
# PARAMETERS(!=required)
#
# dataname      !       =  the name (string) of the target data frame
# monitor       [TRUE]  =  logical. True if you want the description printed
#                          on the screen
# filelog       [""]    =  the name and the path of the log file
# ##############################################################################

mcv<-function(mydata,ord) {


 lastmiss=length(ord)+1
 for (i in length(ord):1) {

  #find any missing value
   anymiss=any(is.na(mydata[,i]))
   if (anymiss==TRUE) {
    lastmiss=i
   }
 }
 if (lastmiss>1) {
  mcvar<-pmatch(ord[lastmiss-1],names(mydata))
 }
 if (lastmiss==1) {
  mcvar=NA
 }
 toreturn<-unique(names(mydata)[mcvar])
 if (is.null(toreturn)) {
  toreturn<-""
 }
 return(toreturn)
}
```

```
# ##############################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_webplots.r                                                        #
#                                                                              #
#
# ##############################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_location.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_webplots
#
# ##############################################################################

# ##############################################################################
#  BIRO_webplots                                                               #
# ##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2010-10-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.0
# ------------------------
# DESCRIPTION
#
# BIRO_webplots draws webplots for a conditionated distribution
#
# DEPENDENCIES
#
#  R packages: grDevice, graphics, Cairo
#
# ##############################################################################
# PARAMETERS(!=required)
#
# data        !             = dataset containing data
# var         !             = the name of the target variable
# labvar      !             = a vector containing the labels of var
# title       !             = a string containing the title of graph
# ext         [csv]         = the extension of the output files
# date        [System date] = date of analysis
# dirout      [current]     = the path of the output text files
# namegraph   !             = the coded name of the output files
# dirgraph    [current]     = the path of the output (jpg,eps,png files)
# dirdataout  !             = the path of the output (data files)
# ##############################################################################


BIRO_spider<-function(data,
                      var,
                      labvars,
                      title,
                      dirout=getwd(),
                      ext="csv",
                      date=Sys.Date(),
                      namegraph,
                      dirgraph,
                      dirdataout) {

# write("Preparing to do SVG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do SVG file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/i",namegraph,"g4_7a.svg",sep="")
 CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
 stars(data[,pmatch(var,names(data))], key.loc=c(2,35),
 main=title, frame=T)

 dev.off()

 write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }


# write("Preparing to do JPEG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do JPEG file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/i",namegraph,"g4_7a.jpeg",sep="")
 jpeg(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent",
      units="px")
```

```
 stars(data[,pmatch(var,names(data))], key.loc=c(2,35),
 main=title, frame=T)

 dev.off()

 write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }

# write("Preparing to do PNG file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do PNG file",file=logfile,append=TRUE)
# }

  file = paste(dirgraph,"/i",namegraph,"g4_7a.png",sep="")
 CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
 stars(data[,pmatch(var,names(data))], key.loc=c(2,35),
 main=title, frame=T)

 dev.off()

 write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }

# write("Preparing to do PDF file",file="")
# if (nchar(logfile)>0) {
#  write("Preparing to do PDF file",file=logfile,append=TRUE)
# }

 file = paste(dirgraph,"/i",namegraph,"g4_7a.pdf",sep="")
 pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
 stars(data[,pmatch(var,names(data))], key.loc=c(2,35),
 main=title, frame=T)

 dev.off()

 write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
 if (nchar(logfile)>0) {
  write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
 }



 write.csv(data[,pmatch(var,names(data))],
           paste(dirdataout,"/i",csvindex,"d4_7a.csv",sep=""),
           row.names = FALSE)

 write("Webplot data stored",file="")
 if (nchar(logfile)>0) {
  write("Webplot data stored",file=logfile,append=TRUE)
 }

 if (struc==TRUE) {
   cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
   cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
   cat("  <tr>",file=sthtml,fill=TRUE,append=TRUE)
   cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/web"
,namegraph,".png\">","</td>",sep=""),file=sthtml,fill=TRUE,append=TRUE)
   cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
   cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
   cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
 }


}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_ce_datastep.r                                                     #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_ce_datastep.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_data_format
#   BIRO_loaddata
#
################################################################################


################################################################################
#  BIRO_data_format
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_data_format is used to format variables in a dataset
#
# DEPENDENCIES
#
# BIRO_ce_setup.r
#
################################################################################
#PARAMETERS:
##data [episode] = tareget dataset
#n_var [NULL]  = a vector containing the names of numeric variables
#c_var [NULL]  = a vector containing the names of character variables
#d_var [NULL]  = a vector containing the names of date variables storing as
#                string
#f_var [NULL]  = a vector containing the names of factor variables
#datefmt  !    = input date format
################################################################################


BIRO_data_format<-function(data=episode,
                           n_var=NULL,
                           c_var=NULL,
                           d_var=NULL,
                           f_var=NULL,datefmt) {



  #convert into numeric variables
   if (is.null(n_var)==FALSE) {
    n_var<-na.omit(n_var)
    if (length(n_var)>0) {
     for (i in 1:length(n_var)) {
      data[,pmatch(n_var[i],names(data))]<-
      as.numeric(as.character(data[,pmatch(n_var[i],names(data))]))
     }
    }
   }


  #convert into string variables
   if (is.null(c_var)==FALSE) {
    c_var<-na.omit(c_var)
    if (length(c_var)>0) {
     for (i in 1:length(c_var)) {
      data[,pmatch(c_var[i],names(data))]<-
      as.character(data[,pmatch(c_var[i],names(data))])
     }
    }
   }


  #convert into date variables
    if (is.null(d_var)==FALSE) {
     d_var<-na.omit(d_var)
     if (length(d_var)>0) {
      for (i in 1:length(d_var)) {
       data[,pmatch(d_var[i],names(data))]<-
       as.Date(as.character(data[,pmatch(d_var[i],names(data))]),format=datefmt)
```

```
      }
     }
    }


  #convert into categorical variables
   if (is.null(d_var)==FALSE) {
    f_var<-na.omit(f_var)
    if (length(f_var)>0) {
     for (i in 1:length(f_var)) {
      data[,pmatch(f_var[i],names(data))]<-
       as.factor(as.character(data[,pmatch(f_var[i],names(data))]))
     }
    }
   }


  return(data)

}


##############################################################################
#  BIRO_loaddata
##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
#-------------------------
# DESCRIPTION
#
# BIRO_loaddata load data from a db or a csv file
#
# DEPENDENCIES
#
##############################################################################
#PARAMETERS:
#dbformat        = the format of the database ("postgres","csv")
#drv             = the JDBC driver to connect R to the database
#                  (if not dbformat="csv")
#pathdb          = the path of the database
#user            = a string containing the username
#password        = a string containing the password
#dbname          = a string containing the database name
#dirdatastore    = a string containing the path of the input csv files
#centre_id       = a string containing the centre id
#yearnow         = a numeric value containing the last possible year of the
#                  episode dates
#fromyear        = a numeric value containing the first possible year where
#                  data are ok
#fromyearl       = a numeric value containing the first possible year where
#                  data are ok for the trends
#refyear         = a string containing year to calculate the indicators with
#                  "in the last 12 months" suffix
#refdate         = a string date to calculare age in the statistics about the
#                  refyear interval
##############################################################################


BIRO_loaddata<-function(dbformat,
                        drv,
                        pathdb,
                        user,
                        password,
                        dbname,
                        dirdatastore,
                        centre_id,
                        yearnow,
                        fromyear,
                        fromyearl,
                        refyear,
                        refdate) {

 if (dbformat=="postgres") {

     assign("conn",dbConnect(drv,pathdb,user,password,dbname),envir=.GlobalEnv)



  ## Import data from Postgres

  patvar<-c("SEX","DOB","DT_DIAG","TYPE_DM")

  patient<-as.data.frame(dbGetQuery(conn,
           "Select DISTINCT patient_id FROM profile"))

  Q<-rep(NA,length(patvar))
  for (i in 1:length(Q)) {
   Q[i]<-paste("Select DISTINCT profile_field_value AS ",patvar[i],
               ",patient_id  FROM profile WHERE profile_field_name='",
               patvar[i],"'",sep="")
  }

  for (i in 1:length(Q)) {
   temp<-as.data.frame(dbGetQuery(conn, Q[i]))
   patient<-merge(patient,temp,by="patient_id",all.x=T)
   rm(temp)
  }

  rm(Q)

  episode<-as.data.frame(dbGetQuery(conn,
           "Select DISTINCT patient_id, episode_data_id FROM episode_data"))

  epivar=c("EPI_DATE","STROKE","MI","AMPUT","DIALYSIS","WEIGHT","BMI",
           "SBP","DBP","HYPERTENSION","HBA1C","CREAT","CHOL","HDL","LDL",
           "DRUG_THERAPY","HEIGHT","MA_TEST","SMOK_STAT","RETIN_L",
           "RETIN_R","ULCER_L","ULCER_R","FOOT_EXAM","ESRF",
           "EYE_EXAM","PUMP_THERAPY","ORAL_THERAPY")
  Q<-rep(NA,length(epivar))

  for (i in 1:length(Q)) {
   Q[i]<-paste("Select DISTINCT data.episode_field_value AS ",epivar[i],
               ",data.episode_data_id FROM data WHERE episode_field_name='",
               epivar[i],"'",sep="")
  }
```

```
    for (i in 1:length(Q)) {
   temp<-as.data.frame(dbGetQuery(conn, Q[i]))
   episode<-merge(episode,temp,by="episode_data_id",all.x=T,all.y=F)
   rm(temp)
  }
  rm(Q)

  site<-as.data.frame(dbGetQuery(conn, "Select * FROM site_profile"))

  centre_id=as.character(
     unique(dbGetQuery(conn, "Select site_header_id from site_header"))[1])

  #Store data read from db
  csvpatient<-paste(dirstoreout,"/patient.csv",sep="")
  csvepisode<-paste(dirstoreout,"/episode.csv",sep="")
  #csvpopulation<-paste(dirstoreout,"/pop.csv",sep="")

  write.csv(patient,csvpatient)
  write.csv(episode,csvepisode)

  #write.csv(pop,csvpopulation)

#####################################################
#####################################################
#####################################################
#Temporarly here (waiting the changes to the XML Schema)
#  pop<-read.csv(paste(dirce,"/","pop.csv",sep=""))
#  write.csv(episode,csvpopulation)

#  assign("pop",pop,envir=.GlobalEnv)
#################
#####################################################
#####################################################
#####################################################
 }

#if (dbformat=="csv") {
# if (length(list.files(dirdatastore)) > 0) {
#  patient<-read.csv(paste(dirdatastore,"/patient.csv",sep=""))
#  episode<-read.csv(paste(dirdatastore,"/episode.csv",sep=""))
#  # pop<-read.csv(paste(dirdatastore,"/pop.csv"))
# }
#}

  patient<-BIRO_data_format(data=patient,
                            n_var="patient_id",
                            d_var=c("dob","dt_diag"),
                            f_var="type_dm",
                            datefmt="%Y-%m-%d")


  episode<-BIRO_data_format(data=episode,
                          n_var=c("episode_data_id","patient_id","weight",
                                  "height","sbp","dbp","creat","chol","hdl",
                                  "ldl","ma_test","bmi","hba1c","stroke","mi",
                                  "hypertension"),
                          c_var=,d_var=c("epi_date"),
                          f_var=c("mi","amput","dialysis","hypertension",
                                  "drug_therapy"),
                          datefmt="%Y-%m-%d")
```

```
  source(paste(dirce,"/source/R/formats/BIRO_ce_recode.r",sep=""))

    #assign patient, episode, population to the global environment
     assign("patient",patient,envir=.GlobalEnv)
     assign("episode",episode,envir=.GlobalEnv)
     assign("pop",pop,envir=.GlobalEnv)
     assign("site",site,envir=.GlobalEnv)

####################################
#   Timeframe for analysis         #
####################################


  episode$epi_date<-as.character(episode$epi_date)
  substr(episode$epi_date,1,4)[which(as.numeric(substr(episode$epi_date,1,4))>
                                yearnow)]<-as.character(yearnow)

  episode$epi_date<-as.Date(as.character(episode$epi_date))

  contr<-(as.numeric(substr(episode$epi_date,1,4))<fromyear)
  if (length(contr[contr==TRUE])>0) {
  episode$epi_date<-as.character(episode$epi_date)
  substr(episode$epi_date,1,4)[which(substr(episode$epi_date,1,4)>yearnow)]<-
  as.character(fromyear)
  }

  episode$epi_date<-as.Date(as.character(episode$epi_date))

  patient$dob<-as.character(patient$dob)
  patient$dob<-ifelse((as.numeric(substr(patient$dob,1,4))<1900),
  paste("19",substr(patient$dob,3,6),"-",substr(patient$dob,8,9)),patient$dob)

  patient$dob<-as.Date(as.character(patient$dob))

  episode<-merge(patient[,c("patient_id","dob")],episode,by="patient_id")

  episode$epi_date[as.numeric(episode$epi_date)<as.numeric(episode$dob)]<-NA

  episode<-episode[,-pmatch("dob",names(episode))]

}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_ce.r                                                      #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_ce.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#
################################################################################
#


################################################################################
#  BIRO_ce
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_ce is the main function of the Central Engine
#
# DEPENDENCIES
#
# BIRO_ce_setup.r
# BIRO_ce_datastep.r
#
################################################################################
#PARAMETERS:
#dbformat        = the format of the database ("postgres","csv")
#driverClass      !  = a string containing the class of the driver
#classPath        !  = a string containing the path of the driver
#identifier.quote !  = a string containing the identifier of quotes
#pathdb           !  = a string containing the path of the database
#user             !  = a string containing the username
#password         !  = a string containing the password
#dbname           !  = a string containing the name of the database
#centre_id        !  = a string containing the centre id
#yearnow          !  = a numeric value containing the last possible year of the
#                      episode dates
#fromyear         !  = a numeric value containing the first possible year where
#                      data are ok
#fromyearl        !  = a numeric value containing the first possible year where
#                      data are ok for the trends
#refyear          !  = a string containing year to calculate the indicators with
#                      "in the last 12 months" suffix
#refdate          !  = a string date to calculare age in the statistics about the
#                      refyear interval
#logfile   [""]      = a string containing the name of the .log file. If "" only
#                      screen print
#cex                 = A numerical value giving the amount by which
#                      plotting text and symbols should be magnified
#                      relative to the default. Note that some graphics
#                      functions such as plot.default have an argument of
#                      this name which multiplies this graphical
#                      parameter, and some functions such as points accept
#                      a vector of values which are recycled.
#                      Other uses will take just the first value if a
#                      vector of length greater than one is supplied.
################################################################################

BIRO_ce<-function(dirce,
                  dbformat, # "postgres" if data are stored in a psotgres database,
                            # "csv" if data are stored in csv files
                  dirout="",
                  driverClass,
                  classPath,
                  identifier.quote,
                  pathdb,
                  user,
                  password,
                  dbname,
                  dirdatastore,
                  centre_id,
                  disaggregation_by="",
                  standardization_by="",
                  dividedprev=TRUE,
                  divisornumber=NULL,
                  where,
```

```
                startdate="",
                enddate="",
                logfile="",
                cex,
                ID_Continent="",
                ID_Countries="",
                ID_Macroregion="",
                ID_Region="",
                ID_Province="",
                ID_District="",
                ID_Postcode="",
                report_list_id="") {


#
#

  engine<-"ce"
  assign("dirout",dirout,envir=.GlobalEnv)
  assign("report_list_id",report_list_id,envir=.GlobalEnv)
  assign("engine",engine,envir=.GlobalEnv)
  if (disaggregation_by=="centre_id") disaggregation_by="id"
  assign("disaggregation_by",disaggregation_by,envir=.GlobalEnv)
  assign("standardization_by",standardization_by,envir=.GlobalEnv)
  assign("dbname",dbname,envir=.GlobalEnv)
  assign("cevar",TRUE,envir=.GlobalEnv)
  assign("centre_id",centre_id,envir=.GlobalEnv)
  assign("cex",cex,envir=.GlobalEnv)
  assign("logfile",logfile,envir=.GlobalEnv)
  #Directory Store Input data (aggregated for in patient and episode datasets)
  assign("dirdatastore",dirdatastore,envir=.GlobalEnv)
  assign("oneyearbf",startdate,envir=.GlobalEnv)
  assign("startdate",startdate,envir=.GlobalEnv)
  assign("checkdate",enddate,envir=.GlobalEnv)
  assign("enddate",enddate,envir=.GlobalEnv)
  assign("dividedprev",dividedprev,envir=.GlobalEnv)
  assign("divisornumber",divisornumber,envir=.GlobalEnv)
  assign("where",where,envir=.GlobalEnv)

  assign("ID_Continent",ID_Continent,envir=.GlobalEnv)
  assign("ID_Countries",ID_Countries,envir=.GlobalEnv)
  assign("ID_Macroregion",ID_Macroregion,envir=.GlobalEnv)
  assign("ID_Region",ID_Region,envir=.GlobalEnv)
  assign("ID_Province",ID_Province,envir=.GlobalEnv)
  assign("ID_District",ID_District,envir=.GlobalEnv)
  assign("ID_Postcode",ID_Postcode,envir=.GlobalEnv)

  timestart<-Sys.time()

  write("processing file: BIRO_ce_setup.r",file="")

  source(paste(dirce,"/source/r/main/biro_ce_setup.r",sep=""))

  write("BIRO_ce_setup.r executed",file="")

  write("calling BIRO_cetenv",file="")

  BIRO_cetenv(driverClass=driverClass,
              classPath=classPath,
              identifier.quote=identifier.quote,
              pathdb=pathdb,
              user=user,
              password=password,
              dbname=dbname)

  write("BIRO_cetenv executed",file="")

  write("calling BIRO_dircreate",file="")

  BIRO_dircreate(dirce=dirce,
                 centre_id=centre_id)

  write("BIRO_dircreate executed",file="")

  logfile<-paste(dirreport,"/",logfile,sep="")
  assign("logfile",logfile,envir=.GlobalEnv)

  write("File .log created",file="")
  if (nchar(logfile)>0) {
   write(paste("Date: ",as.character(Sys.Date()),sep=""),file=logfile,append=FALSE)
   write(paste("Begin ",as.character(timestart),sep=""),file=logfile,append=TRUE)
  }

  write("processing file: BIRO_ce_recode.r",file="")
  if (nchar(logfile)>0) {
   write("processing file: BIRO_ce_recode.r",file=logfile,append=FALSE)
  }

  source(paste(dirce,"/source/r/formats/biro_ce_recode.r",sep=""))

  write("processing file: BIRO_ce_recode.r",file="")
  if (nchar(logfile)>0) {
   write("processing file: BIRO_ce_recode.r",file=logfile,append=FALSE)
  }


  drv <- JDBC(driverClass,classPath, identifier.quote)

   conn <- dbConnect(drv, pathdb,
                     user=user,
                     password=password,
                     dbname=dbname)

  assign("conn",conn,envir=.GlobalEnv)


  write("Connection to Central Database: Established",file="")
  if (nchar(logfile)>0) {
   write("Connection to Central Database: Established",file=logfile,append=FALSE)
  }

  #pop<-read.csv(paste(dirce,"/","pop.csv",sep=""))
  #assign("pop",pop,envir=.GlobalEnv)

  if (file.exists(paste(dirce,"/","mapcod.csv",sep=""))) {
   mapcod<-read.csv(paste(dirce,"/","mapcod.csv",sep=""))
   row.names(mapcod)<-as.numeric(row.names(mapcod))
   mapcod<-as.data.frame(mapcod)
   if (typeof(mapcod)=="list") {
    datamap<-as.vector(as.character(unlist(mapcod[[1]])))
    for (i in 2:dim(mapcod)[2]) {
     thisvector<-as.vector(as.character(unlist(mapcod[[i]])))
     datamap<-as.matrix(cbind(datamap,thisvector))
    }
```

```
    colnames(datamap)<-names(mapcod)
    mapcod<-as.data.frame(datamap)

   }
   mapcod$centre_id<-as.character(mapcod$centre_id)
   assign("mapcod",mapcod,envir=.GlobalEnv)
  }

#  write("Population/Mortality Data Loaded",file="")
#  if (nchar(logfile)>0) {
#   write("Population/Mortality Data Loaded",file=logfile,append=FALSE)
#  }


  open_tex(title="",
           authors="",
           footnote="",
           time="",
           logo="",
           layout=layout,
           final=texfile)

 # Second Title Page
 cat("\\begin{titlepage}",
     file=texfile,fill = TRUE,append=TRUE)


 cat("\\section*{\\huge Parameters used for the production\\\\of this Statistical
Report}",
     file=texfile,fill = TRUE,append=TRUE)

 cat("{\\bf Author}:  The BIRO Consortium,\\\\",
     file=texfile,fill = TRUE,append=TRUE)

 cat("{\\bf Date}: ",format(as.Date(launchtime,format="%d%m%y%H%M%S"),format="%d %B
%Y"), "\\\\",
     file=texfile,fill = TRUE,append=TRUE)

 cat("{\\bf Time}: ",format(strptime(launchtime, "%d%m%y%H%M%S"),format="%H:%M:%S"),
"\\\\",
     file=texfile,fill = TRUE,append=TRUE)

 cat("{\\bf EUBIROD source:}",centre_id,"\\\\",
     file=texfile,fill = TRUE,append=TRUE)
 cat(paste("{\\bf Time interval:}",format(as.Date(startdate,format="%d%m%y%H%M%S"
),format="%d %m %Y"),"-",
     format(as.Date(enddate,format="%d%m%y%H%M%S"),format="%d %m %Y")),"\\\\",
     file=texfile,fill = TRUE,append=TRUE)

 cat("{\\bf Output Directory:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\lstset{breaklines=true,basicstyle=\\footnotesize, basewidth=0.5em}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(dirreport,
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\end{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\framebox[3in][l]{\\bf INPUT DATA}\\\\[0.2 cm]",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\bf Database:}",dbname,"\\\\[0.4 cm]",
     file=texfile,fill = TRUE,append=TRUE)



#Waiting Stefano or Valentina for these parameters
#
merge_table=""
urlacttable=""
urlpoptable=""
urldiabtable=""
urldatastruc=""

if (merge_table!="") {
 cat("{\\bf Merge Table:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(merge_table,
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\end{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
}
if (urlacttable!="") {
 cat("{\\bf Activity  Dataset:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(urlacttable,
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\end{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
}
if (urlpoptable!="") {
 cat("{\\bf Population  Table:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(urlpoptable,
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\end{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
}

if (urldiabtable!="") {
 cat("{\\bf Diabetic Population Table:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(urldiabtable,
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\end{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
}

if (urldatastruc!="") {
 cat("{\\bf Data Source Structure:}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\begin{lstlisting}",
     file=texfile,fill = TRUE,append=TRUE)
 cat(urldatastruc,
     file=texfile,fill = TRUE,append=TRUE)
 cat("{\\end{lstlisting}",
```

```
    file=texfile,fill = TRUE,append=TRUE)
}

 cat("\\framebox[3in][l]{\\bf CONTENTS} \\\\[0.2 cm]",
    file=texfile,fill = TRUE,append=TRUE)


#  cat("Total No. Subjects: ", dim(patient)[1], "\\\\",
#      file=texfile,fill = TRUE,append=TRUE)
#  if ("sub_ds_id" %in% patient) {
#   cat("Sub Centres:  ", unique(patient$sub_ds_id), "\\\\",
#       file=texfile,fill = TRUE,append=TRUE)
#  }
#  cat("Total No. Episodes: ", dim(episode)[1], "\\\\",
#      file=paste(dirreport,"/",dbname,"_",thisyear,".tex",sep=""),fill =
TRUE,append=TRUE)
# }

cat("\\vfill",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\begin{center}",
    file=texfile,fill = TRUE,append=TRUE)
cat("{\\large Copyright the BIRO Consortium 2010}",
    file=texfile,fill = TRUE,append=TRUE)
cat("\\end{center}",
    file=texfile,fill = TRUE,append=TRUE)

 cat("\\end{titlepage}",
     file=texfile,fill = TRUE,append=TRUE)

 cat("\\vspace{8 cm}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\centerline{}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\centerline{}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\centerline{}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\vspace{8 cm}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\centerline{\\bf }",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\centerline{\\bf }",
     file=texfile,fill = TRUE,append=TRUE)

 cat("\\thispagestyle{empty}",
     file=texfile,fill = TRUE,append=TRUE)


 cat("\\pagestyle{plain}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\newpage",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\pagenumbering{roman}",
     file=texfile,fill = TRUE,append=TRUE)
 cat("\\tableofcontents",
     file=texfile,fill = TRUE,append=TRUE)

  write(paste("All Outputs in [OUTDIR]:",dirreport),file="")
  if (nchar(logfile)>0) {
   write(paste("All Outputs in [OUTDIR]:",dirreport),file=logfile,append=TRUE)
  }
  write(paste("Statistical Objects saved into [OUTDATA]:",dirdataout),file="")
  if (nchar(logfile)>0) {
   write(paste("Statistical Objects saved into [OUTDATA]:"
,dirdataout),file=logfile,append=TRUE)
  }

  ################################
  #                              #
  #        Create Report         #
  #                              #
  ################################

  #Demographic characteristics
  new_chapter(title="Demographic characteristics",texfile=texfile)

  cat("\\setcounter{page}{1}",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\pagenumbering{arabic}",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\pagestyle{plain}",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\mbox{}",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\fancyhf{}",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\renewcommand{\\headrulewidth}{0.5pt}",
     file=texfile,fill = TRUE,append=TRUE)

  cat("\\newpage",
     file=texfile,fill = TRUE,append=TRUE)
  cat("\\pagestyle{fancy}",
     file=texfile,fill = TRUE,append=TRUE)

  source(paste(dirce,"/source/r/scripts/","biro_ce_indicator_demographic.r",
         sep=""))

  #Clinical characteristics

  new_chapter(title="Clinical characteristics",texfile=texfile)

  source(paste(dirce,"/source/r/scripts/","biro_ce_indicator_clinical.r",
         sep=""))

  #Health System

  new_chapter(title="Health System",texfile=texfile)

  source(paste(dirce,"/source/r/scripts/","biro_ce_indicator_health_system.r",
         sep=""))

  #Population
  new_chapter(title="Population",texfile=texfile)

  source(paste(dirce,"/source/r/scripts/","biro_ce_indicator_population.r",
         sep=""))


  #Risk Adjusted
  new_chapter(title="Risk Adjusted Indicators",texfile=texfile)

  source(paste(dirce,"/source/r/scripts/","biro_ce_indicator_risk_adjusted.r",
         sep=""))
```

```
#Appendix
 cat("\\chapter{Appendix}",
    file=texfile,fill = TRUE,append=TRUE)

 cat("\\include{appendix}",
    file=texfile,fill = TRUE,append=TRUE)


 close_tex(file=texfile)
 pathtex=substr(texfile,1,nchar(texfile)-4)

 texfile<-paste("\"",texfile,"\"",sep="")
 print(texfile)
 command<-paste("pdflatex ",texfile,sep="")
 print(command)

 write("Preparing to compile .Tex File",file="")
 if (nchar(logfile)>0) {
  write("Preparing to compile .Tex File",file=logfile,append=TRUE)
 }

 setwd(dirreport)

 x<-system(command,intern=TRUE)
 x<-system(command,intern=TRUE)
 x<-system(command,intern=TRUE)

 print(pathtex)
  file.rename(from=paste(pathtex,".aux",sep=""), to=paste(dirreport,"/pdf/",dbname,
".aux",sep=""))
  file.rename(from=paste(pathtex,".out",sep=""), to=paste(dirreport,"/pdf/",dbname,
".out",sep=""))
  file.rename(from=paste(pathtex,".log",sep=""), to=paste(dirreport,"/pdf/",dbname,
".log",sep=""))
  file.rename(from=paste(pathtex,".toc",sep=""), to=paste(dirreport,"/pdf/",dbname,
".toc",sep=""))
  file.rename(from=paste(dirreport,"/appendix.tex",sep=""), to=paste(dirreport,
"/pdf/appendix.tex",sep=""))
  file.rename(from=paste(dirreport,"/appendix.aux",sep=""), to=paste(dirreport,
"/pdf/appendix.aux",sep=""))
  file.rename(from=paste(dirreport,"/",dbname,".tex",sep=""), to=paste(dirreport,
"/pdf/",dbname,".tex",sep=""))

cat("<a href=\" html/appendix.html\"> Appendix </a><br>",sep="",
    file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)



 write("Tex File Compiled",file="")
 if (nchar(logfile)>0) {
  write("Tex File Compiled",file=logfile,append=TRUE)
 }


 unlink(paste(dirreport,"/html/temp",sep=""),recursive=TRUE)
 unlink(paste(dirreport,"/pdf/temp",sep=""),recursive=TRUE)
 timefinish<-Sys.time()

 write(paste("Elapsed time:",timefinish-timestart, "mins",sep=" "),file="")
 if (nchar(logfile)>0) {
  write(paste("Elapsed time:",timefinish-timestart, "mins",sep=" "),
        file=logfile,append=FALSE)
 }
 file.remove(logfile)
 file.copy(from=paste(dirout,"/centralEngineSinkFile.txt",sep=""),
to=paste(dirreport,"/",dbname,".log",sep=""))


}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_ce_setup.r                                                         #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_ce_setup.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
# BIRO_cetenv
# BIRO_dircreate
#
################################################################################

################################################################################
#  BIRO_cetenv
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_cetenv sets the environment of the BIRO Central Engine
#
# DEPENDENCIES
#
# R package: RJDBC
#
################################################################################
#PARAMETERS:
#driverClass      !  = a string containing the class of the driver
#classPath        !  = a string containing the path of the driver
#identifier.quote !  = a string containing the identifier of quotes
#pathdb           !  = a string containing the path of the database
#user             !  = a string containing the username
#password         !  = a string containing the password
#dbname           !  = a string containing the name of the database
################################################################################


BIRO_cetenv<-function(driverClass,
                      classPath,
                      identifier.quote,
                      pathdb,
                      user,
                      password,
                      dbname) {



  library(rJava)
  library(RJDBC)
  library(lattice)
  library(rmeta)
  library(maptools)
  library(Hmisc)
  library(R2HTML)
  library(Cairo)


    assign("drv",JDBC(driverClass ="org.postgresql.Driver",classPath=classPath,
    identifier.quote=identifier.quote),envir=.GlobalEnv)


}

################################################################################
#  BIRO_dircreate
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
```

```
# BIRO_dircreate creates directories according the BIRO directories schema
#
# DEPENDENCIES
#
# r package: base
#
################################################################################
#PARAMETERS:
#dirce      !        = a string containing the path of the Central Engine
#                     directory
#centre_id  !        = a string containing the id of the centre
################################################################################


BIRO_dircreate<-function(dirce,
                         centre_id) {

  dir.create(dirce, showWarnings = FALSE, recursive = TRUE)

  #Directory Reports

  assign("dirce",dirce,envir=.GlobalEnv)

  assign("dirsourceR",paste(dirce,"/source/r",sep=""),envir=.GlobalEnv)

  #Load ".../lib/r/source/biro" functions

  assign("dirlibR",paste(substr(dirce,1,nchar(dirce)-4),
         "lib/r/source/biro",sep=""),envir=.GlobalEnv)

  assign("dirsourcelatex",paste(substr(dirce,1,nchar(dirce)-4),"lib/templates/pdf"
,sep=""),envir=.GlobalEnv)

  for (i in 1:length(list.files(dirlibR))) {
   source(paste(dirlibR,"/",list.files(dirlibR)[i],sep=""))
  }

  assign("dirinclude",paste(dirsourceR,"/include",sep=""),envir=.GlobalEnv)

  for (i in 1:length(list.files(dirinclude))) {
   source(paste(dirinclude,"/",list.files(dirinclude)[i],sep=""))
  }

  dir.create(dirsourceR, showWarnings = FALSE, recursive = TRUE)
  dir.create(dirsourcelatex, showWarnings = FALSE, recursive = TRUE)

  launchtime<-format(Sys.time(),"%d%m%y%H%M%S")
  assign("launchtime",launchtime,envir=.GlobalEnv)

  #Local Repository of DB data in csv format

  dirstoreout<-paste(dirout,"/data/", "#",launchtime,"/",centre_id,sep="")
  dir.create(dirstoreout, showWarnings = FALSE, recursive = TRUE)

  #Directory Local Components

  assign("dirdataout",paste(dirout,"/output/data/", "#",launchtime,"/",centre_id,sep=
""),envir=.GlobalEnv)
  dir.create(dirdataout, showWarnings = FALSE, recursive = TRUE)

  #Directory Reports
  assign("dirreport",paste(dirout,"/output/reports/", "#",launchtime,"/"
,centre_id,sep=""),envir=.GlobalEnv)
  dir.create(dirreport, showWarnings = FALSE, recursive = TRUE)

  #Check and configure

  assign("dirstoreout",dirstoreout,envir=.GlobalEnv)
  assign("dirgraphlatex","graphs",envir=.GlobalEnv)
  assign("dirgraph",paste(dirreport,"/",dirgraphlatex,sep=""),envir=.GlobalEnv)
  assign("dirhtml",paste(dirreport,"/html",sep=""),envir=.GlobalEnv)
  assign("dirwp",paste(dirreport,"/wp",sep=""),envir=.GlobalEnv)
  assign("diraux",paste(dirreport,"/pdf",sep=""),envir=.GlobalEnv)
  assign("dirtables",paste(dirreport,"/tables",sep=""),envir=.GlobalEnv)
  assign("dircsv",paste(dirout,"/output/data/",sep=""),envir=.GlobalEnv)
  dirimages<-paste(dirout,"/output/reports/", "#",launchtime,
                   "/",centre_id,"/images",sep="")
  dir.create(dirimages, showWarnings = FALSE, recursive = TRUE)
  file.copy(paste(substr(dirce,1,nchar(dirce)-4),"lib/templates/html/",
"biro-logo01.jpg",sep=""),
            paste(dirimages,"/","biro-logo01.jpg",sep=""))

#file.copy(paste(substr(dirce,1,nchar(dirce)-4),"lib/templates/pdf/","EUBIROD.eps",sep
="",
  #          paste(dirimages,"/","EUBIROD.eps",sep=""))
  file.copy(paste(substr(dirce,1,nchar(dirce)-4),"lib/templates/pdf/","EUBIROD.pdf"
,sep=""),
            paste(dirimages,"/","EUBIROD.pdf",sep=""))

  dir.create(dirgraph, showWarnings = FALSE, recursive = TRUE)
  dir.create(dirhtml, showWarnings = FALSE, recursive = TRUE)
  dir.create(dirtables, showWarnings = FALSE, recursive = TRUE)
  dir.create(diraux, showWarnings = FALSE, recursive = TRUE)
  dir.create(dirwp, showWarnings = FALSE, recursive = TRUE)

#  if (dirdatastore!="") {
#   dir.create(dirdatastore, showWarnings = FALSE, recursive = TRUE)
#  }

  write("BIRO directories created",file="")
  if (nchar(logfile)>0) {
   write("BIRO directories created",file=logfile,append=FALSE)
  }

  assign("layout",paste(dirsourcelatex,"/layout.tex",sep=""),envir=.GlobalEnv)
  assign("texfile",paste(dirreport,"/",dbname,".tex",sep=""),envir=.GlobalEnv)
  assign("texappendix",paste(dirreport,"/appendix.tex",sep=""),envir=.GlobalEnv)
  assign("htmlappendix",paste(dirreport,"/html/appendix.html",sep=""
),envir=.GlobalEnv)

  assign("dvifile",paste(dirreport,"/",dbname,".dvi",sep=""),envir=.GlobalEnv)
  assign("pdffile",paste(dirreport,"/",dbname,".pdf",sep=""),envir=.GlobalEnv)

  assign("dirmap",paste(substr(dirce,1,nchar(dirce)-4),
         "lib/maps",sep=""),envir=.GlobalEnv)
  assign("maploaded","",envir=.GlobalEnv)

  assign("rephtmlfile",paste(dirreport,"/",dbname,".html",sep=""),envir=.GlobalEnv)

  file.copy(paste(substr(dirce,1,nchar(dirce)-4),"lib/templates/html/",
"layout_open.html",sep=""),
            paste(dirreport,"/",dbname,".html",sep=""))

 cat("<body>",
```

```
    file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat("<table class=\"reportheader\" border=\"0\">",
    file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat("<tbody>",
    file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <tr>",
    file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat("  <td><img src=\"images/biro-logo01.jpg\"></td>",
    file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" </tr>",
    file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat("</tbody>",
    file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat("</table>",
    file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat("<table class=\"reportheader\" border=\"0\">",file=paste(dirreport,"/",dbname,
".html",sep=""),fill = TRUE,append=TRUE)
 cat("<tbody>",file=paste(dirreport,"/",dbname,".html",sep=""),fill =
TRUE,append=TRUE)
# cat(" <tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
# cat(" <td class=\"reportheader\"> Date: ",
#     format(as.Date(launchtime,format="%d%m%y%H%M%S"),format="%d %B %Y"),
#     "</td>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
# cat(" </tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <td class=\"reportheader\"> EUBIROD source: ",centre_id,
    "</td>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
# cat(" <tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
# cat(" <td class=\"reportheader\"> Reference Year: ", thisyear,
#     "</td>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
# cat(" </tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill =
TRUE,append=TRUE)
# cat(" <tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
# cat(" <td class=\"reportheader\"> Time interval: ", intervaldd,
#     "</td>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
# cat(" </tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <td class=\"reportheader\"> Input Database: ", dbname,
    "</td>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <td class=\"reportheader\"> Output Directory Report: ",dirreport,
    "</td>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)

 cat(" </tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 #cat(" <tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 #cat("  <td class=\"reportheader\">Timeframe:
",as.character(format(startdate,format="%d %B
%Y")),"-",as.character(format(enddate,format="%d %B %Y")),"</td>",
 #    file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)
 #cat(" </tr>",file=paste(dirreport,"/",dbname,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" </tbody>",file=paste(dirreport,"/",dbname,".html",sep=""),fill =
TRUE,append=TRUE)
 cat(" </table>",file=paste(dirreport,"/",dbname,".html",sep=""),fill =
TRUE,append=TRUE)

 assign("htmlfile",paste(dirreport,"/",dbname,".html",sep=""),envir=.GlobalEnv)
}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_ce_indicator_clinical.r                                  #
#                                                                              #
#
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-24
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_ce_indicator_clinical.r is part of WP Central Engine of the BIRO
# Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
################################################################################

   new_section(title="2.1 Diabetes Status",texfile=texfile)
   new_section_html(title="2.1 Diabetes Status",htmlfile=htmlfile)

    BIRO_ce_report(ind="2_1_1",
                   numclass=2,
                   condition="",
                   width=1,
                   stratum=c("type_dm","age_c"),
                   lev=list(a=as.vector(names(levtype_dm)),b=classlabel(ageth)),
                   tab=1,
                   namevar=c("Type","Age"),
                   tabvarsum=c("n"),
                   n=c(3,5),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=0,
                   trellis=0,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="2.1.1 Type of diabetes")


    BIRO_ce_report(ind="2_1_2",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("durdiab_c","sex","hba1c_done","type_dm"),
                   lev=list(a=classlabel(durdiabth),b=names(levsex),c=c("with hba1c",
"without hba1c"),d=as.vector(names(levtype_dm))),
                   tab=1,
                   namevar=c("Duration of Diabetes","Gender","HbA1c Done","Type of
Diabetes"),
                   tabvarsum=c("n"),
                   n=c(3,2,2,3),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=0,
                   trellis=0,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="2.1.2. Duration of diabetes (Classes)")

   new_section(title="2.2 Risk Factors",texfile=texfile)
   new_section_html(title="2.2 Risk Factors",htmlfile=htmlfile)

    BIRO_ce_report(ind="2_2_1_1",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("weight_c","sex","age_c","type_dm"),

lev=list(a=c(classlabel(weightth)),b=names(levsex),c=classlabel(ageth),d=as.vector(nam
es(levtype_dm))),
                   tab=1,
                   namevar=c("Weight","Gender","Age","Type of Diabetes"),
                   tabvarsum=c("n"),
                   n=c(6,2,5,3),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=1,
                   trellis=1,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="2.2.1.1. Weight (last episode in 12 months)")
```

```
    BIRO_ce_report(ind="2_2_1_2",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("bmi_c","sex","age_c","type_dm"),

lev=list(a=c(classlabel(bmith)),b=names(levsex),c=classlabel(ageth),d=as.vector(names(
levtype_dm))),
                   tab=1,
                   namevar=c("BMI","Gender","Age","Type of Diabetes"),
                   tabvarsum=c("n"),
                   n=c(5,2,5,3),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=1,
                   trellis=1,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="2.2.1.2. BMI (last episode in 12 months)")

    new_section(title="2.2.2 Lifestyle",texfile=texfile)

    BIRO_ce_report(ind="2_2_2_1",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("smok_stat","sex","age_c","type_dm"),

lev=list(a=c(names(levsmok_stat)),b=names(levsex),c=classlabel(ageth),d=as.vector(name
s(levtype_dm))),
                   tab=1,
                   namevar=c("Smoke","Gender","Age","Type of Diabetes"),
                   tabvarsum=c("n"),
                   n=c(3,2,5,3),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=0,
                   trellis=0,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="2.2.2.1 Smoking status (last episode in 12 months)")

     new_section(title="2.2.3. Clinical measurements",texfile=texfile)

    BIRO_ce_report(ind="2_2_3_1",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("sbp_c","sex","age_c","type_dm"),

lev=list(a=c(classlabel(sbpth)),b=names(levsex),c=classlabel(ageth),d=as.vector(names(
levtype_dm))),
                   tab=1,
                   namevar=c("SBP","Gender","Age","Type of Diabetes"),
                   tabvarsum=c("n"),
                   n=c(3,2,5,3),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=1,
                   trellis=0,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="2.2.3.1. Systolic BP (last episode in 12 months)")


    BIRO_ce_report(ind="2_2_3_2",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("dbp_c","sex","age_c","type_dm"),

lev=list(a=c(classlabel(dbpth)),b=names(levsex),c=classlabel(ageth),d=as.vector(names(
levtype_dm))),
                   tab=1,
                   namevar=c("DBP","Gender","Age","Type of Diabetes"),
                   tabvarsum=c("n"),
                   n=c(4,2,5,3),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=1,
                   trellis=0,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="2.2.3.2. Diastolic BP (last episode in 12 months)")


    BIRO_ce_report(ind="2_2_3_3",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("chol_c","sex","age_c","type_dm"),
                   lev=list(a=c("0 - 2.58","2.59 - 5.17","5.18 - 7.76","7.77 + "
),b=names(levsex),c=classlabel(ageth),d=as.vector(names(levtype_dm))),
                   tab=1,
                   namevar=c("Cholesterol","Gender","Age","Type of Diabetes"),
                   tabvarsum=c("n"),
                   n=c(4,2,5,3),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=1,
```

```
                    trellis=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="2.2.3.3 Total cholesterol (last episode in 12 months)")


    BIRO_ce_report(ind="2_2_3_4",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("hdl_c","sex","age_c","type_dm"),
                    lev=list(a=c("0 - 1.294","1.295 +"
),b=names(levsex),c=classlabel(ageth),d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("Cholesterol","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=1,
                    trellis=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="2.2.3.4. HDL-cholesterol (last episode in 12 months)")


    BIRO_ce_report(ind="2_2_3_5",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("creat_c","sex","age_c","type_dm"),

lev=list(a=c(classlabel(creatth)),b=names(levsex),c=classlabel(ageth),d=as.vector(name
s(levtype_dm))),
                    tab=1,
                    namevar=c("Cratinine","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(4,2,5,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=1,
                    trellis=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="2.2.3.5 Creatinine (last episode in 12 months)")


    BIRO_ce_report(ind="2_2_3_6",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("hba1c_c","sex","age_c","type_dm"),
                    lev=list(a=c("0 - 5.9","6 - 7.9","8 +"
),b=names(levsex),c=classlabel(ageth),d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("Hba1c","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(3,2,5,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=1,
                    trellis=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="2.2.3.6. HbA1c (last episode in 12 months)")

  new_section(title="2.3. Diabetes complications",texfile=texfile)
  new_section_html(title="2.3. Diabetes complications",htmlfile=htmlfile)

   BIRO_ce_report(ind="2_3_1",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("retina","durdiab_c"),
                    lev=list(a=c(names(levretin)),b=classlabel(durdiabth)),
                    tab=1,
                    namevar=c("Rethinopathy","Duration"),
                    tabvarsum=c("n"),
                    n=c(3,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="2.3.1. Retinopathy (first episode in 12 months)")


    BIRO_ce_report(ind="2_3_2",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("esrf","durdiab_c"),
                    lev=list(a=c(names(levesrf)),b=classlabel(durdiabth)),
                    tab=1,
                    namevar=c("ESRF","Duration"),
                    tabvarsum=c("n"),
                    n=c(2,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
```

```
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="2.3.2. End stage renal failure (first episode in 12 months)"
)


    BIRO_ce_report(ind="2_3_3",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("ulcer","durdiab_c"),
                    lev=list(a=c(names(levulcer)),b=classlabel(durdiabth)),
                    tab=1,
                    namevar=c("Foot Ulcer","Duration"),
                    tabvarsum=c("n"),
                    n=c(2,3),
                    bar=0,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="2.3.3. Foot ulcer (first episode in 12 months)")


    BIRO_ce_report(ind="2_3_4",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("amput","durdiab_c"),
                    lev=list(a=c(names(levamput)),b=classlabel(durdiabth)),
                    tab=1,
                    namevar=c("Amputation","Duration"),
                    tabvarsum=c("n"),
                    n=c(2,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="2.3.4. Amputation (first episode in 12 months)")


    BIRO_ce_report(ind="2_3_5",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("stroke","durdiab_c"),
                    lev=list(a=c(names(levstroke)),b=classlabel(durdiabth)),
                    tab=1,
                    namevar=c("Stroke","Duration"),
                    tabvarsum=c("n"),
                    n=c(2,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="2.3.5. Stroke (first episode in 12 months)")


    BIRO_ce_report(ind="2_3_6",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("mi","durdiab_c"),
                    lev=list(a=c(names(levmi)),b=classlabel(durdiabth)),
                    tab=1,
                    namevar=c("Miocardial Infarction","Duration"),
                    tabvarsum=c("n"),
                    n=c(2,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="2.3.6. Myocardal infarction (first episode in 12 months)")


    BIRO_ce_report(ind="2_3_7",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("hypertension","durdiab_c"),
                    lev=list(a=c(names(levhypertension)),b=classlabel(durdiabth)),
                    tab=1,
                    namevar=c("Hypertension","Duration"),
                    tabvarsum=c("n"),
                    n=c(2,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="2.3.7. Hypertension (first episode in 12 months)")
```

```
##############################################################################
#                                                                            #
# Project: BIRO-Project (Funded by European Commission 2005-2008)            #
# File: BIRO_ce_indicator_demographic.r                                      #
#                                                                            #
#
##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-24
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_ce_indicator_clinical.r is part of WP Central Engine of the BIRO
# Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
##############################################################################



  #Age of patient (all the patient in the db)

   new_section(title="1.1 Basic demographics",texfile=texfile)
   new_section_html(title="1.1 Basic demographics",htmlfile=htmlfile)


    BIRO_ce_report(ind="1_1_1",
                   numclass=2,
                   condition="",
                   width=1,
                   stratum=c("age_c","sex"),
                   lev=list(a=classlabel(ageth),b=names(levsex)),
                   tab=1,
                   namevar=c("Age","Gender"),
                   tabvarsum=c("n"),
                   n=c(5,2),
                   chisq=NULL,
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=0,
                   texfile=texfile,
                   cex=cex,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   title="1.1 Age (Classes)")
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_ce_indicator_health_system.r                                      #
#                                                                              #
#
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-24
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_ce_indicator_health_system.r is part of WP Central Engine of the
# BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
################################################################################

new_section(title="3.1. Structure (provider level)",texfile=texfile)
new_section_html(title="3.1. Structure (provider level)",htmlfile=rephtmlfile)

# 3.1.1 Type of Provider


    BIRO_ce_report(ind="3_1_1",
                   numclass=0,
                   condition="",
                   tab=0,
                   bar=0,
                   box=0,
                   lines=0,
                   onlyprint=1,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="3.1.1 Type of Provider")


# 3.1.2 Average diabetes population per center


    BIRO_ce_report(ind="3_1_2",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("age_c","sex",NA,"type_dm"),
                   lev=list(a=classlabel(ageth10),b=names(levsex),c=""
,d=as.vector(names(levtype_dm))),
                   tab=1,
                   namevar=c("Age Classes","Gender",NA,"Type of Diabetes"),
                   tabvarsum=c("n"),
                   n=c(5,2,NA,3),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=0,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="3.1.2 Average diabetes population")

new_section(title="3.2. Structural quality",texfile=texfile)
new_section_html(title="3.2. Structural quality",htmlfile=rephtmlfile)

# 3.2.1 Hospital beds per 100,000 population


    BIRO_ce_report(ind="3_2_1",
                   numclass=0,
                   condition="",
                   tab=0,
                   bar=0,
                   box=0,
                   lines=0,
                   onlyprint=1,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="3.2.1 Hospital beds per 100,000 population")


# 3.2.2 Physicians employed per 100,000 population


    BIRO_ce_report(ind="3_2_2",
                   numclass=0,
                   condition="",
                   tab=0,
                   bar=0,
                   box=0,
                   lines=0,
                   onlyprint=1,
```

```
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.2.2 Physicians employed per 100,000 population")

new_section_html(title="3.3. Processes (individual level)",htmlfile=rephtmlfile)
new_section(title="3.3. Processes (individual level)",texfile=texfile)

new_sub_section_html(title="3.3.1. Foot examination",htmlfile=rephtmlfile)
new_sub_section(title="3.3.1. Foot examination",texfile=texfile)


# 3.3.1.1 Foot examination


    BIRO_ce_report(ind="3_3_1_1",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("foot_exam","age_c",NA,"type_dm"),
                    lev=list(a=c(names(levfoot_exam),"missing"),b=classlabel(ageth),c=
"",d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("Foot Exam","Age Classes",NA,"Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,5,NA,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.1.1 Done (last episode in 12 months)")

   new_sub_section_html(title="3.3.2. Eye examination",htmlfile=rephtmlfile)
   new_sub_section(title="3.3.2. Eye examination",texfile=texfile)


    BIRO_ce_report(ind="3_3_2_1",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("eye_exam","age_c",NA,"type_dm"),
                    lev=list(a=c(names(leveye_exam),"missing"),b=classlabel(ageth),c=""
,d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("Eye Exam","Age Classes",NA,"Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,5,NA,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.2.1 Done (last episode in 12 months)")
  # 3.3.3.1 BP

   new_sub_section_html(title="3.3.3. Measurements done",htmlfile=rephtmlfile)
   new_sub_section(title="3.3.3 Measurements examination",texfile=texfile)


       BIRO_ce_report(ind="3_3_3_1",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("bp","age_c",NA,"type_dm"),
                    lev=list(a=c(names(levbp),"missing"
),b=classlabel(ageth),c=NA,d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("BP Exam","Age Classes",NA,"Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,5,NA,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.3.1 BP (last episode in 12 months)")


  #3.3.3.2 Lipids


    BIRO_ce_report(ind="3_3_3_2",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("lipids","age_c",NA,"type_dm"),
                    lev=list(a=c(names(levlipids),"missing"),b=classlabel(ageth),c=""
,d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("Lipids","Age Classes",NA,"Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,5,NA,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.3.2 Lipids")


  #3.3.3.3 MA_test
```

```
    BIRO_ce_report(ind="3_3_3_3",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("ma_test","age_c",NA,"type_dm"),
                   lev=list(a=c(names(levma_test),"missing"),b=classlabel(ageth),c=""
,d=as.vector(names(levtype_dm))),
                   tab=1,
                   namevar=c("MA test","Age Classes",NA,"Type of Diabetes"),
                   tabvarsum=c("n"),
                   n=c(2,5,NA,3),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=0,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="3.3.3.3 Microalbumin (last episode in 12 months)")

 #
 # 3.3.3.4 Hba1c
 #


    BIRO_ce_report(ind="3_3_3_4",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("hba1c_done","age_c",NA,"type_dm"),
                   lev=list(a=c(names(levhba1c_done),"missing"),b=classlabel(ageth),c=
"",d=as.vector(names(levtype_dm))),
                   tab=1,
                   namevar=c("Hba1c done","Age Classes",NA,"Type of Diabetes"),
                   tabvarsum=c("n"),
                   n=c(2,5,NA,3),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=0,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="3.3.3.4 HbA1c (last episode in 12 months)")


 ##Treatments


 #
 # 3.3.4.1 Antihypertensive Medication
 #

    new_sub_section_html(title="3.3.4 Treatment",htmlfile=rephtmlfile)
    new_sub_section(title="3.3.4 Treatment",texfile=texfile)

    new_sub_section(title="3.3.4.1 Antihypertensive Medication (last episode in 12
months)",texfile=texfile)
    new_sub_section_html(title="3.3.4.1 Antihypertensive Medication (last episode in
12 months)",htmlfile=rephtmlfile)

#    BIRO_ce_report(ind="3_3_4_1",
#                   numclass=4,
#                   condition="",
#                   width=1,
#                   stratum=c("hypert_med","age_c",NA,"type_dm"),
#
lev=list(a=c(names(levhypert_med),"missing"),b=classlabel(ageth),c="",d=as.vector(name
s(levtype_dm))),
#                   tab=1,
#                   namevar=c("Antihypertensive Med","Age Classes",NA,"Type of
Diabetes"),
#                   tabvarsum=c("n"),
#                   n=c(2,5,NA,3),
#                   bar=1,
#                   barvarsum=c("n","sum"),
#                   beside=TRUE,
#                   perc=TRUE,
#                   lines=0,
#                   box=0,
#                   texfile=texfile,
#                   dirgraph=dirgraph,
#                   dirgraphlatex=dirgraphlatex,
#                   cex=cex,
#                   title="3.3.4.1 Antihypertensive Medication (last episode in 12
months)")


 #
 # 3.3.4.2 Lipid Lowering Medication
 #

    new_sub_section(title="3.3.4.2 Lipid Lowering Medication (last episode in 12
months)",texfile=texfile)
    new_sub_section_html(title="3.3.4.2 Lipid Lowering Medication (last episode in 12
months)",htmlfile=rephtmlfile)

    BIRO_ce_report(ind="3_3_4_2",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("lipid_med","age_c",NA,"type_dm"),
                   lev=list(a=c(names(levlipid_med),"missing"),b=classlabel(ageth),c=
"",d=as.vector(names(levtype_dm))),
                   tab=1,
                   namevar=c("Lipid Lowering Med","Age Classes",NA,"Type of Diabetes"
),
                   tabvarsum=c("n"),
                   n=c(2,5,NA,3),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=0,
                   texfile=texfile,
                   dirgraph=dirgraph,
```

```
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.4.2 Lipid Lowering Medication (last episode in 12
months)")


    new_sub_section(title="3.3.4.3 ASA Medication (last episode in 12 months)"
,texfile=texfile)
    new_sub_section_html(title="3.3.4.3 ASA Medication (last episode in 12 months)"
,htmlfile=rephtmlfile)

    BIRO_ce_report(ind="3_3_4_3",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("asa_med","age_c",NA,"type_dm"),
                    lev=list(a=c(names(levasa_med),"missing"),b=classlabel(ageth),c=""
,d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("Lipid Treatment","Age Classes",NA,"Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,5,NA,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.4.3 ASA Medication (last episode in 12 months)")


  #
  # 3.3.4.4 Drug Therapy
  #


    BIRO_ce_report(ind="3_3_4_4_1",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("diet","age_c",NA,"type_dm"),
                    lev=list(a=c(names(levdiet),"missing"),b=classlabel(ageth),c=""
,d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("Diet only","Age Classes",NA,"Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,5,NA,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.4.4.1 Glucose Lowering: Diet Only (last episode in 12
months)")



    BIRO_ce_report(ind="3_3_4_4_2",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("tab","age_c",NA,"type_dm"),
                    lev=list(a=c(names(levtablet),"missing"),b=classlabel(ageth),c=""
,d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("Tablets Only","Age Classes",NA,"Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,5,NA,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.4.4.2 Glucose Lowering: Tablets Only (last episode in
12 months)")


    BIRO_ce_report(ind="3_3_4_4_3",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("ins","age_c",NA,"type_dm"),
                    lev=list(a=c(names(levinsulin),"missing"),b=classlabel(ageth),c=""
,d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("Insulin Only","Age Classes",NA,"Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,5,NA,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.4.4.3 Glucose Lowering: Insulin Only (last episode in
12 months)")


    BIRO_ce_report(ind="3_3_4_4_4",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("instab","age_c",NA,"type_dm"),
                    lev=list(a=c(names(levinstab),"missing"),b=classlabel(ageth),c=""
,d=as.vector(names(levtype_dm))),
                    tab=1,
```

```
                    namevar=c("Instab","Age Classes",NA,"Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,5,NA,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.4.4.4 Glucose Lowering: Insulin and Tablets (last
episode in 12 months)")


    BIRO_ce_report(ind="3_3_4_4_5",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("pump_therapy","age_c",NA,"type_dm"),
                    lev=list(a=c(names(levpump_med),"missing"),b=classlabel(ageth),c=""
,d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("Pump Therapy","Age Classes",NA,"Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,5,NA,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.4.4.5 Glucose Lowering: Insulin Pump (last episode in
12 months)")


 new_section_html(title="3.3.5 Management",htmlfile=rephtmlfile)
 new_section(title="3.3.5 Management",texfile=texfile)


#    BIRO_ce_report(ind="3_3_5_1",
#                    numclass=4,
#                    condition="",
#                    width=1,
#                    stratum=c("self_mon","age_c",NA,"type_dm"),
#
lev=list(a=c(names(levself_mon),"missing"),b=classlabel(ageth),c="",d=as.vector(names(
levtype_dm))),
#                    tab=1,
#                    namevar=c("Self Monitoring","Age Classes",NA,"Type of Diabetes"),
#                    tabvarsum=c("n"),
#                    n=c(3,5,NA,3),
#                    bar=1,
#                    barvarsum=c("n","sum"),
#                    beside=TRUE,
#                    perc=TRUE,
#                    lines=0,
#                    box=0,
#                    texfile=texfile,
#                    dirgraph=dirgraph,
#                    dirgraphlatex=dirgraphlatex,
#                    cex=cex,
#                    title="3.3.5.1 Self monitoring")



 # 3.3.5.2  Visit Frequency

    BIRO_ce_report(ind="3_3_5_2",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("visit_freq","age_c",NA,"type_dm"),
                    lev=list(a=names(levvisit_freq),b=classlabel(ageth),c=""
,d=as.vector(names(levtype_dm))),
                    tab=1,
                    namevar=c("Visit Frequency","Age Classes",NA,"Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(3,5,NA,3),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="3.3.5.2 Visit Frequency")
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_ce_indicator_population.r                                         #
#                                                                              #
#
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-24
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_ce_indicator_population.r is part of WP Central Engine of the BIRO
# Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
################################################################################



# 4.1.1. Total population

   new_section(title="4.1 Vital Statistics",texfile=texfile)
   new_section_html(title="4.1 Vital Statistics",htmlfile=htmlfile)

    BIRO_ce_report(ind="4_1_1",
                   numclass=0,
                   condition="",
                   tab=0,
                   bar=0,
                   box=0,
                   lines=0,
                   onlyprint=1,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="4.1.1. Total population")

# 4.1.2. Life expectancy

    BIRO_ce_report(ind="4_1_2",
                   numclass=0,
                   condition="",
                   tab=0,
                   bar=0,
                   box=0,
                   lines=0,
                   onlyprint=1,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="4.1.2. Life expectancy")


# 4.1.3. Mortality data

    BIRO_ce_report(ind="4_1_3",
                   numclass=0,
                   condition="",
                   tab=0,
                   bar=0,
                   box=0,
                   lines=0,
                   onlyprint=1,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   title="4.1.3. Mortality data")
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_ce_indicator_risk_adjusted.r                                      #
#                                                                              #
#
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-24
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_ce_indicator_risk_adjusted.r is part of WP Central Engine of the
# BIRO Project
#
# GPL Copyright, The BIRO Project
#
# ------------------------
#
################################################################################


 # 5.1.1 Prevalence of diabetes mellitus per 1,000


#  pop$pop=pop$popM+pop$popF
#  poptype_dm<-ifelse(pop$age_band<=7,1,2)
#  poptype<-aggregate(pop,by=list(type_dm=poptype_dm),FUN=sum)
#  rm(poptype_dm)
#  poptype<-poptype[,c("type_dm","pop")]
#  poptype<-rbind(poptype,poptype)
#  poptype$centre_id=c(rep("mycentre2",2),rep("mycentre3",2))


#  BIRO_dframe(dataname="poptype",monitor=TRUE,filelog=logfile)
#  BIRO_dframe(dataname="poptype",monitor=TRUE,filelog="")


   new_section(title="5.1. Epidemiology",texfile=texfile)
   new_section_html(title="5.1. Epidemiology",htmlfile=htmlfile)

    BIRO_ce_report(ind="5_1_1",
                   numclass=0,
                   condition="",
                   width=1,
                   stratum=c("type_dm"),
                   lev=NULL,
                   tab=0,
                   namevar=NULL,
                   tabvarsum=c("n"),
                   n=2,
                   bar=0,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=0,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   cex=cex,
                   stand=1,
                   outcome="diabetes",
                   cov=c("type_dm"),
                   factcov=c(TRUE),
                   strata=standardization_by,
                   levstd=list(a=list("Diabetic"=1),b=levtype_dm),
                   sucvalue=1,
                   per=1000,
                   dirdataout=dirdataout,
                   filename="5_1_1",
                   dirgraphlatex=dirgraphlatex,
                   caption="5_1_1",
                   map=1,
                   mapvar="Adjusted Rate",
                   title="5.1.1 Prevalence of diabetes mellitus per 1,000")


       BIRO_ce_report(ind="5_1_2",
                   numclass=4,
                   condition="",
                   width=1,
                   stratum=c("age_c","sex",NA,"type_dm"),
                   lev=list(a=classlabel(ageth10),b=names(levsex),c=""
,d=names(levtype_dmst)),
                   tab=1,
                   namevar=c("Age","Gender",NA,"Type of Diabetes"),
                   tabvarsum=c("n"),
                   n=c(9,2,NA,2),
                   bar=1,
                   barvarsum=c("n","sum"),
                   beside=TRUE,
                   perc=TRUE,
                   lines=0,
                   box=0,
                   texfile=texfile,
                   dirgraph=dirgraph,
                   cex=cex,
                   stand=1,
                   outcome="diabetes",
                   cov=c("sex","age_c","agesex"),
```

```
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,
                    levstd=list(a=list("Diabetic"
=1),b=levsex,c=classlabellist(ageth5),d=levagesex),
                    sucvalue=1,
                    per=1000,
                    dirdataout=dirdataout,
                    filename="5_1_2",
                    dirgraphlatex=dirgraphlatex,
                    caption="Age at diagnosis by 10 year age bands ",
                    map=1,
                    mapvar="Adjusted Rate",
                    title="5.1.2. Age at diagnosis by 10 year age bands ",
                    chisq=c("type_dm","age_c"))



  #5.2.1.
  #% of subjects with 1+ HbA1c tests
  #in last 12 months

   new_section(title="5.2. Process Quality",texfile=texfile)
   new_section_html(title="5.2. Process Quality",htmlfile=htmlfile)

       BIRO_ce_report(ind="5_2_1",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("hba1c_done","sex","age2_c","type_dm"),

lev=list(a=names(levhba1c_test),b=names(levsex),c=classlabel(ageth),d=names(levtype_dm
st)),
                    tab=1,
                    namevar=c("HbA1c done","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hba1c_done",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhba1c_done,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    dirdataout=dirdataout,
                    filename="5_2_1",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate",
                    title="5.2.1 % of subjects with 1+ HbA1c tests in last 12 months")



       BIRO_ce_report(ind="5_2_2",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("ma_done","sex","age2_c","type_dm"),

lev=list(a=names(levma_test),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst)
),
                    tab=1,
                    namevar=c("Microalbumin done","Age","Gender","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="ma_done12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levma_done,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_2",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate",
                    title="5.2.2 % of subjects with at least one test for
microalbuminuria in last 12 months")



       BIRO_ce_report(ind="5_2_3",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("eye_done","sex","age2_c","type_dm"),

lev=list(a=names(leveye_done),b=classlabel(ageth),b=names(levsex),c=names(levtype_dmst
)),
                    tab=1,
                    namevar=c("Eye examination","Age","Gender","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
```

```
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="eye_done12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=leveye_done,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_3",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate",
                    title="5.2.3 % of subjectswho received a dilated eye examination
or evaluation of retinal photography by a trained caregiver within the last 12 months"
)


       BIRO_ce_report(ind="5_2_4",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("foot_exam","sex","age2_c","type_dm"),

lev=list(a=names(levfoot_done),b=names(levsex),c=classlabel(ageth),d=names(levtype_dms
t)),
                    tab=1,
                    namevar=c("Foot examination","Age","Gender","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="foot_done12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levfoot_done,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_4",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate",
                    title="5.2.4 % of subjectsreceiving at least one foot examination
within the last 12 months")


       BIRO_ce_report(ind="5_2_5",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("smok_stat","sex","age2_c","type_dm"),

lev=list(a=names(levsmoke_doc),b=names(levsex),c=classlabel(ageth),d=names(levtype_dms
t)),
                    tab=1,
                    namevar=c("Smoke Status","Age","Gender","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="smoke_doc12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levsmoke_doc,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_5",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate",
                    title="5.2.5 % of subjectswhose smoking status was ascertained and
documented within the last 12 months")



       BIRO_ce_report(ind="5_2_6",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("creat","sex","age2_c","type_dm"),

lev=list(a=names(levcreat_done),b=names(levsex),c=classlabel(ageth),d=names(levtype_dm
st)),
                    tab=1,
                    namevar=c("Creatinine","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
```

```
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="creat_done12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levcreat_done,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_6",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate",
                    title="5.2.6 % with serum creatinine tested in last 12 months")



      BIRO_ce_report(ind="5_2_7",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("bp","sex","age2_c","type_dm"),

lev=list(a=names(levbp12),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst)),
                    tab=1,
                    namevar=c("BP measurements","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="bp_done12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levbp_done,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_7",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate",
                    title="5.2.7 % of subjects with diabetes and one or more blood
pressure measurements within the last 12 months")



      BIRO_ce_report(ind="5_2_8",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("hypert_med","sex","age2_c","type_dm"),

lev=list(a=names(levhypert),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst))
,
                    tab=1,
                    namevar=c("Hypertension","Age","Gender","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert_done12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_8",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.2.8 % of subjects with
hypertension who receive antihypertensive medication")


      BIRO_ce_report(ind="5_2_9_1",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("sulphonylureas","sex","age2_c"),

lev=list(a=names(levoral_th12),b=names(levsex),c=classlabel(ageth)),
                    tab=1,
                    namevar=c("Sulphonylureas","Gender","Age"),
                    tabvarsum=c("n"),
                    n=c(2,2,5),
```

```
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Sulphonylureas")

       BIRO_ce_report(ind="5_2_9_2",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("biguanides","sex","age2_c"),

lev=list(a=names(levoral_th12),b=names(levsex),c=classlabel(ageth)),
                    tab=1,
                    namevar=c("Biguanides","Gender","Age"),
                    tabvarsum=c("n"),
                    n=c(2,2,5),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Biguanides")

       BIRO_ce_report(ind="5_2_9_3",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("glucosidase_inhibitors","sex","age2_c"),

lev=list(a=names(levoral_th12),b=names(levsex),c=classlabel(ageth)),
                    tab=1,
                    namevar=c("Glucosidase inhibitors","Gender","Age"),
                    tabvarsum=c("n"),
                    n=c(2,2,5),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Glucosidase inhibitors")

       BIRO_ce_report(ind="5_2_9_4",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("glitazones","sex","age2_c"),

lev=list(a=names(levoral_th12),b=names(levsex),c=classlabel(ageth)),
                    tab=1,
                    namevar=c("glitazones","Gender","Age"),
                    tabvarsum=c("n"),
                    n=c(2,2,5),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Glitazones")

       BIRO_ce_report(ind="5_2_9_5",
                    numclass=2,
                    condition="",
                    width=1,
                    stratum=c("glinides","sex","age2_c"),

lev=list(a=names(levoral_th12),b=names(levsex),c=classlabel(ageth)),
                    tab=1,
                    namevar=c("glinides","Gender","Age"),
                    tabvarsum=c("n"),
                    n=c(2,2,5),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    title="Type of oral therapy (distribution of agents) in patients
with diabetes type 2: Glinides")


       BIRO_ce_report(ind="5_2_10",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("ins","sex","age2_c","type_dm"),

lev=list(a=names(levinsulin12),b=names(levsex),c=classlabel(ageth),d=names(levtype_dms
t)),
                    tab=1,
                    namevar=c("insulin","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
```

```
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="insulin12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levinsulin12,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_10",
                    dirgraphlatex=dirgraphlatex,
                    caption="",
                    map=1,
                    mapvar="Adjusted Rate",
                    title="5.2.10 % of subjects treated with insulin")

      BIRO_ce_report(ind="5_2_11",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("insOAD","sex","age2_c","type_dm"),

lev=list(a=names(levinsOAD),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst))
,
                    tab=1,
                    namevar=c("Insulin and OAD","Age","Gender","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="insOAD",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levinsOAD,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_11",
                    dirgraphlatex=dirgraphlatex,
                    caption="",
                    map=1,
                    mapvar="Adjusted Rate",
                    title="5.2.11 % of subjects treated with insulin in combination
with OADs among patients with diabetes")


      BIRO_ce_report(ind="5_2_12",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("pump","sex","age2_c","type_dm"),

lev=list(a=names(levinsulin12),b=names(levsex),c=classlabel(ageth),d=names(levtype_dms
t)),
                    tab=1,
                    namevar=NULL,
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="inspump",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levinspump,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_12",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.2.12 % of subjects with pump
therapy")

      BIRO_ce_report(ind="5_2_13",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("hypert_med","sex","age2_c","type_dm"),

lev=list(a=names(levinsulin12),b=names(levsex),c=classlabel(ageth),d=names(levtype_dms
t)),
                    tab=1,
                    namevar=NULL,
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
```

```
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_13",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate",
                    title="5.2.13 % of subjects with anti hypertensive treatment")

       BIRO_ce_report(ind="5_2_14",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("lipid_med","sex","age2_c","type_dm"),

lev=list(a=names(levlipids),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst))
,
                    tab=1,
                    namevar=NULL,
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_14",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.2.14 % of subjects with lipid
lowering treatment")

       BIRO_ce_report(ind="5_2_15",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("asa_med","sex","age2_c","type_dm"),

lev=list(a=names(levasa_med),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst)
),
                    tab=1,
                    namevar=NULL,
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_15",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.2.15 % of subjects with ASA
treatment")

       BIRO_ce_report(ind="5_2_16",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("self_mon12","sex","age2_c","type_dm"),

lev=list(a=names(levself_mon),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst
)),
                    tab=1,
                    namevar=c("selfmon. blood glucose/ urine","Age","Gender","Type of
Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
```

```
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_2_16",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.2.16 % of subjects performing
selfmonitoring of blood glucose/ urine testing")

   new_section("5.3. Outcome quality - intermediate outcomes",texfile=texfile)
   new_section_html("5.3. Outcome quality - intermediate outcomes",htmlfile=htmlfile)

       BIRO_ce_report(ind="5_3_1",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("hba1c_c9","sex","age2_c","type_dm"),
                    lev=list(a=c("(9 + )","(0 - 9]"
),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst)),
                    tab=1,
                    namevar=c("HbA1c","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_3_1",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.3.1 % of subjects with most
recent HbA1c level greater than 9.0 pct (poor control)")

       BIRO_ce_report(ind="5_3_2",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("hba1c_c75","sex","age2_c","type_dm"),
                    lev=list(a=c("(7.5 + )","(0 - 7.5]"
),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst)),
                    tab=1,
                    namevar=c("HbA1c","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_3_2",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.3.2 % of subjects with most
recent HbA1c level greater than 7,5 pct")

       BIRO_ce_report(ind="5_3_3",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("sbp_c140","sex","age2_c","type_dm"),
                    lev=list(a=c("[0 - 140)","[140 +)"
),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst)),
                    tab=0,
                    namevar=c("SBP","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=0,
                    outcome="sbp_c140",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
```

```
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_3_3",
                    dirgraphlatex=dirgraphlatex,

                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.3.3 % of subjects with most
recent blood pressure less than 140/90 mmHg")

       BIRO_ce_report(ind="5_3_4",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("bmi_c30","sex","age2_c","type_dm"),
                    lev=list(a=c("[30 + )","(0 - 30)"
),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst)),
                    tab=1,
                    namevar=c("BMI","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_3_4",
                    dirgraphlatex=dirgraphlatex,
                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.3.4 % of subjects with BMI
greater than 30")

       BIRO_ce_report(ind="5_3_5",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("ma_test","sex","age2_c","type_dm"),

lev=list(a=names(levma_test),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst)
),
                    tab=1,
                    namevar=c("Microalbuminuria","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_3_5",
                    dirgraphlatex=dirgraphlatex,
                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.3.5 % of subjects with
microalbuminuria in last 12 months (among those who have been tested)")

       BIRO_ce_report(ind="5_3_6",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("current_smoke","sex","age2_c","type_dm"),
                    lev=list(a=c("current smoker","no current smoker"
),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst)),
                    tab=1,
                    namevar=NULL,
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_3_6",
```

```
                    dirgraphlatex=dirgraphlatex,
                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.3.6 Rate of current smokers among
diabetes patients")

       BIRO_ce_report(ind="5_3_7",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("ulcer","sex","age2_c","type_dm"),

lev=list(a=names(levulcer),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst)),
                    tab=1,
                    namevar=c("Foot Ulceration","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_3_7",
                    dirgraphlatex=dirgraphlatex,
                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.3.7 Former or current foot
ulceration")

   new_section("5.4. Outcome quality - terminal outcomes",texfile=texfile)
   new_section_html("5.4. Outcome quality - terminal outcomes",htmlfile=htmlfile)

       BIRO_ce_report(ind="5_4_1",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("diatrans","sex","age2_c","type_dm"),

lev=list(a=names(levdiatrans),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst
)),
                    tab=1,
                    namevar=c("Dialysis and/or Transplantation","Gender","Age","Type
of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_4_1",
                    dirgraphlatex=dirgraphlatex,
                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.4.1 Annual incidence of dialysis
and/or transplantation (renal replacement therapy in patients with diabetes")

       BIRO_ce_report(ind="5_4_2",
                    numclass=4,
                    condition="",
                    width=1,
                    stratum=c("esrf","sex","age2_c","type_dm"),

lev=list(a=names(levesrf),b=names(levsex),c=classlabel(ageth),d=names(levtype_dmst)),
                    tab=1,
                    namevar=c("ESRF","Gender","Age","Type of Diabetes"),
                    tabvarsum=c("n"),
                    n=c(2,2,5,2),
                    bar=1,
                    barvarsum=c("n","sum"),
                    beside=TRUE,
                    perc=TRUE,
                    lines=0,
                    box=0,
                    texfile=texfile,
                    dirgraph=dirgraph,
                    cex=cex,
                    stand=1,
                    outcome="hypert12",
                    cov=c("sex","age_c","agesex"),
                    factcov=c(TRUE,TRUE,TRUE),
                    strata=standardization_by,

levstd=list(a=levhypert,b=classlabellist(agethst),c=levsex,d=levtype_dm),
                    sucvalue=1,
                    per=100,
                    pop=NULL,
                    dirdataout=dirdataout,
                    filename="5_4_2",
                    dirgraphlatex=dirgraphlatex,
                    caption="",
                    map=1,
                    mapvar="Adjusted Rate", title="5.4.2 ESRF in Persons with Diabetes"
)
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_ce.r                                                              #
#                                                                              #
#                                                                              #
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_ce.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
# biro_ce_report
# open_tex
# new_chapter
# new_section
# new_sub_section
# include_tex
# import_pdf
# import_large_pdf
# close_tex
# new_section_html
# new_sub_section_html
# new_chapter_html
#
################################################################################
#

################################################################################
#  BIRO_ce_report
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_ce_report is the function the manage indicator and outputs of the Central
Engine
#
# DEPENDENCIES
#
# open_tex
# new_chapter
# new_section
# new_sub_section
# include_tex
# import_pdf
# import_large_pdf
# new_section_html
# new_sub_section_html
# new_chapter_html
# overtab
# overbar
# overbox
# biro_ce
#
################################################################################
#PARAMETERS:
################################################################################
# ind                           = string containing indicator index
# numclass                      = number of class variables
# condition      [where]        = string containing the where part of a SQL statement
#                               = useful for selection in statistical objects
# width          [1]            = width of graph in PDF report
# stratum        [NULL          = stratum variable name for standardization
# lev                           = levels of the categorical conditionated and
conditionating variables
# tab                           = 1 if output tables is required 0 otherwise
# namevar                       = name of the conditionated variable
# tabvarsum                     = name of the variable must be aggregated
# n                             = number of levels of conditionated and
conditionating variables
# side           [FALSE]        = FALSE for orizontal TRUE for vertical orientation
# bar                           = 1 if output barplot is required 0 otherwise
# barvarsum                     = name of the variable must be aggregated
# beside         [TRUE]         = a logical value. If FALSE the columns of height
#                               = are portrayed as stacked bars
#                                 and if TRUE the columns are portrayed as juxtaposed
#                                 bars.
# perc           [TRUE]         = a logical value. If TRUE percentages are calculated
#                                 in the barplot
# descvar                       = description name of variable represented in graphs
# what           ["patient"]    = description of what in represented in graph
```

```
(patient episode...)
# box                           = 1 if output boxplot is required 0 otherwise
# notch          [FALSE]        = logical. if notch is TRUE a notch is drawn in
#                                 each side of the boxes. If the notches of two
#                                 plots do not overlap this is ï¿½strong evidenceï¿½
that
#                                 the two means differ (Chambers et al. 1983 p. 62).
#                                 See boxplot.stats for the calculations used.
# dirgraph                      = the path of output graphs directory
# texfile                       = the path of the texfile
# cex            [cex]          = expansion factor for words and number printed
# trellis        [0]            = 1 if output boxplot is required 0 otherwise
# stand          [0]            = 1 if standardization output is required 0 otherwise
# title          [""]           = title printed
# outcome                       = the name of outcome variable (for BIRO_standardize)
# cov                           = a vector containing the names of covariates
# levstd                        = a list contaning levels of depvar and indepvar
# factcov        [NULL]         = logical TRUE if relative cov is a factor
# strata         [NULL]          = the name of the variable of stratification
# sucvalue       [NULL]         = the value that represent presence of the outcome
# url            [NULL]         = url of previous estimates to apply at the
standardization model
# pop            [NULL]         = the data.frame containing the population data
# per            [per]          = the moltiplication factor of rates
# dirdataout     [""]           = output directory of Statistical Objects
# stratatname    [""]           = the description name of the strata variable
# filename       [""]           = the name file of standardizione output
# dirgraphlatex  [""]           = the relative path of output graphs directory (for
Latex)
# caption        [""]           =  caption to print after the table
(BIRO_standardize)
# chisq          [NULL]         = a vector containing the name of variable to
calculate
#                                Cochran-Mantel-Haenszel Chi-Squared Test for Count
#                                Data or Pearson Chi-Squared Test
# struc          [TRUE]         = TRUE if you want to create the structure of HTML
pages
# map            [0]            = 1 if you map output for adjusted rates
# mapvar         [Adjusted Rate] = variable name represented in maps
# onlyprint      [0]            = 1 for onlyprint indicator
#
################################################################################
```

```
BIRO_ce_report<-function(ind,
                         numclass,
                         condition=where,
                         width=1,
                         stratum=NULL,
                         lev,
                         tab,
                         namevar,
                         tabvarsum,
                         n,
                         side=FALSE,
                         bar,
                         barvarsum,
                         beside=TRUE,
                         perc=TRUE,
                         lines,
                         descvar,
                         what="prova",
                         box,
                         notch=FALSE,
                         namegraph,
                         dirgraph,
                         texfile,
                         cex=cex,
                         trellis=0,
                         stand=0,
                         title="",
                         outcome="",
                         cov="",
                         levstd=NULL,
                         factcov=NULL,
                         strata=NULL,
                         sucvalue=NULL,
                         url=NULL,
                         pop=NULL,
                         per=per,
                         dirdataout="",
                         stratatname="",
                         filename="",
                         dirgraphlatex="",
                         caption="",
                         chisq=NULL,
                         struc=TRUE,
                         map=0,
                         mapvar="Adjusted Rate",
                         onlyprint=0) {

dirdataout=get("dirdataout",envir=.GlobalEnv)

listtables<-dbListTables(conn)

substrtables<-substr(listtables,1,nchar(ind)+1)

if (paste("i",ind,sep="") %in% substrtables) {

 write(paste("Processing...",title),file="")
 assign(paste("appendix",ind,sep=""),NULL,envir=.GlobalEnv)

 titlelatex<-gsub("\\%","\\\\%",as.character(title))
 titlelatex<-gsub("<","$<$",as.character(titlelatex))
 titlelatex<-gsub(">","$>$",as.character(titlelatex))

 indprint<-gsub("_",".",as.character(ind))
 assign("ind",ind,envir = .GlobalEnv)
 assign("indprint",indprint,envir = .GlobalEnv)


if (onlyprint==0 & stand==0) {
 if ("missing" %in% lev[[1]]) {
  n[1]<-n[1]+1
 }
}
```

```
if (stand==0) {
 if (is.null(stratum)==FALSE) {
  if (is.null(chisq) & length(na.omit(stratum))>=2) {
   chisq<-na.omit(stratum)[1:length(na.omit(stratum))]
  }
 }
}

if (stand==1) {
 if (is.null(stratum)==FALSE) {
  if (is.null(chisq) & length(na.omit(stratum))>2) {
   chisq<-na.omit(stratum)[1:length(na.omit(stratum))]
  }
 }
}


 htmlfile<-paste(dirhtml,"/",ind,".html",sep="")
 assign("ind",ind,envir=.GlobalEnv)
 assign("dirse",dirce,envir=.GlobalEnv)

 assign("htmlfile",htmlfile,envir=.GlobalEnv)

  new_sub_section(title=title,texfile=texfile)
  new_sub_section_html(title=title,htmlfile=rephtmlfile)


 if (struc==TRUE) {
  dirstruc<-paste(dirout,"/output/reports/structure",sep="")
  dir.create(dirstruc,showWarnings = FALSE)
  sthtml<-paste(dirstruc,"/",ind,".html",sep="")

  cat("",file=sthtml,fill = FALSE,append=FALSE)

  assign("struc",TRUE,envir=.GlobalEnv)
  assign("sthtml",sthtml,envir=.GlobalEnv)
 }
 if (struc==FALSE) {

  assign("struc",FALSE,envir=.GlobalEnv)
  assign("sthtml","",envir=.GlobalEnv)
 }

#html
 file.copy(paste(dirce,"/source/html/","layout_open.html",sep=""),
           paste(dirhtml,"/",ind,".html",sep=""),
           overwrite=TRUE)

cat("<body>",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("<table class=\"reportheader\" border=\"0\">",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("<tbody>",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <tr>",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("  <td><img src=\"../images/biro-logo01.jpg\"></td>",
      file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" </tr>",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("</tbody>",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("</table>",
     file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)

 cat("<table class=\"reportheader\" border=\"0\">",file=paste(dirhtml,"/",ind,".html"
,sep=""),fill = TRUE,append=TRUE)
 cat("<tbody>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <td class=\"reportheader\"><a href=\"",paste("../",dbname,".html",sep=""),"\">
BIRO Report </a> site: ",centre_id,", ",
     format(as.Date(launchtime,format="%d%m%y%H%M%S"),format="%d %B %Y"),
     "</td>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("  <td class=\"reportheader\">Reference date: "
,format(as.Date(Sys.Date()),format="%d/%m/%y"),"</td>",file=paste(dirhtml,"/",ind,
".html",sep=""),fill = TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" <tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("  <td class=\"reportheader\">Parameter: ",title,"</td>",file=paste(dirhtml,"/"
,ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat(" </tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("</tbody>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
 cat("</table>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)


 # insert link in report.html
 cat("<a href=\" html/",ind,".html\">",title,"</a><br>",sep="",
     file=paste(dirreport,"/",dbname,".html",sep=""),fill = TRUE,append=TRUE)


   if (tab==1) {
    progtab=0
    if (numclass<4) {
    if (length(na.omit(stratum))==2) tnames<-""
    if (length(na.omit(stratum))==3) tnames<-c(stratum,"")
     for (q in 1:length(tnames)) {

     nametab<-paste("i",ind,tnames[q],"_d1_3a",sep="")
     nametabm<-paste("i",ind,tnames[q],"m_d1_3a",sep="")
    if (tnames[q]=="") codtab<-ind

     if (tnames[q]!="") codtab<-paste(ind,"__",letters[q],sep="")

     levm<-lev
     for (j in 1:length(stratum)) {
      levm[[j]]<-c("Valid Value","NV/NA")
     }

     if (dbExistsTable(conn, nametabm)) {
      progtab=progtab+1
     }
     codtabm<-paste(ind,"m",sep="")


#cat("\\newpage",file=paste(dirreport,"/pdf/temp/table",codtab,".tex",sep=""),fill =
TRUE,append=TRUE)

#cat("\\pagestyle{fancy}",file=paste(dirreport,"/pdf/temp/table",codtab,".tex",sep="")
,fill = TRUE,append=TRUE)
     #cat("\\chead{\\small
",titlelatex,"}",file=paste(dirreport,"/pdf/temp/table",codtab,".tex",sep=""),fill =
TRUE,append=TRUE)
```

```
    overtab(nametab=nametabm,
            firstlab=namevar[1],
            namevar=namevar,
            nvar=numclass,
            condition=condition,
            varsum=tabvarsum,
            by=c("noNA",paste("noNA",stratum[2:length(stratum)],sep="")),
            n=rep(2,length(stratum)),
            side=side,
            width=width,
            lev=levm,
            chisq=NULL,
            caption=paste("Table ",indprint,".",progtab,": Missing Data ",namevar[1],
                          " (by ",paste(na.omit(namevar[2:length(namevar)]),collapse=
","),")",
                          sep=""),
            texfile=texfile,
            codtex=ind,
            codtab=codtabm)

    if (dbExistsTable(conn, nametab)) {
     progtab=progtab+1
    }

    overtab(nametab=nametab,
            firstlab=namevar[1],
            namevar=namevar,
            nvar=numclass,
            condition=condition,
            varsum=tabvarsum,
            by=stratum,
            n=n,
            side=side,
            width=width,
            lev=lev,
            chisq=stratum[2:length(stratum)],
            caption=paste("Table ",indprint,".",progtab,": ",namevar[1],
                          " (by ",paste(na.omit(namevar[2:length(namevar)]),collapse=
","),")",
                          sep=""),
            texfile=texfile,
            codtex=ind,
            codtab=ind)


  file.append(texfile,
              paste(dirreport,"/pdf/temp/table",codtabm,".tex",sep=""))

  file.append(texfile,
              paste(dirreport,"/pdf/temp/table",codtab,".tex",sep=""))

  if (file.exists(paste(dirreport,"/html/temp/table",codtabm,".html",sep=""))) {
   file.append(paste(dirhtml,"/",ind,".html",sep=""),
               paste(dirreport,"/html/temp/table",codtabm,".html",sep=""))
  }

  if (file.exists(paste(dirreport,"/html/temp/table",codtab,".html",sep=""))) {
   file.append(paste(dirhtml,"/",ind,".html",sep=""),
               paste(dirreport,"/html/temp/table",codtab,".html",sep=""))
  }
```

```
  cat("        <br>",file=htmlfile,fill = TRUE,append=TRUE)
  cat("        <br>",file=htmlfile,fill = TRUE,append=TRUE)

    }
  }
   if (numclass==4) {

    newn=n
    newn[4]<-1

   if (length(stratum)==4) {

     if (length(na.omit(stratum))==2) orderstra=2
     if (length(na.omit(stratum))==4) orderstra=c(4,2:length(stratum[-4]))
     if (length(na.omit(stratum))==3 & is.na(stratum[3])) orderstra<-c(4,2)
     if (length(na.omit(stratum))==3 & is.na(stratum[3])==FALSE) orderstra=c(2,3)

     if (file.exists(paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""))) {

      cat("\\newpage",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""),fill
= TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=
""),fill = TRUE,append=TRUE)
      cat("\\chead{\\small ",titlelatex,"}",file=paste(dirreport,"/pdf/temp/table"
,ind,".tex",sep=""),fill = TRUE,append=TRUE)
     }
     for (s in orderstra) {

      if (s != orderstra[1]) {
       if (file.exists(paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""))) {
        cat("\\newpage",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""
),fill = TRUE,append=TRUE)
        cat("\\pagestyle{fancy}",file=paste(dirreport,"/pdf/temp/table",ind,".tex"
,sep=""),fill = TRUE,append=TRUE)
        cat("\\chead{\\small ",titlelatex,"}",file=paste(dirreport,"/pdf/temp/table"
,ind,".tex",sep=""),fill = TRUE,append=TRUE)
       }
      join=paste(stratum[s],"_",sep="")
      joinm=paste("nona",stratum[s],"_",sep="")
      nametab<-paste("i",ind,"_",join,"d1_3a",sep="")
      nametabm<-paste("i",ind,"_",joinm,"d1_3a",sep="")

      newby<-c(stratum[1],stratum[s])
      nnamevar<-c(1,s)

      nnamevar=unique(nnamevar)
      numclass=length(newby)

      newlev<-list()
      for (k in 1:length(nnamevar)) {
       newlev[[k]]<-lev[[nnamevar[k]]]
      }

      if (join=="") codtab<-ind

      if (join!="") codtab<-paste(ind,"__",tolower(join),sep="")

      levm<-lev
      for (j in 1:length(stratum)) {
      levm[[j]]<-c("Valid Value","NV/NA")
      }
```

```
      lev2<-lev
      lev2[[2]]<-lev[[s]]

      if (dbExistsTable(conn, nametabm)) {
       progtab=progtab+1
      }
    codtabm<-paste(codtab,"m",sep="")

    nonaby=c("noNA",paste("noNA",stratum[s],sep=""))

    bycaption=namevar[s]


#cat("\\newpage",file=paste(dirreport,"/pdf/temp/table",codtab,".tex",sep=""),fill =
TRUE,append=TRUE)

#cat("\\pagestyle{fancy}",file=paste(dirreport,"/pdf/temp/table",codtab,".tex",sep="")
,fill = TRUE,append=TRUE)
    #cat("\\chead{\\small
",titlelatex,"}",file=paste(dirreport,"/pdf/temp/table",codtab,".tex",sep=""),fill =
TRUE,append=TRUE)

    overtab(nametab=nametabm,
            firstlab=namevar[1],
            namevar=namevar[nnamevar],
            nvar=numclass[nnamevar],
            condition=condition,
            varsum=tabvarsum,
            by=nonaby,
            n=c(2,2),
            side=side,
            width=width,
            lev=levm,
            chisq=chisq,
            caption=paste("Table ",indprint,".",progtab,": Missing Data ",namevar[1],
                          " (by ",bycaption,")",
                          sep=""),
            texfile=texfile,
            codtex=ind,
            codtab=codtabm)

    if (dbExistsTable(conn, nametab)) {
     progtab=progtab+1
    }
      overtab(nametab=nametab,
              firstlab=namevar[1],
              namevar=namevar[nnamevar],
              nvar=numclass[nnamevar],
              condition=condition,
              varsum=tabvarsum,
              by=newby,
              n=c(n[1],n[s]),
              side=side,
              width=width,
              lev=lev2,
              chisq=stratum[s],
              caption=paste("Table ",indprint,".",progtab,": ",namevar[1],
                            " (by ",bycaption,")",
                            sep=""),
              texfile=texfile,
              codtex=ind,
              codtab=codtab)

     }
    if (stratum[4]=="type_dm") {

     allvalues=c("Type_1", "Type_2" ,"Other_Type")
    }
    allvaluesprint<-allvalues
    for (m in 1:length(allvalues)) {
     allvaluesprint[m]<-paste(unlist(strsplit(as.character(allvalues[m]),"_"
)),collapse=" ")
    }


    for (l in 1:length(allvalues)) {

    if (file.exists(paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""))) {
     cat("\\newpage",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""),fill
= TRUE,append=TRUE)
     cat("\\pagestyle{fancy}",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=
""),fill = TRUE,append=TRUE)
     cat("\\chead{\\small ",titlelatex," \\\\ \\bf ",namevar[4]," = "
,allvaluesprint[l],"}",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""),fill
= TRUE,append=TRUE)
    }
     for (s in 2:length(stratum[-4])) {

      if (allvalues[l]!="") join<-paste(stratum[s],"_",allvalues[l],sep="") else
join<-paste(stratum[s],sep="")
      if (allvalues[l]!="") joinm<-paste("nona",stratum[s],"_",allvalues[l],sep="")
else joinm<-paste("nona",stratum[s],"_",sep="")

      nametab<-paste("i",ind,"_",tolower(join),"d1_3a",sep="")
      if (allvalues[l]=="") nametab<-paste("i",ind,"_",tolower(join),"_d1_3a",sep="")
      nametabm<-paste("i",ind,"_",tolower(joinm),"d1_3a",sep="")

      newby<-c(stratum[1],stratum[s])
      nnamevar<-c(1,s)

      if (allvalues[l]!="") newby<-c(newby,stratum[4])
      if (allvalues[l]!="") nnamevar<-c(nnamevar,4)
      nnamevar=unique(nnamevar)
      numclass=length(newby)
      newlev<-list()
      for (k in 1:length(nnamevar)) {
       newlev[[k]]<-lev[[nnamevar[k]]]
      }

      if (join=="") codtab<-ind

      if (join!="") codtab<-paste(ind,"__",tolower(join),sep="")

      levm<-lev
      for (j in 1:length(stratum)) {
      levm[[j]]<-c("Valid Value","NV/NA")
      }

      if (dbExistsTable(conn, nametabm)) {
       progtab=progtab+1
      }
    codtabm<-paste(codtab,"m",sep="")
```

```
     nonaby=c("noNA",paste("noNA",stratum[s],sep=""))
     if (allvalues[l]!="") nonaby=c(nonaby,stratum[length(stratum)])
     if (allvalues[l]!="") thisnvar=3 else thisnvar=2

     if (allvalues[l]!="") levm[[3]]<-namevar[s]
     if (allvalues[l]!="") newlev[[3]]<-namevar[s]

     allvaluesprint<-allvalues
     for (m in 1:length(allvalues)) {
      allvaluesprint[m]<-paste(unlist(strsplit(as.character(allvalues[m]),"_"
)),collapse=" ")
     }

     bycaption=paste(namevar[s],", ",namevar[4]," = ",allvaluesprint[l],sep="")

     if (file.exists(paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""))) {
      cat("\\newpage",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""),fill
= TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=
""),fill = TRUE,append=TRUE)
      cat("\\chead{\\small ",titlelatex," \\\\ \\bf ",namevar[4]," = "
,allvaluesprint[l],"}",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""),fill
= TRUE,append=TRUE)
     }


     overtab(nametab=nametabm,
             firstlab=namevar[1],
             namevar=namevar[nnamevar],
             nvar=numclass[nnamevar],
             condition=condition,
             varsum=tabvarsum,
             by=nonaby,
             n=c(rep(2,thisnvar-1),1),
             side=side,
             width=width,
             lev=levm,
             chisq=NULL,
             caption=paste("Table ",indprint,".",progtab,": Missing Data ",namevar[1],
                           " (by ",bycaption,")",
                          sep=""),
             texfile=texfile,
             codtex=ind,
             codtab=codtabm)

     if (dbExistsTable(conn, nametab)) {
      progtab=progtab+1
     }

       overtab(nametab=nametab,
               firstlab=namevar[1],
               namevar=namevar[nnamevar],
               nvar=numclass[nnamevar],
               condition=condition,
               varsum=tabvarsum,
               by=newby,
               n=newn[nnamevar],
               side=side,
               width=width,
               lev=newlev,
               chisq=stratum[2:nnamevar],
               caption=paste("Table ",indprint,".",progtab,": ",namevar[1],
                             " (by ",bycaption,")",
                             sep=""),
               texfile=texfile,
               codtex=ind,
               codtab=codtab)
      }
     #all the exposure variables

     join<-paste(stratum[2],"_",stratum[3],"_",allvalues[l],sep="")
     joinm<-paste("nona",stratum[2],"_","nona",stratum[3],"_",allvalues[l],sep="")

     if (join=="") codtab<-ind
     if (join!="") codtab<-paste(ind,"__",tolower(join),sep="")
     #codtabm<-paste(codtab,"m",sep="")

     nametab<-paste("i",ind,"_",tolower(join),"d1_3a",sep="")
     nametabm<-paste("i",ind,"_",tolower(joinm),"d1_3a",sep="")


     nonaby=c("noNA",paste("noNA",stratum[c(2,3)],sep=""),stratum[4])
     newby<-stratum
     newlev<-lev
     newlev[[4]]<-paste(namevar[c(2,3)],collapse=" * ")

     levm<-lev
     for (j in 1:length(stratum)) {
      levm[[j]]<-c("Valid Value","NV/NA")
     }
     levm[[4]]<-paste(namevar[c(2,3)],collapse=" * ")

     namevar<-c(namevar,"")

     if (dbExistsTable(conn, nametabm)) {
      progtab=progtab+1
     }

     allvaluesprint<-allvalues
     for (m in 1:length(allvalues)) {
      allvaluesprint[m]<-paste(unlist(strsplit(as.character(allvalues[m]),"_"
)),collapse=" ")
     }

     bycaption=paste(namevar[2]," * ",namevar[3],", ",namevar[4]," = "
,allvaluesprint[l],sep="")

     if (file.exists(paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""))) {
      cat("\\newpage",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""),fill
= TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=
""),fill = TRUE,append=TRUE)
      cat("\\chead{\\small ",titlelatex," \\\\ \\bf ",namevar[4]," = "
,allvaluesprint[l],"}",file=paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""),fill
= TRUE,append=TRUE)
     }
     overtab(nametab=nametabm,
             firstlab=namevar[1],
             namevar=namevar,
             nvar=4,
             condition=condition,
             varsum=tabvarsum,
             by=nonaby,
             n=c(rep(2,3),1),
```

```
            side=side,
            width=width,
            lev=levm,
            chisq=NULL,
            caption=paste("Table ",indprint,".",progtab,": Missing Data ",namevar[1],
                          " (by ",bycaption,")",
                        sep=""),
            texfile=texfile,
            codtex=ind,
            codtab=codtabm)

     if (dbExistsTable(conn, nametab)) {
      progtab=progtab+1
     }
       overtab(nametab=nametab,
               firstlab=namevar[1],
               namevar=namevar[2:4],
               nvar=4,
               condition=condition,
               varsum=tabvarsum,
               by=newby,
               n=c(n[1:3],1),
               side=side,
               width=width,
               lev=newlev,
               chisq=stratum[2:4],
               caption=paste("Table ",indprint,".",progtab,": ",namevar[1],
                          " (by ",bycaption,")",
                          sep=""),
               texfile=texfile,
               codtex=ind,
               codtab=codtab)


     }
   file.append(texfile,
               paste(dirreport,"/pdf/temp/table",ind,".tex",sep=""))


   if (file.exists(paste(dirreport,"/html/temp/table",ind,".html",sep=""))) {
    file.append(paste(dirhtml,"/",ind,".html",sep=""),
                paste(dirreport,"/html/temp/table",ind,".html",sep=""))
   }

    }


cat("        <br>",file=htmlfile,fill = TRUE,append=TRUE)
cat("        <br>",file=htmlfile,fill = TRUE,append=TRUE)


     }

   }

if (bar==1) {
    progbar=0


     if (is.na(stratum[4])) {
```

```
     newn=n
     newn[4]<-1

      for (s in 2:length(na.omit(stratum))) {

       namebar<-paste("i",ind,"_",stratum[s],"_d4_1a",sep="")
       namebarm<-paste("i",ind,"_m_",stratum[s],"_d4_1a",sep="")
       codbar=ind
       newby<-c(stratum[1],stratum[s])
       nnamevar<-c(1,s)

       nnamevar=unique(nnamevar)
       numclass=length(newby)

       levm<-lev
       for (j in 1:length(stratum)) {
       levm[[j]]<-c("Valid Value","NV/NA")
       }

       lev2<-lev
       lev2[[2]]<-lev[[s]]


     codbarm<-paste(codbar,"m",sep="")

     nonaby=c("noNA",paste("noNA",stratum[s],sep=""))

     bycaption=namevar[s]
     nonaby=c("noNA",paste("noNA",stratum[s],sep=""))

     nonabyprint<-paste("noNA","_",stratum[s],sep="")

      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)

     if (dbExistsTable(conn, namebarm)) {
      progbar=progbar+1
     overbar(namebar=namebarm,
             namevar=namevar[c(1,s)],
             condition="",
             varsum=barvarsum,
             stratum=nonaby,
             lev=levm,
             beside=beside,
             perc=TRUE,
             namegraph=paste(ind,"_",progbar,sep=""),
             dirgraph=dirgraph,
             cex=cex)

    if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".pdf",sep="")))
    import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a",".pdf",sep=""),
               caption=paste("Barplot: ",indprint,".",progbar," - Missing Data "
,namevar[1],
                             " (by ",paste(namevar[s],collapse=","),")",
                             sep=""),
               width=1,
               dirgraph=dirgraphlatex,
               texfile=texfile)

    if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".png",sep="")))
    import_pdf_html(image=paste("../graphs/i",ind,"_",progbar,"g4_1a",".png",sep=""),
```

```
                htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
             caption=paste("Barplot: ",indprint,".",progbar," - Missing Data "
,namevar[1],
                           " (by ",paste(namevar[s],collapse=","),")",
                           sep=""))
     }


     byprint<-paste(stratum[1],"_",stratum[s],sep="")

     if (dbExistsTable(conn, namebar)) {
      progbar=progbar+1

     overbar(namebar=namebar,
             namevar=namevar[c(1,s)],
             condition=condition,
             varsum=barvarsum,
             stratum=stratum[c(1,s)],
             lev=lev2,
             beside=beside,
             perc=TRUE,
             namegraph=paste(ind,"_",progbar,sep=""),
             dirgraph=dirgraph,
             cex=cex)

    if(file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".pdf",sep="")))
    import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a",".pdf",sep=""),
             caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                           " (by ",paste(namevar[s],collapse=","),")",
                           sep=""),
             width=1,
             dirgraph=dirgraphlatex,
             texfile=texfile)

    if(file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".png",sep="")))
    import_pdf_html(image=paste("../graphs/i",ind,"_",progbar,"g4_1a",".png",sep=""),
                htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
             caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                           " (by ",paste(namevar[s],collapse=","),")",
                           sep=""))

     }

    #disaggregated graph
     if (disaggregation_by!="") {

       if (dbExistsTable(conn, namebar)==TRUE) {

        if (where=="")
         statement=paste("Select * from ",namebar,";",sep="")
         if (nchar(where)>0)
          statement=paste("Select * from ",namebar, " where ",where,";",sep="")
          temp<-as.data.frame(dbGetQuery(conn, statement))
          uniquevalues<-unique(temp[,pmatch(stratum[s],names(temp))])
          conditionsource=""
          for (v in 1:length(uniquevalues)){

          condstratum<-paste(stratum[s], "=\'", uniquevalues[v],"\'",sep="")
          if (where=="") condbarsource=paste(condstratum)
          if (where!="") condbarsource=paste(where," and ",condstratum)
```

```
     if (dbExistsTable(conn, namebar)) {
      progbar=progbar+1

         overbar(namebar=namebar,
                 namevar=c(namevar[1],disaggregation_by),
                 condition=condbarsource,
                 varsum=barvarsum,
                 stratum=c(stratum[1],disaggregation_by),

lev=list(a=lev[[1]],b=as.character(unique(temp[,pmatch(disaggregation_by,names(temp))]
))),
                 beside=beside,
                 perc=TRUE,
                 namegraph=paste(ind,"_",progbar,sep=""),
                 dirgraph=dirgraph,
                 cex=cex)

       if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".pdf",sep="")))
       import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a",".pdf",sep=""),
                  caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                                " (by datasources, ",paste(namevar[s]," = "
,uniquevalues[v],sep=""),",)",
                                sep=""),
                  width=1,
                  dirgraph=dirgraphlatex,
                  texfile=texfile)

       if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".png",sep="")))
       import_pdf_html(image=paste("../graphs/i",ind,"_",progbar,"g4_1a",".png",sep=""
),
                       htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                  caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                                " (by datasources, ",paste(namevar[s]," = "
,uniquevalues[v],sep=""),",)",
                                sep=""))
        }
       }
      }
     }
     }

     if (length(na.omit(stratum))==4) {


      if (length(na.omit(stratum))==2) orderstra=2
      if (length(na.omit(stratum))==4) orderstra=c(4,2:length(stratum[-4]))
      if (length(na.omit(stratum))==3 & is.na(stratum[3])) orderstra<-c(4,2)
      if (length(na.omit(stratum))==3 & is.na(stratum[3])==FALSE) orderstra=c(2,3)

       for (s in orderstra) {

       namebar<-paste("i",ind,"_",stratum[s],"_d4_1a",sep="")
       namebarm<-paste("i",ind,"_m_",stratum[s],"_d4_1a",sep="")
       codbar=ind
       newby<-c(stratum[1],stratum[s])
       nnamevar<-c(1,s)

       nnamevar=unique(nnamevar)
       numclass=length(newby)

       levm<-lev
```

```
      for (j in 1:length(stratum)) {
      levm[[j]]<-c("Valid Value","NV/NA")
      }

      lev2<-lev
      lev2[[2]]<-lev[[s]]


    codbarm<-paste(codbar,"m",sep="")

    nonaby=c("noNA",paste("noNA",stratum[s],sep=""))

    bycaption=namevar[s]

    nonabyprint<-paste("noNA","_",stratum[s],sep="")

     cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
     cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
     cat("\\chead{\\small ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)

      if (dbExistsTable(conn, namebarm)) {
       progbar=progbar+1

    overbar(namebar=namebarm,
            namevar=namevar[c(1,s)],
            condition="",
            varsum=barvarsum,
            stratum=nonaby,
            lev=levm,
            beside=beside,
            perc=TRUE,
            namegraph=paste(ind,"_",progbar,sep=""),
            dirgraph=dirgraph,
            cex=cex)

   if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".pdf",sep="")))
   import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a",".pdf",sep=""),
              caption=paste("Barplot: ",indprint,".",progbar," - Missing Data "
,namevar[1],
                            " (by ",paste(namevar[s],collapse=","),")",
                            sep=""),
              width=1,
              dirgraph=dirgraphlatex,
              texfile=texfile)

   if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".png",sep="")))
   import_pdf_html(image=paste("../graphs/i",ind,"_",progbar,"g4_1a",".png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
              caption=paste("Barplot: ",indprint,".",progbar," - Missing Data "
,namevar[1],
                            " (by ",paste(namevar[s],collapse=","),")",
                            sep=""))

      }
     rm(nonabyprint)


    byprint<-paste(stratum[1],"_",stratum[s],sep="")

      if (dbExistsTable(conn, namebarm)) {
       progbar=progbar+1

     overbar(namebar=namebar,
             namevar=namevar[c(1,s)],
             condition=condition,
             varsum=barvarsum,
             stratum=stratum[c(1,s)],
             lev=lev2,
             beside=beside,
             perc=TRUE,
             namegraph=paste(ind,"_",progbar,sep=""),
             dirgraph=dirgraph,
             cex=cex)

   if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".pdf",sep="")))
   import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a",".pdf",sep=""),
              caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                            " (by ",paste(namevar[s],collapse=","),")",
                            sep=""),
              width=1,
              dirgraph=dirgraphlatex,
              texfile=texfile)

   if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".png",sep="")))
   import_pdf_html(image=paste("../graphs/i",ind,"_",progbar,"g4_1a",".png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
              caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                            " (by ",paste(namevar[s],collapse=","),")",
                            sep=""))
     }
   rm(byprint)

    #disaggregated graph
    if (disaggregation_by!="") {

      if (dbExistsTable(conn, namebar)==TRUE) {

       if (where=="")
        statement=paste("Select * from ",namebar,";",sep="")
        if (nchar(where)>0)
         statement=paste("Select * from ",namebar, "where ",where,";",sep="")
         temp<-as.data.frame(dbGetQuery(conn, statement))

        uniquevalues<-unique(temp[,pmatch(stratum[s],names(temp))])
        conditionsource=""
       for (v in 1:length(uniquevalues)){

       condstratum<-paste(stratum[s], "=\'", uniquevalues[v],"\'",sep="")
        if (where=="") condbarsource=paste(condstratum)
        if (where!="") condbarsource=paste("and",condstratum)

       if (dbExistsTable(conn, namebar)) {
        progbar=progbar+1

        overbar(namebar=namebar,
               namevar=c(namevar[1],disaggregation_by),
               condition=condbarsource,
               varsum=barvarsum,
               stratum=c(stratum[1],disaggregation_by),

lev=list(a=lev[[1]],b=as.character(unique(temp[,pmatch(disaggregation_by,names(temp))]
))),
               beside=beside,
               perc=TRUE,
```

```
                namegraph=paste(ind,"_",progbar,sep=""),
                dirgraph=dirgraph,
                cex=cex)

       if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".pdf",sep="")))
       import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a",".pdf",sep=""),
                  caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                                " (by ",paste(namevar[s],collapse=","),",
datasources)",
                                sep=""),
                  width=1,
                  dirgraph=dirgraphlatex,
                  texfile=texfile)

       if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".png",sep="")))
       import_pdf_html(image=paste("../graphs/i",ind,"_",progbar,"g4_1a",".png",sep=""
),
                       htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                  caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                                " (by ",paste(namevar[s],collapse=","),",
datasources))",
                                sep=""))
        }
       }
      }
     }
      }



     if (stratum[4]=="type_dm") {

      allvalues=c("Type_1", "Type_2" ,"Other_Type")
     }

     for (l in 1:length(allvalues)) {

      if (length(na.omit(stratum))==2) orderstra=2
      if (length(na.omit(stratum))==4) orderstra=c(4,2:length(stratum[-4]))
      if (length(na.omit(stratum))==3 & is.na(stratum[3])) orderstra<-c(4,2)
      if (length(na.omit(stratum))==3 & is.na(stratum[3])==FALSE) orderstra=c(2,3)

      for (s in orderstra) {

       if (allvalues[l]!="") join<-paste(stratum[s],"_",allvalues[l],sep="") else
join<-paste(stratum[s],sep="")
       if (allvalues[l]!="") joinm<-paste("m_",stratum[s],"_",allvalues[l],sep="")
else joinm<-paste("m_",stratum[s],"_",sep="")

       namebar<-paste("i",ind,"_",tolower(join),"_d4_1a",sep="")
       if (allvalues[l]=="") nametab<-paste("i",ind,"_",tolower(join),"_d4_1a",sep="")
       namebarm<-paste("i",ind,"_",tolower(joinm),"_d4_1a",sep="")

       newby<-c(stratum[1],stratum[s])
       nnamevar<-c(1,s)

       if (allvalues[l]!="") newby<-c(newby,stratum[4])
       if (allvalues[l]!="") nnamevar<-c(nnamevar,4)
       nnamevar=unique(nnamevar)
       numclass=length(newby)
       newlev<-list()
       for (k in 1:length(nnamevar)) {
        newlev[[k]]<-lev[[nnamevar[k]]]
       }

       if (join=="") codtab<-ind

       if (join!="") codtab<-paste(ind,"__",tolower(join),sep="")

       levm<-lev
       for (j in 1:length(stratum)) {
       levm[[j]]<-c("Valid Value","NV/NA")
       }


     codtabm<-paste(codtab,"m",sep="")

     nonaby=c("noNA",paste("noNA",stratum[s],sep=""))
     if (allvalues[l]!="") nonaby=c(nonaby,stratum[length(stratum)])

     if (allvalues[l]!="") thisnvar=3 else thisnvar=2

     if (allvalues[l]!="") levm[[3]]<-namevar[s]
     if (allvalues[l]!="") newlev[[3]]<-namevar[s]


     allvaluesprint<-allvalues
     for (m in 1:length(allvalues)) {
      allvaluesprint[m]<-paste(unlist(strsplit(as.character(allvalues[m]),"_"
)),collapse=" ")
     }

     bycaption=paste(namevar[s],", ",namevar[4]," = ",allvaluesprint[l],sep="")
     nonaby=c("noNA",paste("noNA",stratum[s],sep=""))

     nonabyprint<-paste("noNA_","noNA","_",stratum[s],"_",allvalues[l],sep="")

      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small ",titlelatex," \\\\ \\bf ",namevar[4]," = "
,allvaluesprint[l],"}",file=texfile,fill = TRUE,append=TRUE)
       if (dbExistsTable(conn, namebarm)) {
        progbar=progbar+1

     overbar(namebar=namebarm,
             namevar=namevar[c(1,s)],
             condition="",
             varsum=barvarsum,
             stratum=nonaby,
             lev=levm,
             beside=beside,
             perc=TRUE,
             namegraph=paste(ind,"_",progbar,sep=""),
             dirgraph=dirgraph,
             cex=cex)

    if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".pdf",sep="")))
    import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a",".pdf",sep=""),
               caption=paste("Barplot: ",indprint,".",progbar," - Missing Data "
,namevar[1],
                             " (by ",paste(namevar[s],collapse=","),", ",
                             namevar[4]," = ",allvaluesprint[l],")",
                             sep=""),
               width=1,
```

```
              dirgraph=dirgraphlatex,
              texfile=texfile)

    if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".png",sep=""))) 
    import_pdf_html(image=paste("../graphs/i",ind,"_",progbar,"g4_1a",".png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
              caption=paste("Barplot: ",indprint,".",progbar," - Missing Data "
,namevar[1],
                            " (by ",paste(namevar[s],collapse=","),", ",
                            namevar[4]," = ",allvaluesprint[l],")",
                            sep=""))

       }
     byprint<-paste(stratum[1],"_",stratum[s],"_",allvalues[l],sep="")


       if (dbExistsTable(conn, namebarm)) {
        progbar=progbar+1


     overbar(namebar=namebar,
             namevar=namevar[c(1,s)],
             condition=condition,
             varsum=barvarsum,
             stratum=stratum[c(1,s)],
             lev=lev2,
             beside=beside,
             perc=TRUE,
             namegraph=paste(ind,"_",progbar,sep=""),
             dirgraph=dirgraph,
             cex=cex)

    if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".pdf",sep=""))) 
    import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a",".pdf",sep=""),
              caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                            " (by ",paste(namevar[s],collapse=","),", ",
                            namevar[4]," = ",allvaluesprint[l],")",
                            sep=""),
              width=1,
              dirgraph=dirgraphlatex,
              texfile=texfile)

    if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".png",sep=""))) 
    import_pdf_html(image=paste("../graphs/i",ind,"_",progbar,"g4_1a",".png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
              caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                            " (by ",paste(namevar[s],collapse=","),", ",
                            namevar[4]," = ",allvaluesprint[l],")",
                            sep=""))
       }
    #disaggregated graph
     if (disaggregation_by!="") {
       if (dbExistsTable(conn, namebar)==TRUE) {

        if (where=="")
         statement=paste("Select * from ",namebar,";",sep="")
         if (nchar(where)>0)
          statement=paste("Select * from ",namebar, "where ",where,";",sep="")
          temp<-as.data.frame(dbGetQuery(conn, statement))
         uniquevalues<-unique(temp[,pmatch(stratum[s],names(temp))])
         conditionsource=""
        for (v in 1:length(uniquevalues)){
```

```
         condstratum<-paste(stratum[s], "='", uniquevalues[v],"'",sep="")
          if (where=="") condbarsource=paste(condstratum)
          if (where!="") condbarsource=paste("and",condstratum)

         if (dbExistsTable(conn, namebar)) {
          progbar=progbar+1

          overbar(namebar=namebar,
                 namevar=c(namevar[1],disaggregation_by),
                 condition=condbarsource,
                 varsum=barvarsum,
                 stratum=c(stratum[1],disaggregation_by),

lev=list(a=lev[[1]],b=as.character(unique(temp[,pmatch(disaggregation_by,names(temp))]
))),
                 beside=beside,
                 perc=TRUE,
                 namegraph=paste(ind,"_",progbar,sep=""),
                 dirgraph=dirgraph,
                 cex=cex)

        if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".pdf",sep=""))) 
        import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_1a",".pdf",sep=""),
                   caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                                 " (by datasources, ",paste(namevar[s]," = "
,uniquevalues[v],sep=""),",)",
                                 sep=""),
                   width=1,
                   dirgraph=dirgraphlatex,
                   texfile=texfile)

        if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_1a",".png",sep=""))) 
        import_pdf_html(image=paste("../graphs/i",ind,"_",progbar,"g4_1a",".png",sep=""
),
                        htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Barplot: ",indprint,".",progbar," - ",namevar[1],
                                 " (by datasources, ",paste(namevar[s]," = "
,uniquevalues[v],sep=""),",)",
                                 sep=""))
         }
        }
       }
     }


   }
     }


   }

  if (bar==1) {
   if (length(stratum)==4) {
    assign("progbar",progbar,envir=.GlobalEnv)

    newlev<-list()
   for (j in 2:length(lev)) {
    newlev[[j-1]]<-lev[[j]]

   }

    overtrellis(namefd=paste("i",ind,"d1_1a",sep=""),
```

```
              condition="",
              namevar=paste(substr(stratum[1],1,nchar(stratum[1])-2),"_i",sep=""),
              stratum=na.omit(stratum[2:length(stratum)]),
              lev=newlev,
              labvar=stratum,
              namegraph=paste("i",ind,sep=""),
              dirgraph=dirgraph,
              hist=1,
              box=0)

 for (k in 1:length(lev[[4]])) {

    progbar=progbar+1
    assign("progbar",progbar,envir=.GlobalEnv)
    values<-lev[[4]]


    #if (stratum[4]=="type_dm") {
    # allvalues=c("Type 1", "Type 2" ,"Other Type")
    # values<-allvalues
    #}

  if (file.exists(paste(dirgraph,"/","i",ind,"_",progbar,"g4_2a.pdf",sep=""))) {

      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small ",titlelatex," \\\\ \\bf ",namevar[4]," = "
,allvaluesprint[k],"}",file=texfile,fill = TRUE,append=TRUE)

   if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_2a",".pdf",sep="")))
   import_pdf(namegraph=paste("i",ind,"_",progbar,"g4_2a.pdf",sep=""),
                    caption=paste("Trellis Barplot: ",indprint,".",progbar," - ",
                           " * ",paste(na.omit(namevar[1:2]),collapse=" * "),
                           " (",namevar[length(namevar[namevar!=""])]," = "
,values[k],")",sep=""),
                    width=1,
                    dirgraph=dirgraphlatex,
                    texfile=texfile)

   if (file.exists(paste(dirgraph,"/i",ind,"_",progbar,"g4_2a",".png",sep="")))
   import_pdf_html(image=paste("../graphs/i",ind,"_",progbar,"g4_2a.png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Trellis Barplot: ",indprint,".",progbar," - "
,namevar[1],
                          " * ",paste(na.omit(namevar[1:2]),collapse=" * "),
                          " (",namevar[length(namevar[namevar!=""])]," = ",values[k],
")",sep=""))

   }
  }
   }
   }
#   if (dbExistsTable(conn, paste("i",ind,"a","d4_1a",sep=""))) {
#    overbar(namebar=paste("i",ind,"a","d4_1a",sep=""),
#            namevar=namevar,
#            condition="",
#            varsum=barvarsum,
#            stratum=na.omit(stratum),
#            lev=lev,
#            beside=beside,
#            perc=perc,
#            namegraph=ind,
#            dirgraph=dirgraph,
#            cex=cex)

#   if (file.exists(paste(dirgraph,"/","i",ind,"ag4_1a",".png",sep=""))) {

#    import_pdf(namegraph=paste("i",ind,"ag4_1a",".png",sep=""),
#               caption=paste("Barplot: ",namevar[1],
#                             " (by ",paste(namevar[2],collapse=","),")",
#                             sep=""),
#               width=1,
#               dirgraph=dirgraphlatex,
#               texfile=texfile)

#    import_pdf_html(image=paste("../graphs/i",ind,"ag4_1a",".png",sep=""),
#                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
#               caption=paste("Barplot: ",namevar[1],
#                             " (by ",paste(namevar[2],collapse=","),")",
#                             sep=""))




#   datacenter
#   for (j in 1:length(lev[[2]])) {
#
#  if (condition=="")
#   statement=paste("Select * from ",paste("i",ind,"a","d4_1a",sep=""),";",sep="")
#  if (nchar(condition)>0)
#   statement=paste("Select * from ",paste("i",ind,"a","d4_1a",sep=""), "where
",condition,";",sep="")
#
#  temp<-as.data.frame(dbGetQuery(conn, statement))
#  header<-data.frame(NULL)
#  for (w in 1:dim(temp)[1]) {
#   for (y in 1:length(stratum)) {
#
header[w,stratum[y]]<-substr(as.character(temp[w,stratum[y]]),2,nchar(as.character(tem
p[w,stratum[y]]))-1)
#   }
#  }
#  for (y in 1:length(stratum)) {
#   temp[,pmatch(stratum[y],names(temp))]<-header[,pmatch(stratum[y],names(header))]
#  }
#  datacenter<-temp[temp[pmatch(stratum[2],names(temp))]==lev[[2]][j] &
is.na(temp[pmatch(stratum[2],names(temp))])==FALSE,]
#  print(datacenter)
#  datacenter$n<-as.numeric(as.character(datacenter$n))
#  datacenter$sum<-as.numeric(as.character(datacenter$sum))
#
#    if (dim(datacenter)[1]>0) {
#     datacenter2<-datacenter
#     datacenter2$id<-centre_id
#
total<-aggregate(datacenter2[,c("n","sum")],by=list(strata<-datacenter2[,stratum[1]]),
FUN=sum)
#     datacenter2<-datacenter2[1:dim(total)[1],]
#     datacenter2[,stratum[1]]<-total$Group.1
#     datacenter2[,stratum[2]]<-datacenter2[,stratum[2]]
#     datacenter2$n<-total$n
#     datacenter2$sum<-total$sum
#
```

```
#     datacenter<-rbind(datacenter,datacenter2)
#     print(datacenter)
#    overbar(dataframe=datacenter,
#            namevar=namevar,
#            condition="",
#            varsum=barvarsum,
#            stratum=c(stratum[1],"id"),
#            lev=lev[[1]],
#            beside=beside,
#            perc=perc,
#            namegraph=ind,
#            dirgraph=dirgraph,
#            cex=cex)
#
#    import_pdf(namegraph=paste("i",ind,letters[i],letters[j],"g4_1a.png",sep=""),
#               caption=paste("Barplot: ",lab_var_cat,
#               " by ","data source (",lab_class[i],"
",names(lev_class[[1]])[j],")",collapse=",",
#               sep=""),
#               width=1,
#               dirgraph=dirgraphlatex,
#               texfile=texfile)
#
#
import_pdf_html(image=paste("../graphs/","i",ind,letters[i],letters[j],"g4_1a.png",sep
="",
#                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
#                    caption=paste("Barplot: ",lab_var_cat,
#               " by ","data source (",lab_class[i],"
",names(lev_class[[1]])[j],")",collapse=",",
#               sep=""))
#    }
#       }
#      }
#   }
  }

  # if (numclass==2) {

   if (lines==1) {
    overlines(namelines=paste("i",ind,"d4_5a",sep=""),
              namevar=namevar,
              condition=condition,
              varsum=c("y","n"),
              stratum=c("times","strata","stat"),
              lev=lev,
              descvar=descvar,
              what=what,
              namegraph=paste("i",ind,"g4_5a",sep=""),
              dirgraph,
              condition=condition)

   if (file.exists(paste(dirgraph,"/","i",ind,"g4_5a",".pdf",sep=""))) {

    import_pdf(namegraph=paste("i",ind,"g4_5a",".pdf",sep=""),
               caption=paste("Lines: ",namevar[1],
                             " (by ",paste(namevar[2],collapse=","),")",
                             sep=""),
               width=1,
               dirgraph=dirgraphlatex,
               texfile=texfile)
    import_pdf_html(image=paste("../graphs/i",ind,"g4_5a",".png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
               caption=paste("Lines: ",namevar[1],
                             " (by ",paste(namevar[2],collapse=","),")",
                             sep=""))


    }
   }

#   if (box==1) {
#
#    overbox(namefd=paste("i",ind,"d1_1a",sep=""),
#            condition="",
#            namevar=paste(substr(stratum[1],1,nchar(stratum[1])-2),"_i",sep=""),
#            stratum=stratum[2],
#            lev=lev[[i]],
#            notch=notch,
#            namegraph=ind,
#            dirgraph=dirgraph)
#
#   if (file.exists(paste(dirgraph,"/","i",ind,"g4_4a",".png",sep=""))) {
#
#    import_pdf(namegraph=paste("i",ind,"g4_4a",".png",sep=""),
#               caption=paste("Boxplot: ",namevar[1],
#                             " (by ",paste(namevar[2],collapse=","),")",
#                             sep=""),
#               width=1,
#               dirgraph=dirgraphlatex,
#               texfile=texfile)
#
#    import_pdf_html(image=paste("../graphs/i",ind,"g4_4a",".png",sep=""),
#                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
#                    caption=paste("Boxplot: ",namevar[1],
#                                  " (by ",paste(namevar[2],collapse=","),")",
#                                  sep=""))
#   }
#  }

if (box==1) {

    progbox=0


     stratum1box<-paste(substr(stratum[1],1,nchar(stratum[1])-2),"_i",sep="")
     newn=n
     newn[4]<-1

      if (length(na.omit(stratum))==2) orderstra=2
      if (length(na.omit(stratum))==4) orderstra=c(4,2:length(stratum[-4]))
      if (length(na.omit(stratum))==3 & is.na(stratum[3])) orderstra<-c(4,2)
      if (length(na.omit(stratum))==3 & is.na(stratum[3])==FALSE) orderstra=c(2,3)


      for (s in orderstra) {

       namebox<-paste("i",ind,"_",stratum[s],"_d1_1a",sep="")
       codbox=ind
       newby<-c(stratum[1],stratum[s])
       nnamevar<-c(1,s)

       nnamevar=unique(nnamevar)
```

```
      numclass=length(newby)

      levm<-lev
      for (j in 1:length(stratum)) {
      levm[[j]]<-c("Valid Value","NV/NA")
      }

      lev2<-lev
      lev2[[2]]<-lev[[s]]

    if (dbExistsTable(conn, namebox)) {
     progbox=progbox+1
    }


    bycaption=namevar[s]

    byprint<-paste(stratum[1],"_",stratum[s],sep="")


     cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
     cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
     cat("\\chead{\\small ",titlelatex,"}",file=texfile,fill = TRUE,append=TRUE)

    overbox(namefd=namebox,
           condition="",
           namevar=stratum1box,
           stratum=stratum[s],
           lev=lev[[s]],
           notch=notch,
           namegraph=paste(ind,"_",progbox,sep=""),
           dirgraph=dirgraph)

   if (file.exists(paste(dirgraph,"/i",ind,"_",progbox,"g4_4a",".pdf",sep="")))
   import_pdf(namegraph=paste("i",ind,"_",progbox,"g4_4a",".pdf",sep=""),
              caption=paste("Boxplot: ",indprint,".",progbox," - ",namevar[1],
                            " (by ",paste(namevar[s],collapse=","),")",
                            sep=""),
              width=1,
              dirgraph=dirgraphlatex,
              texfile=texfile)

   if (file.exists(paste(dirgraph,"/i",ind,"_",progbox,"g4_4a",".png",sep="")))
   import_pdf_html(image=paste("../graphs/i",ind,"_",progbox,"g4_4a",".png",sep=""),
                   htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                   caption=paste("Boxplot: ",indprint,".",progbox," - ",namevar[1],
                                 " (by ",paste(namevar[s],collapse=","),")",
                                sep=""))
     }
   if (length(na.omit(stratum))==4) {

    if (stratum[4]=="type_dm") {

     allvalues=c("Type_1", "Type_2" ,"Other_Type")
    }

    for (l in 1:length(allvalues)) {
     for (s in 2:length(stratum[-4])) {


      if (allvalues[l]!="") join<-paste(stratum[s],"_",allvalues[l],"_",sep="") else
join<-paste(stratum[s],sep="")
      if (allvalues[l]!="") joinm<-paste("nona",stratum[s],"_",allvalues[l],sep="")
else joinm<-paste("nona",stratum[s],"_",sep="")

      namebox<-paste("i",ind,"_",tolower(join),"d1_1a",sep="")

      newby<-c(stratum[1],stratum[s])
      nnamevar<-c(1,s)

      if (allvalues[l]!="") newby<-c(newby,stratum[4])
      if (allvalues[l]!="") nnamevar<-c(nnamevar,4)
      nnamevar=unique(nnamevar)
      numclass=length(newby)
      newlev<-list()
      for (k in 1:length(nnamevar)) {
       newlev[[k]]<-lev[[nnamevar[k]]]
      }

      if (join=="") codtab<-ind

      if (join!="") codtab<-paste(ind,"__",tolower(join),sep="")

      levm<-lev
      for (j in 1:length(stratum)) {
      levm[[j]]<-c("Valid Value","NV/NA")
      }


    codtabm<-paste(codtab,"m",sep="")

    nonaby=c("noNA",paste("noNA",stratum[s],sep=""))
    if (allvalues[l]!="") nonaby=c(nonaby,stratum[length(stratum)])

    if (allvalues[l]!="") thisnvar=3 else thisnvar=2

    if (allvalues[l]!="") levm[[3]]<-namevar[s]
    if (allvalues[l]!="") newlev[[3]]<-namevar[s]

    allvaluesprint<-allvalues
    for (m in 1:length(allvalues)) {
     allvaluesprint[m]<-paste(unlist(strsplit(as.character(allvalues[m]),"_"
)),collapse=" ")
    }

    bycaption=paste(namevar[s],", ",namevar[4]," = ",allvaluesprint[l],sep="")

     cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
     cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
     cat("\\chead{\\small ",titlelatex," \\\\ \\bf ",namevar[4]," = "
,allvaluesprint[l],"}",file=texfile,fill = TRUE,append=TRUE)


    if (dbExistsTable(conn, namebox)) {
     progbox=progbox+1

    overbox(namefd=namebox,
           condition="",
           namevar=stratum1box,
           stratum=stratum[s],
           lev=lev[[s]],
           notch=notch,
           namegraph=paste(ind,"_",progbox,sep=""),
           dirgraph=dirgraph)
```

```
    if (file.exists(paste(dirgraph,"/i",ind,"_",progbox,"g4_4a",".pdf",sep="")))
    import_pdf(namegraph=paste("i",ind,"_",progbox,"g4_4a",".pdf",sep=""),
               caption=paste("Boxplot: ",indprint,".",progbox," - ",namevar[1]," (by "
,
                             paste(namevar[s],collapse=", "),", ",
                             namevar[4]," = ",allvaluesprint[l],")",sep=""),
               width=1,
               dirgraph=dirgraphlatex,
               texfile=texfile)

    if (file.exists(paste(dirgraph,"/i",ind,"_",progbox,"g4_4a",".png",sep="")))
    import_pdf_html(image=paste("../graphs/i",ind,"_",progbox,"g4_4a",".png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
               caption=paste("Boxplot: ",indprint,".",progbox," - ",namevar[1],
                             " (by ",paste(namevar[s],collapse=", "),", ",
                             namevar[4]," = ",allvaluesprint[l],")",
                             sep=""))
     }

   }
     }


   }

  }



  if (box==1) {
   if (length(stratum)==4) {

    assign("progbox",progbox,envir = .GlobalEnv)

    newlev<-list()
   for (j in 2:length(lev)) {
    newlev[[j-1]]<-lev[[j]]

   }

    overtrellis(namefd=paste("i",ind,"d1_1a",sep=""),
               condition="",
               namevar=paste(substr(stratum[1],1,nchar(stratum[1])-2),"_i",sep=""),
               stratum=na.omit(stratum[2:length(stratum)]),
               lev=newlev,
               labvar=stratum,
               namegraph=paste("i",ind,sep=""),
               dirgraph=dirgraph,
               hist=0,
               box=1)

 for (k in 1:length(lev[[4]])) {

     if (dbExistsTable(conn, namebox)) {
      progbox=progbox+1
     }
    values<-lev[[4]]


   # if (stratum[4]=="type_dm") {
   #  allvalues=c("Type 1", "Type 2" ,"Other Type")
   #  values<-allvalues[sort(as.vector(values))]
   # }

  if (file.exists(paste(dirgraph,"/","i",ind,"_",progbox,"g4_2a.pdf",sep=""))) {

      cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
      cat("\\pagestyle{fancy}",file=texfile,fill = TRUE,append=TRUE)
      cat("\\chead{\\small ",titlelatex," \\\\ \\bf ",namevar[4]," = "
,allvaluesprint[k],"}",file=texfile,fill = TRUE,append=TRUE)


    if (file.exists(paste(dirgraph,"/i",ind,"_",progbox,"g4_2a",".pdf",sep="")))
    import_pdf(namegraph=paste("i",ind,"_",progbox,"g4_2a.pdf",sep=""),
                     caption=paste("Trellis Boxplot: ",indprint,".",progbox," - "
,namevar[1],
                                   " * ",paste(na.omit(namevar[2:3]),collapse=" * "),
                                   " (",namevar[length(namevar[namevar!=""])]," = "
,values[k],")",sep=""),
                     width=1,
                     dirgraph=dirgraphlatex,
                     texfile=texfile)

    if (file.exists(paste(dirgraph,"/i",ind,"_",progbox,"g4_2a",".png",sep="")))
    import_pdf_html(image=paste("../graphs/i",ind,"_",progbox,"g4_2a.png",sep=""),
                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
                    caption=paste("Trellis Boxplot: ",indprint,".",progbox," - "
,namevar[1],
                                  " * ",paste(na.omit(namevar[2:3]),collapse=" * "),
                                  " (",namevar[length(namevar[namevar!=""])]," = ",values[k],
")",sep=""))

    }
   }
    }
    }


  if (stand==1) {

   namestd<-paste("i",ind,"d6_1a",sep="")

      if (dbExistsTable(conn, namestd)) {

  if (condition=="")
   statement=paste("Select * from ",namestd, ";",sep="")
  if (nchar(condition)>0)
   statement=paste("Select * from ",namestd, "where ",condition,";",sep="")

  dstd<-as.data.frame(dbGetQuery(conn, statement))
  dstd$Successes<-as.numeric(as.character(dstd$Successes))
  dstd$Insuccesses<-as.numeric(as.character(dstd$Insuccesses))

  dstd<-na.omit(dstd)

  if (length(unique(dstd[,pmatch(standardization_by,names(dstd))]))>1) {


  if (length(unique(dstd$target))>1) {

   for (i in 1:length(unique(na.omit(dstd[,pmatch("target",names(dstd))])))) {
    target<-unique(na.omit(dstd[,pmatch("target",names(dstd))]))[i]
    datatype<-na.omit(dstd[dstd[,pmatch("target",names(dstd))]==target,])
```

```
  if (dim(datatype)[1]>0) {


 # if ((all(datatype$Successes)==0)==FALSE) {

   if (exists("progtab")==FALSE) progtab=0
   progtab=progtab+1
   assign("progtab",progtab,envir=.GlobalEnv)
   if (exists("progbar")==FALSE) progbar=0
   progbar=progbar+1
   assign("progbar",progbar,envir=.GlobalEnv)
   if (exists("progfor")==FALSE) progfor=0
   progfor=progfor+1
   assign("progfor",progfor,envir=.GlobalEnv)


   mystd<-BIRO_standardize(data=NULL,
                           datafreq=datatype,
                           target=target,
                           title=title,
                           outcome=outcome,
                           cov=cov,
                           factcov=factcov,
                           strata=standardization_by,
                           lev=levstd,
                           sucvalue=sucvalue,
                           url=url,
                           pop=NULL,
                           per=per,
                           csvindex=ind,
                           dirdataout=dirdataout,
                           strataname=strataname,
                           filename=filename,
                           dirgraph=dirgraph,
                           dirgraphlatex=dirgraphlatex,
                           texfile=texfile,
                           caption=caption,
                           chisq=NULL,
                           map=map,
                           mapvar=mapvar,
                           std=1,
                           progtab=progtab,
                           progbar=progbar,
                           progfor=progfor)


  #}
    #latex(mystd,file=texfile,append=TRUE,table.env=FALSE)

#    BIRO_df2html(data=mystd,
#                 htmlfile=paste(dirtables,"/i",ind,"d1_3a.html",sep=""),
#                 catcol=NULL)
# file.append(paste(dirhtml,"/",ind,".html",sep=""),
#             paste(dirtables,"/i",ind,"d1_3a.html",sep=""))




#    import_pdf(namegraph=paste("map",ind,".png",sep=""),
#               caption=paste("Map: ",title,"(",mapvar,")",sep=""),
#               width=1,
#               dirgraph=dirgraphlatex,
#               texfile=texfile)


      }
     }
    } else {
  if ((all(dstd$Successes)==0)==FALSE) {
   if (exists("progtab")==FALSE) progtab=0
   progtab=progtab+1
   assign("progtab",progtab,envir=.GlobalEnv)
   if (exists("progbar")==FALSE) progbar=0
   progbar=progbar+1
   assign("progbar",progbar,envir=.GlobalEnv)
   if (exists("progfor")==FALSE) progfor=0
   progfor=progfor+1
   assign("progfor",progfor,envir=.GlobalEnv)

   mystd<-BIRO_standardize(data=NULL,
                           datafreq=dstd,
                           target="",
                           title=title,
                           outcome=outcome,
                           cov=cov,
                           factcov=factcov,
                           strata=standardization_by,
                           lev=levstd,
                           sucvalue=sucvalue,
                           url=url,
                           pop=pop,
                           per=per,
                           csvindex=ind,
                           dirdataout=dirdataout,
                           strataname=strataname,
                           filename=filename,
                           dirgraph=dirgraph,
                           dirgraphlatex=dirgraphlatex,
                           texfile=texfile,
                           caption=caption,
                           chisq=NULL,
                           map=map,
                           mapvar=mapvar,
                           std=1,
                           progtab=progtab,
                           progbar=progbar,
                           progfor=progfor)
     }
    }
   } else {

    targetval<-unique(na.omit(dstd[,pmatch("target",names(dstd))]))

    if (length(targetval)==1) {
     dstd$N<-dstd$Successes+dstd$Insuccesses
     dstd$'Crude Rate'<-round((dstd$Successes/dstd$N)*per,1)
     dstdprint<-dstd
     names(dstdprint)[pmatch('type_dm',names(dstdprint))]<-'Type of Diabetes'
     dstdprint$'Type of Diabetes'<-as.character(dstdprint$'Type of Diabetes')
     names(dstdprint)[pmatch('Successes',names(dstdprint))]<-'Obs'
     names(dstdprint)[pmatch('N',names(dstdprint))]<-'Population'
     lastrow<-c("T",sum(dstdprint$'Obs'),unique(dstdprint$'Population'
),round(sum(dstd$'Crude Rate'),1))
```

```
     dstdprint<-dstdprint[,pmatch(c('Type of Diabetes','Obs','Population','Crude Rate'
),names(dstdprint))]
     dstdprint<-rbind(dstdprint,lastrow)
     latex(dstdprint[,pmatch(c('Type of Diabetes','Obs','Population','Crude Rate'
),names(dstdprint))],col.just=c("l","r","r","r"
),file=texfile,append=TRUE,table.env=FALSE,na.blank=TRUE,rowname="",title="")

     BIRO_df2html(data=dstdprint[,pmatch(c('Type of Diabetes','Obs','Population',
'Crude Rate'),names(dstdprint))],
                  htmlfile=paste(dirtables,"/",ind,".html",sep=""),
                  catcol=NULL)


    }
   }
   }

#   }
#  if (patmap==1) {
#   namemap<-paste("i",ind,"d6_1a",sep="")
#   dmap<-as.data.frame(dbReadTable(conn, namemap))

#   for i in 1:length(unique(



#  }


  write(paste("Indicator:",title,"Done"),file="")
  if (nchar(logfile)>0) {
   write(paste("Indicator:",title,"Done"),file=logfile,append=TRUE)
  }
 }

 if (onlyprint==1) {
  if (ind=="4_1_1" | ind=="4_1_3") {
   namestd=paste("i",ind,"_d1_1",sep="")
  } else namestd=paste("i",ind,"d1_3a",sep="")

  if (dbExistsTable(conn, namestd)) {
   if (condition=="")
    statement=paste("Select * from ",namestd, ";",sep="")
   if (nchar(condition)>0)
    statement=paste("Select * from ",namestd, "where ",condition,";",sep="")

   dataprint<-as.data.frame(dbGetQuery(conn, statement))
   dataprintagg<-dataprint

  if (ind!="4_1_2") {
   csvdata<-dataprintagg
   csvdata$centre_id<-centre_id
   csvdata$dbname<-dbname
   if (ind=="4_1_1" | ind=="4_1_3") {
    extension="_d1_1"
   } else extension="d1_3a"

   write.csv(csvdata,paste(dirdataout,"/","i",ind,extension,".csv",sep=""
),row.names=FALSE)

  }
   if (ind=="4_1_1" |  ind=="4_1_3") {
```

```
    dataprintagg<-dataprint[,1:4]
     for (r in 2:4) {
      dataprintagg[,r]<-as.numeric(as.character(dataprintagg[,r]))
     }
    if ('Age_in_class' %in% names(dataprintagg)) names(dataprintagg)[pmatch(
'Age_in_class',names(dataprintagg))]<-'Age'
    dataprintagg<-aggregate(dataprintagg[,-pmatch('Age',names(dataprintagg))],by
=list('Age' = dataprintagg$'Age'),sum)
    if (dividedprev==TRUE) {
     for (r in 2:4) {
      dataprintagg[,r]<-dataprintagg[,r]/divisornumber
     }
    dataprintagg[,1]<-as.character(dataprintagg[,1])
    lastrow=c("Overall"
,sum(dataprintagg[,2]),sum(dataprintagg[,3]),sum(dataprintagg[,4]))
    names(dataprintagg)<-c("Age",'Males (N)','Females (N)','Total (N)')
    dataprintagg<-rbind(dataprintagg,lastrow)

    }

       for (i in 1:length(names(dataprintagg))) {
      names(dataprintagg)[i]<-paste(unlist(strsplit(names(dataprintagg)[i],"_"
)),collapse="\\_")
     }
     latex(dataprintagg,col.just=c("l","r","r","r"
),file=texfile,append=TRUE,na.blank=TRUE,table.env=FALSE,rowlabel="",rowname="",title=
"")

         BIRO_df2html(data=dataprintagg,
                htmlfile=htmlfile,
                catcol=NULL)

  }


 }

  if (ind=="3_1_1") {

    for (i in 1:length(names(dataprintagg))) {
     names(dataprintagg)[i]<-paste(unlist(strsplit(names(dataprintagg)[i],"_"
)),collapse="\\_")
    }


latex(dataprintagg[,1:3],file=texfile,append=TRUE,na.blank=TRUE,table.env=FALSE,rowlab
el="",rowname="",title="")

    BIRO_df2html(data=dataprintagg,
                 htmlfile=htmlfile,
                 catcol=NULL)

  }

  if (ind=="3_2_1" | ind=="3_2_2") {

    for (i in 1:length(names(dataprintagg))) {
     names(dataprintagg)[i]<-paste(unlist(strsplit(names(dataprintagg)[i],"_"
)),collapse="\\_")
    }
    year<-paste(unique(as.character(dataprintagg[,1])),collapse="-")
```

```
    if (ind=="3_2_1")
newdataprintagg<-as.data.frame(cbind(year,sum(as.numeric(as.character(dataprintagg[,2]
))),sum(as.numeric(as.character(dataprintagg[,3])))))
    if (ind=="3_2_2")
newdataprintagg<-as.data.frame(cbind(year,sum(as.numeric(as.character(dataprintagg[,3]
))),sum(as.numeric(as.character(dataprintagg[,4])))))

    if (dividedprev==TRUE)
newdataprintagg[,3]=as.numeric(as.character(newdataprintagg[,3]))/divisornumber
    if (ind=="3_2_1") names(newdataprintagg)<-c("year","beds","pop")
    if (ind=="3_2_2") names(newdataprintagg)<-c("year","ds phisicians","pop")


newdataprintagg$Rate<-as.numeric(as.character(newdataprintagg[,2]))/as.numeric(as.char
acter(newdataprintagg[,3]))
    newdataprintagg$Rate<-round(newdataprintagg$Rate*100000,2)


latex(newdataprintagg,file=texfile,append=TRUE,na.blank=TRUE,table.env=FALSE,rowlabel=
"",rowname="",title="")

    BIRO_df2html(data=dataprintagg,
                 htmlfile=htmlfile,
                 catcol=NULL)

    newdataprintagg$centre_id=centre_id
    newdataprintagg$dbname=dbname
    newdataprintagg$start=startdate
    newdataprintagg$end=enddate



  }

  if (ind=="4_1_2") {
 if (dbExistsTable(conn, "i4_1_1_d1_1") & dbExistsTable(conn, "i4_1_3_d1_1")) {

  if (condition=="")
   statement=paste("Select * from i4_1_1_d1_1;",sep="")
  if (nchar(condition)>0)
   statement=paste("Select * from i4_1_1_d1_1 where ",condition,";",sep="")

  pop<-as.data.frame(dbGetQuery(conn, statement))
  names(pop)<-c("age_band","popM","popF","Total")
  pop[,2]<-as.numeric(as.character(pop[,2]))
  pop[,3]<-as.numeric(as.character(pop[,3]))
  pop[,4]<-as.numeric(as.character(pop[,4]))

  if (condition=="")
   statement=paste("Select * from i4_1_3_d1_1;",sep="")
  if (nchar(condition)>0)
   statement=paste("Select * from i4_1_3_d1_1 where ",condition,";",sep="")

  mor<-as.data.frame(dbGetQuery(conn, statement))
  names(mor)<-c("age_band","morM","morF","Total")

  mor[,2]<-as.numeric(as.character(mor[,2]))
  mor[,3]<-as.numeric(as.character(mor[,3]))
  mor[,4]<-as.numeric(as.character(mor[,4]))

  exm<-BIRO_explife(pop=pop,
                    mor=mor,
                    agewidth=c(15,10,10,10,10,10,10,10,12.5),
                    ageth=c(15,25,35,45,55,65,75,85),
                    sex="M",
                    morname="mor",
                    popname="pop",
                    labsex='Males')

  exf<-BIRO_explife(pop=pop,
                    mor=mor,
                    agewidth=c(15,10,10,10,10,10,10,10,12.5),
                    ageth=c(15,25,35,45,55,65,75,85),
                    sex="F",
                    morname="mor",
                    popname="pop",
                    labsex='Females')
  dataprintagg<-as.data.frame(merge(exm,exf))
  dataprintagg$Males<-round(as.numeric(as.character(dataprintagg$Males,2)))
  dataprintagg$Females<-round(as.numeric(as.character(dataprintagg$Females,2)))
  dataprintagg<-na.omit(dataprintagg)
  names(dataprintagg)[pmatch("Males",names(dataprintagg))]<-'Males (years)'
  names(dataprintagg)[pmatch("Females",names(dataprintagg))]<-'Females (years)'

  csvdata<-dataprintagg
  csvdata$centre_id<-centre_id
  csvdata$dbname<-dbname
  write.csv(csvdata,paste(dirdataout,"/","i","4_1_2","_d1_1.csv",sep=""
),row.names=FALSE)
  rm(csvdata)

       for (i in 1:length(names(dataprintagg))) {
      names(dataprintagg)[i]<-paste(unlist(strsplit(names(dataprintagg)[i],"_"
)),collapse="\\_")
     }
     latex(dataprintagg,col.just=c("l","r","r","r"
),file=texfile,append=TRUE,na.blank=TRUE,table.env=FALSE,rowlabel="",rowname="",title=
"")

         BIRO_df2html(data=dataprintagg,
                htmlfile=htmlfile,
                catcol=NULL)

   }
  }

  }
 file.append(paste(dirhtml,"/",ind,".html",sep=""),
             paste(dirce,"/source/html/","layout_close.html",sep=""))

  #write appendix section

  if (is.null(get(paste("appendix",ind,sep="")))==FALSE) {
   cat("\\newpage",file=texappendix,fill = TRUE,append=TRUE)
   cat("\\pagestyle{plain}",file=texappendix,fill = TRUE,append=TRUE)
   new_sub_section(title=title,texfile=texappendix)
   new_sub_section_html(title=title,htmlfile=htmlappendix)

   myapp<-get(paste("appendix",ind,sep=""))
   for (namapp in 1:length(names(myapp))) {
    names(myapp)[namapp]<-gsub("_"," ",names(myapp)[namapp])
   }
   latex(myapp,title="",col.just=rep("r"
,length(names(myapp))),file=texappendix,append=TRUE,table.env=FALSE,na.blank=TRUE,rowl
```

```
abel="",rowname="")

   BIRO_df2html(data=myapp,
                htmlfile=htmlappendix,
                catcol=NULL)

   }
  }
}

################################################################################
#  open_tex                                                                    #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# open_tex manages the creation and style of the tex file
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#title     [""] = a string containing the title of the report
#authors   [""] = a vector of strings containing the authors of the report
#
#footnote  [""] = a string containing the footnote of the report
#
#time      [""] = a string containing the date of the report
#
#logo      [""] = a string containing the path of the logo (PNG format)
#layout     !   = a string containing the name of the layout .tex file
#final      !   = a string containing the name of the report .tex file
################################################################################

open_tex<-function(title="",
                   authors="",
                   footnote="",
                   time="",
                   logo="",
                   layout,
                   final) {

  x<-FALSE
  if (is.null(layout)) {
        print(layout)
        stop("layout not recognized")
    }

  if (is.null(final)) {
        print(final)
        stop("final not recognized")
    }

  if (file.exists(layout)==FALSE) {
     print(layout)
     stop("layout not exists")
 }

  if (file.exists(layout)==TRUE) {

  x<-  file.copy(layout,final,overwrite=TRUE)


  }

  invisible(x)
}

################################################################################
#  new_chapter                                                                 #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# new_chapter creates a new chapter in the report PDF file
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#title        !   = a string containing the title of the chapter
#texfile      !   = a string containing the name of the report .tex file
################################################################################

new_chapter<-function(title,
                      texfile) {


 title<-paste(strsplit(title,"<"),collapse="\\<")
 title<-paste(strsplit(title,"%"),collapse="\\%")
 title<-paste(strsplit(title,">"),collapse="\\>")
 cat("\\chapter{",title,"}",file=texfile,fill = TRUE,append=TRUE)
}

################################################################################
#  new_section                                                                 #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
```

```
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista
# Programming Language: R 2.8.1, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
# ------------------------
# DESCRIPTION
#
# new_section creates a new section in the report PDF file
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#title        !   = a string containing the title of the section
#texfile      !   = a string containing the name of the report .tex file
################################################################################

new_section<-function(title,
                      texfile) {

 title<-gsub("%","\\\\%",as.character(title))
 title<-gsub("<","$<$",as.character(title))
 title<-gsub(">","$>$",as.character(title))

 cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
 cat("\\pagestyle{plain}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\section{",title,"}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)

}

################################################################################
#  new_sub_section                                                             #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# new_sub_section creates a new sub section in the report PDF file
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#title        !   = a string containing the title of the sub section
#texfile      !   = a string containing the name of the report .tex file
################################################################################

new_sub_section<-function(title,texfile) {

 title<-gsub("%","\\\\%",as.character(title))
 title<-gsub("<","$<$",as.character(title))
 title<-gsub(">","$>$",as.character(title))


 cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
 cat("\\pagestyle{plain}",file=texfile,fill = TRUE,append=TRUE)
 cat(paste("\\subsection{",title,"}",sep=""),file=texfile,fill = TRUE,append=TRUE)
 #cat("Reference Date:
",format(checkdate,format="%d/%m/%Y"),"\\par",file=texfile,fill = TRUE,append=TRUE)

}

################################################################################
#  include_tex                                                                 #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# include_tex include a tex file into another
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#incfile      !   = a string containing the name of the report .tex file
#texfile      !   = a string containing the name of the report .tex file
#caption      !   = a string containing the caption
################################################################################

include_tex<-function(incfile,
                      texfile,
                      caption) {

cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
cat("\\parbox[2]{",1,"\\textwidth}{",file=texfile,fill = TRUE,append=TRUE)

cat("\\include{",incfile,"}",file=texfile,fill = TRUE,append=TRUE)
cat("}",file=texfile,fill = TRUE,append=TRUE)

}

################################################################################
#  import_pdf                                                                  #
################################################################################
# ------------------------
```

```
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# import_pdf include a PNG file into a te
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#namegraph    !   = a string containing the name of the PNG file
#caption      !   = a string containing the caption
#width        !   = a numeric value between 0 and 1 indicating the width in
#                   the report
#dirgraph     !   = the directory where is the PGN file (latex format)
#texfile      !   = a string containing the name of the report .tex file
################################################################################

import_pdf<-function(namegraph,
                     caption,
                     width,
                     dirgraph,
                     texfile) {


 for (i in 1:nchar(dirgraph)) {
  if (substr(dirgraph,i,i)=="\\") {
   substr(dirgraph,i,i)<-"/"
  }
 }

 caption<-gsub("%","\\%",as.character(caption))
 caption<-gsub("<","$<$",as.character(caption))
 caption<-gsub(">","$>$",as.character(caption))
 caption<-gsub("\\%","\\\\%",as.character(caption))



 cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\parbox[2]{",width,"\\textwidth}{",file=texfile,fill = TRUE,append=TRUE)
 cat("\\fbox{",file=texfile,fill = TRUE,append=TRUE)
 cat("\\includegraphics[width=",width,"\\textwidth]{",file=texfile,fill =
FALSE,append=TRUE,sep="")
 cat(dirgraph,"/",namegraph,"}",file=texfile,fill = TRUE,append=TRUE,sep="")
 cat("}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\newline",file=texfile,fill = TRUE,append=TRUE)
 cat(paste(caption,sep=""),file=texfile,fill = TRUE,append=TRUE)
 cat("}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)

}
```

```
################################################################################
#  import_large_png                                                            #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# import_pdf include a PNG file into a te, layout landscape
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#namegraph    !   = a string containing the name of the PNG file
#caption      !   = a string containing the caption
#dirgraph     !   = the directory where is the PGN file (latex format)
#texfile      !   = a string containing the name of the report .tex file
################################################################################

import_large_pdf<-function(namegraph,
                           caption,
                           dirgraph,
                           texfile) {


 for (i in 1:nchar(dirgraph)) {
  if (substr(dirgraph,i,i)=="\\") {
   substr(dirgraph,i,i)<-"/"
  }
 }

 caption<-gsub("%","\\\\%",as.character(caption))
 caption<-gsub("<","$<$",as.character(caption))
 caption<-gsub(">","$>$",as.character(caption))



 cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\fbox{",file=texfile,fill = TRUE,append=TRUE)
 cat("\\includegraphics[width=0.90\\textwidth]",
     file=texfile,fill = TRUE,append=TRUE,sep="")
 cat("{",paste(dirgraph,"/",namegraph,sep=""),"}",file=texfile,fill = TRUE,
     append=TRUE,sep="")
 cat("}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\",file=texfile,fill = TRUE,append=TRUE)
 #cat("\\begin{rotate}{90}",file=texfile,fill = TRUE,append=TRUE)
 cat("\\parbox[2]{1\\textheight}{",file=texfile,fill = TRUE,append=TRUE)
 #cat("\\scriptsize",file=texfile,fill = TRUE,append=TRUE)
 cat(paste(caption,sep=""),file=texfile,fill = TRUE,append=TRUE)
 cat("}",file=texfile,fill = TRUE,append=TRUE)
 #cat("\\end{rotate}",file=texfile,fill = TRUE,append=TRUE)
```

```
 cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)
}

##############################################################################
#  close_tex                                                                 #
##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
## ------------------------
# DESCRIPTION
#
# BIRO_df2html writes the last rows of a .tex file (closes comands like
#  \document,...)
#
# DEPENDENCIES
#
# R package: base
#
##############################################################################
# PARAMETERS(!=required)
#
#file       !              =  string containig the name of the file that must
#                             be closed
#
##############################################################################

close_tex<-function(file) {

  x<-FALSE
  if (is.null(file)) {
        print(file)
        stop("file not recognized")
    }


  if (file.exists(file)==TRUE) {

  cat("\\end{document}",file=file,fill = TRUE,append=TRUE)



  }

  invisible(x)
}

##############################################################################
#  import_pdf_html                                                           #
##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# import_pdf include a PNG file into a te, layout landscape
#
# DEPENDENCIES
#
#  R package: base
#
##############################################################################
# PARAMETERS(!=required)
#
#namegraph    !   = a string containing the name of the PNG file
#caption      !   = a string containing the caption
#dirgraph     !   = the directory where is the PGN file (latex format)
#texfile      !   = a string containing the name of the report .tex file
##############################################################################

import_pdf_html<-function(image,
                          htmlfile,caption) {

  cat("<BR><BR><BR><BR><BR><BR>",
      file=htmlfile,fill=TRUE,append=TRUE)

  cat("<table style=\"width: 100%;\" border=\"1\">",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<tbody>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<tr>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<td class=\"graph\">",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat(paste("<img src=\"",image,"\">",sep=""),
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</td>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</tr>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</tbody>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<caption>",caption,"</caption>",
      file=htmlfile,fill = TRUE,append=TRUE)

  cat("</table>",
      file=htmlfile,fill=TRUE,append=TRUE)
}


##############################################################################
#  new_section_html                                                          #
##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista
```

```
# Programming Language: R 2.8.1, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
# ------------------------
# DESCRIPTION
#
# new_section_html creates a new section in a html file
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#title        !   = a string containing the title of the section
#htmlfile      !   = a string containing the name of the report .html file
################################################################################

new_section_html<-function(title,htmlfile) {

 title<-paste(strsplit(title,"<"),collapse="&lt;")
 title<-paste(strsplit(title,"%"),collapse="\\%")
 title<-paste(strsplit(title,">"),collapse="&gt;")

 cat("<table >",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<tbody>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<tr>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<td class=\"section\">",paste(title),
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</td>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</tr>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</tbody>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</table>",
      file=htmlfile,fill=TRUE,append=TRUE)
}

################################################################################
#  new_chapter_html                                                            #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# new_chapter_html creates a new chapter in the report html file
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#title        !   = a string containing the title of the chapter
#texfile      !   = a string containing the name of the report .html file
################################################################################

new_chapter_html<-function(title,
                           htmlfile) {

 title<-paste(strsplit(title,"<"),collapse="&lt;")
 title<-paste(strsplit(title,"%"),collapse="\\%")
 title<-paste(strsplit(title,">"),collapse="&gt;")

 cat("<table >",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<tbody>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<tr>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<td class=\"chapter\">",paste(title),
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</td>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</tr>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</tbody>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</table>",
      file=htmlfile,fill=TRUE,append=TRUE)
}

################################################################################
#  new_sub_section_html                                                        #
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2008-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# new_sub_section_html creates a new sub section in the report HTML file
#
# DEPENDENCIES
#
#  R package: base
#
################################################################################
# PARAMETERS(!=required)
#
#title        !   = a string containing the title of the sub section
#texfile      !   = a string containing the name of the report .html file
################################################################################

new_sub_section_html<-function(title,
                               htmlfile) {
```

```
 title<-paste(strsplit(title,"<"),collapse="&lt;")
# title<-paste(strsplit(title,"%"),collapse="\\%")
 title<-paste(strsplit(title,">"),collapse="&gt;")
 cat("<table >",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<tbody>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<tr>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("<td class=\"subsection\">",paste(title),
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</td>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</tr>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</tbody>",
      file=htmlfile,fill=TRUE,append=TRUE)
  cat("</table>",
      file=htmlfile,fill=TRUE,append=TRUE)
}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_overbar.r                                                         #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_overbar.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
# overbar
#
################################################################################


################################################################################
#  overbar
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_overbar is the function that aggregate data for output barplot
#
# DEPENDENCIES
#
# biro_plots.r
#
################################################################################
#PARAMETERS:
################################################################################
# nametab             = name of the table that you want to read
# dataframe           = data.frame loaded instead of postgres table name
# condition           = SQL where condition
# varsum              = name of the variabiles to aggregate
# stratum             = names of the stratum variable
# lev                 = levels of the variables
# number              = logical if you want numbers printed on the graphs
# beside              = a logical value. If FALSE the columns of height
# perc                = a logical value. If TRUE percentages are calculated
# what                = description of what in represented in graph (patient
episode...)
# namegraph           = output file name
# dirgraph            = the path of output graphs directory
# cex                 = expansion factor for words and number printed
#
################################################################################


overbar<-function(namebar,
                  dataframe=NULL,
                  namevar,
                  condition,
                  varsum,
                  stratum,
                  lev,
                  number=FALSE,
                  perc=TRUE,
                  beside,
                  what,
                  namegraph,
                  dirgraph,
                  cex) {


 if (dbExistsTable(conn, namebar)) {
 if (is.null(dataframe)==TRUE) {

  if (condition=="")
   statement=paste("Select * from ",namebar,";",sep="")
  if (nchar(condition)>0)
   statement=paste("Select * from ",namebar, " where ",condition,";",sep="")

  temp<-as.data.frame(dbGetQuery(conn, statement))
  write.csv(temp,file=paste(dirdataout,"/",namebar,".csv",sep=""))

  write(paste("Barplot data saved into: [OUTDATA]","/",namebar,".csv",sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("Barplot data saved into: [OUTDATA]","/",namebar,".csv",sep=""
),file=logfile,append=TRUE)
  }
```

```
 }
 if (is.null(dataframe)==FALSE) {
  temp<-dataframe
 }


  if (dim(temp)[1]>0) {

  for (i in 1:length(varsum)) {

temp[,pmatch(varsum[i],names(temp))]<-as.numeric(as.character(temp[,pmatch(varsum[i],n
ames(temp))]))
  }


bar<-as.data.frame(aggregate(temp[,pmatch(varsum,names(temp))],as.list(temp[,c(stratum
)]),FUN="sum"))
  names(bar)<-c(stratum,varsum)
  for (i in 1:length(stratum)) {

bar[,pmatch(stratum[i],names(bar))]<-as.character(bar[,pmatch(stratum[i],names(bar))])
   bar$codist=centre_id
   bar$splitvar<-"over_all"

   bar<-bar[,c("splitvar",stratum,varsum,"codist")]
  }
      names(bar)<-c("splitvar",letters[1:length(stratum)],varsum,"codist")
      bar<-bar[,-pmatch("splitvar",names(bar))]

   if (bar[1,1]=="NV/NA") {
    bar$key<-paste(bar$b,".",bar$a,sep="")
    bar<-bar[order(bar$key,decreasing=TRUE),]
    bar<-bar[,-pmatch("key",names(bar))]
   }


    if (stratum[2]==disaggregation_by) {

    sums<-aggregate(bar$sum,by=list(b=bar$b),FUN=max)
    names(sums)<-c("b","sum2")
    bar[,pmatch("b",names(bar))]<-as.character(bar[,pmatch("b",names(bar))])
    assign("bar",bar,envir=.GlobalEnv)

schema<-as.data.frame(cbind(as.character(rep(lev[[1]],length(lev[[2]]))),as.character(
rep(lev[[2]],each=length(lev[[1]])))))
    names(schema)<-c("a","b")

    bar<-merge(bar,schema,by=c("a","b"),all.x=TRUE,all.y=TRUE)
    bar<-merge(bar,sums,by=c("b"),all.x=TRUE,all.y=TRUE)

    bar$n[is.na(bar$n)]<-0

    bar$sum[is.na(bar$sum)]<-bar$sum2
    bar<-bar[,-pmatch("sum2",names(bar))]
    bar$codist[is.na(bar$codist)]=centre_id
    bar<-bar[,c(2,1,3:dim(bar)[2])]
    dataall<-aggregate(bar$n,by=list(a=bar$a),sum)
    names(dataall)<-c("a","n")
    dataall$sum=sum(dataall$n)
    dataall$b="overall"
    dataall$codist=centre_id
    dataall<-dataall[,c(1,4,2,3,5)]
    bar<-rbind(bar,dataall)
    lev[[2]]<-c(lev[[2]],"overall")
   }
  BIRO_drawbars(tab=bar,
              # namevar=namevar,
               lev=lev,
               labelvar=namevar[1],
               beside=TRUE,
               number=FALSE,
               perc=TRUE,
               what="patients",
               namegraph=namegraph,
               dirgraph=dirgraph,
               cex=cex)



 }
 }
}
```

```
##############################################################################
#                                                                            #
# Project: BIRO-Project (Funded by European Commission 2005-2008)            #
# File: BIRO_overbox.r                                                       #
#                                                                            #
#
##############################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_overbox.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
# overbox
#
##############################################################################


##############################################################################
#  overbox
##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_overbox is the function that menage the overall output boxplot
#
# DEPENDENCIES
#
# biro_plots.r
#
##############################################################################
#PARAMETERS:
##############################################################################
# namefd                  = name of the table that you want to read
# namevar                 = name of the target variabile
# condition               = SQL where condition
# stratum                 = names of the stratum variable
# lev                     = levels of the variables
# notch          [FALSE]  = logical. if notch is TRUE a notch is drawn in
#                           each side of the boxes. If the notches of two
#                           plots do not overlap this is ï¿½strong evidenceï¿½ that
#                           the two means differ (Chambers et al. 1983 p. 62).
#                           See boxplot.stats for the calculations used.
# namegraph               = output file name
# dirgraph                = the path of output graphs directory
#
##############################################################################



overbox<-function(namefd,
                  namevar,
                  condition,
                  stratum,
                  lev,
                  notch,
                  dirgraph,
                  namegraph) {


 if (dbExistsTable(conn, namefd)) {
  if (condition=="")
   statement=paste("Select * from ",namefd, ";",sep="")
  if (nchar(condition)>0)
   statement=paste("Select * from ",namefd, "where ",condition,";",sep="")

  temp<-as.data.frame(dbGetQuery(conn, statement))

  write.csv(temp,file=paste(dirdataout,"/",namefd,".csv",sep=""))

  write(paste("Boxplot data saved into: [OUTDATA]","/i",namefd,".csv",sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("Boxplot data saved into: [OUTDATA]","/i",namefd,".csv",sep=""
),file=logfile,append=TRUE)
  }

  if (dim(temp)[1]>0) {
  targetvar<-temp[,pmatch(namevar,names(temp))]
  targetvarnum<-rep(NA,length(targetvar))

 for (i in 1:dim(temp)[1]) {
   targetvarnum[i]<-as.numeric(as.character(targetvar[i]))
  }
```

```
#   targetvarnum<-as.numeric(as.character(targetvarnum))

  #rebuilding
  #expands the data
  expdata<-rep(targetvarnum,as.numeric(as.character(temp$n)))


  for (i in 1:length(stratum)) {

v<-as.data.frame(rep(temp[,pmatch(stratum[i],names(temp))],as.numeric(as.character(tem
p$n))))
   expdata<-cbind(expdata,v)
  }
  names(expdata)<-c(namevar,stratum)


  unistra<-unique(expdata[,pmatch(stratum,names(expdata))])
  lev<-lev[unistra[order(unistra)]]


  #assign levels to categorical variable (stratum) and calculate how many boxes
  nlev<-1
  for (i in 1:length(stratum)) {
   levels(expdata[,pmatch(stratum[i],names(expdata))])<-lev
   nlev<-nlev*nlevels(expdata[,pmatch(stratum[i],names(expdata))])
  }

  #write("Preparing to do SVG file",file="")
  #if (nchar(logfile)>0) {
  # write("Preparing to do SVG file",file=logfile,append=TRUE)
  #}
  file = paste(dirgraph,"/","i",namegraph,"g4_4a.svg",sep="")
  CairoSVG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
  boxplot(formula=as.formula(paste(namevar,"~",paste(stratum,collapse="+"),sep=""
)),data=expdata,notch=notch,col=rainbow(nlev),outline=FALSE)
  dev.off()
  write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

  #write("Preparing to do JPEG file",file="")
  #if (nchar(logfile)>0) {
  # write("Preparing to do JPEG file",file=logfile,append=TRUE)
  #}
  file = paste(dirgraph,"/","i",namegraph,"g4_4a.jpeg",sep="")
  jpeg(file, width= 813.6, height= 348.48, units="px",pointsize = 9,bg = "transparent"
)
  boxplot(formula=as.formula(paste(namevar,"~",paste(stratum,collapse="+"),sep=""
)),data=expdata,notch=notch,col=rainbow(nlev),outline=FALSE)
  dev.off()
  write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

  #write("Preparing to do PNG file",file="")
  #if (nchar(logfile)>0) {
  # write("Preparing to do PNG file",file=logfile,append=TRUE)
  #}
  file = paste(dirgraph,"/","i",namegraph,"g4_4a.png",sep="")
  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
  boxplot(formula=as.formula(paste(namevar,"~",paste(stratum,collapse="+"),sep=""
)),data=expdata,notch=notch,col=rainbow(nlev),outline=FALSE)
  dev.off()
  write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

  #write("Preparing to do PDF file",file="")
  #if (nchar(logfile)>0) {
  # write("Preparing to do PDF file",file=logfile,append=TRUE)
  #}
  file = paste(dirgraph,"/","i",namegraph,"g4_4a.pdf",sep="")
  pdf(file, width = 11.33, height =4.84 , pointsize = 7, bg = "transparent")
  boxplot(formula=as.formula(paste(namevar,"~",paste(stratum,collapse="+"),sep=""
)),data=expdata,notch=notch,col=rainbow(nlev),outline=FALSE)
  dev.off()
  write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }
  }
 }
 }
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_overlines.r                                                         #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_overlines.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
# overlines
#
################################################################################


################################################################################
#  overlines
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_overlines is the function that menage the overall output trend
#
# DEPENDENCIES
#
# biro_plots.r
#
################################################################################
#PARAMETERS:
################################################################################
# namelines               = name of the table that you want to read
# namevar                 = name of the target variabile
# condition               = SQL where condition
# varsum                  = name of the variabiles to aggregate
# stratum                 = names of the stratum variable
# lev                     = levels of the variables
# descvar                 = description of the represented variable
# what                    = description of what in represented in graph (patient
episode...)
# namegraph               = output file name
# dirgraph                = the path of output graphs directory
#
################################################################################


#  nametab= #name of the tab that you want to read
#  condition=     #SQL where condition
#  varsum=    #name of the variabile that you want to sum
#  stratum= #names of the stratum variable
#  n=



overlines<-function(namelines,
                    namevar,
                    condition,
                    varsum,
                    stratum,
                    lev,
                    descvar,
                    what,
                    namegraph,
                    dirgraph) {

 if (dbExistsTable(conn, namelines)) {

  if (condition=="")
   statement=paste("Select * from ",namelines, ";",sep="")
  if (nchar(condition)>0)
   statement=paste("Select * from ",namelines, "where ",condition,";",sep="")

  temp<-as.data.frame(dbGetQuery(conn, statement))

  if (dim(temp)[1]>0) {
  y<-as.numeric(as.character(temp$y))
  n<-as.numeric(as.character(temp$n))

  tot<-y*n

  temp$statistics<-as.character(noquote(temp$stat))
```

```
  temp$codistitute<-as.character(noquote(temp$codist))
  for (i in 1:dim(temp)[1]) {
   temp$statistics[i]<-substr(temp$statistics[i],2,nchar(temp$statistics[i])-1)
   temp$codistitute[i]<-substr(temp$codistitute[i],2,nchar(temp$codistitute[i])-1)
  }


  y[temp$statistics=="mean"]<-tot[temp$statistics=="mean"]

  temp$y<-y
  temp$n<-n
  temp$stat=temp$statistics
  temp$codist=temp$codistitute
  temp<-temp[,-pmatch(c("statistics","codistitute"),names(temp))]


  rm(tot)
  rm(y)
  rm(n)

  for (i in 1:length(varsum)) {

temp[,pmatch(varsum[i],names(temp))]<-as.numeric(as.character(temp[,pmatch(varsum[i],n
ames(temp))]))
  }

  #calculate sd

  n1<-length(unique(temp$times))
  n2<-length(unique(temp$strata))

  f<-length(unique(temp$codist))

  dsd<-NULL
  if (f>1) {
   for (i in 1:n1) {
    for (j in 1:n2) {

     sdi<-as.numeric(as.character(temp$sd[temp$times==unique(temp$times)[i] &
temp$strata==unique(temp$strata)[j]]))

     vari=sdi^2
     medie<-as.numeric(as.character(temp$y[temp$times==unique(temp$times)[i] &
temp$strata==unique(temp$strata)[j]]))
     medie<-medie/as.numeric(as.character(temp$n[temp$times==unique(temp$times)[i] &
temp$strata==unique(temp$strata)[j]]))
     n<-as.numeric(as.character(temp$n[temp$times==unique(temp$times)[i] &
temp$strata==unique(temp$strata)[j]]))
     N<-sum(n)


thisvar<-(weighted.mean(vari,n)*(N-f)+sum(((medie-weighted.mean(medie,n))^2)%*%n))/(N-
1)


dsd<-as.data.frame(rbind(dsd,c(as.numeric(as.character(temp$times[i])),as.numeric(as.c
haracter(temp$strata[j])),sqrt(thisvar))))
     dsd<-as.data.frame(dsd)
     names(dsd)<-c("times","strata","sd")
    }
   }
  }
```

```
  if (f==1) {

   dsd<-temp[,c("times","strata","sd")]
   #names(dsd)<-c("times","strata","sd")
  }


  #aggregate


lines<-as.data.frame(aggregate(temp[,pmatch(varsum,names(temp))],as.list(temp[,c(strat
um)]),FUN="sum"))

  names(lines)<-c(stratum,varsum)

  lines<-merge(lines,dsd,by=c("times","strata"))
  for (i in 1:length(stratum)) {

lines[,pmatch(stratum[i],names(lines))]<-as.character(lines[,pmatch(stratum[i],names(l
ines))])
  }


  for (i in 1:length(varsum)) {

lines[,pmatch(varsum[i],names(lines))]<-as.numeric(as.character(lines[,pmatch(varsum[i
],names(lines))]))
  }
  lines<-as.data.frame(lines)
  lines$codist=as.vector(as.character(centre_id))
  lines$times<-as.vector(as.numeric(as.character(lines$times)))
  lines$strata<-as.vector(as.numeric(as.character(lines$strata)))
  lines$stat<-as.vector(as.character(lines$stat))
  lines$codist<-as.vector(as.character(lines$codist))

  lines$y[as.character(lines$stat)=="mean"
]<-as.numeric(as.character(lines$y))/as.numeric(as.character(lines$n))

  lines$strata<-as.factor(lines$strata)
  levels(lines$strata)<-lev


BIRO_plotlines(data=lines,lev=lev,descvar=descvar,namegraph=namegraph,dirgraph=dirgrap
h)
 }
 }
}
```

```
##############################################################################
#                                                                            #
# Project: BIRO-Project (Funded by European Commission 2005-2008)            #
# File: BIRO_overmean.r                                                       #
#                                                                            #
#
##############################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_overmean.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
# overmean
#
##############################################################################


##############################################################################
#  overmean
##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_overmean is the function that menage overall mean
#
# DEPENDENCIES
#
# biro_aggregate.r
#
##############################################################################
#PARAMETERS:
##############################################################################
# namemean            = name of the table that you want to read
# condition           = SQL where condition
# varsum              = name of the variabiles to aggregate
# stratum             = names of the stratum variable
# lev                 = levels of the variables
#
##############################################################################

#  nametab= #name of the tab that you want to read
#  condition=     #SQL where condition
#  varsum=    #name of the variabile that you want to sum
#  stratum= #names of the stratum variable
#  n=



overmean<-function(namemean,
                   namevar,
                   condition,
                   varsum,
                   stratum,
                   lev) {


  temp<-as.data.frame(dbReadTable(conn, namemean))
  for (i in 1:length(varsum)) {

temp[,pmatch(varsum[i],names(temp))]<-as.numeric(as.character(temp[,pmatch(varsum[i],n
ames(temp))]))
  }


temp<-as.data.frame(temp)


  mean<-as.data.frame(aggregate(temp[,pmatch(varsum,names(temp))],as.list(temp[,is
.na(pmatch(names(temp),"sex"))]),FUN="sum"))

  return(mean)

 }
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_overtab.r                                                         #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_overtab.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
# overtab
#
################################################################################


################################################################################
#  overtab
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_overtab is the function that aggregate data for output table
#
# DEPENDENCIES
#
# biro_plots.r
#
################################################################################
#PARAMETERS:
################################################################################
# nametab             = name of the table that you want to read
# dataframe           = data.frame loaded instead of postgres table name
# condition           = SQL where condition
# varsum              = name of the variabiles to aggregate
# stratum             = names of the stratum variable
# lev                 = levels of the variables
# number              = logical if you want numbers printed on the graphs
# beside              = a logical value. If FALSE the columns of height
# perc                = a logical value. If TRUE percentages are calculated
# what                = description of what in represented in graph (patient
episode...)
# namegraph           = output file name
# dirgraph            = the path of output graphs directory
# cex                 = expansion factor for words and number printed
#
################################################################################
#  nametab= #name of the tab that you want to read
#  condition=     #SQL where condition
#  varsum=    #name of the variabile that you want to sum
#  stratum= #names of the stratum variable
#  nvar=               = number of variables
# by                   = names of the variables
# n                    = numbers of levels of each variables
# lev                  = levels of the variables
# side                 = FALSE for orizontal TRUE for vertical orientation
# width                = width in the PDF report
# texfile              = path of the indicator tex file
# firstlab             = description of the outcome
# caption              = caption of the output table
# chisq   [NULL]       = a vector containing the name of variable to calculate
#                        Cochran-Mantel-Haenszel Chi-Squared Test for Count
#                        Data or Pearson Chi-Squared Test
# namevar              = descriptive names of variables
# codtex               = code of the temporary tex file
# codtab               = code of the temporary table
#
################################################################################
overtab<-function(nametab,
                  nvar,
                  condition,
                  varsum,
                  by,
                  n,
                  lev,
                  side,
                  width,
                  texfile,
                  firstlab,
                  caption,
                  chisq,
```

```
                namevar,
                codtex,
                codtab) {

 if (dbExistsTable(conn, nametab)) {

  if (condition=="")
   statement=paste("Select * from ",nametab, ";",sep="")
  if (nchar(condition)>0)
   statement=paste("Select * from ",nametab, "where ",condition,";",sep="")

  data<-as.data.frame(dbGetQuery(conn, statement))

  write.csv(data,file=paste(dirdataout,"/",nametab,".csv",sep=""))

  write(paste("Boxplot data saved into: [OUTDATA]","/i",nametab,".csv",sep=""),file=""
)
  if (nchar(logfile)>0) {
   write(paste("Boxplot data saved into: [OUTDATA]","/i",nametab,".csv",sep=""
),file=logfile,append=TRUE)
  }


   #write Appendix

   appvar<-report_list_id
   for (app in 1:length(appvar)) {
    if (appvar[app]=="centre_id") appvar[app]="id"
   }
   appendix<-as.data.frame(data[,pmatch(na.omit(appvar),names(data))])
   names(appendix)<-report_list_id
   oldappendix<-get(paste("appendix",ind,sep=""))
   oldappendix<-rbind(oldappendix,appendix)
   oldappendix<-unique(oldappendix)
   assign(paste("appendix",ind,sep=""),oldappendix,envir = .GlobalEnv)


  if (dim(data)[1]>0) {
  if (nvar==4) {
   values4<-unique(data[,pmatch(by[4],names(data))])
   for (z in 1:length(values4)) {
      #n[is.na(n)]<-1
      n[4]<-1

   temp<-data[data[,pmatch(by[4],names(data))]==values4[z],]


   by<-by[by %in% names(temp)]


   for (i in 1:length(na.omit(varsum))) {

temp[,pmatch(na.omit(varsum)[i],names(temp))]<-as.numeric(as.character(temp[,pmatch(na
.omit(varsum)[i],names(temp))]))
   }



tab<-aggregate(temp[,pmatch(na.omit(varsum),names(temp))],as.list(temp[,pmatch(na.omit
(by),names(temp))]),FUN="sum")

   names(tab)<-c(na.omit(by),varsum)
```

```
   for (i in 1:length(na.omit(by))) {

tab[,pmatch(na.omit(by)[i],names(tab))]<-as.character(tab[,pmatch(na.omit(by)[i],names
(tab))])
    }
    #names(tab)<-c(letters[1:length(na.omit(by))],varsum)

   # print(paste("z",z))
   # print(length(unique(tab[,pmatch("d",names(tab))])))
  if (z==length(values4)) {
   newcaption=caption
  }
  if (z<length(values4)) {
   newcaption=""
  }

# for (k in 1:length(unique(tab[,pmatch("d",names(tab))]))) {
  #values of the 4th categorical variable

  if (length(na.omit(n))==4) {
  values<-unique(tab[,pmatch(by[length(na.omit(by))],names(tab))])
  temptab<-tab[tab[,pmatch(by[length(na.omit(by))],names(tab))]==values4[z],]
  }

  if (length(na.omit(n))==3) {
  values<-unique(tab[,pmatch(by[length(na.omit(by))],names(tab))])
  temptab<-tab[tab[,pmatch(by[length(na.omit(by))],names(tab))]==values4[z],]
  }


  #newlev[[4]]<-as.character(values[1])

  n[4]<-1
  n<-na.omit(n)

  names(temptab)[pmatch(varsum,names(temptab))]<-"Freq"

  #var<-na.omit(var)[na.omit(var) %in% names(tab)]


  mychisq<-na.omit(chisq[1:3])

  if (lev[[1]][1]=="NV/NA") mychisq<-NULL

  BIRO_table_print(tab=temptab,
                   var=by,
                   n=na.omit(n),
                   FirstLab=firstlab,
                   lev=na.omit(lev),
                   texfile=texfile,
                   codtab=codtab,
                   dirtables=dirtables,
                   dircsv=dirdataout,
                   caption=caption,
                   side=side,
                   width=width,
                   ind=ind,
                   perc=TRUE,
                   chisq=mychisq,
                   codtex=codtex)


#  }
```

```
  }
   }

 else {
    temp<-data

      for (i in 1:length(varsum)) {

temp[,pmatch(varsum[i],names(temp))]<-as.numeric(as.character(temp[,pmatch(varsum[i],n
ames(temp))]))
      }




tab<-aggregate(temp[,pmatch(varsum,names(temp))],as.list(temp[,pmatch(by,names(temp))]
),FUN="sum")
   names(tab)<-c(by,varsum)
   for (i in 1:length(by)) {
    tab[,pmatch(by[i],names(tab))]<-as.character(tab[,pmatch(by[i],names(tab))])
    }

   names(tab)<-c(namevar,varsum)
   tab<-as.data.frame(tab)


  if (nvar<4) {


  names(tab)[pmatch(varsum,names(tab))]<-"Freq"

 newlev<-lev
 newlev[[length(lev)]]<-paste(na.omit(namevar[2]),collapse=" * ")



  for (y in 1:(dim(tab)[2]-1)) {
   for (k in 1:(dim(tab)[1])) {
    tab[k,y]<-paste(pmatch(tab[k,y],lev[[y]]),tab[k,y])
   }
  }

  for (y in 1:dim(tab)[2]) {
   names(tab)[y]<-paste(unlist(strsplit(as.character(names(tab)[y]),"_")),collapse=
"__")
  }

  formula<-paste("~`",names(tab)[1],"`",sep="")
  tab<-sort.data.frame(tab,by=as.formula(formula))

  for (y in 1:dim(tab)[2]) {
   names(tab)[y]<-paste(unlist(strsplit(as.character(names(tab)[y]),"__")),collapse="
")
  }

  levout<-unique(tab[,1])

  for (s in 1:length(levout)) {
   levout[s]<-substr(levout[s],3,nchar(levout[s]))
  }


var=names(tab)[1:(length(names(tab))-1)]
if (length(na.omit(var))==3) {
 newlev[[1]]<-levout
 newlev[[2]]<-lev[[2]]
 newlev[[3]]<-lev[[3]]
 newlev[[4]]<-paste(na.omit(namevar[2:3]),collapse=" * ")
 var[4]<-var[2]
 n[4]<-1
 n<-n[1:3]       #add
 var<-var[1:2]   #add
}

if (length(na.omit(var))==2) {
 newlev[[1]]<-levout
 newlev[[2]]<-lev[[2]]
 newlev[[3]]<-namevar[2]
 newlev[[4]]<-namevar[2]
 var[3]<-NA
 var[4]<-var[2]
 n[3]<-1
 n[4]<-1

}

tab$id<-centre_id
n[1]<-length(na.omit(unique(tab[,1])))
assign("engine","se",envir = .GlobalEnv)

names(newlev)<-var

 for (y in 1:dim(tab)[2]) {
  names(tab)[y]<-paste(unlist(strsplit(as.character(names(tab)[y]),"_")),collapse=
"__")
 }

 for (y in 1:length(var)) {
  var[y]<-paste(unlist(strsplit(as.character(var[y]),"_")),collapse="__")
  if (var[y]=="NA") var[y]<-NA
 }


 mychisq<-var[2:length(var)]

 if (newlev[[1]][2]=="NV/NA") mychisq<-NULL

 var<-na.omit(var)[na.omit(var) %in% names(tab)]



 BIRO_table_print(tab=tab,
                  var=var,
                  n=na.omit(n),
                  FirstLab=firstlab,
                  lev=newlev,
                  texfile=texfile,
                  codtab=codtab,
                  dirtables=dirtables,
                  dircsv=dirdataout,
                  caption=caption,
                  side=side,
```

```
                width=width,
                ind=ind,
                perc=TRUE,
                chisq=mychisq,
                codtex=codtex)

  }
  }
  }
 }

}
```

```
##############################################################################
#                                                                            #
# Project: BIRO-Project (Funded by European Commission 2005-2008)            #
# File: BIRO_overtrellis.r                                                   #
#                                                                            #
#
##############################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_overtrellis.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
# overtrellis
#
##############################################################################


##############################################################################
#  overtrellis
##############################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_overtrellis is the function that aggregate data for output trellis plot
#
##############################################################################
#PARAMETERS:
##############################################################################
# namefd              = name of the table that you want to read
# namevar             = name of the target variabile
# condition           = SQL where condition
# stratum             = names of the stratum variable
# lev                 = levels of the variables
# labvar              = description of the target variabile
# notch          [FALSE]  = logical. if notch is TRUE a notch is drawn in
#                           each side of the boxes. If the notches of two
#                           plots do not overlap this is ï¿½strong evidenceï¿½ that
#                           the two means differ (Chambers et al. 1983 p. 62).
#                           See boxplot.stats for the calculations used.
# namegraph           = output file name
# dirgraph            = the path of output graphs directory
# cex                 = expansion factor for words and number printed
# hist                = 1 for trellis histgrams, 0 otherwise
# box                 = 1 for trellis histgrams, 0 otherwise
#
##############################################################################



overtrellis<-function(namefd,
                      namevar,
                      condition,
                      stratum,
                      lev,
                      labvar,
                      notch,
                      dirgraph,
                      namegraph,
                      hist=0,
                      box=0) {

 if (dbExistsTable(conn, namefd)) {

  if (condition=="")
   statement=paste("Select * from ",namefd, ";",sep="")
  if (nchar(condition)>0)
   statement=paste("Select * from ",namefd, "where ",condition,";",sep="")

  temp<-as.data.frame(dbGetQuery(conn, statement))

  if (dim(temp)[1]>0) {

  targetvar<-temp[,pmatch(namevar,names(temp))]
  targetvarnum<-rep(NA,length(targetvar))

#  for (i in 1:dim(temp)[1]) {
#
targetvarnum[i]<-as.numeric(as.character(substr(as.character(targetvar[i]),2,nchar(as.
character(targetvar[i]))-1)))
#  }

   targetvarnum<-as.numeric(as.character(targetvar))
```

```
  #rebuilding

  #expands the data
  expdata<-rep(targetvarnum,as.numeric(as.character(temp$n)))


  for (i in 1:length(stratum)) {

v<-as.data.frame(rep(temp[,pmatch(stratum[i],names(temp))],as.numeric(as.character(tem
p$n))))
   expdata<-cbind(expdata,v)
  }

  names(expdata)<-c(namevar,stratum)


   if (hist==1) {
  if (is.null(stratum)==FALSE) {


   for (i in 1:length(stratum)) {
    expdata[,pmatch(stratum[i],names(expdata))]<-
    as.factor(expdata[,pmatch(stratum[i],names(expdata))])
    levels(expdata[,pmatch(stratum[i],names(expdata))])=lev[[i]]
   }


  formula<-paste(namevar,"~",paste(stratum,collapse="+"),sep="")

  n<-NULL
  for (i in 1:length(lev)) {
   n<-c(n,length(lev[[i]]))
  }
   firstlay<-prod(n)/4

   formula<-paste("~ ",namevar," | ",paste(stratum[-3],collapse="+"),sep="")

  # write("preparing to do trellis function histogram",file="")


  for (k in 1:length(unique(expdata[,pmatch(stratum[3],names(expdata))]))) {
   # values of the 4th categorical variable
   values<-unique(expdata[,pmatch(stratum[3],names(expdata))])

   if (stratum[3]=="type_dm") {
     allvalues=c("Type 1", "Type 2" ,"Other Type")
     values=c("Other Type","Type 2","Type 1")
     values<-allvalues[sort(na.omit(pmatch(allvalues,values)))]

   }
   tempdata<-expdata[expdata[,pmatch(stratum[3],names(expdata))]==values[k],]
   targetprint<-paste(unlist(strsplit(values[k]," ")),collapse="_")

   if (dim(tempdata)[1]>0) {
   trellis<-histogram(as.formula(as.character(formula)),
                      data=tempdata,
                      #layout=c(firstlay,4),
                      xlab=labvar,
                      type = "density",
                      panel = function(x, ...) {
                        panel.histogram(x, ...)
                        panel.mathdensity(dmath = dnorm, col = "black",
                        args = list(mean=mean(x),sd=sd(x)))
                      } )

   #write("function histogram done",file="")
   progbar=progbar+1

   file = paste(dirgraph,"/",namegraph,"_",progbar,"g4_2a.svg",sep="")
   CairoSVG(file, width = 11.33, height =11.33, pointsize=9, bg = "transparent")
   plot(trellis)
   dev.off()
  write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

   file = paste(dirgraph,"/",namegraph,"_",progbar,"g4_2a.jpeg",sep="")
   jpeg(file, width= 813.6, height= 813.6, pointsize=9, units="px",
        bg = "transparent")
   plot(trellis)
   dev.off()
  write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

   file = paste(dirgraph,"/",namegraph,"_",progbar,"g4_2a.png",sep="")
   CairoPNG(file, width = 480, height = 480,
         pointsize = 9, bg = "transparent")
   plot(trellis)
   dev.off()
  write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

   file = paste(dirgraph,"/",namegraph,"_",progbar,"g4_2a.pdf",sep="")
   pdf(file, width = 11.33, height =11.33 , pointsize = 9, bg = "transparent")
   plot(trellis)
   dev.off()
  write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
  if (nchar(logfile)>0) {
   write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
  }

  #write("trellis histogram saved",file="")
    }
   }
  }
  if (is.null(stratum)==TRUE) {

   #write("preparing to do trellis function histogram",file="")

   histogram(data[,pmatch(namevar,names(expdata))],
             xlab=labvar,
             type = "density",
             panel = function(x, ...) {
                panel.histogram(x, ...)
```

```
              panel.mathdensity(dmath = dnorm, col = "black",
              args = list(mean=mean(x),sd=sd(x)))
            } )

   #write("function histogram done",file="")
  }
    }
  if (box==1) {
  if (is.null(stratum)==FALSE) {

   for (i in 1:length(stratum)) {
    expdata[,pmatch(stratum[i],names(expdata))]<-
    as.factor(expdata[,pmatch(stratum[i],names(expdata))])
    levels(expdata[,pmatch(stratum[i],names(expdata))])=lev[[i]]
   }


  formula<-paste(namevar,"~",paste(stratum,collapse="+"),sep="")

  n<-NULL

  for (i in 1:length(lev)) {
   n<-c(n,length(lev[[i]]))

  }

   firstlay<-prod(n)/4

  for (k in 1:length(unique(expdata[,pmatch(stratum[3],names(expdata))]))) {
   # values of the 4th categorical variable
   values<-unique(expdata[,pmatch(stratum[3],names(expdata))])

   if (stratum[3]=="type_dm") {
     allvalues=c("Type 1", "Type 2" ,"Other Type")
     values=c("Other Type","Type 2","Type 1")
     values<-allvalues[sort(na.omit(pmatch(allvalues,values)))]

   }
   tempdata<-expdata[expdata[,pmatch(stratum[3],names(expdata))]==values[k],]
   targetprint<-paste(unlist(strsplit(values[k]," ")),collapse="_")
   if (dim(tempdata)[1]>0) {
   formula<-paste("~ ",namevar," | ",paste(stratum[-3],collapse="+"),sep="")
   #write("preparing to do trellis function boxplot",file="")

   trellis<-bwplot(as.formula(as.character(formula)),data=tempdata)
   #write("function boxplot done",file="")
   progbox=progbox+1

   file = paste(dirgraph,"/",namegraph,"_",progbox,"g4_2a.svg",sep="")
   CairoSVG(file, width = 11.33, height =11.33, pointsize=9, bg = "transparent")
   plot(trellis)
   dev.off()
   write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
   if (nchar(logfile)>0) {
    write(paste("SVG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
   }

   file = paste(dirgraph,"/",namegraph,"_",progbox,"g4_2a.jpeg",sep="")
   jpeg(file, width= 813.6, height= 813.6, pointsize=9, units="px",
        bg = "transparent")
   plot(trellis)
   dev.off()
   write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
   if (nchar(logfile)>0) {
    write(paste("JPEG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
   }

   file = paste(dirgraph,"/",namegraph,"_",progbox,"g4_2a.png",sep="")
   CairoPNG(file, width = 480, height = 480,
         pointsize = 9, bg = "transparent")
   plot(trellis)
   dev.off()
   write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
   if (nchar(logfile)>0) {
    write(paste("PNG file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
   }

   file = paste(dirgraph,"/",namegraph,"_",progbox,"g4_2a.pdf",sep="")
   pdf(file, width = 11.33, height =11.33 , pointsize = 9, bg = "transparent")
   plot(trellis)
   dev.off()
   write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""),file="")
   if (nchar(logfile)>0) {
    write(paste("PDF file: [OUTDIR]/graphs/",basename(file),sep=""
),file=logfile,append=TRUE)
   }

   #write("trellis boxplot saved",file="")
    }
   }
  }
  }
#  if (is.null(stratum)==TRUE) {
#
#   write("preparing to do trellis function boxplot",file="")
#
#   histogram(data[,pmatch(namevar,names(expdata))],
#             xlab=labvar,
#             type = "density",
#             panel = function(x, ...) {
#                panel.histogram(x, ...)
#                panel.mathdensity(dmath = dnorm, col = "black",
#                args = list(mean=mean(x),sd=sd(x)))
#             } )
#
#   write("function boxplot done",file="")
#  }
  }
  }
}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_overufd.r                                                         #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_overufd.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
# overufd
#
################################################################################


################################################################################
#  overufd
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_overufd is the function that menage overall ufd
#
# DEPENDENCIES
#
# biro_aggregate.r
#
################################################################################
#PARAMETERS:
################################################################################
# nametab             = name of the table that you want to read
# condition           = SQL where condition
# varsum              = name of the variabiles to aggregate
# stratum             = names of the stratum variable
# lev                 = levels of the variables
#
################################################################################



overufd<-function(nametab,condition,varsum,stratum,n,lev) {


  temp<-as.data.frame(dbReadTable(conn, nametab))
  for (i in 1:length(varsum)) {

temp[,pmatch(varsum[i],names(temp))]<-as.numeric(as.character(temp[,pmatch(varsum[i],n
ames(temp))]))
  }


  tab<-aggregate(temp[,pmatch(varsum,names(temp))],as.list(temp[,c(stratum)]),FUN=
"sum")

  names(tab)<-c(stratum,varsum)

  for (i in 1:length(stratum)) {

tab[,pmatch(stratum[i],names(tab))]<-as.character(tab[,pmatch(stratum[i],names(tab))])

  }


  BIROprinttab(tab=tab,nvar=2,n=n,FirstLab="Age",lev=lev,texfile=texfile,codtab="1_1"
,dirhtml=dirhtml,dircsv=dircsv,
               caption="Age (by Gender)",side=0,width=0.5,perc=TRUE)

 }
```

```
#  nametab= #name of the tab that you want to read
#  condition=     #SQL where condition
#  varsum=    #name of the variabile that you want to sum
#  stratum= #names of the stratum variable
#  n=
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_overvar.r                                                         #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_overvar.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
# overvar
#
################################################################################


################################################################################
#  overvar
################################################################################
# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2007-11-24
# Version: 2008-11-21
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# DESCRIPTION
#
# BIRO_overvar is the function that menage overall variance
#
# DEPENDENCIES
#
# biro_aggregate.r
#
################################################################################
#PARAMETERS:
################################################################################
# namevar             = name of the table that you want to read
# condition           = SQL where condition
# varsum              = name of the variabiles to aggregate
# stratum             = names of the stratum variable
# lev                 = levels of the variables
#
################################################################################



overvar<-function(namebar,namevar,condition,varsum,stratum,lev) {


  tab<-as.data.frame(dbReadTable(conn, namebar))
  for (i in 1:length(varsum)) {

tab[,pmatch(varsum[i],names(tab))]<-as.numeric(as.character(tab[,pmatch(varsum[i],name
s(tab))]))
  }



  for (i in 1:length(unique(tab$variable))) {

   temp<-tab[tab$variable==unique(tab$variable)[i],]


  }


(weighted.var(temp$var,temp$n)*(sum(temp$n)-length(unique(f)))+sum(((medie-weighted.va
r(medie,n))^2)%*%n))/(length(x)-1)



(weighted.var(vari,n)*(length(x)-length(unique(f)))+sum(((medie-weighted.var(medie,n))
^2)%*%n))/(length(x)-1)

}
```

```
################################################################################
#                                                                              #
# Project: BIRO-Project (Funded by European Commission 2005-2008)              #
# File: BIRO_ce_recode.r                                                         #
#                                                                              #
#
################################################################################

# ------------------------
# Authors:
# Luca Rossi       <redsluke@gmail.com>
# Fabrizio Carinci <research@fabcarinci.net>
# Institutions: Serectrix snc
# Created: 2007-11-21
# Version: 2010-10-31
# OS: Windows XP/Vista, Fedora Core 13
# Programming Language: R 2.8.1
# ------------------------
# COPYRIGHT INFORMATION
#
# This file is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This file is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this file; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
# In short: you may use this file any way you like, as long as you
# don't charge money for it, remove this notice, or hold anyone liable
# for its results.
#
# BIRO_ce_recode.r is part of WP Central Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
################################################################################

  #Thresholds

  ageth<-c(18,35,55,75)
  ageth5<-seq(5,85,5)
  ageth10<-c(10,20,30,40,50,60,70,80)
  durdiabth<-c(10,20)
  weightth<-c(50,70,90,110,130)
  bmith<-c(25,27,30,40)
  sbpth<-c(130,160)
  dbpth<-c(70,100,120)
  choith<-c(2.59,5.18,7.77)
  hdlth<-c(1.295)
  ldlth<-c(2.59,5.18)
  creatth<-c(50,100,150)
  hba1cth<-c(7,8)
  agethst<-c(15,25,35,45,55,65,75,85)

  #Limits

  bmilimit<-c(4,44)
  weightlimit<-c(25,200)
  heightlimit<-c(60,250)
  chollimit<-c(70,400)
  hdllimit<-c(10,100)
  ldllimit<-c(30,300)
  creatlimit<-c(0.1,10)
  hba1climit<-c(2.5,20)
  ma_testlimit<-c(0,300)
  dbplimit<-c(30,150)
  sbplimit<-c(60,250)
  trigllimit<-c(10,1000)
  trigl12limit<-c(10,1000)

  #Levels of Categorical Variables

  levhba1c9<-list("Yes"=1,"No"=2)
  levhba1c75<-list("Yes"=1,"No"=2)
  levtype_dm<-list("Type 1"="1","Type 2"="2","Other Type"="3")
  levtype_dmst<-list("Type 1"="1","Type 2"="2")
  levsex<-list("Male"=1,"Female"=2)
  levagesex<-list("Male"=0,"0 - 14"=1,"15 - 24"=2,"25 - 34"=3,"35 - 44"=4,"45 - 54"=5,
"55 - 64"=6,"65 - 74"=7,"75 - 84"=8,"85 +"=9)
  levma_test<-list("No MA Test Recorded"=0,"MA Test Normal"=1,"MA Test Abnormal"=2)
  levsmok_stat<-list("Current smoker"=1,"Non-Smoker"=2,"Ex-Smoker"=3)
  levretin<-list("No Retinopathy"=1,"Background Retinopathy"=2,"Referable Retinopathy"
=3)
  levmacul<-list("No Maculopathy"=1,"Maculopathy"=2)
  levfoot_exam<-list("No"=0,"Yes"=1)
  levesrf<-list("Yes"=1,"No"=2)
  levulcer<-list("Yes"=1,"No"=2)
  levamput<-list("Yes"=1,"No"=2)
  levstroke<-list("Yes"=1,"No"=2)
  levmi<-list("Yes"=1,"No"=2)
  levhypertension<-list("Yes"=1,"No"=2)
  leveye_exam<-list("No"=0,"Yes"=1)
  levbp<-list("Yes"=1,"No"=2)
  levma_testdone<-list("Recorded"=1,"No Recorded"=2)
  levhba1c_done=list("Yes"=1,"No"=2)
  levma_done=list("Yes"=1,"No"=2)
  levfoot_done=list("Yes"=1,"No"=2)
  leveye_done=list("Yes"=1,"No"=2)
  levsmoke_doc=list("Documented"=1,"Not Documented"=2)
  levcreat_done=list("Yes"=1,"No"=2)
  levlipids=list("Yes"=1,"No"=2)
  levhypert_med=list("Yes"=1,"No"=2)
  levpump<-list("Yes"=1,"No"=2)
  levdiet<-list("Yes"=1,"No"=2)
  levtablet<-list("Yes"=1,"No"=2)
  levinsulin<-list("Yes"=1,"No"=2)
  levinstab<-list("Yes"=1,"No"=2)
  levpump_med<-list("Yes"=1,"No"=2)
  levbp12<-list("Yes"=1,"No"=2)
  levoral_th12<-list("Sulphonylureas"=1,"Biguanides"=2,"Glucosidase Inhib."=3,
"Glitazones"=4,"Glinides"=5)
  levinsulin12<-list("Treated"=1,"Not Treated"=2)
  levinsOAD<-list("Treated"=1,"Not Treated"=2)
  levinspump<-list("Yes"=1,"No"=2)
```

```
  levhypert12<-list("Yes"=1,"No"=2)
  levhba1c_c9<-list("Yes"=1,"No"=2)
  levcurrentsmoker<-list("Yes"=1,"No"=2)
  levlipid_med<-list("Yes"=1,"No"=2)
  levasa_med<-list("Yes"=1,"No"=2)
  levself_mon<-list("Urine"=1,"Blood Glucose"=2,"Both"=3)
  levvisit_freq<-list("Only one"=1,"More than one"=2)
  levhypert<-list("Yes"=1,"No"=2)
  levbp_done<-list("Yes"=1,"No"=2)
  levdiatrans<-list("Yes"=1,"No"=2)

# tests

  levhba1c_test<-list("at least one test"=1,"no test"=2)
  levma_test<-list("at least one test"=1,"no test"=2)
```

Table of Contents