```
rm(list=ls())

source("/home/fabrizio/biro/software/statengine010409/biro/software/_se_/source/r/main/biro_se_.r")

BIRO_se(dirse="/home/fabrizio/biro/software/statengine010409/biro/software/_se_",
        dbformat="postgres", # "postgres" if data are stored in a postgres database, "csv" if data are stored in csv files
        driverClass="org.postgresql.Driver",
        classPath="/home/fabrizio/biro/software/statengine010409/biro/software/lib/db/postgresql-8.2-504.jdbc3.jar",
        identifier.quote="'",
        pathdb="jdbc:postgresql://localhost/rrdm_centre1",
        user="postgres",
        password="postgres",
        dbname="rrdm_centre1",
        dirdatastore="",
        centre_id="rrdmcentre1",
        startdate="01/01/2002", #Bergen meeting this must be a date (the start date interval for data analysis)
        enddate="31/12/2002",   #Bergen meeting include a final date interval for data analysis
        yearnow=2009,
        refanadate="31/12",
        logfile="rrdmcentre1.log",
        cex=1.5,
        wide=1)

BIRO_se(dirse="/home/fabrizio/biro/software/statengine010409/biro/software/_se_",
        dbformat="postgres", # "postgres" if data are stored in a postgres database, "csv" if data are stored in csv files
        driverClass="org.postgresql.Driver",
        classPath="/home/fabrizio/biro/software/statengine010409/biro/software/lib/db/postgresql-8.2-504.jdbc3.jar",
        identifier.quote="'",
        pathdb="jdbc:postgresql://localhost/rrdm_centre2",
        user="postgres",
        password="postgres",
        dbname="rrdm_centre2",
        dirdatastore="",
        centre_id="rrdmcentre2",
        startdate="01/01/2002", #Bergen meeting this must be a date (the start date interval for data analysis)
        enddate="31/12/2002",   #Bergen meeting include a final date interval for data analysis
        yearnow=2009,
        refanadate="31/12",
        logfile="rrdmcentre2.log",
        cex=1.5,
        wide=1)
```

```
   42:
   43: BIRO_se(dirse="/home/fabrizio/biro/software/statengine010409/biro/software/_se_",
   44:         dbformat="postgres", # "postgres" if data are stored in a postgres database, "csv" if data are stored in
csv files
   45:         driverClass="org.postgresql.Driver",
   46:         classPath="/home/fabrizio/biro/software/statengine010409/biro/software/lib/db/postgresql-8.2-504.jdbc3.jar"
,
   47:         identifier.quote="'",
   48:         pathdb="jdbc:postgresql://localhost/rrdm_centre3",
   49:         user="postgres",
   50:         password="postgres",
   51:         dbname="rrdm_centre3",
   52:         dirdatastore="",
   53:         centre_id="rrdmcentre3",
   54:         startdate="01/01/2002", #Bergen meeting this must be a date (the start date interval for data analysis)
   55:         enddate="31/12/2002",   #Bergen meeting include a final date interval for data analysis
   56:         yearnow=2009,
   57:         refanadate="31/12",
   58:         logfile="rrdmcentre3.log",
   59:         cex=1.5,
   60:         wide=1)
   61:
   62: BIRO_se(dirse="/home/fabrizio/biro/software/statengine010409/biro/software/_se_",
   63:         dbformat="postgres", # "postgres" if data are stored in a postgres database, "csv" if data are stored in
csv files
   64:         driverClass="org.postgresql.Driver",
   65:         classPath="/home/fabrizio/biro/software/statengine010409/biro/software/lib/db/postgresql-8.2-504.jdbc3.jar"
,
   66:         identifier.quote="'",
   67:         pathdb="jdbc:postgresql://localhost/rrdm_centre4",
   68:         user="postgres",
   69:         password="postgres",
   70:         dbname="rrdm_centre4",
   71:         dirdatastore="",
   72:         centre_id="rrdmcentre4",
   73:         startdate="01/01/2002", #Bergen meeting this must be a date (the start date interval for data analysis)
   74:         enddate="31/12/2002",   #Bergen meeting include a final date interval for data analysis
   75:         yearnow=2009,
   76:         refanadate="31/12",
   77:         logfile="rrdmcentre4.log",
   78:         cex=1.5,
   79:         wide=1)
   80:
   81: BIRO_se(dirse="/home/fabrizio/biro/software/statengine010409/biro/software/_se_",
   82:         dbformat="postgres", # "postgres" if data are stored in a postgres database, "csv" if data are stored in
```

```
csv files
   83:         driverClass="org.postgresql.Driver",
   84:         classPath="/home/fabrizio/biro/software/statengine010409/biro/software/lib/db/postgresql-8.2-504.jdbc3.jar"
,
   85:         identifier.quote="'",
   86:         pathdb="jdbc:postgresql://localhost/rrdm_centre5",
   87:         user="postgres",
   88:         password="postgres",
   89:         dbname="rrdm_centre5",
   90:         dirdatastore="",
   91:         centre_id="rrdmcentre5",
   92:         startdate="01/01/2002", #Bergen meeting this must be a date (the start date interval for data analysis)
   93:         enddate="31/12/2002",   #Bergen meeting include a final date interval for data analysis
   94:         yearnow=2009,
   95:         refanadate="31/12",
   96:         logfile="rrdmcentre5.log",
   97:         cex=1.5,
   98:         wide=1)
```

```
    1: ##############################################################################
    2: #                                                                            #
    3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
    4: # File: biro_se_datastep.r                                                   #
    5: #                                                                            #
#
    6: ##############################################################################
    7:
    8: # ------------------------
    9: # Authors:
   10: # Luca Rossi       <redsluke@gmail.com>
   11: # Fabrizio Carinci <research@fabcarinci.net>
   12: # Institutions: Serectrix snc
   13: # Created: 2007-11-21
   14: # Version: 2008-10-23
   15: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
   16: # Programming Language: R 2.8.0
   17: # ------------------------
   18: # COPYRIGHT INFORMATION
   19: #
   20: # This file is free software; you can redistribute it and/or modify
   21: # it under the terms of the GNU General Public License as published by
   22: # the Free Software Foundation; either version 2, or (at your option)
   23: # any later version.
   24: #
   25: # This file is distributed in the hope that it will be useful,
   26: # but WITHOUT ANY WARRANTY; without even the implied warranty of
   27: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   28: # GNU General Public License for more details.
   29: #
   30: # You should have received a copy of the GNU General Public License
   31: # along with this file; see the file COPYING.  If not, write to
   32: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
   33: #
   34: # In short: you may use this file any way you like, as long as you
   35: # don't charge money for it, remove this notice, or hold anyone liable
   36: # for its results.
   37: #
   38: # biro_se_datastep.r is part of WP Statistical Engine of the BIRO Project
   39: # GPL Copyright, The BIRO Project
   40: #
   41: # ------------------------
   42: #
   43: # CONTENT
   44: #
```

```
45: #   BIRO_data_format
46: #   BIRO_loaddata
47: #
48: ##############################################################################
49:
50:
51: ##############################################################################
52: #  BIRO_data_format
53: ##############################################################################
54: # ------------------------
55: # Authors:
56: # Luca Rossi       <redsluke@gmail.com>
57: # Fabrizio Carinci <research@fabcarinci.net>
58: # Institutions: Serectrix snc
59: # Created: 2007-11-24
60: # Version: 2008-11-21
61: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
62: # Programming Language: R 2.8.0
63: # ------------------------
64: # DESCRIPTION
65: #
66: # BIRO_data_format is used to format variables in a dataset
67: #
68: # DEPENDENCIES
69: #
70: # biro_se_setup.r
71: #
72: ##############################################################################
73: #PARAMETERS:
74: ##data [episode] = tareget dataset
75: #n_var [NULL]  = a vector containing the names of numeric variables
76: #c_var [NULL]  = a vector containing the names of character variables
77: #d_var [NULL]  = a vector containing the names of date variables storing as
78: #                 string
79: #f_var [NULL]  = a vector containing the names of factor variables
80: #datefmt  !    = input date format
81: ##############################################################################
82:
83:
84: BIRO_data_format<-function(data=episode,
85:                            n_var=NULL,
86:                            c_var=NULL,
87:                            d_var=NULL,
88:                            f_var=NULL,
89:                            datefmt) {
```

```
 90:
 91:
 92:
 93:
 94:  #convert into numeric variables
 95:  if (is.null(n_var)==FALSE) {
 96:   n_var<-na.omit(n_var)
 97:   if (length(n_var)>0) {
 98:    for (i in 1:length(n_var)) {
 99:     data[,pmatch(n_var[i],names(data))]<-
100:     as.numeric(as.character(data[,pmatch(n_var[i],names(data))]))
101:    }
102:   }
103:  }
104:  #convert into string variables
105:  if (is.null(c_var)==FALSE) {
106:   c_var<-na.omit(c_var)
107:   if (length(c_var)>0) {
108:    for (i in 1:length(c_var)) {
109:     data[,pmatch(c_var[i],names(data))]<-
110:     as.character(data[,pmatch(c_var[i],names(data))])
111:    }
112:   }
113:  }
114:  #convert into date variables
115:  if (is.null(d_var)==FALSE) {
116:   d_var<-na.omit(d_var)
117:   if (length(d_var)>0) {
118:    for (i in 1:length(d_var)) {
119:     data[,pmatch(d_var[i],names(data))]<-
120:     as.Date(as.character(data[,pmatch(d_var[i],names(data))]),format=datefmt)
121:    }
122:   }
123:  }
124:  #convert into categorical variables
125:  if (is.null(d_var)==FALSE) {
126:   f_var<-na.omit(f_var)
127:   if (length(f_var)>0) {
128:    for (i in 1:length(f_var)) {
129:     data[,pmatch(f_var[i],names(data))]<-
130:      as.factor(as.character(data[,pmatch(f_var[i],names(data))]))
131:    }
132:   }
133:  }
134:  return(data)
```

```
135: }
136:
137:
138: ##############################################################################
139: #  BIRO_loaddata
140: ##############################################################################
141: # ------------------------
142: # Authors:
143: # Luca Rossi       <redsluke@gmail.com>
144: # Fabrizio Carinci <research@fabcarinci.net>
145: # Institutions: Serectrix snc
146: # Created: 2007-11-21
147: # Version: 2008-10-23
148: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
149: # Programming Language: R 2.8.0
150: #-------------------------
151: # DESCRIPTION
152: #
153: # BIRO_loaddata load data from a db or a csv file
154: #
155: # DEPENDENCIES
156: #
157: ##############################################################################
158: #PARAMETERS:
159: #dbformat        = the format of the database ("postgres","csv")
160: #drv             = the JDBC driver to connect R to the database
161: #                  (if not dbformat="csv")
162: #pathdb          = the path of the database
163: #user            = a string containing the username
164: #password        = a string containing the password
165: #dbname          = a string containing the database name
166: #dirdatastore    = a string containing the path of the input csv files
167: #centre_id       = a string containing the centre id
168: #yearnow         = a numeric value containing the last possible year of the
169: #                  episode dates
170: #fromyear        = a numeric value containing the first possible year where
171: #                  data are ok
172: #fromyearl       = a numeric value containing the first possible year where
173: #                  data are ok for the trends
174: #refyear         = a string containing year to calculate the indicators with
175: #                  "in the last 12 months" suffix
176: #refdate         = a string date to calculare age in the statistics about the
177: #                  refyear interval
178: ##############################################################################
179:
```

```
180:
181: BIRO_loaddata<-function(dbformat,
182:                         drv,
183:                         pathdb,
184:                         user,
185:                         password,
186:                         dbname,
187:                         dirdatastore,
188:                         centre_id,
189:                         yearnow,
190:                         fromyear,
191:                         fromyearl,
192:                         startdate,
193:                         enddate,
194:                         refdate) {
195:
196:  if (dbformat=="postgres") {
197:   assign("conn",dbConnect(drv,pathdb,user,password,dbname),envir=.GlobalEnv)
198:
199:   # Gets data from PGSQL and creates 4 csv s with demographic,
200:   # clinical, mortality, population and site data
201:
202:   if (wide==0) {
203:   ## Import data from Postgres
204:
205:    x <- "SELECT DISTINCT profile_field_name from profile"
206:
207:    patient<-as.data.frame(dbGetQuery(conn, x))
208:
209:
210:    tabpatvar<-as.data.frame(dbGetQuery(conn, x))
211:    patvar<-tabpatvar$profile_field_name
212:    rm(tabpatvar)
213:
214:   x <- (dbSendUpdate(conn, "DROP TABLE IF EXISTS t1"))
215:   x <- (dbSendUpdate(conn,
216:                      "CREATE TABLE t1 AS Select DISTINCT patient_id FROM profile"))
217:   print("First query executed")
218:
219:   ## Create index
220:   x <- paste("create index patient_id ON t1 (patient_id)")
221:   dbSendUpdate(conn, x)
222:   for (i in 1:length(patvar)) {
223:    x <- paste("CREATE TEMPORARY TABLE ", patvar[i], " AS ",
224:               "Select DISTINCT profile.profile_field_value AS ",patvar[i],
```

```
225:              ",profile.patient_id FROM profile WHERE profile_field_name='",
226:              patvar[i],"'",sep="")
227:    dbSendUpdate(conn, x)
228:
229:    x <- paste("create index ", patvar[i], "patient_id ON ", patvar[i], " (patient_id) ", sep = "")
230:    print(paste("About to create index: ", x))
231:    dbSendUpdate(conn, x)
232:    print(paste(i, "th index created",sep=""))
233:
234:   }
235:
236:   x <- (dbSendUpdate(conn, "DROP TABLE IF EXISTS xxx"))
237:   select.part <- NULL
238:   join.part <- NULL
239:   commavar=","
240:   for (i in 1:length(patvar)) {
241:    if (i == length(patvar)) {
242:     commavar=""
243:    }
244:    select.part <- paste(select.part, patvar[i], ".", patvar[i], commavar, sep = "" )
245:    join.part <- paste(join.part, "LEFT JOIN ", patvar[i], " ON ",
246:                       patvar[i], ".patient_id = ", "t1.patient_id ",
247:                       sep = "" )
248:   }
249:   full.query <- paste("CREATE TABLE xxx AS select t1.patient_id, ", select.part,
250:                       "from t1 ",
251:                       join.part)
252:   dbSendUpdate(conn, full.query)
253:
254:   x<-"Select * FROM xxx"
255:
256:   res<-dbSendQuery(conn,x)
257:   patient<-NULL
258:   nofinished<-TRUE
259:   i=0
260:   while(nofinished) {
261:     i=i+1
262:     temp<-as.data.frame(fetch(res,400))
263:     if ((dim(temp)[1]>0) & i==1) {
264:      patient<-temp
265:     }
266:     if ((dim(temp)[1]>0) & i>1) {
267:      patient<-rbind(patient,temp)
268:     }
269:     if (dim(temp)[1]==0) {
```

```
270:       nofinished<-FALSE
271:      }
272:    }
273:    rm(temp)
274:
275:    x <- (dbSendUpdate(conn, "DROP TABLE IF EXISTS xxx"))
276:    x <- "SELECT DISTINCT episode_field_name from data"
277:
278:    tabepivar<-as.data.frame(dbGetQuery(conn, x))
279:    epivar<-tabepivar$episode_field_name
280:    rm(tabepivar)
281:
282: ## Create temporary tables for each variable
283:    for (i in 1:length(epivar)) {
284:     x <- paste("CREATE TEMPORARY TABLE ", epivar[i], " AS ",
285:                "Select DISTINCT data.episode_field_value AS ",epivar[i],
286:                ",data.episode_data_id FROM data WHERE episode_field_name='",
287:                epivar[i],"'",sep="")
288:
289:
290:     dbSendUpdate(conn, x)
291:     print(paste(i, "th query done",sep=""))
292:     ## Create indexes
293:     x <- paste("create index ", epivar[i], "episode_data_id ON ", epivar[i], " (episode_data_id) ", sep = "")
294:     print(paste("About to create index: ", x))
295:     dbSendUpdate(conn, x)
296:     print(paste(i, "th index created",sep=""))
297:
298:    }
299:
300:    x <- (dbSendUpdate(conn, "DROP TABLE IF EXISTS xxx"))
301:    select.part <- NULL
302:    join.part <- NULL
303:    commavar=","
304:    for (i in 1:length(epivar)) {
305:     if (i == length(epivar)) {
306:      commavar=""
307:     }
308:     select.part <- paste(select.part, epivar[i], ".", epivar[i], commavar, sep = "" )
309:     join.part <- paste(join.part, "LEFT JOIN ", epivar[i], " ON ",
310:                        epivar[i], ".episode_data_id = ", "t2.episode_data_id ",
311:                        sep = "" )
312:    }
313:    full.query <- paste("CREATE TABLE xxx AS select t2.patient_id, t2.episode_data_id, ", select.part,
314:                        "from t2 ",
```

```
315:                        join.part)
316:   dbSendUpdate(conn, full.query)
317:
318:   x<-"Select * FROM xxx"
319:
320:   res<-dbSendQuery(conn,x)
321:   episode<-NULL
322:   nofinished<-TRUE
323:   i=0
324:   while(nofinished) {
325:     i=i+1
326:     temp<-as.data.frame(fetch(res,400))
327:     if ((dim(temp)[1]>0) & i==1){
328:      episode<-temp
329:     }
330:     if ((dim(temp)[1]>0) & i>1){
331:      episode<-rbind(episode,temp)
332:     }
333:     if (dim(temp)[1]==0) {
334:      nofinished<-FALSE
335:     }
336:   }
337:   rm(temp)
338:
339:   x <- (dbSendUpdate(conn, "DROP TABLE IF EXISTS xxx"))
340:
341:  }
342:
343:  if (wide==1) {
344:
345:   x<-"Select * FROM profile_wide RIGHT JOIN activity_data ON profile_wide.patient_id=activity_data.patient_id"
346:
347:   res<-dbSendQuery(conn,x)
348:   patient<-NULL
349:   nofinished<-TRUE
350:   i=0
351:   while(nofinished) {
352:     i=i+1
353:     temp<-as.data.frame(fetch(res,400))
354:     if ((dim(temp)[1]>0) & i==1){
355:      patient<-temp
356:     }
357:     if ((dim(temp)[1]>0) & i>1){
358:      patient<-rbind(patient,temp)
359:     }
```

```
360:     if (dim(temp)[1]==0) {
361:      nofinished<-FALSE
362:     }
363:   }
364:   rm(temp)
365:
366:
367:   x<-"Select * FROM episode_wide"
368:
369:   res<-dbSendQuery(conn,x)
370:   episode<-NULL
371:   nofinished<-TRUE
372:   i=0
373:   while(nofinished) {
374:     i=i+1
375:     temp<-as.data.frame(fetch(res,400))
376:
377:     if ((dim(temp)[1]>0) & i==1){
378:      episode<-temp
379:     }
380:     if ((dim(temp)[1]>0) & i>1){
381:      episode<-rbind(episode,temp)
382:     }
383:     if (dim(temp)[1]==0) {
384:      nofinished<-FALSE
385:     }
386:   }
387:   rm(temp)
388:
389:  }
390:  allepivar <- c("EPI_DATE","STROKE","MI","AMPUT","DIALYSIS","WEIGHT","BMI",
391:                 "SBP","DBP","HYPERTENSION","HBA1C","CREAT","CHOL","HDL","LDL",
392:                 "DRUG_THERAPY","HEIGHT","MA_TEST","SMOK_STAT","RETIN_L",
393:                 "RETIN_R","ULCER_L","ULCER_R","FOOT_EXAM","ESRF",
394:                 "EYE_EXAM","PUMP_THERAPY","ORAL_THERAPY","LIPID_MED","ASA_MED",
395:                 "SELF_MON","TRANSPLANT")
396:
397:  assign("episode",episode,envir=.GlobalEnv)
398:
399:  site<-as.data.frame(dbGetQuery(conn, "Select * FROM site_profile"))
400:
401:  centre_id=as.character(
402:      unique(dbGetQuery(conn, "Select site_header_id from site_header"))[1])
403:
404:  #Simulating missing RDU variables
```

```
405:
406:  patient$end_reason=""
407:  patient$start_reason=""
408:
409:  patient$area<-"IT521"
410:  patient$area[1:1000]<-"IT522"
411:
412:  episode$weight=rnorm(dim(episode)[1],100,11)
413:  episode$bmi=rnorm(dim(episode)[1],30,3)
414:  episode$chol=rnorm(dim(episode)[1],200,100)
415:  episode$hdl=rnorm(dim(episode)[1],50,20)
416:  episode$hba1c=rnorm(dim(episode)[1],7,2)
417:  episode$hba1c[episode$hba1c<4]<-NA
418:
419:  #mi
420:  episode$mi=as.character(rbinom(dim(episode)[1],1,0.3)+1)
421:
422:  #creat
423:  episode$creat=as.character(rbinom(dim(episode)[1],1,0.3)+1)
424:
425:  #ldl
426:  episode$ldl=as.character(rbinom(dim(episode)[1],1,0.3)+1)
427:
428:  #oral_therapy
429:  episode$oral_therapy=as.character(rbinom(dim(episode)[1],4,0.3)+1)
430:
431:  #drug_therapy
432:  episode$drug_therapy=as.character(rbinom(dim(episode)[1],3,0.3)+1)
433:
434:  #ma_test
435:  episode$ma_test=as.character(rbinom(dim(episode)[1],1,0.3)+1)
436:
437:  #smok_stat
438:  episode$smok_stat=as.character(rbinom(dim(episode)[1],2,0.3)+1)
439:
440:  #retin
441:  episode$retin_l=as.character(rbinom(dim(episode)[1],1,0.3)+1)
442:  episode$retin_r=as.character(rbinom(dim(episode)[1],1,0.3)+1)
443:
444:  #ulcer
445:  episode$ulcer_l=as.character(rbinom(dim(episode)[1],1,0.3)+1)
446:  episode$ulcer_r=as.character(rbinom(dim(episode)[1],1,0.3)+1)
447:
448:  #amput
449:  episode$amput=as.character(rbinom(dim(episode)[1],1,0.3)+1)
```

```
450:
451:  #hypertension
452:  episode$hypertension=as.character(rbinom(dim(episode)[1],1,0.3)+1)
453:
454:  #stroke
455:  episode$stroke=as.character(rbinom(dim(episode)[1],1,0.3)+1)
456:
457:  #foot_exam
458:  episode$foot_exam=as.character(rbinom(dim(episode)[1],1,0.3)+1)
459:
460:  #esrf
461:  episode$esrf=as.character(rbinom(dim(episode)[1],1,0.3)+1)
462:
463:  #eye_exam
464:  episode$eye_exam=as.character(rbinom(dim(episode)[1],1,0.3)+1)
465:
466:  #pump_therapy
467:  episode$pump_therapy=as.character(rbinom(dim(episode)[1],1,0.3)+1)
468:
469:  #lipid_med
470:  episode$lipid_med=as.character(rbinom(dim(episode)[1],1,0.3)+1)
471:
472:  #asa_med
473:  episode$asa_med=as.character(rbinom(dim(episode)[1],1,0.3)+1)
474:
475:  #self_mon
476:  episode$self_mon=as.character(rbinom(dim(episode)[1],2,0.3)+1)
477:
478:  #hypert_med
479:  episode$hypert_med=as.character(rbinom(dim(episode)[1],1,0.3)+1)
480:
481:  #microal
482:  episode$microal=rnorm(dim(episode)[1],30,11)
483:
484:  #transplant
485:  episode$transplant=as.character(rbinom(dim(episode)[1],1,0.3)+1)
486:
487:  ##########################################
488: #popdiab
489:
490: popdiab<-as.data.frame(rbind(c(1,1,1,350),c(2,1,1,1200)))
491: names(popdiab)<-c("sex","ageband","type_dm","Freq")
492:
493:  #Store data read from db
494:  csvpatient<-paste(dirstoreout,"/patient.csv",sep="")
```

```
495:  csvepisode<-paste(dirstoreout,"/episode.csv",sep="")
496:  csvpopulation<-paste(dirstoreout,"/pop.csv",sep="")
497:
498:  write.csv(patient,csvpatient)
499:  write.csv(episode,csvepisode)
500:
501:  #write.csv(pop,csvpopulation)
502:
503: ####################################################
504: ####################################################
505: ####################################################
506:  #Temporarly here (waiting the changes to the XML Schema)
507:  pop<-read.csv(paste(dirse,"/","pop.csv",sep=""))
508:  write.csv(episode,csvpopulation)
509: ################
510: ####################################################
511: ####################################################
512: ####################################################
513:  }
514:
515:  if (dbformat=="csv") {
516:   if (length(list.files(dirdatastore)) > 0) {
517:    patient<-read.csv(paste(dirdatastore,"/patient.csv",sep=""))
518:    episode<-read.csv(paste(dirdatastore,"/episode.csv",sep=""))
519:    pop<-read.csv(paste(dirdatastore,"/pop.csv"))
520:   }
521:  }
522:  patient<-BIRO_data_format(data=patient,
523:                            n_var=c("patient_id","start_reason","end_reason"),
524:                            d_var=c("dob","dt_diag","start_date","end_date"),
525:                            f_var="type_dm",
526:                            datefmt="%Y-%m-%d")
527:
528:  allnvar<-c("episode_data_id","patient_id","weight",
529:             "height","sbp","dbp","creat","chol","hdl",
530:             "ldl","ma_test","bmi","hba1c","stroke","mi",
531:             "hypertension","smok_stat","retin_l",
532:             "retin_r","ulcer_l","ulcer_r","foot_exam",
533:             "esrf","eye_exam","pump_therapy",
534:             "oral_therapy","lipid_med","asa_med",
535:             "self_mon","transplant","dialysis","amput",
536:                                "dialysis","drug_therapy")
537:
538:  episode<-BIRO_data_format(data=episode,
539:                            n_var=names(episode[na.omit(pmatch(allnvar,names(episode)))]),
```

```
540:                             c_var=,d_var=c("epi_date"),
541:                             datefmt="%Y-%m-%d")
542:  site<-BIRO_data_format(data=site,
543:                          n_var=c("ds_denom","ds_beds","ds_physicians",
544:                                  "ds_diabetologists","ds_doctors"))
545:
546:  write("processing file: biro_se_recode.r",file="")
547:  if (nchar(logfile)>0) {
548:   write("processing file: biro_se_recode.r",file=logfile,append=FALSE)
549:  }
550:
551:  source(paste(dirse,"/source/r/formats/biro_se_recode.r",sep=""))
552:
553:  write("biro_se_recode.r executed",file="")
554:  if (nchar(logfile)>0) {
555:   write("biro_se_recode.r executed",file=logfile,append=FALSE)
556:  }
557:
558:  #assign patient, episode, population to the global environment
559:  assign("patient",patient,envir=.GlobalEnv)
560:  assign("episode",episode,envir=.GlobalEnv)
561:  assign("pop",pop,envir=.GlobalEnv)
562:  assign("site",site,envir=.GlobalEnv)
563:  assign("popdiab",popdiab,envir=.GlobalEnv)
564:
565: ######################################
566: #   Timeframe for analysis           #
567: ######################################
568:
569:  patient$dob<-as.character(patient$dob)
570:  patient$dob<-ifelse((as.numeric(substr(patient$dob,1,4))<1900),
571:  paste("19",substr(patient$dob,3,10),sep=""),patient$dob)
572:
573:  patient$dob<-as.character(patient$dob)
574:  patient$dob<-ifelse((as.numeric(substr(patient$dob,1,4))>yearnow),
575:  paste("19",substr(patient$dob,3,10),sep=""),patient$dob)
576:
577:  patient$dob<-as.Date(as.character(patient$dob))
578:  episode<-merge(patient[,c("patient_id","dob")],episode,by="patient_id")
579:  episode$epi_date[as.numeric(episode$epi_date)<as.numeric(episode$dob)]<-NA
580:  episode<-episode[,-pmatch("dob",names(episode))]
581: }
582:
```

```
  1: # ############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: biro_se.r                                                            #
  5: #                                                                            #
#
  6: # ############################################################################
  7:
  8: # ------------------------
  9: # Authors:
 10: # Luca Rossi       <redsluke@gmail.com>
 11: # Fabrizio Carinci <research@fabcarinci.net>
 12: # Institutions: Serectrix snc
 13: # Created: 2007-11-21
 14: # Version: 2008-10-23
 15: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 16: # Programming Language: R 2.8.0
 17: # ------------------------
 18: # COPYRIGHT INFORMATION
 19: #
 20: # This file is free software; you can redistribute it and/or modify
 21: # it under the terms of the GNU General Public License as published by
 22: # the Free Software Foundation; either version 2, or (at your option)
 23: # any later version.
 24: #
 25: # This file is distributed in the hope that it will be useful,
 26: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 27: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 28: # GNU General Public License for more details.
 29: #
 30: # You should have received a copy of the GNU General Public License
 31: # along with this file; see the file COPYING.  If not, write to
 32: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 33: #
 34: # In short: you may use this file any way you like, as long as you
 35: # don't charge money for it, remove this notice, or hold anyone liable
 36: # for its results.
 37: #
 38: # biro_se.r is part of WP Statistical Engine of the BIRO Project
 39: # GPL Copyright, The BIRO Project
 40: #
 41: # ------------------------
 42: #
 43: # CONTENT
 44: #
```

```
45: #
46: # ##############################################################################
47:
48: # ##############################################################################
49: #  BIRO_se                                                                     #
50: # ##############################################################################
51: # ------------------------
52: # Authors:
53: # Luca Rossi       <redsluke@gmail.com>
54: # Fabrizio Carinci <research@fabcarinci.net)
55: # Institutions: Serectrix snc
56: # Created: 2007-11-24
57: # Version: 2008-11-21
58: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
59: # Programming Language: R 2.8.0
60: # ------------------------
61: # DESCRIPTION
62: #
63: # BIRO_se is the main function of the Statistical Engine
64: #
65: # DEPENDENCIES
66: #
67: # biro_se_setup.r
68: # biro_se_datastep.r
69: #
70: # ##############################################################################
71: # PARAMETERS:
72: # dbformat        !   = the format of the database ("postgres","csv")
73: # driverClass      !  = a string containing the class of the driver
74: # classPath        !  = a string containing the path of the driver
75: # identifier.quote !  = a string containing the identifier of quotes required by
76: #                       JDBC package
77: # pathdb           !  = a string containing the path of the database
78: # user             !  = a string containing the username
79: # password         !  = a string containing the password
80: # dbname           !  = a string containing the name of the database
81: # dirdatastore     !  = a string containing the path of the input csv files
82: # centre_id        !  = a string containing the centre id
83: # yearnow          !  = a numeric value containing the last possible year of the
84: #                       episode dates
85: # fromyear         !  = a numeric value containing the first possible year where
86: #                       data are reliable
87: # fromyearl        !  = a numeric value containing the first possible year where
88: #                       data are reliable for the trends
89: # endyearl         !  = a numeric value containing the last possible year where
```

```
 90: #                        data are reliable for the trends
 91: # refyear          !  = a numeric containing year to calculate the indicators with
 92: #                        "in the last 12 months" suffix
 93: # refdate          !  = a string date to calculare age in the statistics about the
 94: #                        refyear interval
 95: # logfile   [""]      = a string containing the name of the .log file. If "" only
 96: #                        screen print
 97: # cex                 = A numerical value giving the amount by which
 98: #                        plotting text and symbols should be magnified
 99: #                        relative to the default. Note that some graphics
100: #                        functions such as plot.default have an argument of
101: #                        this name which multiplies this graphical
102: #                        parameter, and some functions such as points accept
103: #                        a vector of values which are recycled.
104: #                        Other uses will take just the first value if a
105: #                        vector of length greater than one is supplied.
106: # ##############################################################################
107:
108: BIRO_se<-function(dirse,
109:                   dbformat, # "postgres" if data are stored in a psotgres database,
110:                             # "csv" if data are stored in csv files
111:                   driverClass,
112:                   classPath,
113:                   identifier.quote,
114:                   pathdb,
115:                   user,
116:                   password,
117:                   dbname,
118:                   dirdatastore,
119:                   centre_id,
120:         startdate="01/01/1997", #Bergen meeting this must be a date (the start date interval for data analysis)
121:         enddate="01/01/2005",   #Bergen meeting include a final date interval for data analysis
122:         yearnow,
123:                   fromyear,
124:                   refanadate,
125:                   refdate,
126:                   logfile="",
127:                   cex,
128:                   wide=1) {
129:
130:  mylog<-logfile
131:  startdate=as.Date(startdate,format="%d/%m/%Y")
132:  enddate=as.Date(enddate,format="%d/%m/%Y")
133:  anayears<-format(startdate,format="%Y"):format(enddate,format="%Y")
134:  assign("anayears",anayears,envir=.GlobalEnv)
```

```
135:  assign("wide",wide,envir=.GlobalEnv)
136:  assign("dirse",dirse,envir=.GlobalEnv)
137:  assign("centre_id",centre_id,envir=.GlobalEnv)
138:  assign("startdate",startdate,envir=.GlobalEnv)
139:  assign("enddate",enddate,envir=.GlobalEnv)
140:  fromyearl=as.character(format(as.Date(startdate,"%d/%m/%Y"),"%Y"))
141:  enddate=as.character(format(as.Date(enddate,"%d/%m/%Y"),"%Y"))
142:  assign("yearnow",yearnow,envir=.GlobalEnv)
143:  assign("fromyearl",startdate,envir=.GlobalEnv)
144:  assign("endyearl",enddate,envir=.GlobalEnv)
145:  assign("refanadate",refanadate,envir=.GlobalEnv)
146:  assign("cex",cex,envir=.GlobalEnv)
147:  assign("logfile",logfile,envir=.GlobalEnv)
148:  #Directory Store Input data (aggregated for in patient and episode datasets)
149:  assign("dirdatastore",dirdatastore,envir=.GlobalEnv)
150:
151:  timestart<-Sys.time()
152:
153:  write("processing file: biro_se_setup.r",file="")
154:  source(paste(dirse,"/source/r/main/biro_se_setup.r",sep=""))
155:  write("biro_se_setup.r executed",file="")
156:  write("calling BIRO_dircreate",file="")
157:
158:  BIRO_dircreate(dirse=dirse,
159:                 centre_id=centre_id)
160:  write("BIRO_dircreate executed",file="")
161:
162:  write("processing file: biro_se_datastep.r",file="")
163:  source(paste(dirse,"/source/r/main/biro_se_datastep.r",sep=""))
164:  write("biro_se_datastep.r executed",file="")
165:
166:  logfile<-paste(dirse,"/output/reports/",logfile,sep="")
167:  assign("logfile",logfile,envir=.GlobalEnv)
168:
169:  write("File .log created",file="")
170:  if (nchar(logfile)>0) {
171:   write("File .log created",file=logfile,append=FALSE)
172:  }
173:
174:  write(paste("Indicator Number: ",length(anayears)*69,sep=""),file="")
175:  if (nchar(logfile)>0) {
176:   write(paste("Indicator Number: ",length(anayears)*69,sep=""),file=logfile,append=FALSE)
177:  }
178:
179:  assign("indnum",0,envir=.GlobalEnv)
```

```
180:  write("calling BIRO_setenv",file="")
181:  if (nchar(logfile)>0) {
182:   write("calling BIRO_setenv",file=logfile,append=FALSE)
183:  }
184:
185:  BIRO_setenv(driverClass=driverClass,
186:              classPath=classPath,
187:              identifier.quote=identifier.quote,
188:              pathdb=pathdb,
189:              user=user,
190:              password=password,
191:              dbname=dbname)
192:
193:  write("BIRO_setenv executed",file="")
194:  if (nchar(logfile)>0) {
195:   write("BIRO_setenv executed",file=logfile,append=FALSE)
196:  }
197:  if (nchar(logfile)>0) {
198:   write(paste("Date: ",as.character(Sys.Date()),sep=""),file=logfile,append=TRUE)
199:   write(paste("Begin ",as.character(timestart),sep=""),file=logfile,append=TRUE)
200:  }
201:  write(paste("Directory CSV ",as.character(dirdataout),sep=""),file=logfile,append=TRUE)
202:  write("calling BIRO_loaddata",file="")
203:  if (nchar(logfile)>0) {
204:   write("calling BIRO_loaddata",file=logfile,append=FALSE)
205:  }
206:  BIRO_loaddata(dbformat=dbformat,
207:                drv=drv,
208:                pathdb=pathdb,
209:                user=user,
210:                password=password,
211:                dbname=dbname,
212:                dirdatastore=dirdatastore,
213:                centre_id=centre_id,
214:                yearnow=yearnow,
215:                fromyear=fromyear,
216:                fromyearl=fromyearl,
217:                startdate=startdate,
218:                enddate=enddate,
219:                refdate=refdate)
220:
221:  write("BIRO_loaddata executed",file="")
222:  if (nchar(logfile)>0) {
223:   write("BIRO_loaddata executed",file=logfile,append=FALSE)
224:  }
```

```
225:
226:  assign("patient",patient,envir=.GlobalEnv)
227:  assign("episode",episode,envir=.GlobalEnv)
228:  assign("site",site,envir=.GlobalEnv)
229:
230:  BIRO_dframe(dataname="patient",monitor=TRUE,filelog=logfile)
231:  BIRO_dframe(dataname="patient",monitor=TRUE,filelog="")
232:  BIRO_dframe(dataname="episode",monitor=TRUE,filelog=logfile)
233:  BIRO_dframe(dataname="episode",monitor=TRUE,filelog="")
234:  BIRO_dframe(dataname="site",monitor=TRUE,filelog=logfile)
235:  BIRO_dframe(dataname="site",monitor=TRUE,filelog="")
236:
237:  for (thisyear in anayears) {
238:   assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
239:          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
240:   assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
241:
242:   cat("opening .Tex file. year:", thisyear,file="",append=TRUE)
243:   if (nchar(logfile)>0) {
244:    cat("opening .Tex file. year:", thisyear,file="",append=TRUE)
245:   }
246:
247:   open_tex(title="",
248:            authors="",
249:            footnote="",
250:            time="",
251:            logo="",
252:            layout=layout,
253:            final=texfile)
254:
255:  }
256:
257:   # ################################
258:   #                              #
259:   #        Create Reports        #
260:   #                              #
261:   # ################################
262:
263:   #Demographic characteristics
264:
265:  for (thisyear in anayears) {
266:   assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
267:          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
268:   assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
269:   new_chapter(title="Demographic characteristics",texfile=texfile)
```

```
270:   new_chapter_html(title="Demographic characteristics",htmlfile=htmlfile)
271:  }
272:
273:  write("processing file: biro_se_indicator_demographic.r",file="")
274:  if (nchar(logfile)>0) {
275:   write("processing file: biro_se_indicator_demographic.r",file=logfile,append=FALSE)
276:  }
277:  source(paste(dirse,"/source/r/scripts/","biro_se_indicator_demographic.r",
278:         sep=""))
279:  write("biro_se_indicator_demographic.r executed",file="")
280:  if (nchar(logfile)>0) {
281:   write("biro_se_indicator_demographic.r executed",file=logfile,append=FALSE)
282:  }
283:
284:  #Clinical characteristics
285:  for (thisyear in anayears) {
286:   assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
287:          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
288:   assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
289:   new_chapter(title="Clinical characteristics",texfile=texfile)
290:   new_chapter_html(title="Clinical characteristics",htmlfile=htmlfile)
291:  }
292:  write("processing file: biro_se_indicator_clinical.r",file="")
293:  if (nchar(logfile)>0) {
294:   write("processing file: biro_se_indicator_clinical.r",file=logfile,append=FALSE)
295:  }
296:  source(paste(dirse,"/source/r/scripts/","biro_se_indicator_clinical.r",
297:         sep=""))
298:  write("biro_se_indicator_clinical.r executed",file="")
299:  if (nchar(logfile)>0) {
300:   write("biro_se_indicator_clinical.r executed",file=logfile,append=FALSE)
301:  }
302:
303:  #Health System
304:  for (thisyear in anayears) {
305:   assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
306:          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
307:   assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
308:   new_chapter(title="Health System",texfile=texfile)
309:   new_chapter_html(title="Health System",htmlfile=htmlfile)
310:  }
311:  write("processing file: biro_se_indicator_health_system.r",file="")
312:  if (nchar(logfile)>0) {
313:   write("processing file: biro_se_indicator_health_system.r",file=logfile,append=FALSE)
314:  }
```

```
315:  source(paste(dirse,"/source/r/scripts/","biro_se_indicator_health_system.r",
316:         sep=""))
317:  write("biro_se_indicator_health_system.r executed",file="")
318:  if (nchar(logfile)>0) {
319:   write("biro_se_indicator_health_system.r executed",file=logfile,append=FALSE)
320:  }
321:
322:  #Population
323:  for (thisyear in anayears) {
324:   assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
325:          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
326:   assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
327:   new_chapter(title="Population",texfile=texfile)
328:   new_chapter_html(title="Population",htmlfile=htmlfile)
329:  }
330:  write("processing file: biro_se_indicator_population.r",file="")
331:  if (nchar(logfile)>0) {
332:   write("processing file: biro_se_indicator_population.r",file=logfile,append=FALSE)
333:  }
334:  source(paste(dirse,"/source/r/scripts/","biro_se_indicator_population.r",
335:         sep=""))
336:  write("biro_se_indicator_population.r executed",file="")
337:  if (nchar(logfile)>0) {
338:   write("biro_se_indicator_population.r executed",file=logfile,append=FALSE)
339:  }
340:  #Risk Adjusted
341:  for (thisyear in anayears) {
342:   assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
343:          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
344:   assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
345:   new_chapter(title="Risk Adjusted Indicators",texfile=texfile)
346:   new_chapter_html(title="Risk Adjusted Indicators",htmlfile=htmlfile)
347:  }
348:  write("processing file: biro_se_indicator_risk_adjusted.r",file="")
349:  if (nchar(logfile)>0) {
350:   write("processing file: biro_se_indicator_risk_adjusted.r",file=logfile,append=FALSE)
351:  }
352:  source(paste(dirse,"/source/r/scripts/","biro_se_indicator_risk_adjusted.r",
353:         sep=""))
354:  write("biro_se_indicator_risk_adjusted.r executed",file="")
355:  if (nchar(logfile)>0) {
356:   write("biro_se_indicator_risk_adjusted.r executed",file=logfile,append=FALSE)
357:  }
358:
359:  for (thisyear in anayears) {
```

```
360:   assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
361:          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
362:   assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
363:   close_tex(file=texfile)
364:   setwd(dirreport)
365:   assign("layout",paste(dirsourcelatex,"/layout.tex",sep=""),envir=.GlobalEnv)
366:   assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
367:   assign("dvifile",paste(dirreport,"/report.dvi",sep=""),envir=.GlobalEnv)
368:   assign("pdffile",paste(dirreport,"/,report",thisyear,".pdf",sep=""),envir=.GlobalEnv)
369:
370:   command<-paste("pdflatex",texfile)
371:
372:   assign("diraux",paste(dirse,"/output/reports/", "#",launchtime,
373:          "/",thisyear,"/",centre_id,"/pdf",sep=""),envir=.GlobalEnv)
374:   write("Preparing to compile .Tex File",file="")
375:   if (nchar(logfile)>0) {
376:    write("Preparing to compile .Tex File",file=logfile,append=TRUE)
377:   }
378:
379:   x<-system(command,intern=TRUE)
380:   x<-system(command,intern=TRUE)
381:   x<-system(command,intern=TRUE)
382:
383:   pathtex=substr(texfile,1,nchar(texfile)-4)
384:   file.rename(from=paste(pathtex,".aux",sep=""), to=paste(dirreport,"/pdf/report.aux",sep=""))
385:   file.rename(from=paste(pathtex,".out",sep=""), to=paste(dirreport,"/pdf/report.out",sep=""))
386:   file.rename(from=paste(pathtex,".log",sep=""), to=paste(dirreport,"/pdf/report.log",sep=""))
387:   file.rename(from=paste(pathtex,".toc",sep=""), to=paste(dirreport,"/pdf/report.toc",sep=""))
388:   file.rename(from=texfile, to=paste(dirreport,"/pdf/report.tex",sep=""))
389:
390:   write("Tex File Compiled",file="")
391:   if (nchar(logfile)>0) {
392:    write("Tex File Compiled",file=logfile,append=TRUE)
393:   }
394:
395:  }
396:  timefinish<-Sys.time()
397:
398:  write(paste("Elapsed time:",timefinish-timestart, "mins",sep=" "),file="")
399:  if (nchar(logfile)>0) {
400:   write(paste("Elapsed time:",timefinish-timestart, "mins",sep=" "),
401:         file=logfile,append=TRUE)
402:  }
403:  newlogfile<-paste(dirreport,"/",mylog,sep="")
404:  file.rename(logfile,newlogfile)
```

```
405: }
406:
407:
408:
```

```
# ##############################################################################
#                                                                              #
# 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: 2008-10-23
# OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
# 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_setup.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
```

```
45: # BIRO_setenv
46: # BIRO_dircreate
47: #
48: # ##############################################################################
49:
50: # ##############################################################################
51: #  BIRO_setenv
52: # ##############################################################################
53: # ------------------------
54: # Authors:
55: # Luca Rossi       <redsluke@gmail.com>
56: # Fabrizio Carinci <research@fabcarinci.net)
57: # Institutions: Serectrix snc
58: # Created: 2007-11-24
59: # Version: 2008-11-21
60: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
61: # Programming Language: R 2.8.0
62: # ------------------------
63: # DESCRIPTION
64: #
65: # BIRO_setenv sets the environment of the BIRO Statistical Engine
66: #
67: # DEPENDENCIES
68: #
69: # R package: RJDBC
70: #
71: # ##############################################################################
72: # PARAMETERS:
73: # driverClass      !  = a string containing the class of the driver
74: # classPath        !  = a string containing the path of the driver
75: # identifier.quote !  = a string containing the identifier of quotes
76: # pathdb           !  = a string containing the path of the database
77: # user             !  = a string containing the username
78: # password         !  = a string containing the password
79: # dbname           !  = a string containing the name of the database
80: # ##############################################################################
81:
82:
83: BIRO_setenv<-function(driverClass,
84:                       classPath,
85:                       identifier.quote,
86:                       pathdb,
87:                       user,
88:                       password,
89:                       dbname) {
```

```
 90:
 91: #R Packages installation
 92:
 93:  mySys<-R.Version()
 94:
 95:  if (substr(mySys$os,1,5)=="mingw") {
 96:   write("O.S. Windows",file="")
 97:   if (nchar(logfile)>0) {
 98:    write("O.S. Windows",file=logfile,append=TRUE)
 99:   }
100:   dirpack<-paste(substr(dirse,1,nchar(dirse)-4),"lib/r/source/packages",sep="")
101:
102:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="Cairo"])==0) {
103:    install.packages(paste(dirpack,"/win/Cairo_1.4-4.zip",sep=""), repos = NULL)
104:    write("Cairo package installed from binary zip file... O.S. Windows",file="")
105:    if (nchar(logfile)>0) {
106:     write("Cairo package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
107:    }
108:   }
109:
110:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="rJava"])==0) {
111:    install.packages(paste(dirpack,"/win/rJava_0.6-0.zip",sep=""), repos = NULL)
112:    write("rJava package installed from binary zip file... O.S. Windows",file="")
113:    if (nchar(logfile)>0) {
114:     write("rJava package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
115:    }
116:   }
117:
118:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="DBI"])==0) {
119:    install.packages(paste(dirpack,"/win/DBI_0.2-4.zip",sep=""), repos = NULL)
120:    write("DBI package installed from binary zip file... O.S. Windows",file="")
121:    if (nchar(logfile)>0) {
122:     write("DBI package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
123:    }
124:   }
125:
126:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="RJDBC"])==0) {
127:    install.packages(paste(dirpack,"/win/RJDBC_0.1-5.zip",sep=""), repos = NULL)
128:    write("RJDBC package installed from binary zip file... O.S. Windows",file="")
129:    if (nchar(logfile)>0) {
130:     write("RJDBC package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
131:    }
132:   }
133:
134:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="lattice"])==0) {
```

```
135:    install.packages(paste(dirpack,"/win/lattice_0.17-15.zip",sep=""), repos = NULL)
136:    write("lattice package installed from binary zip file... O.S. Windows",file="")
137:    if (nchar(logfile)>0) {
138:     write("lattice package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
139:    }
140:   }
141:
142:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="rmeta"])==0) {
143:    install.packages(paste(dirpack,"/win/rmeta_2.14.zip",sep=""), repos = NULL)
144:    write("rmeta package installed from binary zip file... O.S. Windows",file="")
145:    if (nchar(logfile)>0) {
146:     write("rmeta package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
147:    }
148:   }
149:
150:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="sp"])==0) {
151:    install.packages(paste(dirpack,"/win/sp_0.9-28.zip",sep=""), repos = NULL)
152:    write("sp package installed from binary zip file... O.S. Windows",file="")
153:    if (nchar(logfile)>0) {
154:     write("sp package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
155:    }
156:   }
157:
158:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="maptools"])==0) {
159:    install.packages(paste(dirpack,"/win/maptools_0.7-15.zip",sep=""), repos = NULL)
160:    write("maptools package installed from binary zip file... O.S. Windows",file="")
161:    if (nchar(logfile)>0) {
162:     write("maptools package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
163:    }
164:   }
165:
166:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="Hmisc"])==0) {
167:    install.packages(paste(dirpack,"/win/Hmisc_3.4-4.zip",sep=""), repos = NULL)
168:    write("Hmisc package installed from binary zip file... O.S. Windows",file="")
169:    if (nchar(logfile)>0) {
170:     write("Hmisc package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
171:    }
172:   }
173:
174:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="R2HTML"])==0) {
175:    install.packages(paste(dirpack,"/win/R2HTML_1.59.zip",sep=""), repos = NULL)
176:    write("R2HTML package installed from binary zip file... O.S. Windows",file="")
177:    if (nchar(logfile)>0) {
178:     write("R2HTML package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
179:    }
```

```
180:   }
181:
182:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="Epi"])==0) {
183:    install.packages(paste(dirpack,"/win/Epi_1.0.8.zip",sep=""), repos = NULL)
184:    write("Epi package installed from binary zip file... O.S. Windows",file="")
185:    if (nchar(logfile)>0) {
186:     write("Epi package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
187:    }
188:   }
189:
190:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="epicalc"])==0) {
191:    install.packages(paste(dirpack,"/win/epicalc_2.8.0.0.zip",sep=""), repos = NULL)
192:    write("epicalc package installed from binary zip file... O.S. Windows",file="")
193:    if (nchar(logfile)>0) {
194:     write("epicalc package installed from binary zip file... O.S. Windows",file=logfile,append=TRUE)
195:    }
196:   }
197:
198:  }
199:
200:  if (substr(mySys$os,1,5)=="unix") {
201:   write("O.S. Linux",file="")
202:   if (nchar(logfile)>0) {
203:    write("O.S. Linux",file=logfile,append=TRUE)
204:   }
205:
206:   dirpack<-paste(substr(dirse,1,nchar(dirse)-4),"lib/r/source/packages",sep="")
207:
208:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="Cairo"])==0) {
209:    install.packages(paste(dirpack,"/linux/Cairo_1.4-4.tar.gz",sep=""), repos = NULL)
210:    write("Cairo package installed from tar.gz file... O.S. Linux",file="")
211:    if (nchar(logfile)>0) {
212:     write("Cairo package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
213:    }
214:   }
215:
216:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="rJava"])==0) {
217:    install.packages(paste(dirpack,"/linux/rJava_0.6-0.tar.gz",sep=""), repos = NULL)
218:    write("rJava package installed from tar.gz file... O.S. Linux",file="")
219:    if (nchar(logfile)>0) {
220:     write("rJava package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
221:    }
222:   }
223:
224:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="DBI"])==0) {
```

```
225:    install.packages(paste(dirpack,"/linux/DBI_0.2-4.tar.gz",sep=""), repos = NULL)
226:    write("DBI package installed from tar.gz file... O.S. Linux",file="")
227:    if (nchar(logfile)>0) {
228:     write("DBI package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
229:    }
230:   }
231:
232:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="RJDBC"])==0) {
233:    install.packages(paste(dirpack,"/linux/RJDBC_0.1-5.tar.gz",sep=""), repos = NULL)
234:    write("RJDBC package installed from tar.gz file... O.S. Linux",file="")
235:    if (nchar(logfile)>0) {
236:     write("RJDBC package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
237:    }
238:   }
239:
240:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="lattice"])==0) {
241:    install.packages(paste(dirpack,"/linux/lattice_0.17-17.tar.gz",sep=""), repos = NULL)
242:    write("lattice package installed from tar.gz file... O.S. Linux",file="")
243:    if (nchar(logfile)>0) {
244:     write("lattice package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
245:    }
246:   }
247:
248:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="rmeta"])==0) {
249:    install.packages(paste(dirpack,"/linux/rmeta_2.14.tar.gz",sep=""), repos = NULL)
250:    write("rmeta package installed from tar.gz file... O.S. Linux",file="")
251:    if (nchar(logfile)>0) {
252:     write("rmeta package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
253:    }
254:   }
255:
256:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="sp"])==0) {
257:    install.packages(paste(dirpack,"/linux/sp_0.9-28.tar.gz",sep=""), repos = NULL)
258:    write("sp package installed from tar.gz file... O.S. Linux",file="")
259:    if (nchar(logfile)>0) {
260:     write("sp package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
261:    }
262:   }
263:
264:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="maptools"])==0) {
265:    install.packages(paste(dirpack,"/linux/maptools_0.7-16.tar.gz",sep=""), repos = NULL)
266:    write("maptools package installed from tar.gz file... O.S. Linux",file="")
267:    if (nchar(logfile)>0) {
268:     write("maptools package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
269:    }
```

```
270:   }
271:
272:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="Hmisc"])==0) {
273:    install.packages(paste(dirpack,"/linux/Hmisc_3.4-4.tar.gz",sep=""), repos = NULL)
274:    write("Hmisc package installed from tar.gz file... O.S. Linux",file="")
275:    if (nchar(logfile)>0) {
276:     write("Hmisc package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
277:    }
278:   }
279:
280:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="R2HTML"])==0) {
281:    install.packages(paste(dirpack,"/linux/R2HTML_1.59.tar.gz",sep=""), repos = NULL)
282:    write("R2HTML package installed from tar.gz file... O.S. Linux",file="")
283:    if (nchar(logfile)>0) {
284:     write("R2HTML package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
285:    }
286:   }
287:
288:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="Epi"])==0) {
289:    install.packages(paste(dirpack,"/linux/Epi_1.0.8.tar.gz",sep=""), repos = NULL)
290:    write("Epi package installed from tar.gz file... O.S. Linux",file="")
291:    if (nchar(logfile)>0) {
292:     write("Epi package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
293:    }
294:   }
295:
296:   if (length(sort(.packages(all.available = TRUE))[sort(.packages(all.available = TRUE))=="epicalc"])==0) {
297:    install.packages(paste(dirpack,"/linux/epicalc_2.8.0.0.tar.gz",sep=""), repos = NULL)
298:    write("epicalc package installed from tar.gz zip file... O.S. Linux",file="")
299:    if (nchar(logfile)>0) {
300:     write("epicalc package installed from tar.gz file... O.S. Linux",file=logfile,append=TRUE)
301:    }
302:   }
303:  }
304:
305:  library(rJava)
306:  library(RJDBC)
307:  library(lattice)
308:  library(rmeta)
309:  library(maptools)
310:  library(Hmisc)
311:  library(R2HTML)
312:  library(Cairo)
313:
314:  assign("drv",JDBC(driverClass ="org.postgresql.Driver",classPath=classPath,
```

```
315:         identifier.quote=identifier.quote),envir=.GlobalEnv)
316: }
317:
318: # ############################################################################
319: #  BIRO_dircreate
320: # ############################################################################
321: # ------------------------
322: # Authors:
323: # Luca Rossi       <redsluke@gmail.com>
324: # Fabrizio Carinci <research@fabcarinci.net)
325: # Institutions: Serectrix snc
326: # Created: 2007-11-24
327: # Version: 2008-11-21
328: # OS: Windows XP/Vista
329: # Programming Language: R 2.8.0
330: # ------------------------
331: # DESCRIPTION
332: #
333: # BIRO_dircreate creates directories according the BIRO directories schema
334: #
335: # DEPENDENCIES
336: #
337: # r package: base
338: #
339: # ############################################################################
340: # PARAMETERS:
341: # dirse      !        = a string containing the path of the Statistical Engine
342: #                      directory
343: # centre_id  !        = a string containing the id of the centre
344: # ############################################################################
345:
346:
347: BIRO_dircreate<-function(dirse,
348:                          centre_id) {
349:  dir.create(dirse, showWarnings = FALSE, recursive = TRUE)
350:  #Directory Reports
351:  assign("dirse",dirse,envir=.GlobalEnv)
352:  assign("dirsourceR",paste(dirse,"/source/r",sep=""),envir=.GlobalEnv)
353:  assign("dirsourcelatex",paste(substr(dirse,1,nchar(dirse)-4),
354:         "lib/templates/pdf",sep=""),envir=.GlobalEnv)
355:  #Load ".../lib/R/source/biro" functions
356:
357:  assign("dirlibR",paste(substr(dirse,1,nchar(dirse)-4),
358:         "lib/r/source/biro",sep=""),envir=.GlobalEnv)
359:
```

```
360:  listlib<-c("biro_aggregate.r",
361:             "biro_demographic.r",
362:             "biro_explife.r",
363:             "biro_forest.r",
364:             "biro_maps.r",
365:             "biro_plots.r",
366:             "biro_regression.r",
367:             "biro_standardize.r",
368:             "biro_trellis.r",
369:             "biro_util.r",
370:             "biro_util.r",
371:             "biro_webplots.r")
372:
373:  for (i in 1:length(list.files(dirlibR))) {
374:   source(paste(dirlibR,"/",listlib[i],sep=""))
375:  }
376:
377:  #Load ".../_se_/source/R/include" functions
378:  assign("dirinclude",paste(dirsourceR,"/include",sep=""),envir=.GlobalEnv)
379:
380:  #list of files to include
381:  listinc<-c("biro_se_boxplots.r",
382:             "biro_se_dispersion.r",
383:             "biro_se_histograms.r",
384:             "biro_se_location.r",
385:             "biro_se_report.r",
386:             "biro_se_tables.r",
387:             "biro_se_trend.r")
388:  for (i in 1:length(listinc)) {
389:   source(paste(dirinclude,"/",listinc[i],sep=""))
390:  }
391:
392:  dir.create(dirsourceR, showWarnings = FALSE, recursive = TRUE)
393:  dir.create(dirsourcelatex, showWarnings = FALSE, recursive = TRUE)
394:  #Local Repository of DB data in csv format
395:
396:  launchtime<-format(Sys.time(),"%d%m%y%H%M%S")
397:  assign("launchtime",launchtime,envir=.GlobalEnv)
398:  dirstoreout<-paste(dirse,"/data/", "#",launchtime,"/",centre_id,sep="")
399:  assign("dirstoreout",dirstoreout,envir=.GlobalEnv)
400:  dir.create(dirstoreout, showWarnings = FALSE, recursive = TRUE)
401:  #Directory Local Components
402:  for (thisyear in anayears) {
403:   assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
404:          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
405:   dir.create(dirdataout, showWarnings = FALSE, recursive = TRUE)
406:   assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
407:          "/",thisyear,"/",centre_id,"/","graphs",sep=""),envir=.GlobalEnv)
408:   dir.create(dirgraph, showWarnings = FALSE, recursive = TRUE)
409:   assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
410:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
411:   dir.create(dirtables, showWarnings = FALSE, recursive = TRUE)
412:   assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
413:          "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
414:   assign("diraux",paste(dirse,"/output/reports/", "#",launchtime,
415:          "/",thisyear,"/",centre_id,"/pdf",sep=""),envir=.GlobalEnv)
416:   dirimages<-paste(dirse,"/output/reports/", "#",launchtime,
417:                    "/",thisyear,"/",centre_id,"/images",sep="")
418:   dir.create(dirimages, showWarnings = FALSE, recursive = TRUE)
419:   assign("dirlibR",paste(substr(dirse,1,nchar(dirse)-4),
420:          "lib/r/source/biro",sep=""),envir=.GlobalEnv)
421:   file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/html/","biro-logo01.jpg",sep=""),
422:             paste(dirimages,"/","biro-logo01.jpg",sep=""))
423:   dir.create(diraux, showWarnings = FALSE, recursive = TRUE)
424:   dir.create(dirhtml, showWarnings = FALSE, recursive = TRUE)
425:   assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
426:          "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
427:   assign("layout",paste(dirsourcelatex,"/layout.tex",sep=""),envir=.GlobalEnv)
428:   assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
429:   assign("dvifile",paste(dirreport,"/report.dvi",sep=""),envir=.GlobalEnv)
430:   assign("pdffile",paste(dirreport,"/report.pdf",sep=""),envir=.GlobalEnv)
431:
432:   #create html report index
433:
434:   file.copy(paste(substr(dirse,1,nchar(dirse)-4),"lib/templates/html/","layout_open.html",sep=""),
435:             paste(dirreport,"/","report.html",sep=""))
436:
437:  }
438:
439:  assign("dirmap",paste(substr(dirse,1,nchar(dirse)-4),
440:         "lib/maps",sep=""),envir=.GlobalEnv)
441:  assign("maploaded","",envir=.GlobalEnv)
442:  assign("dirgraphlatex","graphs",envir=.GlobalEnv)
443:  write("BIRO directories created",file="")
444:  if (nchar(logfile)>0) {
445:   write("BIRO directories created",file=logfile,append=TRUE)
446:  }
447:
448:  cat("<body>",
449:      file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
```

```
  450:  cat("<table class=\"reportheader\" border=\"0\">",
  451:      file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  452:  cat("<tbody>",
  453:      file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  454:  cat(" <tr>",
  455:      file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  456:  cat("  <td><img src=\"images/biro-logo01.jpg\"></td>",
  457:      file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  458:  cat(" </tr>",
  459:      file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  460:  cat("</tbody>",
  461:      file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  462:  cat("</table>",
  463:      file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  464:  cat("<table class=\"reportheader\" border=\"0\">",file=paste(dirreport,"/","report.html",sep=""),fill =
TRUE,append=TRUE)
  465:  cat("<tbody>",file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  466:  cat(" <tr>",file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  467:  cat(" <td class=\"reportheader\"> BIRO Report site: ",centre_id,", ",
  468:      format(as.Date(launchtime,format="%d%m%y%H%M%S"),format="%d %B %Y"),
  469:      "</td>",file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  470:  cat(" </tr>",file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  471:  cat(" <tr>",file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  472:  cat("  <td class=\"reportheader\">Timeframe: ",startdate,"-",enddate,"</td>",
  473:      file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  474:  cat(" </tr>",file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  475:  cat("</tbody>",file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  476:  cat("</table>",file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  477:  assign("htmlfile",paste(dirreport,"/","report.html",sep=""),envir=.GlobalEnv)
  478:
  479: }
  480:
  481:
  482:
  483:
```

```
    1: ##############################################################################
    2: #                                                                            #
    3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
    4: # File: BIRO_se_indicator_clinical.r                                  #
    5: #                                                                            #
#
    6: ##############################################################################
    7: # ------------------------
    8: # Authors:
    9: # Luca Rossi       <redsluke@gmail.com)
   10: # Fabrizio Carinci <research@fabcarinci.net)
   11: # Institutions: Serectrix snc
   12: # Created: 2007-11-24
   13: # Version: 2008-11-24
   14: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
   15: # Programming Language: R 2.8.0
   16: # ------------------------
   17: # COPYRIGHT INFORMATION
   18: #
   19: # This file is free software; you can redistribute it and/or modify
   20: # it under the terms of the GNU General Public License as published by
   21: # the Free Software Foundation; either version 2, or (at your option)
   22: # any later version.
   23: #
   24: # This file is distributed in the hope that it will be useful,
   25: # but WITHOUT ANY WARRANTY; without even the implied warranty of
   26: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   27: # GNU General Public License for more details.
   28: #
   29: # You should have received a copy of the GNU General Public License
   30: # along with this file; see the file COPYING.  If not, write to
   31: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
   32: #
   33: # In short: you may use this file any way you like, as long as you
   34: # don't charge money for it, remove this notice, or hold anyone liable
   35: # for its results.
   36: #
   37: # BIRO_se_indicator_clinical.r is part of WP Statistical Engine of the BIRO
   38: # Project
   39: # GPL Copyright, The BIRO Project
   40: #
   41: # ------------------------
   42: #
   43: ##############################################################################
   44:
```

```
45:
46:   new_section(title="2.1 Diabetes Status",texfile=texfile)
47:   new_section_html(title="2.1 Diabetes Status",htmlfile=htmlfile)
48:
49:   #
50:   # 2.1.1 Type of Diabetes (All the patient of the db)
51:   #
52:
53:   if (length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0) {
54:
55:
56:     for (thisyear in anayears) {
57:
58:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
59:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
60:
61:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
62:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
63:
64:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
65:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
66:
67:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
68:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
69:
70:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
71:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
72:
73:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
74:
75:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
76:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
77:
78:     data2_1_1<-patient[patient$end_date>checkdate | is.na(patient$end_date),]
79:     data2_1_1$age<-(checkdate-patient$dob)/365.25
80:     data2_1_1$age_c<-varclass(data2_1_1$age,ageth)
81:
82:     BIRO_dframe(dataname="data2_1_1",monitor=TRUE,filelog=logfile)
83:     BIRO_dframe(dataname="data2_1_1",monitor=TRUE,filelog="")
84:
85:        BIRO_report(ind="2_1_1",
86:                    title="2.1.1. Type of diabetes",
87:                    var_cat="type_dm",
88:                    class=c("age_c"),
89:                    lev_var_cat=levtype_dm,
```

```
                    lev_class=list(a=classlabellist(ageth)),
                    lab_var_cat="Type of Diabetes",
                    lab_class=c("Age Classes"),
                    tab=1,
                    tabside="H",
                    tabperc=TRUE,
                    tabwidth=1,
                    barbeside=TRUE,
                    barnumber=TRUE,
                    barperc=TRUE,
                    box=0,
                    bar=1,
                    lines=0,
                    trellis=0,
                    pie=0,
                    texfile=texfile,
                    dirtables=dirtables,
                    dircsv=dirdataout,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    chisq=NULL)

    rm(data2_1_1)
    }
    } else BIRO_report_toc(title="2.1.1. Type of diabetes")
  #
  # 2.1.2 Duration of Diabetes (All the patient in the db)
  #


  if (length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0 &
      length(na.omit(patient$sex))>0 & length(na.omit(episode$hba1c))>0) {


    for (thisyear in anayears) {

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

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

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

```
  135:
  136:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  137:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  138:
  139:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  140:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  141:
  142:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  143:
  144:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  145:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  146:
  147:     data2_1_2<-BIRO_demographic(patient=patient[patient$end_date>checkdate | is.na(patient$end_date),],
  148:                                 episode=episode[episode$patient_id %in% patient[patient$end_date>checkdate | is
.na(patient$end_date),"patient_id"],],
  149:                                 var=c("sex","type_dm","durdiab_c"),
  150:                                 date="first episode date",
  151:                                 startvar="dt_diag",
  152:                                 th=durdiabth,
  153:                                 hold=TRUE,
  154:                                 varname="durdiab")
  155:
  156:     date2_1_2<-as.data.frame(cbind(data2_1_2$patient_id,checkdate))
  157:
  158:     names(date2_1_2)<-c("patient_id","checkdate")
  159:     #Find the first date of hba1c
  160:     firsthba1c<-as.data.frame(episode[is.na(episode$hba1c)==FALSE,
  161:                                 c("patient_id","epi_date")])
  162:     #select hba1c done
  163:     firsthba1c<-aggregate(firsthba1c$epi_date,
  164:                           list(firsthba1c$patient_id),FUN=min) #the first hba1c
  165:     names(firsthba1c)<-c("patient_id","firsthba1c")
  166:     data2_1_2<-merge(data2_1_2,firsthba1c,by="patient_id",all.x=TRUE)
  167:     data2_1_2$hba1c_done<-
  168:       ifelse(as.numeric(data2_1_2$first_epi_date)>=data2_1_2$firsthba1c,1,2)
  169:     data2_1_2$hba1c_done[is.na(data2_1_2$hba1c_done)]<-2
  170:
  171:
  172:     BIRO_dframe(dataname="data2_1_2",monitor=TRUE,filelog=logfile)
  173:     BIRO_dframe(dataname="data2_1_2",monitor=TRUE,filelog="")
  174:
  175:        BIRO_report(ind="2_1_2",
  176:                    title="2.1.2. Duration of diabetes (Classes)",
  177:                    var_cat="hba1c_done",
  178:                    class=c("sex","durdiab_c","type_dm"),
```

```
179:                     lev_var_cat=levhba1c_done,
180:                     lev_class=list(a=levsex,
181:                                    b=classlabellist(durdiabth),
182:                                    c=levtype_dm),
183:                     lab_var_cat="HbA1c done",
184:                     lab_class=c("Type of Diabetes","Duration of Diabetes","Gender"),
185:                     tab=1,
186:                     tabside="H",
187:                     tabperc=TRUE,
188:                     tabwidth=1,
189:                     barbeside=TRUE,
190:                     barnumber=TRUE,
191:                     barperc=TRUE,
192:                     box=0,
193:                     bar=1,
194:                     lines=0,
195:                     trellis=0,
196:                     pie=0,
197:                     texfile=texfile,
198:                     dirtables=dirtables,
199:                     dircsv=dirdataout,
200:                     dirgraph=dirgraph,
201:                     dirgraphlatex=dirgraphlatex,
202:                     cex=cex)
203:
204:     rm(data2_1_2)
205:   }
206:   } else BIRO_report_toc(title="2.1.2. Duration of diabetes (Classes)")
207:
208:   new_section(title="2.2 Risk Factors",texfile=texfile)
209:   new_section_html(title="2.2 Risk Factors",htmlfile=htmlfile)
210:
211:   new_sub_section_html(title="2.2.1 Obesity",htmlfile=htmlfile)
212:
213:   #
214:   # 2.2.1.1 Weight (the last episode, not only refyear)
215:   #
216:
217:   if (length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0 &
218:       length(na.omit(patient$sex))>0 & length(na.omit(episode$weight))>0) {
219:
220:     episode$weight<-
221:       ifelse(episode$weight>=weightlimit[1] & episode$weight<=weightlimit[2],
222:              episode$weight,NA)
223:     episode$weight_c<-varclass(episode$weight,weightth)
```

```
  224:
  225:
  226:     for (thisyear in anayears) {
  227:
  228:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  229:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  230:
  231:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  232:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  233:
  234:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  235:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  236:
  237:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  238:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  239:
  240:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  241:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  242:
  243:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  244:
  245:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  246:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  247:
  248:     if (dim(episode[is.na(episode$weight)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
  249:
  250:     data2_2_1_1<-BIRO_demographic(patient=patient,
  251:                                   episode=episode[is.na(episode$weight)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  252:                                   var=c("sex","type_dm","age_c"),
  253:                                   date="last episode date",
  254:                                   startvar="dob",
  255:                                   th=ageth,
  256:                                   hold=TRUE,
  257:                                   varname="age")
  258:
  259:     data2_2_1_1<-merge(episode[is.na(episode$weight)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date<=checkdate,
  260:                           c("patient_id","epi_date","weight","weight_c")],
  261:                        data2_2_1_1,
  262:                        by.x=c("patient_id","epi_date"),
  263:                        by.y=c("patient_id","last_epi_date"),
  264:                        all.y=TRUE)
  265:
```

```
266:     #BIRO_mean
267: #    BIRO_mean(data=data2_2_1_1,
268: #              var="weight",
269: #              stratum="sex",
270: #              dirout=dirdataout,
271: #              out="m_2_2_1_1a",
272: #              ext="csv",
273: #              date=Sys.Date())
274: #    BIRO_mean(data=data2_2_1_1,
275: #              var="weight",
276: #              stratum="type_dm",
277: #              dirout=dirdataout,
278: #              out="m_2_2_1_1b",
279: #              ext="csv",
280: #              date=Sys.Date())
281: #    BIRO_mean(data=data2_2_1_1,
282: #              var="weight",
283: #              stratum="age_c",
284: #              dirout=dirdataout,
285: #              out="m_2_2_1_1c",
286: #              ext="csv",
287: #              date=Sys.Date())
288:
289:     BIRO_dframe(dataname="data2_2_1_1",monitor=TRUE,filelog=logfile)
290:     BIRO_dframe(dataname="data2_2_1_1",monitor=TRUE,filelog="")
291:
292:     }
293:
294:
295:    if (thisyear==anayears[1]) {
296:     data2_2_1_1l<-data2_2_1_1
297:     data2_2_1_1l$year_epi=thisyear
298:    } else {
299:     data2_2_1_1$year_epi=thisyear
300:     data2_2_1_1l<-rbind(data2_2_1_1l,data2_2_1_1)
301:    }
302:
303:
304:
305: #    data2_2_1_1l<-BIRO_demographic(patient=patient,
306: #                                  episode=episode[is.na(episode$weight)==FALSE,],
307: #                                    var=c("sex","type_dm","age_c"),
308: #                                    date="last episode date for year",
309: #                                    startvar="dob",
310: #                                    th=ageth,
```

```
311: #                                    hold=TRUE,
312: #                                    varname="age")
313:
314: #    data2_2_1_1l<-merge(data2_2_1_1l,
315: #                        BIRO_aggregate(data=episode,
316: #                                       request=c("patient_id","year_epi"),
317: #                                       fun="mean",
318: #                                       vard="weight",
319: #                                       isdate=FALSE,
320: #                                       year_epi="year_epi",
321: #                                       create_year=TRUE),
322: #                                       by=c("patient_id","year_epi"))
323: #    data2_2_1_1l<-data2_2_1_1l[data2_2_1_1l$year_epi>=fromyearl,]
324:
325:     BIRO_dframe(dataname="data2_2_1_1l",monitor=TRUE,filelog=logfile)
326:     BIRO_dframe(dataname="data2_2_1_1l",monitor=TRUE,filelog="")
327:
328:     printlines=0
329:     if (thisyear==anayears[length(anayears)]) {
330:       lines=1
331:       printlines=1
332:       if (length(anayears)==1) {
333:         printlines=0
334:       }
335:     }
336:
337:     if (dim(episode[is.na(episode$weight)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
338:
339:        BIRO_report(ind="2_2_1_1",
340:                    title="2.2.1.1. Weight (last episode in 12 months)",
341:                    var="weight",
342:                    var_cat="weight_c",
343:                    class=c("sex","age_c","type_dm"),
344:                    lev_var_cat=classlabel(weightth),
345:                    lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
346:                    lab_var_cat="Weight",
347:                    lab_class=c("Gender","Age Classes","Type of Diabetes"),
348:                    tab=1,
349:                    tabside="H",
350:                    tabperc=TRUE,
351:                    tabwidth=1,
352:                    barbeside=TRUE,
353:                    barnumber=TRUE,
354:                    barperc=TRUE,
```

```
355:                    box=1,
356:                    bar=1,
357:                    lines=printlines,
358:                    varlines="weight",
359:                    timelines="year_epi",
360:                    trellis=0,
361:                    pie=0,
362:                    texfile=texfile,
363:                    dirtables=dirtables,
364:                    dircsv=dirdataout,
365:                    dirgraph=dirgraph,
366:                    dirgraphlatex=dirgraphlatex,
367:                    cex=cex
368:                    )
369:
370:
371:     rm(data2_2_1_1)
372:     }
373:   }
374:      rm(data2_2_1_1l)
375:   }  else BIRO_report_toc(title="2.2.1.1. Weight (last episode in 12 months)")
376:
377:
378:   #
379:   # 2.2.1.2 BMI (the last episode, not only refyear)
380:   #
381:
382:  if  ((length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0 &
383:        length(na.omit(patient$sex))>0 & length(na.omit(episode$bmi))>0) |
384:       (length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0 &
385:        length(na.omit(patient$sex))>0 & length(na.omit(episode$weight))>0 &
386:        length(na.omit(episode$height))>0)) {
387:
388:     for (thisyear in anayears) {
389:
390:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
391:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
392:
393:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
394:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
395:
396:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
397:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
398:
399:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
```

```
  400:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  401:
  402:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  403:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  404:
  405:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  406:
  407:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  408:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  409:
  410:     if (dim(episode[is.na(episode$bmi)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
  411:
  412:     episode$bmi<-ifelse(is.na(episode$bmi),
  413:                         episode$weight/(episode$height)^2,
  414:                         episode$bmi)
  415:     episode$bmi<-ifelse(episode$bmi>=bmilimit[1] & episode$bmi<=bmilimit[2],
  416:                         episode$bmi,
  417:                         NA)
  418:     episode$bmi_c<-varclass(episode$bmi,c(25,27,30,40))
  419:
  420:     data2_2_1_2<-BIRO_demographic(patient=patient,
  421:                                   episode=episode[is.na(episode$bmi)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  422:                                   var=c("sex","type_dm","age_c"),
  423:                                   date="last episode date",
  424:                                   startvar="dob",
  425:                                   th=ageth,
  426:                                   hold=TRUE,
  427:                                   varname="age")
  428:
  429:     data2_2_1_2<-merge(episode[is.na(episode$bmi)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,
  430:                           c("patient_id","epi_date","bmi","bmi_c")],
  431:                        data2_2_1_2,
  432:                        by.x=c("patient_id","epi_date"),
  433:                        by.y=c("patient_id","last_epi_date"),
  434:                        all.y=TRUE)
  435:
  436:     BIRO_dframe(dataname="data2_2_1_2",monitor=TRUE,filelog=logfile)
  437:     BIRO_dframe(dataname="data2_2_1_2",monitor=TRUE,filelog="")
  438:
  439: }
  440:
  441:    if (thisyear==anayears[1]) {
```

```
  442:     data2_2_1_2l<-data2_2_1_2
  443:     data2_2_1_2l$year_epi=thisyear
  444:    } else {
  445:     data2_2_1_2$year_epi=thisyear
  446:     data2_2_1_2l<-rbind(data2_2_1_2l,data2_2_1_2)
  447:    }
  448:
  449:     BIRO_dframe(dataname="data2_2_1_2l",monitor=TRUE,filelog=logfile)
  450:     BIRO_dframe(dataname="data2_2_1_2l",monitor=TRUE,filelog="")
  451:
  452:     printlines=0
  453:     if (thisyear==anayears[length(anayears)]) {
  454:       lines=1
  455:       printlines=1
  456:       if (length(anayears)==1) {
  457:         printlines=0
  458:       }
  459:     }
  460:
  461:     if (dim(episode[is.na(episode$bmi)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
  462:
  463:        BIRO_report(ind="2_2_1_2",
  464:                    title="2.2.1.2. BMI (last episode in 12 months)",
  465:                    var="bmi",
  466:                    var_cat="bmi_c",
  467:                    class=c("sex","age_c","type_dm"),
  468:                    lev_var_cat=classlabel(bmith),
  469:                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
  470:                    lab_var_cat="BMI",
  471:                    lab_class=c("Gender","Age Classes","Type of Diabetes"),
  472:                    tab=1,
  473:                    tabside="H",
  474:                    tabperc=TRUE,
  475:                    tabwidth=1,
  476:                    barbeside=TRUE,
  477:                    barnumber=TRUE,
  478:                    barperc=TRUE,
  479:                    box=1,
  480:                    bar=1,
  481:                    lines=printlines,
  482:                    varlines="bmi",
  483:                    timelines="year_epi",
  484:                    trellis=1,
  485:                    pie=0,
```

```
                    texfile=texfile,
                    dirtables=dirtables,
                    dircsv=dirdataout,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex
                    )

    rm(data2_2_1_2)
  }
  }
    rm(data2_2_1_2l)

  } else BIRO_report_toc(title="2.2.1.2. BMI (last episode in 12 months)")

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

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


  if (length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0 &
      length(na.omit(patient$sex))>0 & length(na.omit(episode$smok_stat))>0) {

    for (thisyear in anayears) {

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

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

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

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

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

   assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)

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

```
  531:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  532:
  533:     if (dim(episode[is.na(episode$smok_stat)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
  534:
  535:   data2_2_2_1<-BIRO_demographic(patient=patient,
  536:                                 episode=episode[is.na(episode$smok_stat)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  537:                                 var=c("sex","type_dm","age_c"),
  538:                                 date="last episode date",
  539:                                 startvar="dob",
  540:                                 th=ageth,
  541:                                 hold=TRUE,
  542:                                 varname="age")
  543:
  544:   data2_2_2_1<-merge(episode[is.na(episode$smok_stat)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,
  545:                         c("patient_id","epi_date","smok_stat")],
  546:                      data2_2_2_1,
  547:                      by.x=c("patient_id","epi_date"),
  548:                      by.y=c("patient_id","last_epi_date"),
  549:                      all.y=TRUE)
  550:
  551:   BIRO_dframe(dataname="data2_2_2_1",monitor=TRUE,filelog=logfile)
  552:   BIRO_dframe(dataname="data2_2_2_1",monitor=TRUE,filelog="")
  553:
  554:
  555:       BIRO_report(ind="2_2_2_1",
  556:                   title="2.2.2.1 Smoking status (last episode in 12 months)",
  557:                   var_cat="smok_stat",
  558:                   class=c("sex","age_c","type_dm"),
  559:                   lev_var_cat=levsmok_stat,
  560:                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
  561:                   lab_var_cat="Smoking status",
  562:                   lab_class=c("Gender","Age Classes","Type of Diabetes"),
  563:                   tab=1,
  564:                   tabside="H",
  565:                   tabperc=TRUE,
  566:                   tabwidth=1,
  567:                   barbeside=TRUE,
  568:                   barnumber=TRUE,
  569:                   barperc=TRUE,
  570:                   box=1,
  571:                   bar=1,
  572:                   lines=0,
```

```
573:                     trellis=1,
574:                     pie=0,
575:                     texfile=texfile,
576:                     dirtables=dirtables,
577:                     dircsv=dirdataout,
578:                     dirgraph=dirgraph,
579:                     dirgraphlatex=dirgraphlatex,
580:                     cex=cex
581:                     )
582:
583:   rm(data2_2_2_1)
584:   }
585:   }
586:   } else BIRO_report_toc(title="2.2.2.1 Smoking status (last episode in 12 months)")
587:
588: new_sub_section_html(title="2.2.3. Clinical measurements",htmlfile=htmlfile)
589:   #
590:   # 2.2.3.1 Systolic blood pressure (the last episode, not only refyear)
591:   #
592:
593:
594:   if (length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0 &
595:       length(na.omit(patient$sex))>0 & length(na.omit(episode$sbp))>0) {
596:
597:     for (thisyear in anayears) {
598:
599:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
600:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
601:
602:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
603:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
604:
605:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
606:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
607:
608:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
609:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
610:
611:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
612:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
613:
614:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
615:
616:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
617:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
```

```
  618:
  619:     if (dim(episode[is.na(episode$sbp)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
  620:
  621:     episode$sbp<-ifelse(episode$sbp>=sbplimit[1] &
  622:                           episode$sbp<=sbplimit[2],episode$sbp,NA)
  623:     episode$sbp_c<-varclass(episode$sbp,sbpth)
  624:     data2_2_3_1<-BIRO_demographic(patient=patient,
  625:                                   episode=episode[is.na(episode$sbp)==FALSE,],
  626:                                   var=c("sex","type_dm","age_c"),
  627:                                   date="last episode date",
  628:                                   startvar="dob",
  629:                                   th=ageth,
  630:                                   hold=TRUE,
  631:                                   varname="age")
  632:     data2_2_3_1<-merge(episode[is.na(episode$sbp)==FALSE,
  633:                           c("patient_id","epi_date","sbp","sbp_c")],
  634:                        data2_2_3_1,
  635:                        by.x=c("patient_id","epi_date"),
  636:                        by.y=c("patient_id","last_epi_date"),
  637:                        all.y=TRUE)
  638:
  639:     BIRO_dframe(dataname="data2_2_3_1",monitor=TRUE,filelog=logfile)
  640:     BIRO_dframe(dataname="data2_2_3_1",monitor=TRUE,filelog="")
  641:
  642:     }
  643:
  644:    if (thisyear==anayears[1]) {
  645:     data2_2_3_1l<-data2_2_3_1
  646:     data2_2_3_1l$year_epi=thisyear
  647:    } else {
  648:     data2_2_3_1$year_epi=thisyear
  649:     data2_2_3_1l<-rbind(data2_2_3_1l,data2_2_3_1)
  650:    }
  651:
  652:
  653:     BIRO_dframe(dataname="data2_2_3_1l",monitor=TRUE,filelog=logfile)
  654:     BIRO_dframe(dataname="data2_2_3_1l",monitor=TRUE,filelog="")
  655:
  656:     printlines=0
  657:     if (thisyear==anayears[length(anayears)]) {
  658:       lines=1
  659:       printlines=1
  660:       if (length(anayears)==1) {
  661:         printlines=0
```

```
  662:       }
  663:     }
  664:
  665:     if (dim(episode[is.na(episode$sbp)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
  666:
  667:        BIRO_report(ind="2_2_3_1",
  668:                    title="2.2.3.1. Systolic BP (last episode in 12 months)",
  669:                    var="sbp",
  670:                    var_cat="sbp_c",
  671:                    class=c("sex","age_c","type_dm"),
  672:                    lev_var_cat=classlabel(sbpth),
  673:                    lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
  674:                    lab_var_cat="SBP",
  675:                    lab_class=c("Gender","Age Classes","Type of Diabetes"),
  676:                    tab=1,
  677:                    tabside="H",
  678:                    tabperc=TRUE,
  679:                    tabwidth=1,
  680:                    barbeside=TRUE,
  681:                    barnumber=TRUE,
  682:                    barperc=TRUE,
  683:                    box=1,
  684:                    bar=1,
  685:                    lines=lines,
  686:                    printlines=printlines,
  687:                    varlines="sbp",
  688:                    timelines="year_epi",
  689:                    trellis=1,
  690:                    pie=0,
  691:                    texfile=texfile,
  692:                    dirtables=dirtables,
  693:                    dircsv=dirdataout,
  694:                    dirgraph=dirgraph,
  695:                    dirgraphlatex=dirgraphlatex,
  696:                    cex=cex
  697:                    )
  698:
  699:     rm(data2_2_3_1)
  700:   }
  701:     }
  702:     rm(data2_2_3_1l)
  703:
  704:   } else BIRO_report_toc(title="2.2.3.1. Systolic BP (last episode in 12 months)")
  705:
```

```
  706:   #
  707:   # 2.2.3.2 Diastolic Blood Pressure (the last episode, not only refyear)
  708:   #
  709:
  710:   if (length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0 &
  711:       length(na.omit(patient$sex))>0 & length(na.omit(episode$dbp))>0) {
  712:
  713:     for (thisyear in anayears) {
  714:
  715:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  716:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  717:
  718:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  719:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  720:
  721:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  722:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  723:
  724:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  725:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  726:
  727:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  728:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  729:
  730:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  731:
  732:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  733:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  734:
  735:     if (dim(episode[is.na(episode$dbp)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
  736:
  737:     episode$dbp<-ifelse(episode$dbp>=dbplimit[1] &
  738:                         episode$dbp<=dbplimit[2],episode$dbp,NA)
  739:
  740:     episode$dbp_c<-varclass(episode$dbp,dbpth)
  741:
  742:     data2_2_3_2<-BIRO_demographic(patient=patient,
  743:                                   episode=episode[is.na(episode$dbp)==FALSE,],
  744:                                   var=c("sex","type_dm","age_c"),
  745:                                   date="last episode date",
  746:                                   startvar="dob",
  747:                                   th=ageth,
  748:                                   hold=TRUE,
  749:                                   varname="age")
```

```
  750:     data2_2_3_2<-merge(episode[is.na(episode$dbp)==FALSE,
  751:                           c("patient_id","epi_date","dbp","dbp_c")],
  752:                        data2_2_3_2,
  753:                        by.x=c("patient_id","epi_date"),
  754:                        by.y=c("patient_id","last_epi_date"),
  755:                        all.y=TRUE)
  756:
  757:     BIRO_dframe(dataname="data2_2_3_2",monitor=TRUE,filelog=logfile)
  758:     BIRO_dframe(dataname="data2_2_3_2",monitor=TRUE,filelog="")
  759:
  760:     }
  761:
  762:    if (thisyear==anayears[1]) {
  763:     data2_2_3_2l<-data2_2_3_2
  764:     data2_2_3_2l$year_epi=thisyear
  765:    } else {
  766:     data2_2_3_2$year_epi=thisyear
  767:     data2_2_3_2l<-rbind(data2_2_3_2l,data2_2_3_2)
  768:    }
  769:
  770:     BIRO_dframe(dataname="data2_2_3_2l",monitor=TRUE,filelog=logfile)
  771:     BIRO_dframe(dataname="data2_2_3_2l",monitor=TRUE,filelog="")
  772:
  773:     printlines=0
  774:     if (thisyear==anayears[length(anayears)]) {
  775:       lines=1
  776:       printlines=1
  777:       if (length(anayears)==1) {
  778:         printlines=0
  779:       }
  780:     }
  781:
  782:     if (dim(episode[is.na(episode$dbp)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
  783:
  784:        BIRO_report(ind="2_2_3_2",
  785:                    title="2.2.3.2. Diastolic BP (last episode in 12 months)",
  786:                    var="dbp",
  787:                    var_cat="dbp_c",
  788:                    class=c("sex","age_c","type_dm"),
  789:                    lev_var_cat=classlabel(dbpth),
  790:                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
  791:                    lab_var_cat="DBP",
  792:                    lab_class=c("Gender","Age Classes","Type of Diabetes"),
  793:                    tab=1,
```

```
794:                     tabside="H",
795:                     tabperc=TRUE,
796:                     tabwidth=1,
797:                     barbeside=TRUE,
798:                     barnumber=TRUE,
799:                     barperc=TRUE,
800:                     box=1,
801:                     bar=1,
802:                     lines=lines,
803:                     printlines=printlines,
804:                     varlines="dbp",
805:                     timelines="year_epi",
806:                     trellis=1,
807:                     pie=0,
808:                     texfile=texfile,
809:                     dirtables=dirtables,
810:                     dircsv=dirdataout,
811:                     dirgraph=dirgraph,
812:                     dirgraphlatex=dirgraphlatex,
813:                     cex=cex
814:                     )
815:
816:     rm(data2_2_3_2)
817:   }
818:   }
819:     rm(data2_2_3_2l)
820:
821:   } else BIRO_report_toc(title="2.2.3.2. Diastolic BP")
822:
823:   #
824:   # 2.2.3.2 Total cholesterol (the last episode, not only refyear)
825:   #
826:
827:   if (length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0 &
828:       length(na.omit(patient$sex))>0 & length(na.omit(episode$chol))>0) {
829:
830:     episode$chol<-ifelse(episode$chol>=chollimit[1] &
831:                            episode$chol<=chollimit[2],episode$chol,NA)
832:     episode$chol_c<-varclass(episode$chol,cholth)
833:
834:
835:     for (thisyear in anayears) {
836:
837:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
838:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
  839:
  840:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  841:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  842:
  843:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  844:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  845:
  846:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  847:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  848:
  849:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  850:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  851:
  852:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  853:
  854:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  855:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  856:
  857:     if (dim(episode[is.na(episode$chol)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
  858:
  859:     data2_2_3_3<-BIRO_demographic(patient=patient,
  860:                                   episode=episode[is.na(episode$chol)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  861:                                   var=c("sex","type_dm","age_c"),
  862:                                   date="last episode date",
  863:                                   startvar="dob",
  864:                                   th=ageth,
  865:                                   hold=TRUE,
  866:                                   varname="age")
  867:
  868:     data2_2_3_3<-merge(episode[is.na(episode$chol)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,
  869:                           c("patient_id","epi_date","chol","chol_c")],
  870:                        data2_2_3_3,
  871:                        by.x=c("patient_id","epi_date"),
  872:                        by.y=c("patient_id","last_epi_date"),
  873:                        all.y=TRUE)
  874:
  875:     BIRO_dframe(dataname="data2_2_3_3",monitor=TRUE,filelog=logfile)
  876:     BIRO_dframe(dataname="data2_2_3_3",monitor=TRUE,filelog="")
  877:
  878:     }
  879:
  880:    if (thisyear==anayears[1]) {
```

```
  881:     data2_2_3_3l<-data2_2_3_3
  882:     data2_2_3_3l$year_epi=thisyear
  883:    } else {
  884:     data2_2_3_3$year_epi=thisyear
  885:     data2_2_3_3l<-rbind(data2_2_3_3l,data2_2_3_3)
  886:    }
  887:
  888:
  889:     BIRO_dframe(dataname="data2_2_3_3l",monitor=TRUE,filelog=logfile)
  890:     BIRO_dframe(dataname="data2_2_3_3l",monitor=TRUE,filelog="")
  891:
  892:
  893:     printlines=0
  894:     if (thisyear==anayears[length(anayears)]) {
  895:       lines=1
  896:       printlines=1
  897:       if (length(anayears)==1) {
  898:         printlines=0
  899:       }
  900:     }
  901:
  902:     if (dim(episode[is.na(episode$chol)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
  903:
  904:        BIRO_report(ind="2_2_3_3",
  905:                    title="2.2.3.3. Total cholesterol (last episode in 12 months)",
  906:                    var="chol",
  907:                    var_cat="chol_c",
  908:                    class=c("sex","age_c","type_dm"),
  909:                    lev_var_cat=classlabel(cholth),
  910:                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
  911:                    lab_var_cat="Cholesterol",
  912:                    lab_class=c("Gender","Age Classes","Type of Diabetes"),
  913:                    tab=1,
  914:                    tabside="H",
  915:                    tabperc=TRUE,
  916:                    tabwidth=1,
  917:                    barbeside=TRUE,
  918:                    barnumber=TRUE,
  919:                    barperc=TRUE,
  920:                    box=1,
  921:                    bar=1,
  922:                    lines=lines,
  923:                    printlines=printlines,
  924:                    varlines="chol",
```

```
925:                     timelines="year_epi",
926:                     trellis=1,
927:                     pie=0,
928:                     texfile=texfile,
929:                     dirtables=dirtables,
930:                     dircsv=dirdataout,
931:                     dirgraph=dirgraph,
932:                     dirgraphlatex=dirgraphlatex,
933:                     cex=cex
934:                     )
935:
936:      rm(data2_2_3_3)
937:      }
938:      }
939:      rm(data2_2_3_3l)
940:        } else BIRO_report_toc(title="2.2.3.3. Total cholesterol (last episode in 12 months)")
941:
942:    #
943:    # 2.2.3.4 HDL-cholesterol (the last episode, not only refyear)
944:    #
945:
946:    if (length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0 &
947:        length(na.omit(patient$sex))>0 & length(na.omit(episode$hdl))>0) {
948:
949:      episode$hdl<-ifelse(episode$hdl>=hdllimit[1] & episode$hdl<=hdllimit[2],
950:                          episode$hdl,NA)
951:
952:      episode$hdl_c<-varclass(episode$hdl,hdlth)
953:
954:      for (thisyear in anayears) {
955:
956:     assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
957:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
958:
959:     assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
960:           "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
961:
962:     assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
963:           "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
964:
965:     assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
966:            "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
967:
968:     assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
969:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
  970:
  971:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  972:
  973:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  974:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  975:
  976:     if (dim(episode[is.na(episode$hdl)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
  977:
  978:
  979:     data2_2_3_4<-BIRO_demographic(patient=patient,
  980:                                   episode=episode[is.na(episode$hdl)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  981:                                   var=c("sex","type_dm","age_c"),
  982:                                   date="last episode date",
  983:                                   startvar="dob",
  984:                                   th=ageth,
  985:                                   hold=TRUE,
  986:                                   varname="age")
  987:
  988:     data2_2_3_4<-merge(episode[is.na(episode$hdl)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,
  989:                           c("patient_id","epi_date","hdl","hdl_c")],
  990:                        data2_2_3_4,
  991:                        by.x=c("patient_id","epi_date"),
  992:                        by.y=c("patient_id","last_epi_date"),
  993:                        all.y=TRUE)
  994:
  995:     BIRO_dframe(dataname="data2_2_3_4",monitor=TRUE,filelog=logfile)
  996:     BIRO_dframe(dataname="data2_2_3_4",monitor=TRUE,filelog="")
  997:
  998:     }
  999:
 1000:    if (thisyear==anayears[1]) {
 1001:     data2_2_3_4l<-data2_2_3_4
 1002:     data2_2_3_4l$year_epi=thisyear
 1003:    } else {
 1004:     data2_2_3_4$year_epi=thisyear
 1005:     data2_2_3_4l<-rbind(data2_2_3_4l,data2_2_3_4)
 1006:    }
 1007:
 1008:     BIRO_dframe(dataname="data2_2_3_4l",monitor=TRUE,filelog=logfile)
 1009:     BIRO_dframe(dataname="data2_2_3_4l",monitor=TRUE,filelog="")
 1010:
 1011:     printlines=0
```

```
1012:     if (thisyear==anayears[length(anayears)]) {
1013:       lines=1
1014:       printlines=1
1015:       if (length(anayears)==1) {
1016:         printlines=0
1017:       }
1018:     }
1019:
1020:     if (dim(episode[is.na(episode$hdl)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
1021:
1022:        BIRO_report(ind="2_2_3_4",
1023:                    title="2.2.3.4. HDL-cholesterol (last episode in 12 months)",
1024:                    var="hdl",
1025:                    var_cat="hdl_c",
1026:                    class=c("sex","age_c","type_dm"),
1027:                    lev_var_cat=classlabel(hdlth),
1028:                    lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
1029:                    lab_var_cat="HDL",
1030:                    lab_class=c("Gender","Age Classes","Type of Diabetes"),
1031:                    tab=1,
1032:                    tabside="H",
1033:                    tabperc=TRUE,
1034:                    tabwidth=1,
1035:                    barbeside=TRUE,
1036:                    barnumber=TRUE,
1037:                    barperc=TRUE,
1038:                    box=1,
1039:                    bar=1,
1040:                    lines=lines,
1041:                    printlines=printlines,
1042:                    varlines="hdl",
1043:                    timelines="year_epi",
1044:                    trellis=1,
1045:                    pie=0,
1046:                    texfile=texfile,
1047:                    dirtables=dirtables,
1048:                    dircsv=dirdataout,
1049:                    dirgraph=dirgraph,
1050:                    dirgraphlatex=dirgraphlatex,
1051:                    cex=cex
1052:                    )
1053:
1054:
1055:     rm(data2_2_3_4)
```

```
1056:     }
1057:     }
1058:     rm(data2_2_3_4l)
1059:
1060:   } else BIRO_report_toc(title="2.2.3.4. HDL-cholesterol (last episode in 12 months)")
1061:
1062:   #
1063:   # 2.2.3.5 creatinine (the last episode, not only refyear)
1064:   #
1065:
1066:   if (length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0 &
1067:       length(na.omit(patient$sex))>0 & length(na.omit(episode$creat))>0) {
1068:
1069:     episode$creat<-ifelse(episode$creat>=creatlimit[1] &
1070:                           episode$creat<=creatlimit[2],episode$creat,NA)
1071:     episode$creat_c<-varclass(episode$creat,creatth)
1072:
1073:     for (thisyear in anayears) {
1074:
1075:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1076:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1077:
1078:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1079:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1080:
1081:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1082:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1083:
1084:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1085:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
1086:
1087:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
1088:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1089:
1090:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
1091:
1092:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1093:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1094:
1095:     if (dim(episode[is.na(episode$chol)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
1096:
1097:     data2_2_3_5<-BIRO_demographic(patient=patient,
1098:                                   episode=episode[is.na(episode$creat)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
```

```
1099:                                    var=c("sex","type_dm","age_c"),
1100:                                    date="last episode date",
1101:                                    startvar="dob",
1102:                                    th=ageth,
1103:                                    hold=TRUE,
1104:                                    varname="age")
1105:
1106:     data2_2_3_5<-merge(episode[is.na(episode$creat)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,
1107:                                c("patient_id","epi_date","creat","creat_c")],
1108:                        data2_2_3_5,
1109:                        by.x=c("patient_id","epi_date"),
1110:                        by.y=c("patient_id","last_epi_date"),
1111:                        all.y=TRUE)
1112:
1113:     BIRO_dframe(dataname="data2_2_3_5",monitor=TRUE,filelog=logfile)
1114:     BIRO_dframe(dataname="data2_2_3_5",monitor=TRUE,filelog="")
1115:
1116:     }
1117:
1118:    if (thisyear==anayears[1]) {
1119:     data2_2_3_5l<-data2_2_3_5
1120:     data2_2_3_5l$year_epi=thisyear
1121:    } else {
1122:     data2_2_3_5$year_epi=thisyear
1123:     data2_2_3_5l<-rbind(data2_2_3_5l,data2_2_3_5)
1124:    }
1125:
1126:
1127:     BIRO_dframe(dataname="data2_2_3_5l",monitor=TRUE,filelog=logfile)
1128:     BIRO_dframe(dataname="data2_2_3_5l",monitor=TRUE,filelog="")
1129:
1130:     if (thisyear==anayears[length(anayears)]) printlines=1 else printlines=0
1131:     printlines=0
1132:     if (thisyear==anayears[length(anayears)]) {
1133:       lines=1
1134:       printlines=1
1135:       if (length(anayears)==1) {
1136:         printlines=0
1137:       }
1138:     }
1139:
1140:     if (dim(episode[is.na(episode$hdl)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
1141:
```

```
1142:
1143:        BIRO_report(ind="2_2_3_5",
1144:                    title="2.2.3.5. Creatinine (last episode in 12 months)",
1145:                    var="creat",
1146:                    var_cat="creat_c",
1147:                    class=c("sex","age_c","type_dm"),
1148:                    lev_var_cat=classlabel(creatth),
1149:                   lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
1150:                    lab_var_cat="CREATININE",
1151:                    lab_class=c("Gender","Age Classes","Type of Diabetes"),
1152:                    tab=1,
1153:                    tabside="H",
1154:                    tabperc=TRUE,
1155:                    tabwidth=1,
1156:                    barbeside=TRUE,
1157:                    barnumber=TRUE,
1158:                    barperc=TRUE,
1159:                    box=1,
1160:                    bar=1,
1161:                    lines=lines,
1162:                    printlines=printlines,
1163:                    varlines="creat",
1164:                    timelines="year_epi",
1165:                    trellis=1,
1166:                    pie=0,
1167:                    texfile=texfile,
1168:                    dirtables=dirtables,
1169:                    dircsv=dirdataout,
1170:                    dirgraph=dirgraph,
1171:                    dirgraphlatex=dirgraphlatex,
1172:                    cex=cex
1173:                    )
1174:
1175:     rm(data2_2_3_5)
1176:     }
1177:     }
1178:     rm(data2_2_3_5l)
1179:
1180:   } else BIRO_report_toc(title="2.2.3.5. Creatinine")
1181:
1182:   #
1183:   # 2.2.3.6 hba1c (the last episode, not only refyear)
1184:   #
1185:
1186:
```

```
 1187:   if (length(na.omit(patient$type_dm))>0 & length(na.omit(patient$dob))>0 &
 1188:       length(na.omit(patient$sex))>0 & length(na.omit(episode$hba1c))>0) {
 1189:
 1190:     episode$hba1c<-as.numeric(as.character(episode$hba1c))
 1191:     episode$hba1c<-ifelse(episode$hba1c>=hba1climit[1] &
 1192:                           episode$hba1c<=hba1climit[2],episode$hba1c,NA)
 1193:     episode$hba1c_c<-varclass(episode$hba1c,hba1cth)
 1194:
 1195:     for (thisyear in anayears) {
 1196:
 1197:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
 1198:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1199:
 1200:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
 1201:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 1202:
 1203:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 1204:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 1205:
 1206:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1207:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1208:
 1209:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1210:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1211:
 1212:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1213:
 1214:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 1215:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 1216:
 1217:     if (dim(episode[is.na(episode$hba1c)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
 1218:
 1219:     data2_2_3_6<-BIRO_demographic(patient=patient,
 1220:                                   episode=episode[is.na(episode$hba1c)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
 1221:                                   var=c("sex","type_dm","age_c"),
 1222:                                   date="last episode date",
 1223:                                   startvar="dob",
 1224:                                   th=ageth,
 1225:                                   hold=TRUE,
 1226:                                   varname="age")
 1227:
 1228:     data2_2_3_6<-merge(episode[is.na(episode$hba1c)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,
```

```
1229:                          c("patient_id","epi_date","hba1c","hba1c_c")],
1230:                        data2_2_3_6,
1231:                        by.x=c("patient_id","epi_date"),
1232:                        by.y=c("patient_id","last_epi_date"),
1233:                        all.y=TRUE)
1234:
1235:    if (thisyear==anayears[1]) {
1236:     data2_2_3_6l<-data2_2_3_6
1237:     data2_2_3_6l$year_epi=thisyear
1238:    } else {
1239:     data2_2_3_6$year_epi=thisyear
1240:     data2_2_3_6l<-rbind(data2_2_3_6l,data2_2_3_6)
1241:    }
1242:
1243:
1244:
1245:     BIRO_dframe(dataname="data2_2_3_6",monitor=TRUE,filelog=logfile)
1246:     BIRO_dframe(dataname="data2_2_3_6",monitor=TRUE,filelog="")
1247:
1248:     }
1249:
1250:
1251:     BIRO_dframe(dataname="data2_2_3_6l",monitor=TRUE,filelog=logfile)
1252:     BIRO_dframe(dataname="data2_2_3_6l",monitor=TRUE,filelog="")
1253:
1254:     printlines=0
1255:     if (thisyear==anayears[length(anayears)]) {
1256:       lines=1
1257:       printlines=1
1258:       if (length(anayears)==1) {
1259:         printlines=0
1260:       }
1261:     }
1262:
1263:     if (dim(episode[is.na(episode$hba1c)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
1264:
1265:        BIRO_report(ind="2_2_3_6",
1266:                    title="2.2.3.6. HbA1c (last episode in 12 months)",
1267:                    var="hba1c",
1268:                    var_cat="hba1c_c",
1269:                    class=c("sex","age_c","type_dm"),
1270:                    lev_var_cat=classlabel(hba1cth),
1271:                    lev_class=list(a=levsex,b=classlabellist(ageth),c=levtype_dm),
1272:                    lab_var_cat="HbA1c",
```

```
1273:                     lab_class=c("Gender","Age Classes","Type of Diabetes"),
1274:                     tab=1,
1275:                     tabside="H",
1276:                     tabperc=TRUE,
1277:                     tabwidth=1,
1278:                     barbeside=TRUE,
1279:                     barnumber=TRUE,
1280:                     barperc=TRUE,
1281:                     box=1,
1282:                     bar=1,
1283:                     lines=lines,
1284:                     printlines=printlines,
1285:                     varlines="hba1c",
1286:                     timelines="year_epi",
1287:                     trellis=1,
1288:                     pie=0,
1289:                     texfile=texfile,
1290:                     dirtables=dirtables,
1291:                     dircsv=dirdataout,
1292:                     dirgraph=dirgraph,
1293:                     dirgraphlatex=dirgraphlatex,
1294:                     cex=cex
1295:                     )
1296:
1297:     rm(data2_2_3_6)
1298:     }
1299:     }
1300:     rm(data2_2_3_6l)
1301:
1302:   } else BIRO_report_toc(title="2.2.3.6. HbA1c (last episode in 12 months)")
1303:
1304:   new_section(title="2.3. Diabetes complications",texfile=texfile)
1305:   new_section_html(title="2.3. Diabetes complications",htmlfile=htmlfile)
1306:   #
1307:   # 2.3.1 retinopathy (the first episode, not only refyear)
1308:   #
1309:
1310:   if ((is.null(patient$dt_diag)==F) & (is.null(episode$retin_l)==F)& (is.null(episode$retin_r)==F)) {
1311:   if ((length(na.omit(patient$dt_diag))>0) & (length(na.omit(episode$retin_l))>0) &
1312:       (length(na.omit(episode$retin_r))>0)) {
1313:
1314:
1315:   for (thisyear in anayears) {
1316:
1317:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
```

```
 1318:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1319:
 1320:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
 1321:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 1322:
 1323:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 1324:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 1325:
 1326:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1327:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1328:
 1329:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1330:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1331:
 1332:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1333:
 1334:   checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 1335:   oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 1336:
 1337:   if (dim(episode[is.na(episode$hba1c)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
 1338:
 1339:
 1340:   episode$retin<-ifelse(((is.na(episode$retin_l)==FALSE & episode$retin_l==1)) |
 1341:                         ((is.na(episode$retin_r)==FALSE & episode$retin_r==1)),1,2)
 1342:
 1343:
 1344:   data2_3_1<-BIRO_demographic(patient=patient,
 1345:                               episode=episode[episode$retin==1 & is.na(episode$retin)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
 1346:                               var=c("durdiab_c"),
 1347:                               date="first episode date",
 1348:                               startvar="dt_diag",
 1349:                               th=durdiabth,
 1350:                               hold=TRUE,
 1351:                               varname="durdiab")
 1352:
 1353:   data2_3_1$retin=1;
 1354:
 1355:   if (length(episode$retin[episode$retin==2])>0) {
 1356:
 1357:     noretin<-BIRO_demographic(patient=patient,
 1358:                               episode=episode[episode$retin==2 & is.na(episode$retin)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
 1359:                               var=c("durdiab_c"),
```

```
1360:                                 date="first episode date",
1361:                                 startvar="dt_diag",
1362:                                 th=durdiabth,
1363:                                 hold=TRUE,
1364:                                 varname="durdiab")
1365:
1366:     noretin$retin=2;
1367:
1368:     data2_3_1<-merge(data2_3_1,noretin,by="patient_id",all.x=TRUE,all.y=TRUE)
1369:
1370:
1371:     data2_3_1$retin=ifelse(is.na(data2_3_1$retin.x)==TRUE,data2_3_1$retin.y,data2_3_1$retin.x)
1372:     data2_3_1$durdiab_c=ifelse(is.na(data2_3_1$durdiab_c.x)==TRUE,data2_3_1$durdiab_c.y,data2_3_1$durdiab_c.x)
1373:     data2_3_1$first_epi_date=ifelse(is
.na(data2_3_1$first_epi_date.x)==TRUE,data2_3_1$first_epi_date.y,data2_3_1$first_epi_date.x)
1374:     data2_3_1$first_epi_date<-format(data2_3_1$first_epi_date,format="%Y-%m-%d")
1375:   }
1376:   data2_3_1<-data2_3_1[,c("patient_id","first_epi_date","retin","durdiab_c")]
1377:
1378: # Delete duplicated rows
1379:
1380:   unique2_3_1<-aggregate(data2_3_1$retin,by=list(patient_id= data2_3_1$patient_id),min)
1381:   names(unique2_3_1)<-c("patient_id","retin")
1382:
1383:   data2_3_1<-merge(unique2_3_1,unique(data2_3_1[-pmatch("retin",names(data2_3_1))]),by=c("patient_id"
),all.x=TRUE,all.y=FALSE)
1384:
1385:   BIRO_dframe(dataname="data2_3_1",monitor=TRUE,filelog=logfile)
1386:   BIRO_dframe(dataname="data2_3_1",monitor=TRUE,filelog="")
1387:
1388:     if (dim(episode[is.na(episode$retin)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
1389:
1390:       BIRO_report(ind="2_3_1",
1391:                   title="2.3.1. Retinopathy (first episode in 12 months)",
1392:                   var_cat="retin",
1393:                   class=c("durdiab_c"),
1394:                   lev_var_cat=levretin,
1395:                   lev_class=list(a=classlabellist(durdiabth)),
1396:                   lab_var_cat="Retinopathy",
1397:                   lab_class=c("Duration"),
1398:                   tab=1,
1399:                   tabside="H",
1400:                   tabperc=TRUE,
1401:                   tabwidth=1,
```

```
1402:                     barbeside=TRUE,
1403:                     barnumber=TRUE,
1404:                     barperc=TRUE,
1405:                     box=0,
1406:                     bar=1,
1407:                     lines=0,
1408:                     trellis=0,
1409:                     pie=0,
1410:                     texfile=texfile,
1411:                     dirtables=dirtables,
1412:                     dircsv=dirdataout,
1413:                     dirgraph=dirgraph,
1414:                     dirgraphlatex=dirgraphlatex,
1415:                     cex=cex
1416:                     )
1417:
1418:
1419:   rm(data2_3_1)
1420:     }
1421:   }
1422:   }
1423:       }
1424:   } else BIRO_report_toc(title="2.3.1. Retinopathy (first episode in 12 months)")
1425:
1426:
1427:   #
1428:   # 2.3.2 esrf (the first episode, not only refyear)
1429:   #
1430:
1431:   if ((is.null(patient$dt_diag)==F) & (is.null(episode$esrf)==F)) {
1432:   if (length(na.omit(patient$dt_diag))>0 & length(na.omit(episode$esrf))>0) {
1433:
1434:   for (thisyear in anayears) {
1435:
1436:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1437:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1438:
1439:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1440:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1441:
1442:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1443:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1444:
1445:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1446:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
```

```
 1447:
 1448:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1449:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1450:
 1451:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1452:
 1453:   checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 1454:   oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 1455:
 1456:   if (dim(episode[is.na(episode$esrf)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
 1457:
 1458:
 1459:   data2_3_2<-BIRO_demographic(patient=patient,
 1460:                               episode=episode[episode$esrf==1 & is.na(episode$esrf)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
 1461:                               var=c("durdiab_c"),
 1462:                               date="first episode date",
 1463:                               startvar="dt_diag",
 1464:                               th=durdiabth,
 1465:                               hold=TRUE,
 1466:                               varname="durdiab")
 1467:
 1468:   data2_3_2$esrf=1;
 1469:   if (length(episode$esrf[episode$esrf==2 & is.na(episode$hypertension)==FALSE])>0) {
 1470:
 1471:     noesrf<-BIRO_demographic(patient=patient,
 1472:                              episode=episode[episode$esrf==2 & is.na(episode$esrf)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
 1473:                              var=c("durdiab_c"),
 1474:                              date="first episode date",
 1475:                              startvar="dt_diag",
 1476:                              th=durdiabth,
 1477:                              hold=TRUE,
 1478:                              varname="durdiab")
 1479:
 1480:     noesrf$esrf=2;
 1481:     data2_3_2<-merge(data2_3_2,noesrf,by="patient_id",all.x=TRUE,all.y=TRUE)
 1482:
 1483:
 1484:     data2_3_2$esrf=ifelse(is.na(data2_3_2$esrf.x)==TRUE,data2_3_2$esrf.y,data2_3_2$esrf.x)
 1485:     data2_3_2$durdiab_c=ifelse(is.na(data2_3_2$durdiab_c.x)==TRUE,data2_3_2$durdiab_c.y,data2_3_2$durdiab_c.x)
 1486:     data2_3_2$first_epi_date=ifelse(is
.na(data2_3_2$first_epi_date.x)==TRUE,data2_3_2$first_epi_date.y,data2_3_2$first_epi_date.x)
 1487:     data2_3_2$first_epi_date<-format(data2_3_2$first_epi_date,format="%Y-%m-%d")
```

```
 1488:   }
 1489:   data2_3_2<-data2_3_2[,c("patient_id","first_epi_date","esrf","durdiab_c")]
 1490:
 1491: # Delete duplicated rows
 1492:
 1493:   unique2_3_2<-aggregate(data2_3_2$esrf,by=list(patient_id= data2_3_2$patient_id),min)
 1494:   names(unique2_3_2)<-c("patient_id","esrf")
 1495:
 1496:   data2_3_2<-merge(unique2_3_2,unique(data2_3_2[,-pmatch("esrf",names(data2_3_2))]),by=c("patient_id"
),all.x=TRUE,all.y=FALSE)
 1497:
 1498:   BIRO_dframe(dataname="data2_3_2",monitor=TRUE,filelog=logfile)
 1499:   BIRO_dframe(dataname="data2_3_2",monitor=TRUE,filelog="")
 1500:
 1501:   if (dim(episode[is.na(episode$esrf)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
 1502:
 1503:        BIRO_report(ind="2_3_2",
 1504:                    title="2.3.2. End stage renal failure (first episode in 12 months)",
 1505:                    var_cat="esrf",
 1506:                    class=c("durdiab_c"),
 1507:                    lev_var_cat=levesrf,
 1508:                    lev_class=list(a=classlabellist(durdiabth)),
 1509:                    lab_var_cat="E.S.R.F.",lab_class="Duration",
 1510:                    tab=1,
 1511:                    tabside="H",
 1512:                    tabperc=TRUE,
 1513:                    tabwidth=1,
 1514:                    barbeside=TRUE,
 1515:                    barnumber=TRUE,
 1516:                    barperc=TRUE,
 1517:                    box=0,
 1518:                    bar=1,
 1519:                    lines=0,
 1520:                    trellis=0,
 1521:                    pie=0,
 1522:                    texfile=texfile,
 1523:                    dirtables=dirtables,
 1524:                    dircsv=dirdataout,
 1525:                    dirgraph=dirgraph,
 1526:                    dirgraphlatex=dirgraphlatex,
 1527:                    cex=cex
 1528:                    )
 1529:
 1530:   rm(data2_3_2)
```

```
1531:    }
1532:    }
1533:    }
1534:    }
1535:    } else BIRO_report_toc(title="2.3.2. End stage renal failure (first episode in 12 months)")
1536:
1537:    #
1538:    # 2.3.3 foot ulcer (the first episode, not only refyear)
1539:    #
1540:
1541:    if ((is.null(patient$dt_diag)==F) & (is.null(episode$ulcer_l)==F) & (is.null(episode$ulcer_r)==F)) {
1542:    if (((length(na.omit(patient$dt_diag)))>0) & (length(na.omit(episode$ulcer_l))>0) &
1543:        ((length(na.omit(episode$ulcer_r)))>0)) {
1544:
1545:    episode$ulcer<-ifelse(((is.na(episode$ulcer_l)==FALSE & episode$ulcer_l==1)) |
1546:                          ((is.na(episode$ulcer_r)==FALSE & episode$ulcer_r==1)),1,2)
1547:
1548:    for (thisyear in anayears) {
1549:
1550:     assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1551:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1552:
1553:     assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1554:           "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1555:
1556:     assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1557:           "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1558:
1559:     assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1560:            "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
1561:
1562:     assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
1563:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1564:
1565:     assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
1566:
1567:    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1568:    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1569:
1570:    if (dim(episode[is.na(episode$ulcer)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
1571:
1572:
1573:    data2_3_3<-BIRO_demographic(patient=patient,
1574:                                episode=episode[episode$ulcer==1 & is.na(episode$ulcer)==FALSE & episode$epi_date >=
```

```
oneyearbf & episode$epi_date <= checkdate,],
 1575:                                  var=c("durdiab_c"),
 1576:                                  date="first episode date",
 1577:                                  startvar="dt_diag",
 1578:                                  th=durdiabth,
 1579:                                  hold=TRUE,
 1580:                                  varname="durdiab")
 1581:
 1582:    data2_3_3$ulcer=1;
 1583:
 1584:    if (length(episode$stroke[episode$stroke==2 & is.na(episode$hypertension)==FALSE])>0) {
 1585:
 1586:      noulcer<-BIRO_demographic(patient=patient,
 1587:                                  episode=episode[episode$ulcer==2 & is.na(episode$ulcer)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
 1588:                                  var=c("durdiab_c"),
 1589:                                  date="first episode date",
 1590:                                  startvar="dt_diag",
 1591:                                  th=durdiabth,
 1592:                                  hold=TRUE,
 1593:                                  varname="durdiab")
 1594:
 1595:      noulcer$ulcer=2;
 1596:      data2_3_3<-merge(data2_3_3,noulcer,by="patient_id",all.x=TRUE,all.y=TRUE)
 1597:
 1598:
 1599:      data2_3_3$ulcer=ifelse(is.na(data2_3_3$ulcer.x)==TRUE,data2_3_3$ulcer.y,data2_3_3$ulcer.x)
 1600:      data2_3_3$durdiab_c=ifelse(is.na(data2_3_3$durdiab_c.x)==TRUE,data2_3_3$durdiab_c.y,data2_3_3$durdiab_c.x)
 1601:      data2_3_3$first_epi_date=ifelse(is
.na(data2_3_3$first_epi_date.x)==TRUE,data2_3_3$first_epi_date.y,data2_3_3$first_epi_date.x)
 1602:      data2_3_3$first_epi_date<-format(data2_3_3$first_epi_date,format="%Y-%m-%d")
 1603:    }
 1604:    data2_3_3<-data2_3_3[,c("patient_id","first_epi_date","ulcer","durdiab_c")]
 1605:
 1606: # Delete duplicated rows
 1607:
 1608:    unique2_3_3<-aggregate(data2_3_3$ulcer,by=list(patient_id= data2_3_3$patient_id),min)
 1609:    names(unique2_3_3)<-c("patient_id","ulcer")
 1610:
 1611:    data2_3_3<-merge(unique2_3_3,unique(data2_3_3[,-pmatch("ulcer",names(data2_3_3))]),by=c("patient_id"
),all.x=TRUE,all.y=FALSE)
 1612:
 1613:
 1614:    BIRO_dframe(dataname="data2_3_3",monitor=TRUE,filelog=logfile)
 1615:    BIRO_dframe(dataname="data2_3_3",monitor=TRUE,filelog="")
```

```
1616:
1617:   if (dim(episode[is.na(episode$ulcer)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
1618:
1619:        BIRO_report(ind="2_3_3",
1620:                    title="2.3.3. Foot ulcer (first episode in 12 months)",
1621:                    var_cat="ulcer",
1622:                    class=c("durdiab_c"),
1623:                    lev_var_cat=levulcer,
1624:                    lev_class=list(a=classlabellist(durdiabth)),
1625:                    lab_var_cat="Foot Ulcer",lab_class="Duration",
1626:                    tab=1,
1627:                    tabside="H",
1628:                    tabperc=TRUE,
1629:                    tabwidth=1,
1630:                    barbeside=TRUE,
1631:                    barnumber=TRUE,
1632:                    barperc=TRUE,
1633:                    box=0,
1634:                    bar=1,
1635:                    lines=0,
1636:                    trellis=0,
1637:                    pie=0,
1638:                    texfile=texfile,
1639:                    dirtables=dirtables,
1640:                    dircsv=dirdataout,
1641:                    dirgraph=dirgraph,
1642:                    dirgraphlatex=dirgraphlatex,
1643:                    cex=cex
1644:                    )
1645:
1646:
1647:
1648:   rm(data2_3_3)
1649:
1650:   }
1651:   }
1652:   }
1653:   }
1654:   } else BIRO_report_toc(title="2.3.3. Foot ulcer (first episode in 12 months)")
1655:
1656:   #
1657:   # 2.3.4 amputation (the first episode, not only refyear)
1658:   #
1659:
```

```
 1660:   if ((is.null(patient$dt_diag)==F) & (is.null(episode$amput)==F)) {
 1661:   if (length(na.omit(patient$dt_diag))>0 & length(na.omit(episode$amput))>0) {
 1662:
 1663:   for (thisyear in anayears) {
 1664:
 1665:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
 1666:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1667:
 1668:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
 1669:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 1670:
 1671:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 1672:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 1673:
 1674:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1675:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1676:
 1677:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1678:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1679:
 1680:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1681:
 1682:   checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 1683:   oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 1684:
 1685:   if (dim(episode[is.na(episode$amput)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
 1686:
 1687:   data2_3_4<-BIRO_demographic(patient=patient,
 1688:                               episode=episode[episode$amput==1 & is.na(episode$amput)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
 1689:                               var=c("durdiab_c"),
 1690:                               date="first episode date",
 1691:                               startvar="dt_diag",
 1692:                               th=durdiabth,
 1693:                               hold=TRUE,
 1694:                               varname="durdiab")
 1695:
 1696:   data2_3_4$amput=1;
 1697:
 1698:   if (length(episode$amput[episode$amput==2 & is.na(episode$hypertension)==FALSE])>0) {
 1699:
 1700:     noamput<-BIRO_demographic(patient=patient,
 1701:                               episode=episode[episode$amput==2 & is.na(episode$amput)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
```

```
1702:                                 var=c("durdiab_c"),
1703:                                 date="first episode date",
1704:                                 startvar="dt_diag",
1705:                                 th=durdiabth,
1706:                                 hold=TRUE,
1707:                                 varname="durdiab")
1708:
1709:     noamput$amput=2;
1710:     data2_3_4<-merge(data2_3_4,noamput,by="patient_id",all.x=TRUE,all.y=TRUE)
1711:
1712:
1713:     data2_3_4$amput=ifelse(is.na(data2_3_4$amput.x)==TRUE,data2_3_4$amput.y,data2_3_4$amput.x)
1714:     data2_3_4$durdiab_c=ifelse(is.na(data2_3_4$durdiab_c.x)==TRUE,data2_3_4$durdiab_c.y,data2_3_4$durdiab_c.x)
1715:     data2_3_4$first_epi_date=ifelse(is
.na(data2_3_4$first_epi_date.x)==TRUE,data2_3_4$first_epi_date.y,data2_3_4$first_epi_date.x)
1716:     data2_3_4$first_epi_date<-format(data2_3_4$first_epi_date,format="%Y-%m-%d")
1717:   }
1718:   data2_3_4<-data2_3_4[,c("patient_id","first_epi_date","amput","durdiab_c")]
1719:
1720: # Delete duplicated rows
1721:
1722:   unique2_3_4<-aggregate(data2_3_4$amput,by=list(patient_id= data2_3_4$patient_id),min)
1723:   names(unique2_3_4)<-c("patient_id","amput")
1724:
1725:   data2_3_4<-merge(unique2_3_4,unique(data2_3_4[,-pmatch("amput",names(data2_3_4))]),by=c("patient_id"
),all.x=TRUE,all.y=FALSE)
1726:
1727:
1728:   BIRO_dframe(dataname="data2_3_4",monitor=TRUE,filelog=logfile)
1729:   BIRO_dframe(dataname="data2_3_4",monitor=TRUE,filelog="")
1730:
1731:   if (dim(episode[is.na(episode$ulcer)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
1732:
1733:       BIRO_report(ind="2_3_4",
1734:                   title="2.3.4. Amputation (first episode in 12 months)",
1735:                   var_cat="amput",
1736:                   class=c("durdiab_c"),
1737:                   lev_var_cat=levamput,
1738:                   lev_class=list(a=classlabellist(durdiabth)),
1739:                   lab_var_cat="Amputation",lab_class="Duration",
1740:                   tab=1,
1741:                   tabside="H",
1742:                   tabperc=TRUE,
1743:                   tabwidth=1,
```

```
1744:                     barbeside=TRUE,
1745:                     barnumber=TRUE,
1746:                     barperc=TRUE,
1747:                     box=0,
1748:                     bar=1,
1749:                     lines=0,
1750:                     trellis=0,
1751:                     pie=0,
1752:                     texfile=texfile,
1753:                     dirtables=dirtables,
1754:                     dircsv=dirdataout,
1755:                     dirgraph=dirgraph,
1756:                     dirgraphlatex=dirgraphlatex,
1757:                     cex=cex
1758:                     )
1759:
1760:
1761:
1762:   rm(data2_3_4)
1763:   }
1764:   }
1765:   }
1766:   }
1767:   } else BIRO_report_toc(title="2.3.4. Amputation (first episode in 12 months)")
1768:
1769:   #
1770:   # 2.3.5 stroke (the first episode, not only refyear)
1771:   #
1772:
1773:   if ((is.null(patient$dt_diag)==F) & (is.null(episode$stroke)==F)) {
1774:   if (length(na.omit(patient$dt_diag))>0 & length(na.omit(episode$stroke))>0) {
1775:
1776:   for (thisyear in anayears) {
1777:
1778:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1779:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1780:
1781:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1782:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1783:
1784:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1785:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1786:
1787:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1788:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
```

```
 1789:
 1790:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1791:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1792:
 1793:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1794:
 1795:   checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 1796:   oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 1797:
 1798:   if (dim(episode[is.na(episode$stroke)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
 1799:
 1800:   data2_3_5<-BIRO_demographic(patient=patient,
 1801:                               episode=episode[episode$stroke==1 & is.na(episode$stroke)==FALSE & episode$epi_date
>= oneyearbf & episode$epi_date <= checkdate,],
 1802:                               var=c("durdiab_c"),
 1803:                               date="first episode date",
 1804:                               startvar="dt_diag",
 1805:                               th=durdiabth,
 1806:                               hold=TRUE,
 1807:                               varname="durdiab")
 1808:
 1809:   data2_3_5$stroke=1;
 1810:
 1811:   if (length(episode$stroke[episode$stroke==2 & is.na(episode$stroke)==FALSE])>0) {
 1812:     nostroke<-BIRO_demographic(patient=patient,
 1813:                                episode=episode[episode$stroke==2 & is.na(episode$stroke)==FALSE & episode$epi_date
>= oneyearbf & episode$epi_date <= checkdate,],
 1814:                                var=c("durdiab_c"),
 1815:                                date="first episode date",
 1816:                                startvar="dt_diag",
 1817:                                th=durdiabth,
 1818:                                hold=TRUE,
 1819:                                varname="durdiab")
 1820:
 1821:     nostroke$stroke=2;
 1822:     data2_3_5<-merge(data2_3_5,nostroke,by="patient_id",all.x=TRUE,all.y=TRUE)
 1823:
 1824:
 1825:     data2_3_5$stroke=ifelse(is.na(data2_3_5$stroke.x)==TRUE,data2_3_5$stroke.y,data2_3_5$stroke.x)
 1826:     data2_3_5$durdiab_c=ifelse(is.na(data2_3_5$durdiab_c.x)==TRUE,data2_3_5$durdiab_c.y,data2_3_5$durdiab_c.x)
 1827:     data2_3_5$first_epi_date=ifelse(is
.na(data2_3_5$first_epi_date.x)==TRUE,data2_3_5$first_epi_date.y,data2_3_5$first_epi_date.x)
 1828:     data2_3_5$first_epi_date<-format(data2_3_5$first_epi_date,format="%Y-%m-%d")
 1829:
```

```
1830:    }
1831:
1832:    data2_3_5<-data2_3_5[,c("patient_id","first_epi_date","stroke","durdiab_c")]
1833:
1834: # Delete duplicated rows
1835:
1836:    unique2_3_5<-aggregate(data2_3_5$stroke,by=list(patient_id= data2_3_5$patient_id),min)
1837:    names(unique2_3_5)<-c("patient_id","stroke")
1838:
1839:    data2_3_5<-merge(unique2_3_5,unique(data2_3_5[,-pmatch("stroke",names(data2_3_5))]),by=c("patient_id"
),all.x=TRUE,all.y=FALSE)
1840:
1841:    BIRO_dframe(dataname="data2_3_5",monitor=TRUE,filelog=logfile)
1842:    BIRO_dframe(dataname="data2_3_5",monitor=TRUE,filelog="")
1843:
1844:         BIRO_report(ind="2_3_5",
1845:                     title="2.3.5. Stroke (first episode in 12 months)",
1846:                     var_cat="stroke",
1847:                     class=c("durdiab_c"),
1848:                     lev_var_cat=levstroke,
1849:                     lev_class=list(a=classlabellist(durdiabth)),
1850:                     lab_var_cat="Stroke",
1851:                     lab_class="Duration",
1852:                     tab=1,
1853:                     tabside="H",
1854:                     tabperc=TRUE,
1855:                     tabwidth=1,
1856:                     barbeside=TRUE,
1857:                     barnumber=TRUE,
1858:                     barperc=TRUE,
1859:                     box=0,
1860:                     bar=1,
1861:                     lines=0,
1862:                     trellis=0,
1863:                     pie=0,
1864:                     texfile=texfile,
1865:                     dirtables=dirtables,
1866:                     dircsv=dirdataout,
1867:                     dirgraph=dirgraph,
1868:                     dirgraphlatex=dirgraphlatex,
1869:                     cex=cex
1870:                     )
1871:
1872:    rm(data2_3_5)
1873:    }
```

```
1874:    }
1875:    }
1876:    } else BIRO_report_toc(title="2.3.5. Stroke (first episode in 12 months)")
1877:
1878:    #
1879:    # 2.3.6 myocardial infarctions (the first episode, not only refyear)
1880:    #
1881:
1882:    if ((is.null(patient$dt_diag)==F) & (is.null(episode$mi)==F)) {
1883:    if (length(na.omit(patient$dt_diag))>0 & length(na.omit(episode$mi))>0) {
1884:
1885:    for (thisyear in anayears) {
1886:
1887:     assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1888:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1889:
1890:     assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1891:           "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1892:
1893:     assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1894:           "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1895:
1896:     assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1897:            "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
1898:
1899:     assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
1900:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1901:
1902:     assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
1903:
1904:    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1905:    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1906:
1907:    if (dim(episode[is.na(episode$mi)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
1908:
1909:    data2_3_6<-BIRO_demographic(patient=patient,
1910:                                episode=episode[episode$mi==1 & is.na(episode$mi)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
1911:                                var=c("durdiab_c"),
1912:                                date="first episode date",
1913:                                startvar="dt_diag",
1914:                                th=durdiabth,
1915:                                hold=TRUE,
1916:                                varname="durdiab")
```

```
 1917:
 1918:   data2_3_6$mi=1;
 1919:   if (length(episode$mi[episode$mi==2 & is.na(episode$hypertension)==FALSE])>0) {
 1920:
 1921:     nomi<-BIRO_demographic(patient=patient,
 1922:                            episode=episode[episode$mi==2 & is.na(episode$mi)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
 1923:                            var=c("durdiab_c"),
 1924:                            date="first episode date",
 1925:                            startvar="dt_diag",
 1926:                            th=durdiabth,
 1927:                            hold=TRUE,
 1928:                            varname="durdiab")
 1929:
 1930:     nomi$mi=2;
 1931:     data2_3_6<-merge(data2_3_6,nomi,by="patient_id",all.x=TRUE,all.y=TRUE)
 1932:
 1933:     data2_3_6$mi=ifelse(is.na(data2_3_6$mi.x)==TRUE,data2_3_6$mi.y,data2_3_6$mi.x)
 1934:     data2_3_6$durdiab_c=ifelse(is.na(data2_3_6$durdiab_c.x)==TRUE,data2_3_6$durdiab_c.y,data2_3_6$durdiab_c.x)
 1935:     data2_3_6$first_epi_date=ifelse(is
.na(data2_3_6$first_epi_date.x)==TRUE,data2_3_6$first_epi_date.y,data2_3_6$first_epi_date.x)
 1936:     data2_3_6$first_epi_date<-format(data2_3_6$first_epi_date,format="%Y-%m-%d")
 1937:   }
 1938:   data2_3_6<-data2_3_6[,c("patient_id","first_epi_date","mi","durdiab_c")]
 1939:
 1940: # Delete duplicated rows
 1941:
 1942:   unique2_3_6<-aggregate(data2_3_6$mi,by=list(patient_id= data2_3_6$patient_id),min)
 1943:   names(unique2_3_6)<-c("patient_id","mi")
 1944:
 1945:   data2_3_6<-merge(unique2_3_6,unique(data2_3_6[,-pmatch("mi",names(data2_3_6))]),by=c("patient_id"
),all.x=TRUE,all.y=FALSE)
 1946:
 1947:
 1948:   BIRO_dframe(dataname="data2_3_6",monitor=TRUE,filelog=logfile)
 1949:   BIRO_dframe(dataname="data2_3_6",monitor=TRUE,filelog="")
 1950:
 1951:       BIRO_report(ind="2_3_6",
 1952:                   title="2.3.6. Myocardial infarction (first episode in 12 months)",
 1953:                   var_cat="mi",
 1954:                   class=c("durdiab_c"),
 1955:                   lev_var_cat=levmi,
 1956:                   lev_class=list(a=classlabellist(durdiabth)),
 1957:                   lab_var_cat="Myocardial Infarction",
 1958:                   lab_class="Duration",
```

```
1959:                     tab=1,
1960:                     tabside="H",
1961:                     tabperc=TRUE,
1962:                     tabwidth=1,
1963:                     barbeside=TRUE,
1964:                     barnumber=TRUE,
1965:                     barperc=TRUE,
1966:                     box=0,
1967:                     bar=1,
1968:                     lines=0,
1969:                     trellis=0,
1970:                     pie=0,
1971:                     texfile=texfile,
1972:                     dirtables=dirtables,
1973:                     dircsv=dirdataout,
1974:                     dirgraph=dirgraph,
1975:                     dirgraphlatex=dirgraphlatex,
1976:                     cex=cex
1977:                     )
1978:
1979:   rm(data2_3_6)
1980:   }
1981:   }
1982:   }
1983:   } else BIRO_report_toc(title="2.3.6. Myocardal infarction (first episode in 12 months)")
1984:
1985:   #
1986:   # 2.3.7 hypertension (the first episode, not only refyear)
1987:   #
1988:
1989:   if ((is.null(patient$dt_diag)==F) & (is.null(episode$hypertension)==F)) {
1990:   if (length(na.omit(patient$dt_diag))>0 & length(na.omit(episode$hypertension))>0) {
1991:
1992:   for (thisyear in anayears) {
1993:
1994:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1995:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1996:
1997:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1998:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1999:
2000:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
2001:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
2002:
2003:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
```

```
 2004:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 2005:
 2006:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 2007:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 2008:
 2009:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 2010:
 2011:   checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 2012:   oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 2013:
 2014:   if (dim(episode[is.na(episode$hypertension)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,])[1]>0) {
 2015:
 2016:   data2_3_7<-BIRO_demographic(patient=patient,
 2017:                               episode=episode[episode$hypertension==1 & is.na(episode$hypertension)==FALSE &
episode$epi_date >= oneyearbf & episode$epi_date <= checkdate,],
 2018:                               var=c("durdiab_c"),
 2019:                               date="first episode date",
 2020:                               startvar="dt_diag",
 2021:                               th=durdiabth,
 2022:                               hold=TRUE,
 2023:                               varname="durdiab")
 2024:
 2025:   data2_3_7$hypertension=1;
 2026:   if (length(episode$hypertension[episode$hypertension==2 & is.na(episode$hypertension)==FALSE])>0) {
 2027:
 2028:     nohypertension<-BIRO_demographic(patient=patient,
 2029:                                      episode=episode[episode$hypertension==2 & is.na(episode$hypertension)==FALSE
& episode$epi_date >= oneyearbf & episode$epi_date <= checkdate,],
 2030:                                      var=c("durdiab_c"),
 2031:                                      date="first episode date",
 2032:                                      startvar="dt_diag",
 2033:                                      th=durdiabth,
 2034:                                      hold=TRUE,
 2035:                                      varname="durdiab")
 2036:
 2037:     nohypertension$hypertension=2;
 2038:     data2_3_7<-merge(data2_3_7,nohypertension,by="patient_id",all.x=TRUE,all.y=TRUE)
 2039:
 2040:     data2_3_7$hypertension=ifelse(is
.na(data2_3_7$hypertension.x)==TRUE,data2_3_7$hypertension.y,data2_3_7$hypertension.x)
 2041:     data2_3_7$durdiab_c=ifelse(is.na(data2_3_7$durdiab_c.x)==TRUE,data2_3_7$durdiab_c.y,data2_3_7$durdiab_c.x)
 2042:     data2_3_7$first_epi_date=ifelse(is
.na(data2_3_7$first_epi_date.x)==TRUE,data2_3_7$first_epi_date.y,data2_3_7$first_epi_date.x)
 2043:     data2_3_7$first_epi_date<-format(data2_3_7$first_epi_date,format="%Y-%m-%d")
```

```
2044:    }
2045:    data2_3_7<-data2_3_7[,c("patient_id","first_epi_date","hypertension","durdiab_c")]
2046:
2047: # Delete duplicated rows
2048:
2049:    unique2_3_7<-aggregate(data2_3_7$hypertension,by=list(patient_id= data2_3_7$patient_id),min)
2050:    names(unique2_3_7)<-c("patient_id","hypertension")
2051:
2052:    data2_3_7<-merge(unique2_3_7,unique(data2_3_7[,-pmatch("hypertension",names(data2_3_7))]),by=c("patient_id"
),all.x=TRUE,all.y=FALSE)
2053:
2054:
2055:    BIRO_dframe(dataname="data2_3_7",monitor=TRUE,filelog=logfile)
2056:    BIRO_dframe(dataname="data2_3_7",monitor=TRUE,filelog="")
2057:
2058:         BIRO_report(ind="2_3_7",
2059:                     title="2.3.7. Hypertension (first episode in 12 months)",
2060:                     var_cat="hypertension",
2061:                     class=c("durdiab_c"),
2062:                     lev_var_cat=levhypertension,
2063:                     lev_class=list(a=classlabellist(durdiabth),b=NA,c=NA),
2064:                     lab_var_cat="Hypertension",
2065:                     lab_class="Duration",
2066:                     tab=1,
2067:                     tabside="H",
2068:                     tabperc=TRUE,
2069:                     tabwidth=1,
2070:                     barbeside=TRUE,
2071:                     barnumber=TRUE,
2072:                     barperc=TRUE,
2073:                     box=0,
2074:                     bar=1,
2075:                     lines=0,
2076:                     trellis=0,
2077:                     pie=0,
2078:                     texfile=texfile,
2079:                     dirtables=dirtables,
2080:                     dircsv=dirdataout,
2081:                     dirgraph=dirgraph,
2082:                     dirgraphlatex=dirgraphlatex,
2083:                     cex=cex
2084:                     )
2085:
2086:
2087:    rm(data2_3_7)
```

```
2088:   }
2089:   }
2090:   }
2091:   } else BIRO_report_toc(title="2.3.7. Hypertension (first episode in 12 months)")
2092:
```

```
  1: ##############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: BIRO_se_indicator_demographic.r                                      #
  5: #                                                                            #
#
  6: ##############################################################################
  7: # ------------------------
  8: # Authors:
  9: # Luca Rossi       <redsluke@gmail.com)
 10: # Fabrizio Carinci <research@fabcarinci.net)
 11: # Institutions: Serectrix snc
 12: # Created: 2007-11-24
 13: # Version: 2008-11-24
 14: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 15: # Programming Language: R 2.8.0
 16: # ------------------------
 17: # COPYRIGHT INFORMATION
 18: #
 19: # This file is free software; you can redistribute it and/or modify
 20: # it under the terms of the GNU General Public License as published by
 21: # the Free Software Foundation; either version 2, or (at your option)
 22: # any later version.
 23: #
 24: # This file is distributed in the hope that it will be useful,
 25: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 26: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 27: # GNU General Public License for more details.
 28: #
 29: # You should have received a copy of the GNU General Public License
 30: # along with this file; see the file COPYING.  If not, write to
 31: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 32: #
 33: # In short: you may use this file any way you like, as long as you
 34: # don't charge money for it, remove this notice, or hold anyone liable
 35: # for its results.
 36: #
 37: # BIRO_se_indicator_clinical.r is part of WP Statistical Engine of the BIRO
 38: # Project
 39: # GPL Copyright, The BIRO Project
 40: #
 41: # ------------------------
 42: #
 43: ##############################################################################
 44:
```

```
45:
46:   #1.1 Age of patients (all the patient in the db)
47:
48:   if (length(na.omit(patient$sex))>0 & length(na.omit(patient$dob))>0) {
49:
50:
51:     for (thisyear in anayears) {
52:
53:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
54:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
55:
56:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
57:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
58:
59:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
60:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
61:
62:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
63:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
64:
65:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
66:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
67:
68:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
69:
70:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
71:
72:     data1_1<-patient[patient$end_date>checkdate | is.na(patient$end_date),]
73:     data1_1$age<-(checkdate-patient$dob)/365.25
74:     data1_1<-as.data.frame(data1_1)
75:     data1_1$age_c<-varclass(data1_1$age,ageth)
76:
77: #    data1_1<-BIRO_demographic(patient=patient[patient],
78: #                              episode=episode,
79: #                              var=c("sex","age_c"),
80: #                              date=refdate,
81: #                              startvar="dob",
82: #                              th=ageth,
83: #                              hold=FALSE,
84: #                              varname="age")
85:
86:     #data1.2 is the same of data1.1
87:
88:     BIRO_dframe(dataname="data1_1",monitor=TRUE,filelog="")
89:     BIRO_dframe(dataname="data1_1",monitor=TRUE,filelog=logfile)
```

```
 90:
 91:     BIRO_report(ind="1_1",
 92:                 title="1.1. Age (Classes)",
 93:                 var_cat="age_c",
 94:                 class=c("sex"),
 95:                 lev_var_cat=classlabel(ageth),
 96:                 lev_class=list(levsex),
 97:                 lab_var_cat="Age Classes",
 98:                 lab_class=c("Gender"),
 99:                 tab=1,
100:                 tabside="H",
101:                 tabperc=TRUE,
102:                 tabwidth=1,
103:                 barbeside=TRUE,
104:                 barnumber=TRUE,
105:                 barperc=TRUE,
106:                 box=0,
107:                 bar=1,
108:                 lines=0,
109:                 trellis=0,
110:                 pie=0,
111:                 texfile=texfile,
112:                 dirtables=dirtables,
113:                 dircsv=dirdataout,
114:                 dirgraph=dirgraph,
115:                 dirgraphlatex=dirgraphlatex,
116:                 cex=cex,
117:                 chisq=NULL,
118:                 patmap=1,
119:                 shapefile="eurnuts3",
120:                 mapvar="sex",
121:                 maplev=levsex,
122:                 mapstat="freq",
123:                 maplab="Patients",
124:                 shapearea_id="ID",
125:                 valuesarea_id="area")
126:
127:     rm(data1_1)
128:   }
129:   } else BIRO_report_toc(title="1.1. Age (Classes)")
130:
```

```
  1: ##############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: BIRO_se_indicator_health_system.r                                   #
  5: #                                                                            #
#
  6: ##############################################################################
  7: # ------------------------
  8: # Authors:
  9: # Luca Rossi       <redsluke@gmail.com)
 10: # Fabrizio Carinci <research@fabcarinci.net)
 11: # Institutions: Serectrix snc
 12: # Created: 2007-11-24
 13: # Version: 2008-11-24
 14: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 15: # Programming Language: R 2.8.0
 16: # ------------------------
 17: # COPYRIGHT INFORMATION
 18: #
 19: # This file is free software; you can redistribute it and/or modify
 20: # it under the terms of the GNU General Public License as published by
 21: # the Free Software Foundation; either version 2, or (at your option)
 22: # any later version.
 23: #
 24: # This file is distributed in the hope that it will be useful,
 25: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 26: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 27: # GNU General Public License for more details.
 28: #
 29: # You should have received a copy of the GNU General Public License
 30: # along with this file; see the file COPYING.  If not, write to
 31: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 32: #
 33: # In short: you may use this file any way you like, as long as you
 34: # don't charge money for it, remove this notice, or hold anyone liable
 35: # for its results.
 36: #
 37: # BIRO_se_indicator_health_system.r is part of WP Statistical Engine of the
 38: # BIRO Project
 39: # GPL Copyright, The BIRO Project
 40: #
 41: # ------------------------
 42: #
 43: ##############################################################################
 44:
```

```
45:
46:
47: new_section(title="3.1. Structure (provider level)",texfile=texfile)
48: new_section_html(title="3.1. Structure (provider level)",htmlfile=htmlfile)
49:
50: #
51: # 3.1.1 Type of Provider
52: #
53:   if (dim(site)[1]>0) {
54:
55:    for (thisyear in anayears) {
56:
57:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
58:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
59:
60:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
61:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
62:
63:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
64:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
65:
66:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
67:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
68:
69:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
70:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
71:
72:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
73:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
74:
75:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
76:
77:    data3_1_1<-as.data.frame(cbind(as.character(centre_id),as.character(site$ds_type)))
78:    names(data3_1_1)<-c("centre_id","type")
79:    row.names(data3_1_1)=""
80:    data3_1_1$centre_id<-as.character(data3_1_1$centre_id)
81:
82:    write.csv(data3_1_1,paste(dirdataout,"/","i3_1_1d1_3a.csv",sep=""),row.names=FALSE)
83:
84:        BIRO_report(ind="3_1_1",
85:                    title="3.1.1 Type of Provider",
86:                    var=NULL,
87:                    var_cat=NULL,
88:                    lev_var_cat=NULL,
89:                    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,
                    dirtables=dirtables,
                    dircsv=dirdataout,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    stand=0)
    }
   } else BIRO_report_toc(title="3.1.1 Type of Provider")

#
#Popolation by to be implemented
# TODO
#

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

    data3_1_2<-as.data.frame(popdiab)
    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"

 } else {
    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(dirse,"/output/data/", "#",launchtime,
           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)

```

```
135:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
136:           "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
137:
138:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
139:           "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
140:
141:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
142:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
143:
144:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
145:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
146:
147:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
148:
149:     popdiab<-patient[patient$end_date>checkdate | is.na(patient$end_date),]
150:     popdiab$age<-(checkdate-popdiab$dob)/365.25
151:
152:     popdiab$begin<-ifelse(popdiab$start_date<oneyearbf,oneyearbf,as.Date(popdiab$start_date))
153:     popdiab$wperson<-(as.numeric(checkdate)-popdiab$begin)/365
154:     popdiab$wperson<-ifelse(popdiab$wperson>1,1,popdiab$wperson)
155:     popdiab$age_c<-varclass(popdiab$age,ageth5)
156:
157:     popdiab<-aggregate(popdiab$wperson,by=list(type=popdiab$type_dm,sex=popdiab$sex,ageband=popdiab$age_c),FUN=sum)
158:
159:     popdiabf<-popdiab[popdiab$sex==1,-pmatch("sex",names(popdiab))]
160:     names(popdiabf)<-c("type","ageband","popdiaf")
161:
162:     popdiabm<-popdiab[popdiab$sex==2,-pmatch("sex",names(popdiab))]
163:     names(popdiabm)<-c("type","ageband","popdiam")
164:
165:     data3_1_2<-merge(popdiabm,popdiabf,by=c("type","ageband"),all=TRUE)
166:
167:     rm(popdiabm)
168:     rm(popdiabf)
169:
170:    data3_1_2$sex<-factor(data3_1_2$sex)
171:    data3_1_2$type_dm<-factor(data3_1_2$type_dm)
172:    data3_1_2$ageband<-factor(data3_1_2$ageband)
173:    levels(data3_1_2$sex)<-levsex
174:    levels(data3_1_2$type_dm)<-levtype_dm
175:    levels(data3_1_2$ageband)<-classlabellist(ageth5)
176:    data3_1_2$centre_id=centre_id
177:    names(data3_1_2)[pmatch("Freq",names(data3_1_2))]<-"n"
178:    }
179:
```

```
180:  }
181:
182:    write.csv(data3_1_2,paste(dirdataout,"/","i3_1_2d1_3a.csv",sep=""),row.names=FALSE)
183:
184:    data3_1_2<-data3_1_2[,pmatch(c("sex","ageband","type_dm","n"),names(data3_1_2))]
185:
186:        BIRO_report(ind="3_1_2",
187:                    title="3.1.2 Average diabetes population per center",
188:                    var=NULL,
189:                    var_cat=NULL,
190:                    lev_var_cat=NULL,
191:                    lev_class=NULL,
192:                    lab_var=NULL,
193:                    lab_var_cat=NULL,
194:                    lab_class=NULL,
195:                    tab=0,
196:                    box=0,
197:                    bar=0,
198:                    lines=0,
199:                    trellis=0,
200:                    stand=0,
201:                    pie=0,
202:                    texfile=texfile,
203:                    dirtables=dirtables,
204:                    dircsv=dirdataout,
205:                    dirgraph=dirgraph,
206:                    dirgraphlatex=dirgraphlatex,
207:                    cex=cex)
208:   # }
209:
210:    #else BIRO_report_toc(title="3.1.2 Average diabetes population per center")
211:
212: new_section(title="3.2. Structural quality",texfile=texfile)
213: new_section_html(title="3.2. Structural quality",htmlfile=htmlfile)
214:
215: #
216: # 3.2.1 Hospital beds per 100,000 population
217: #
218:    if (length(is.na(site$ds_beds))>0) {
219:
220: if (dim(pop)[1]>0) {
221:
222:    for (thisyear in anayears) {
223:
224:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
```

```
225:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
226:
227:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
228:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
229:
230:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
231:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
232:
233:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
234:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
235:
236:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
237:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
238:
239:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
240:
241:    totpop<-sum(pop[pop$year==thisyear,"popM"])+sum(pop[pop$year==thisyear,"popF"])
242:
243:    data3_2_1<-as.data.frame(cbind(thisyear,site$ds_beds,totpop,round((site$ds_beds/totpop)*100000,digits=2)))
244:    names(data3_2_1)<-c("year","beds","pop","rate")
245:    row.names(data3_2_1)=""
246:    data3_2_1$centre_id=centre_id
247:
248: }
249:
250:    write.csv(data3_2_1,paste(dirdataout,"/","i3_2_1d1_3a.csv",sep=""),row.names=FALSE)
251:
252:    data3_2_1<-data3_2_1[,pmatch(c("year","beds","pop","rate"),names(data3_2_1))]
253:
254:        BIRO_report(ind="3_2_1",
255:                    title="3.2.1 Hospital beds per 100,000 population",
256:                    var=NULL,
257:                    var_cat=NULL,
258:                    lev_var_cat=NULL,
259:                    lev_class=NULL,
260:                    lab_var=NULL,
261:                    lab_var_cat=NULL,
262:                    lab_class=NULL,
263:                    tab=0,
264:                    box=0,
265:                    bar=0,
266:                    lines=0,
267:                    trellis=0,
268:                    pie=0,
269:                    texfile=texfile,
```

```
  270:                     dirtables=dirtables,
  271:                     dircsv=dirdataout,
  272:                     dirgraph=dirgraph,
  273:                     dirgraphlatex=dirgraphlatex,
  274:                     cex=cex)
  275:   }
  276:   } else BIRO_report_toc(title="3.2.1 Hospital beds per 100,000 population")
  277:
  278:
  279: #
  280: # 3.2.2 Physicians employed per 100,000 population
  281: #
  282:   if ((length(is.na(site$ds_physicians))>0) & (dim(pop)[1]>0)) {
  283:
  284:    for (thisyear in anayears) {
  285:
  286:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  287:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  288:
  289:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  290:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  291:
  292:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  293:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  294:
  295:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  296:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  297:
  298:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  299:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  300:
  301:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  302:
  303:    totpop<-sum(pop[pop$year==thisyear,"popM"])+sum(pop[pop$year==thisyear,"popF"])
  304:
  305:
data3_2_2<-as.data.frame(cbind(thisyear,centre_id,site$ds_physicians,totpop,round((site$ds_physicians/totpop)*100000,digit
s=2)))
  306:    names(data3_2_2)<-c("year","centre_id","ds_physicians","pop","rate")
  307:    row.names(data3_2_2)=""
  308:    data3_2_2$centre_id=centre_id
  309:
  310:
  311:    write.csv(data3_2_2,paste(dirdataout,"/","i3_2_2d1_3a.csv",sep=""),row.names=FALSE)
  312:
```

```
313:    data3_2_2<-data3_2_2[,pmatch(c("year","ds_physicians","pop","rate"),names(data3_2_2))]
314:
315:        BIRO_report(ind="3_2_2",
316:                    title="3.2.2 Physicians employed per 100,000 population",
317:                    var=NULL,
318:                    var_cat=NULL,
319:                    lev_var_cat=NULL,
320:                    lev_class=NULL,
321:                    lab_var=NULL,
322:                    lab_var_cat=NULL,
323:                    lab_class=NULL,
324:                    tab=0,
325:                    box=0,
326:                    bar=0,
327:                    lines=0,
328:                    trellis=0,
329:                    pie=0,
330:                    texfile=texfile,
331:                    dirtables=dirtables,
332:                    dircsv=dirdataout,
333:                    dirgraph=dirgraph,
334:                    dirgraphlatex=dirgraphlatex,
335:                    cex=cex)
336:
337:    }
338:   } else BIRO_report_toc(title="3.2.2 Physicians employed per 100,000 population")
339:
340: #
341: # Creating epirefyear
342: #
343:
344:
345: #  epirefyear=episode
346: #  epirefyear$year_epi=
347: #           as.numeric(format(as.Date(as.character(epirefyear[,
348: #           pmatch("epi_date",names(epirefyear))]),format="%Y-%m-%d"),format="%Y"))
349: #  refanadate<-as.Date(as.character(refanadate),format="%d/%m/%Y")
350: #  refanadate<-refanadate-365
351: #  epirefyear<-epirefyear[epirefyear$epi_date>=refanadate,]
352: ##  epirefyear<-epirefyear[is.na(epirefyear$year_epi)==FALSE,]
353: #
354: #  patrefyear<-patient[patient$patient_id %in% unique(epirefyear$patient_id),]
355:
356: #stop()
357:
```

```
  358:
  359: new_section_html(title="3.3. Processes (individual level)",htmlfile=htmlfile)
  360: new_section(title="3.3. Processes (individual level)",texfile=texfile)
  361:
  362: new_sub_section_html(title="3.3.1. Foot examination",htmlfile=htmlfile)
  363: new_sub_section(title="3.3.1. Foot examination",texfile=texfile)
  364:
  365: #
  366: # 3.3.1.1 Foot examination
  367: #
  368:
  369:
  370:   if (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
  371:       length(na.omit(episode$foot_exam))>0) {
  372:
  373:     episode$foot_exam<-as.numeric(as.character(episode$foot_exam))
  374:
  375:    for (thisyear in anayears) {
  376:
  377:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  378:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  379:
  380:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  381:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  382:
  383:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  384:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  385:
  386:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  387:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  388:
  389:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  390:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  391:
  392:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  393:
  394:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  395:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  396:
  397:     data3_3_1_1<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  398:                                   episode=episode[is.na(episode$foot_exam)==FALSE & episode$epi_date >= oneyearbf
& episode$epi_date <= checkdate,],
  399:                                   var=c("age_c","type_dm"),
  400:                                   date="last episode date",
```

```
401:                                    startvar="dob",
402:                                    th=ageth,
403:                                    hold=TRUE,
404:                                    varname="age")
405:
406:     foot_e_status<-BIRO_aggregate(data=episode[is.na(episode$foot_exam)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
407:                                    request=c("patient_id"),
408:                                    fun="min",
409:                                    vard="foot_exam",
410:                                    isdate=FALSE)
411:
412:     names(foot_e_status)<-c("patient_id","foot_exam")
413:     data3_3_1_1<-merge(data3_3_1_1,foot_e_status,by="patient_id",all.x=TRUE)
414:     data3_3_1_1$foot_exam[is.na(data3_3_1_1$foot_exam)==TRUE]<-2
415:     data3_3_1_1<-na.omit(data3_3_1_1[,c("age_c","type_dm","foot_exam")])
416:
417:        BIRO_report(ind="3_3_1_1",
418:                    title="3.3.1.1 Done (last episode in 12 months)",
419:                    var_cat="foot_exam",
420:                    class=c("age_c",NA,"type_dm"),
421:                    lev_var_cat=names(levfoot_exam),
422:                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
423:                    lab_var_cat="Foot Examination",
424:                    lab_class=c("Age Classes","Type of Diabetes"),
425:                    tab=1,
426:                    tabside="H",
427:                    tabperc=TRUE,
428:                    tabwidth=1,
429:                    barbeside=TRUE,
430:                    barnumber=TRUE,
431:                    barperc=TRUE,
432:                    box=0,
433:                    bar=1,
434:                    lines=0,
435:                    trellis=0,
436:                    pie=0,
437:                    texfile=texfile,
438:                    dirtables=dirtables,
439:                    dircsv=dirdataout,
440:                    dirgraph=dirgraph,
441:                    dirgraphlatex=dirgraphlatex,
442:                    cex=cex)
443:
444:   rm(data3_3_1_1)
```

```
  445:   }
  446:   } else BIRO_report_toc(title="3.3.1.1 Done (last episode in 12 months)")
  447:
  448:
  449:  # 3.3.2.1 Eye examination
  450:
  451:    new_sub_section_html(title="3.3.2. Eye examination",htmlfile=htmlfile)
  452:    new_sub_section(title="3.3.2. Eye examination",texfile=texfile)
  453:
  454:    if (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
  455:        length(na.omit(episode$eye_exam))>0) {
  456:
  457:     episode$eye_exam<-as.numeric(as.character(episode$eye_exam))
  458:
  459:    for (thisyear in anayears) {
  460:
  461:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  462:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  463:
  464:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  465:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  466:
  467:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  468:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  469:
  470:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  471:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  472:
  473:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  474:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  475:
  476:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  477:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  478:
  479:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  480:
  481:     data3_3_2_1<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  482:                                   episode=episode[is.na(episode$eye_exam)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  483:                                   var=c("age_c","type_dm"),
  484:                                   date="last episode date",
  485:                                   startvar="dob",
  486:                                   th=ageth,
  487:                                   hold=TRUE,
```

```
                                    varname="age")

    eye_e_status<-BIRO_aggregate(data=episode[is.na(episode$eye_exam)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
                                    request=c("patient_id"),
                                    fun="min",
                                    vard="eye_exam",
                                    isdate=FALSE)

    names(eye_e_status)<-c("patient_id","eye_exam")
    data3_3_2_1<-merge(data3_3_2_1,eye_e_status,by="patient_id",all.x=TRUE)
    data3_3_2_1$foot_exam[is.na(data3_3_2_1$eye_exam)==TRUE]<-2
    data3_3_2_1<-na.omit(data3_3_2_1[,c("age_c","type_dm","eye_exam")])


       BIRO_report(ind="3_3_2_1",
                   title="3.3.2.1 Done (last episode in 12 months)",
                   var_cat="eye_exam",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=names(leveye_exam),
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Eye Examination",
                   lab_class=c("Age Classes","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dirtables=dirtables,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex)


  rm(data3_3_2_1)
```

```
  532:   }
  533:   } else BIRO_report_toc(title="3.3.2.1 Done (last episode in 12 months)")
  534:
  535: # Measurements done
  536:
  537:    new_sub_section_html(title="3.3.3. Measurements done",htmlfile=htmlfile)
  538:    new_sub_section(title="3.3.3 Measurements examination",texfile=texfile)
  539:
  540:
  541:   # 3.3.3.1 BP
  542:
  543: if ((length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
  544:       length(na.omit(episode$sbp))>0) |
  545:      (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
  546:       length(na.omit(episode$dbp))>0)) {
  547:
  548:
  549:      episode$bp<-ifelse(is.na(episode$sbp)==FALSE |
  550:                                 is.na(episode$dbp)==FALSE,1,2)
  551:
  552:    for (thisyear in anayears) {
  553:
  554:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  555:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  556:
  557:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  558:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  559:
  560:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  561:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  562:
  563:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  564:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  565:
  566:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  567:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  568:
  569:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  570:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  571:
  572:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  573:
  574:     data3_3_3_1<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  575:                                   episode=episode[is.na(episode$bp)==FALSE & episode$epi_date >= oneyearbf &
```

```
episode$epi_date <= checkdate,],
  576:                                    var=c("age_c","type_dm"),
  577:                                    date="last episode date",
  578:                                    startvar="dob",
  579:                                    th=ageth,
  580:                                    hold=TRUE,
  581:                                    varname="age")
  582:
  583:     bp_status<-BIRO_aggregate(data=episode[is.na(episode$bp)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  584:                                  request=c("patient_id"),
  585:                                  fun="min",
  586:                                  vard="bp",
  587:                                  isdate=FALSE)
  588:
  589:     names(bp_status)<-c("patient_id","bp_exam")
  590:     data3_3_3_1<-merge(data3_3_3_1,bp_status,by="patient_id",all.x=TRUE)
  591:     data3_3_3_1$bp[is.na(data3_3_3_1$bp)==TRUE]<-2
  592:     data3_3_3_1<-na.omit(data3_3_3_1[,c("age_c","type_dm","bp_exam")])
  593:
  594:        BIRO_report(ind="3_3_3_1",
  595:                    title="3.3.3.1 BP (last episode in 12 months)",
  596:                    var_cat="bp_exam",
  597:                    class=c("age_c",NA,"type_dm"),
  598:                    lev_var_cat=names(levbp),
  599:                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
  600:                    lab_var_cat="BP measurements",
  601:                    lab_class=c("Age Classes","Type of Diabetes"),
  602:                    tab=1,
  603:                    tabside="H",
  604:                    tabperc=TRUE,
  605:                    tabwidth=1,
  606:                    barbeside=TRUE,
  607:                    barnumber=TRUE,
  608:                    barperc=TRUE,
  609:                    box=0,
  610:                    bar=1,
  611:                    lines=0,
  612:                    trellis=0,
  613:                    pie=0,
  614:                    texfile=texfile,
  615:                    dirtables=dirtables,
  616:                    dircsv=dirdataout,
  617:                    dirgraph=dirgraph,
  618:                    dirgraphlatex=dirgraphlatex,
```

```
  619:                     cex=cex)
  620:
  621:   rm(data3_3_3_1)
  622:      }
  623:   } else BIRO_report_toc(title="3.3.3.1 Done (last episode in 12 months)")
  624:
  625:
  626:   #3.3.3.2 Lipids
  627:
  628:   if ((length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
  629:        length(na.omit(episode$chol))>0) |
  630:     (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
  631:      length(na.omit(episode$hdl))>0)) {
  632:
  633:      episode$lipids<-ifelse(is.na(episode$chol)==FALSE | is.na(episode$hdl)==FALSE,1,2)
  634:
  635:    for (thisyear in anayears) {
  636:
  637:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  638:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  639:
  640:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  641:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  642:
  643:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  644:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  645:
  646:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  647:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  648:
  649:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  650:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  651:
  652:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  653:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  654:
  655:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  656:
  657:     data3_3_3_2<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  658:                                   episode=episode[is.na(episode$lipids)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  659:                                   var=c("age_c","type_dm"),
  660:                                   date="last episode date",
  661:                                   startvar="dob",
```

```
                                    th=ageth,
                                    hold=TRUE,
                                    varname="age")

    lipids_status<-BIRO_aggregate(data=episode[is.na(episode$lipids)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
                                    request=c("patient_id"),
                                    fun="min",
                                    vard="lipids",
                                    isdate=FALSE)

    names(lipids_status)<-c("patient_id","lipid_exam")

    data3_3_3_2<-merge(data3_3_3_2,lipids_status,by="patient_id",all.x=TRUE)
    data3_3_3_2$lipids[is.na(data3_3_3_2$lipid_exam)]=2
    data3_3_3_2<-na.omit(data3_3_3_2[,c("age_c","type_dm","lipid_exam")])


       BIRO_report(ind="3_3_3_2",
                   title="3.3.3.2 Lipids",
                   var_cat="lipid_exam",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=names(levlipids),
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Lipids measurements",
                   lab_class=c("Age Classes","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dirtables=dirtables,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex)

```

```
  706:   rm(data3_3_3_2)
  707:   }
  708:   } else BIRO_report_toc(title="3.3.3.2 Lipids (last episode in 12 months)")
  709:
  710:
  711:   #3.3.3.3 MA_test
  712:
  713:
  714:  if (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
  715:      length(na.omit(episode$ma_test))>0) {
  716:
  717:    for (thisyear in anayears) {
  718:
  719:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  720:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  721:
  722:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  723:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  724:
  725:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  726:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  727:
  728:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  729:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  730:
  731:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  732:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  733:
  734:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  735:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  736:
  737:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  738:
  739:     data3_3_3_3<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  740:                                   episode=episode[is.na(episode$ma_test)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  741:                                   var=c("age_c","type_dm"),
  742:                                   date="last episode date",
  743:                                   startvar="dob",
  744:                                   th=ageth,
  745:                                   hold=TRUE,
  746:                                   varname="age")
  747:
  748:     ma_test_status<-BIRO_aggregate(data=episode[is.na(episode$ma_test)==FALSE & episode$epi_date >= oneyearbf &
```

```
episode$epi_date <= checkdate,],
  749:                                    request=c("patient_id"),
  750:                                    fun="min",
  751:                                    vard="ma_test",
  752:                                    isdate=FALSE)
  753:
  754:     names(ma_test_status)<-c("patient_id","ma_test")
  755:     data3_3_3_3<-merge(data3_3_3_3,ma_test_status,by="patient_id",all.x=TRUE)
  756:     data3_3_3_3$ma_test[is.na(data3_3_3_3$ma_test)]<-2
  757:     data3_3_3_3<-na.omit(data3_3_3_3[,c("ma_test","age_c","type_dm")])
  758:
  759:        BIRO_report(ind="3_3_3_3",
  760:                    title="3.3.3.3 Microalbumin (last episode in 12 months)",
  761:                    var_cat="ma_test",
  762:                    class=c("age_c",NA,"type_dm"),
  763:                    lev_var_cat=names(levma_test),
  764:                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
  765:                    lab_var_cat="Microalbumin done",
  766:                    lab_class=c("Age Classes","Type of Diabetes"),
  767:                    tab=1,
  768:                    tabside="H",
  769:                    tabperc=TRUE,
  770:                    tabwidth=1,
  771:                    barbeside=TRUE,
  772:                    barnumber=TRUE,
  773:                    barperc=TRUE,
  774:                    box=0,
  775:                    bar=1,
  776:                    lines=0,
  777:                    trellis=0,
  778:                    pie=0,
  779:                    texfile=texfile,
  780:                    dirtables=dirtables,
  781:                    dircsv=dirdataout,
  782:                    dirgraph=dirgraph,
  783:                    dirgraphlatex=dirgraphlatex,
  784:                    cex=cex)
  785:
  786:
  787:     rm(data3_3_3_3)
  788:   }
  789:   } else BIRO_report_toc(title="3.3.3.3 Microalbumin (last episode in 12 months)")
  790:
  791:  #
  792:  # 3.3.3.4 Hba1c
```

```
  793:  #
  794:  if (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
  795:      length(na.omit(episode$hba1c))>0) {
  796:
  797:    for (thisyear in anayears) {
  798:
  799:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  800:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  801:
  802:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  803:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  804:
  805:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  806:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  807:
  808:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  809:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  810:
  811:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  812:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  813:
  814:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  815:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  816:
  817:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  818:
  819:   data3_3_3_4<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  820:                                 episode=episode[is.na(episode$hba1c)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  821:                                 var=c("age_c","type_dm"),
  822:                                 date="last episode date",
  823:                                 startvar="dob",
  824:                                 th=ageth,
  825:                                 hold=TRUE,
  826:                                 varname="age")
  827:
  828:   hba1c_status<-BIRO_aggregate(data=episode[is.na(episode$hba1c)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  829:                                request=c("patient_id"),
  830:                                fun="min",
  831:                                vard="hba1c",
  832:                                isdate=FALSE)
  833:
  834:   names(hba1c_status)<-c("patient_id","hba1c_done")
```

```
  835:
  836:   hba1c_status<-merge(patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],hba1c_status,by="patient_id",
  837:         all.x=TRUE,all.y=TRUE)
  838:   hba1c_status<-hba1c_status[,c("patient_id","hba1c_done")]
  839:
  840:   hba1c_status$hba1c_done[is.na(hba1c_status$hba1c_done)]<-2
  841:
  842:   data3_3_3_4<-merge(data3_3_3_4,hba1c_status,by="patient_id",all=TRUE)
  843:   data3_3_3_4$hba1c_done[is.na(data3_3_3_4$hba1c_done)==FALSE]=1
  844:   data3_3_3_4$hba1c_done[is.na(data3_3_3_4$hba1c_done)]<-2
  845:   data3_3_3_4<-na.omit(data3_3_3_4[,c("age_c","type_dm","hba1c_done")])
  846:
  847:        BIRO_report(ind="3_3_3_4",
  848:                    title="3.3.3.4 HbA1c (last episode in 12 months)",
  849:                    var_cat="hba1c_done",
  850:                    class=c("age_c",NA,"type_dm"),
  851:                    lev_var_cat=names(levhba1c_done),
  852:                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
  853:                    lab_var_cat="HbA1c done",
  854:                    lab_class=c("Age Classes","Type of Diabetes"),
  855:                    tab=1,
  856:                    tabside="H",
  857:                    tabperc=TRUE,
  858:                    tabwidth=1,
  859:                    barbeside=TRUE,
  860:                    barnumber=TRUE,
  861:                    barperc=TRUE,
  862:                    box=0,
  863:                    bar=1,
  864:                    lines=0,
  865:                    trellis=0,
  866:                    pie=0,
  867:                    texfile=texfile,
  868:                    dirtables=dirtables,
  869:                    dircsv=dirdataout,
  870:                    dirgraph=dirgraph,
  871:                    dirgraphlatex=dirgraphlatex,
  872:                    cex=cex)
  873:
  874:   rm(data3_3_3_4)
  875:   }
  876:   } else BIRO_report_toc(title="3.3.3.4 HbA1c (last episode in 12 months)")
  877:
  878:
```

```
  879:  ##Treatment
  880:
  881:  new_sub_section_html(title="3.3.4 Treatment",htmlfile=htmlfile)
  882:  new_sub_section(title="3.3.4 Treatment",texfile=texfile)
  883:
  884:  #
  885:  # 3.3.4.1 Antihypertensive Medication (only epirefyear)
  886:  #
  887:
  888:  if ((is.null(patient$dob)==F) & (is.null(patient$type_dm)==F) &
  889:      (is.null(episode$hypert_med)==F)) {
  890:
  891:  if (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
  892:      length(na.omit(episode$hypert_med))>0) {
  893:
  894:    for (thisyear in anayears) {
  895:
  896:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  897:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  898:
  899:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  900:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  901:
  902:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  903:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  904:
  905:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  906:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  907:
  908:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  909:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  910:
  911:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  912:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  913:
  914:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  915:
  916:   data3_3_4_1<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  917:                                 episode=episode[is.na(episode$hypert_med)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
  918:                                 var=c("patient_id","age_c","type_dm"),
  919:                                 date="last episode date",
  920:                                 startvar="dob",
  921:                                 th=ageth,
```

```
                                    hold=FALSE,
                                    varname="age")

  hypert_med_status<-BIRO_aggregate(data=episode[is.na(episode$hypert_med)==FALSE & episode$epi_date >= oneyearbf
& episode$epi_date <= checkdate,],
                                        request=c("patient_id"),
                                        fun="min",
                                        vard="hypert_med",
                                        isdate=FALSE)

  names(hypert_med_status)<-c("patient_id","hypert_med")
  data3_3_4_1<-merge(data3_3_4_1,hypert_med_status,by="patient_id",all.x=TRUE)
  data3_3_4_1$hypert_med[is.na(data3_3_4_1$hypert_med)]<-2
  data3_3_4_1<-na.omit(data3_3_4_1[,c("age_c","type_dm","hypert_med")])

       BIRO_report(ind="3_3_4_1",
                   title="3.3.4.1 Antihypertensive Medication (last episode in 12 months)",
                   var_cat="hypert_med",
                   class=c("age_c",NA,"type_dm"),
                   lev_var_cat=names(levhypert_med),
                   lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                   lab_var_cat="Antihypertensive Medication",
                   lab_class=c("Age Classes","Type of Diabetes"),
                   tab=1,
                   tabside="H",
                   tabperc=TRUE,
                   tabwidth=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   bar=1,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dirtables=dirtables,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex)


  rm(data3_3_4_1)
  }
```

```
  966:   }
  967:   } else BIRO_report_toc(title="3.3.4.1 Antihypertensive Medication (last episode in 12 months)")
  968:
  969:  #
  970:  # 3.3.4.2 Lipid Lowering Treatment
  971:  # URGENT: this must be specified in the Dictionary and DATABASE
  972:  # TODO Create variable lipid_med
  973:
  974:  if ((is.null(patient$dob)==F) & (is.null(patient$type_dm)==F) &
  975:      (is.null(episode$lipid_med)==F)) {
  976:
  977:  if (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
  978:      length(na.omit(episode$lipid_med))>0) {
  979:
  980:    for (thisyear in anayears) {
  981:
  982:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  983:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  984:
  985:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  986:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  987:
  988:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  989:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  990:
  991:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  992:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  993:
  994:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  995:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  996:
  997:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  998:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  999:
 1000:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1001:
 1002:   data3_3_4_2<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1003:                                 episode=episode[is.na(episode$lipid_med)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
 1004:                                 var=c("patient_id","age_c","type_dm"),
 1005:                                 date="last episode date",
 1006:                                 startvar="dob",
 1007:                                 th=ageth,
 1008:                                 hold=FALSE,
```

```
1009:                                   varname="age")
1010:   lipid_med_status<-BIRO_aggregate(data=episode[is.na(episode$lipid_med)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
1011:                                       request=c("patient_id"),
1012:                                       fun="min",
1013:                                       vard="lipid_med",
1014:                                       isdate=FALSE)
1015:   names(lipid_med_status)<-c("patient_id","lipid_med")
1016:   data3_3_4_2<-merge(data3_3_4_2,lipid_med_status,by="patient_id",all.x=TRUE)
1017:   data3_3_4_2$lipid_med[is.na(data3_3_4_2$lipid_med)]<-2
1018:   data3_3_4_2<-na.omit(data3_3_4_2[,c("age_c","type_dm","lipid_med")])
1019:
1020:        BIRO_report(ind="3_3_4_2",
1021:                    title="3.3.4.2 Lipid Lowering Medication (last episode in 12 months)",
1022:                    var_cat="lipid_med",
1023:                    class=c("age_c",NA,"type_dm"),
1024:                    lev_var_cat=names(levlipid_med),
1025:                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
1026:                    lab_var_cat="Lipid Lowering Medication",
1027:                    lab_class=c("Age Classes","Type of Diabetes"),
1028:                    tab=1,
1029:                    tabside="H",
1030:                    tabperc=TRUE,
1031:                    tabwidth=1,
1032:                    barbeside=TRUE,
1033:                    barnumber=TRUE,
1034:                    barperc=TRUE,
1035:                    box=0,
1036:                    bar=1,
1037:                    lines=0,
1038:                    trellis=0,
1039:                    pie=0,
1040:                    texfile=texfile,
1041:                    dirtables=dirtables,
1042:                    dircsv=dirdataout,
1043:                    dirgraph=dirgraph,
1044:                    dirgraphlatex=dirgraphlatex,
1045:                    cex=cex
1046:                    )
1047:
1048:
1049:   rm(data3_3_4_2)
1050:   }
1051:   }
1052:   } else BIRO_report_toc(title="3.3.4.2 Lipid Lowering Medication (last episode in 12 months)")
```

```
1053:
1054:  #
1055:  # 3.3.4.3 ASA Treatment
1056:  # URGENT: this must be specified in the Dictionary and DATABASE
1057:  # TODO Create variable asa_med
1058:
1059:  if ((is.null(patient$dob)==F) & (is.null(patient$type_dm)==F) &
1060:      (is.null(episode$asa_med)==F)) {
1061:
1062:  if (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
1063:      length(na.omit(episode$asa_med))>0) {
1064:
1065:    for (thisyear in anayears) {
1066:
1067:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1068:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1069:
1070:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1071:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1072:
1073:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1074:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1075:
1076:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1077:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1078:
1079:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1080:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
1081:
1082:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
1083:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1084:
1085:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
1086:
1087:   data3_3_4_3<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
1088:                                 episode=episode[is.na(episode$asa_med)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
1089:                                 var=c("patient_id","age_c","type_dm"),
1090:                                 date="last episode date",
1091:                                 startvar="dob",
1092:                                 th=ageth,
1093:                                 hold=FALSE,
1094:                                 varname="age")
1095:   asa_med_status<-BIRO_aggregate(data=episode[is.na(episode$asa_med)==FALSE & episode$epi_date >= oneyearbf &
```

```
episode$epi_date <= checkdate,],
 1096:                                    request=c("patient_id"),
 1097:                                    fun="min",
 1098:                                    vard="asa_med",
 1099:                                    isdate=FALSE)
 1100:   names(asa_med_status)<-c("patient_id","asa_med")
 1101:   data3_3_4_3<-merge(data3_3_4_3,asa_med_status,by="patient_id",all.x=TRUE)
 1102:   data3_3_4_3$asa_med[is.na(data3_3_4_3$asa_med)]=2
 1103:   data3_3_4_3<-na.omit(data3_3_4_3[,c("age_c","type_dm","asa_med")])
 1104:
 1105:        BIRO_report(ind="3_3_4_3",
 1106:                    title="3.3.4.3 ASA Medication (last episode in 12 months)",
 1107:                    var_cat="asa_med",
 1108:                    class=c("age_c",NA,"type_dm"),
 1109:                    lev_var_cat=names(levasa_med),
 1110:                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
 1111:                    lab_var_cat="ASA Medication",
 1112:                    lab_class=c("Age Classes","Type of Diabetes"),
 1113:                    tab=1,
 1114:                    tabside="H",
 1115:                    tabperc=TRUE,
 1116:                    tabwidth=1,
 1117:                    barbeside=TRUE,
 1118:                    barnumber=TRUE,
 1119:                    barperc=TRUE,
 1120:                    box=0,
 1121:                    bar=1,
 1122:                    lines=0,
 1123:                    trellis=0,
 1124:                    pie=0,
 1125:                    texfile=texfile,
 1126:                    dirtables=dirtables,
 1127:                    dircsv=dirdataout,
 1128:                    dirgraph=dirgraph,
 1129:                    dirgraphlatex=dirgraphlatex,
 1130:                    cex=cex
 1131:                    )
 1132:
 1133:
 1134:   rm(data3_3_4_3)
 1135:   }
 1136:   }
 1137:   } else BIRO_report_toc(title="3.3.4.3 ASA Medication (last episode in 12 months)")
 1138:
 1139:   #
```

```
1140:   # 3.3.4.4 Drug Therapy (only epirefyear)
1141:   #
1142:   if ((length(na.omit(patient$dob))>0) & (length(na.omit(patient$type_dm))>0) &
1143:       (length(na.omit(episode$drug_therapy))>0)) {
1144:
1145:   episode$drug_therapy=as.numeric(as.character(episode$drug_therapy))
1146:
1147:    for (thisyear in anayears) {
1148:
1149:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1150:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1151:
1152:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1153:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1154:
1155:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1156:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1157:
1158:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1159:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1160:
1161:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1162:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
1163:
1164:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
1165:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1166:
1167:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
1168:
1169:   data3_3_4_4_1<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
1170:                                  episode=episode[is.na(episode$drug_therapy)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
1171:                                  var=c("patient_id","age_c","type_dm"),
1172:                                  date="last episode date",
1173:                                  startvar="dob",
1174:                                  th=ageth,
1175:                                  hold=FALSE,
1176:                                  varname="age")
1177:
1178:   diet_only<-cbind(unique(episode[episode$drug_therapy==4 & is.na(episode$drug_therapy)==FALSE & episode$epi_date
>= oneyearbf & episode$epi_date <= checkdate,
1179:                       c("patient_id")]),1)
1180:   diet_only<-as.data.frame(diet_only)
1181:   names(diet_only)<-c("patient_id","diet_only")
```

```
1182:   data3_3_4_4_1<-merge(data3_3_4_4_1,
1183:                        diet_only,
1184:                        by="patient_id",
1185:                        all.x=TRUE,
1186:                        all.y=FALSE)
1187:
1188:   data3_3_4_4_1$diet_only[is.na(data3_3_4_4_1$diet_only)==TRUE]<-2
1189:
1190:        BIRO_report(ind="3_3_4_4_1",
1191:                    title="3.3.4.4.1 Glucose Lowering: Diet Only (last episode in 12 months)",
1192:                    var_cat="diet_only",
1193:                    class=c("age_c",NA,"type_dm"),
1194:                    lev_var_cat=names(levdiet),
1195:                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
1196:                    lab_var_cat="Diet Only",
1197:                    lab_class=c("Age Classes","Type of Diabetes"),
1198:                    tab=1,
1199:                    tabside="H",
1200:                    tabperc=TRUE,
1201:                    tabwidth=1,
1202:                    barbeside=TRUE,
1203:                    barnumber=TRUE,
1204:                    barperc=TRUE,
1205:                    box=0,
1206:                    bar=1,
1207:                    lines=0,
1208:                    trellis=0,
1209:                    pie=0,
1210:                    texfile=texfile,
1211:                    dirtables=dirtables,
1212:                    dircsv=dirdataout,
1213:                    dirgraph=dirgraph,
1214:                    dirgraphlatex=dirgraphlatex,
1215:                    cex=cex
1216:                    )
1217:
1218:
1219:  rm(data3_3_4_4_1)
1220:
1221:   data3_3_4_4_2<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
1222:                                  episode=episode[is.na(episode$drug_therapy)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
1223:                                  var=c("patient_id","age_c","type_dm"),
1224:                                  date="last episode date",
```

```
1225:                                   startvar="dob",
1226:                                   th=ageth,
1227:                                   hold=FALSE,
1228:                                   varname="age")
1229:
1230:   tablets_only<-cbind(unique(episode[episode$drug_therapy==2 & is.na(episode$drug_therapy)==FALSE &
episode$epi_date >= oneyearbf & episode$epi_date <= checkdate,
1231:                       c("patient_id")]),1)
1232:   tablets_only<-as.data.frame(tablets_only)
1233:   names(tablets_only)<-c("patient_id","tablets_only")
1234:
1235:   data3_3_4_4_2<-merge(data3_3_4_4_2,
1236:                        tablets_only,
1237:                        by="patient_id",
1238:                        all.x=TRUE,
1239:                        all.y=FALSE)
1240:
1241:   data3_3_4_4_2$tablets_only[is.na(data3_3_4_4_2$tablets_only)==TRUE]<-2
1242:
1243:
1244:        BIRO_report(ind="3_3_4_4_2",
1245:                    title="3.3.4.4.2 Glucose Lowering: Tablets Only (last episode in 12 months)",
1246:                    var_cat="tablets_only",
1247:                    class=c("age_c",NA,"type_dm"),
1248:                    lev_var_cat=names(levtablet),
1249:                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
1250:                    lab_var_cat="Tablets Only",
1251:                    lab_class=c("Age Classes","Type of Diabetes"),
1252:                    tab=1,
1253:                    tabside="H",
1254:                    tabperc=TRUE,
1255:                    tabwidth=1,
1256:                    barbeside=TRUE,
1257:                    barnumber=TRUE,
1258:                    barperc=TRUE,
1259:                    box=0,
1260:                    bar=1,
1261:                    lines=0,
1262:                    trellis=0,
1263:                    pie=0,
1264:                    texfile=texfile,
1265:                    dirtables=dirtables,
1266:                    dircsv=dirdataout,
1267:                    dirgraph=dirgraph,
1268:                    dirgraphlatex=dirgraphlatex,
```

```
 1269:                     cex=cex
 1270:                     )
 1271:
 1272:  rm(data3_3_4_4_2)
 1273:
 1274:
 1275:   data3_3_4_4_3<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1276:                                   episode=episode[is.na(episode$drug_therapy)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
 1277:                                   var=c("patient_id","age_c","type_dm"),
 1278:                                   date="last episode date",
 1279:                                   startvar="dob",
 1280:                                   th=ageth,
 1281:                                   hold=FALSE,
 1282:                                   varname="age")
 1283:   insulin_only<-cbind(unique(episode[episode$drug_therapy==1 & is.na(episode$drug_therapy)==FALSE &
episode$epi_date >= oneyearbf & episode$epi_date <= checkdate,
 1284:                         c("patient_id")]),1)
 1285:   insulin_only<-as.data.frame(insulin_only)
 1286:   names(insulin_only)<-c("patient_id","insulin_only")
 1287:
 1288:   data3_3_4_4_3<-merge(data3_3_4_4_3,
 1289:                        insulin_only,
 1290:                        by="patient_id",
 1291:                        all.x=TRUE,
 1292:                        all.y=FALSE)
 1293:
 1294:   data3_3_4_4_3$insulin_only[is.na(data3_3_4_4_3$insulin_only)==TRUE]<-2
 1295:
 1296:        BIRO_report(ind="3_3_4_4_3",
 1297:                    title="3.3.4.4.3 Glucose Lowering: Insulin Only (last episode in 12 months)",
 1298:                    var_cat="insulin_only",
 1299:                    class=c("age_c",NA,"type_dm"),
 1300:                    lev_var_cat=names(levinsulin),
 1301:                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
 1302:                    lab_var_cat="Insulin Only",
 1303:                    lab_class=c("Age Classes","Type of Diabetes"),
 1304:                    tab=1,
 1305:                    tabside="H",
 1306:                    tabperc=TRUE,
 1307:                    tabwidth=1,
 1308:                    barbeside=TRUE,
 1309:                    barnumber=TRUE,
 1310:                    barperc=TRUE,
```

```
 1311:                     box=0,
 1312:                     bar=1,
 1313:                     lines=0,
 1314:                     trellis=0,
 1315:                     pie=0,
 1316:                     texfile=texfile,
 1317:                     dirtables=dirtables,
 1318:                     dircsv=dirdataout,
 1319:                     dirgraph=dirgraph,
 1320:                     dirgraphlatex=dirgraphlatex,
 1321:                     cex=cex
 1322:                     )
 1323:
 1324:  rm(data3_3_4_4_3)
 1325:
 1326:   data3_3_4_4_4<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1327:                                  episode=episode[is.na(episode$drug_therapy)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
 1328:                                  var=c("patient_id","age_c","type_dm"),
 1329:                                  date="last episode date",
 1330:                                  startvar="dob",
 1331:                                  th=ageth,
 1332:                                  hold=FALSE,
 1333:                                  varname="age")
 1334:   instab<-cbind(unique(episode[episode$drug_therapy==3 &is.na(episode$drug_therapy)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,
 1335:           c("patient_id")]),1)
 1336:   instab<-as.data.frame(instab)
 1337:   names(instab)<-c("patient_id","instab")
 1338:
 1339:   data3_3_4_4_4<-merge(data3_3_4_4_4,
 1340:                        instab,
 1341:                        by="patient_id",
 1342:                        all.x=TRUE,all.y=FALSE)
 1343:
 1344:   data3_3_4_4_4$instab[is.na(data3_3_4_4_4$instab)==TRUE]<-2
 1345:
 1346:        BIRO_report(ind="3_3_4_4_4",
 1347:                    title="3.3.4.4.4 Glucose Lowering: Insulin and Tablets (last episode in 12 months)",
 1348:                    var_cat="instab",
 1349:                    class=c("age_c",NA,"type_dm"),
 1350:                    lev_var_cat=names(levinstab),
 1351:                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
 1352:                    lab_var_cat="Insulin And Tablets",
```

```
1353:                     lab_class=c("Age Classes","Type of Diabetes"),
1354:                     tab=1,
1355:                     tabside="H",
1356:                     tabperc=TRUE,
1357:                     tabwidth=1,
1358:                     barbeside=TRUE,
1359:                     barnumber=TRUE,
1360:                     barperc=TRUE,
1361:                     box=0,
1362:                     bar=1,
1363:                     lines=0,
1364:                     trellis=0,
1365:                     pie=0,
1366:                     texfile=texfile,
1367:                     dirtables=dirtables,
1368:                     dircsv=dirdataout,
1369:                     dirgraph=dirgraph,
1370:                     dirgraphlatex=dirgraphlatex,
1371:                     cex=cex
1372:                     )
1373:
1374:  rm(data3_3_4_4_4)
1375: }
1376: } else { BIRO_report_toc(title="3.3.4.4.1 Glucose Lowering: Diet Only (last episode in 12 months)")
1377:          BIRO_report_toc(title="3.3.4.4.2 Glucose Lowering: Tablets Only (last episode in 12 months)")
1378:          BIRO_report_toc(title="3.3.4.4.3 Glucose Lowering: Insulin Only (last episode in 12 months)")
1379:          BIRO_report_toc(title="3.3.4.4.4 Glucose Lowering: Insulin and Tablets (last episode in 12 months)")
1380:        }
1381:
1382:   if (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
1383:       length(na.omit(episode$pump_therapy))>0) {
1384:
1385:    for (thisyear in anayears) {
1386:
1387:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1388:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1389:
1390:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1391:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1392:
1393:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1394:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1395:
1396:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1397:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
```

```
 1398:
 1399:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1400:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1401:
 1402:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1403:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1404:
 1405:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1406:
 1407:   episode$pump_therapy<-as.numeric(as.character(episode$pump_therapy))
 1408:
 1409:   data3_3_4_4_5<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1410:                                  episode=episode[is.na(episode$pump_therapy)==FALSE & episode$epi_date >=
oneyearbf & episode$epi_date <= checkdate,],
 1411:                                  var=c("patient_id","age_c","type_dm"),
 1412:                                  date="last episode date",
 1413:                                  startvar="dob",
 1414:                                  th=ageth,
 1415:                                  hold=FALSE,
 1416:                                  varname="age")
 1417:
 1418:   pump_status<-BIRO_aggregate(data=episode[is.na(episode$hba1c)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
 1419:                               request=c("patient_id"),
 1420:                               fun="min",
 1421:                               vard="pump_therapy",
 1422:                               isdate=FALSE)
 1423:
 1424:   names(pump_status)<-c("patient_id","pump_therapy")
 1425:   data3_3_4_4_5<-merge(data3_3_4_4_5,pump_status,by="patient_id",all.x=TRUE)
 1426:   data3_3_4_4_5$pump_therapy[is.na(data3_3_4_4_5$pump_therapy)]=2
 1427:
 1428:        BIRO_report(ind="3_3_4_4_5",
 1429:                    title="3.3.4.4.5 Glucose Lowering: Insulin Pump (last episode in 12 months)",
 1430:                    var_cat="pump_therapy",
 1431:                    class=c("age_c",NA,"type_dm"),
 1432:                    lev_var_cat=names(levpump_med),
 1433:                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
 1434:                    lab_var_cat="Pump Therapy",
 1435:                    lab_class=c("Age Classes","Type of Diabetes"),
 1436:                    tab=1,
 1437:                    tabside="H",
 1438:                    tabperc=TRUE,
 1439:                    tabwidth=1,
```

```
1440:                     barbeside=TRUE,
1441:                     barnumber=TRUE,
1442:                     barperc=TRUE,
1443:                     box=0,
1444:                     bar=1,
1445:                     lines=0,
1446:                     trellis=0,
1447:                     pie=0,
1448:                     texfile=texfile,
1449:                     dirtables=dirtables,
1450:                     dircsv=dirdataout,
1451:                     dirgraph=dirgraph,
1452:                     dirgraphlatex=dirgraphlatex,
1453:                     cex=cex
1454:                    )
1455:
1456:
1457:   rm(data3_3_4_4_5)
1458:  }
1459:  } else BIRO_report_toc(title="3.3.4.4.5 Glucose Lowering: Insulin Pump (last episode in 12 months)")
1460:
1461:
1462:  ##Management
1463:
1464:  new_sub_section_html(title="3.3.5 Management",htmlfile=htmlfile)
1465:  new_sub_section(title="3.3.5 Management",texfile=texfile)
1466:
1467:  #
1468:  # 3.3.5.1 Selfmonitoring (only epirefyear)
1469:  #
1470:
1471:   if (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
1472:       length(na.omit(episode$self_mon))>0) {
1473:
1474:     episode$self_mon=as.numeric(as.character(episode$self_mon))
1475:
1476:    for (thisyear in anayears) {
1477:
1478:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1479:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
1480:
1481:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
1482:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1483:
1484:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
```

```
1485:
1486:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1487:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1488:
1489:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1490:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1491:
1492:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1493:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1494:
1495:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1496:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1497:
1498:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1499:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
1500:
1501:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
1502:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1503:
1504:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
1505:
1506:     data3_3_5_1<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
1507:                                   episode=episode[is.na(episode$self_mon)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
1508:                                   var=c("age_c","type_dm"),
1509:                                   date="last episode date",
1510:                                   startvar="dob",
1511:                                   th=ageth,
1512:                                   hold=TRUE,
1513:                                   varname="age")
1514:
1515:     self_mon_status<-BIRO_aggregate(data=episode[is.na(episode$self_mon)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
1516:                                   request=c("patient_id"),
1517:                                   fun="max",
1518:                                   vard="self_mon",
1519:                                   isdate=FALSE)
1520:
1521:     names(self_mon_status)<-c("patient_id","self_mon")
1522:     data3_3_5_1<-merge(data3_3_5_1,self_mon_status,by="patient_id",all.x=TRUE)
1523:     data3_3_5_1<-na.omit(data3_3_5_1[,c("age_c","type_dm","self_mon")])
1524:
1525:
1526:        BIRO_report(ind="3_3_5_1",
```

```
1527:                     title="3.3.5.1 Self monitoring",
1528:                     var_cat="self_mon",
1529:                     class=c("age_c",NA,"type_dm"),
1530:                     lev_var_cat=names(levself_mon),
1531:                     lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
1532:                     lab_var_cat="Self Monitoring",
1533:                     lab_class=c("Age Classes","Sex"),
1534:                     tab=1,
1535:                     tabside="H",
1536:                     tabperc=TRUE,
1537:                     tabwidth=1,
1538:                     barbeside=TRUE,
1539:                     barnumber=TRUE,
1540:                     barperc=TRUE,
1541:                     box=0,
1542:                     bar=1,
1543:                     lines=0,
1544:                     trellis=0,
1545:                     pie=0,
1546:                     texfile=texfile,
1547:                     dirtables=dirtables,
1548:                     dircsv=dirdataout,
1549:                     dirgraph=dirgraph,
1550:                     dirgraphlatex=dirgraphlatex,
1551:                     cex=cex
1552:                     )
1553:
1554:   rm(data3_3_5_1)
1555:    }
1556:   } else BIRO_report_toc(title="3.3.5.1 Self monitoring")
1557:
1558:
1559:  #
1560:  # 3.3.5.2  Visit Frequency
1561:  # URGENT: this must be specified in the Dictionary and DATABASE
1562:  # TODO Create variable visit_freq (1=Low,2=High)
1563:
1564:  if ((is.null(patient$dob)==F) & (is.null(patient$type_dm)==F) &
1565:      (is.null(patient$sex)==F)) {
1566:
1567:  if (length(na.omit(patient$dob))>0 & length(na.omit(patient$type_dm))>0 &
1568:      length(na.omit(patient$sex))>0) {
1569:
1570:    for (thisyear in anayears) {
1571:
```

```
 1572:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 1573:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 1574:        sixmbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")+180
 1575:
 1576:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
 1577:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1578:
 1579:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
 1580:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 1581:
 1582:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 1583:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 1584:
 1585:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1586:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1587:
 1588:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1589:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1590:
 1591:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1592:
 1593:   data3_3_5_2<-BIRO_demographic(patient=patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1594:                                 episode=episode[is.na(episode$self_mon)==FALSE & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,],
 1595:                                 var=c("patient_id","age_c","sex","type_dm"),
 1596:                                 date=checkdate,
 1597:                                 startvar="dob",
 1598:                                 th=ageth,
 1599:                                 hold=TRUE,
 1600:                                 varname="age")
 1601:
 1602:   episode$visits=1
 1603:
 1604:   visit_freq_status6m<-BIRO_aggregate(data=episode[episode$epi_date >= sixmbf & episode$epi_date <= checkdate,],
 1605:                                       request=c("patient_id"),
 1606:                                       fun="sum",
 1607:                                       vard="visits",
 1608:                                       isdate=FALSE)
 1609:   visit_freq_status6m$visits<-3
 1610:
 1611:   visit_freq_status12m<-BIRO_aggregate(data=episode[episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,],
 1612:                                       request=c("patient_id"),
 1613:                                       fun="sum",
```

```
                                    vard="visits",
                                    isdate=FALSE)

   visit_freq_status12m$visits<-2

   visit_freq_status<-merge(visit_freq_status12m,visit_freq_status12m,by="patient_id",all=TRUE)
   visit_freq_status$visits<-ifelse(is
.na(visit_freq_status$visits.x)==TRUE,visit_freq_status$visits.y,visit_freq_status$visits.x)
   visit_freq_status<-visit_freq_status[,c("patient_id","visits")]

   data3_3_5_2<-merge(data3_3_5_2,visit_freq_status,by="patient_id",all.x=TRUE)
   data3_3_5_2$visits[is.na(data3_3_5_2$visits)]=1

   names(data3_3_5_2)[pmatch("visits",names(data3_3_5_2))]<-"visit_freq"

   data3_3_5_2<-data3_3_5_2[,c("age_c","type_dm","visit_freq")]

        BIRO_report(ind="3_3_5_2",
                    title="3.3.5.2 Visit Frequency",
                    var_cat="visit_freq",
                    class=c("age_c",NA,"type_dm"),
                    lev_var_cat=names(levvisit_freq),
                    lev_class=list(a=classlabellist(ageth),b=NA,c=levtype_dm),
                    lab_var_cat="Visit Frequency",
                    lab_class=c("Age Classes","Sex"),
                    tab=1,
                    tabside="H",
                    tabperc=TRUE,
                    tabwidth=1,
                    barbeside=TRUE,
                    barnumber=TRUE,
                    barperc=TRUE,
                    box=0,
                    bar=1,
                    lines=0,
                    trellis=0,
                    pie=0,
                    texfile=texfile,
                    dirtables=dirtables,
                    dircsv=dirdataout,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex
                    )

```

```
1658:
1659:   rm(data3_3_5_2)
1660:   }
1661:      }
1662:   } else BIRO_report_toc(title="3.3.5.2 Visit Frequency")
1663:
```

```
#############################################################################
#                                                                           #
# 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 9 running Kernel:2.6.27.5-37.fc9.i686
# 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_indicator_population.r is part of WP Statistical Engine of the BIRO
# Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
#############################################################################

```

```
45:
46: # 4.1.1. Total population
47:
48: if (dim(pop)[1]>0) {
49:
50:    for (thisyear in anayears) {
51:
52:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
53:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
54:
55:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
56:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
57:
58:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
59:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
60:
61:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
62:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
63:
64:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
65:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
66:
67:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
68:
69:   data4_1_1<-pop[pop$year==thisyear,pmatch(c("age_band","popM","popF"),names(pop))]
70:   names(data4_1_1)<-c('Age in class','Males','Females')
71:   data4_1_1[,pmatch('Age in class',names(data4_1_1))]<-classlabel(seq(5,85,5))
72:
73:   BIRO_dframe(dataname="data4_1_1",monitor=TRUE,filelog=logfile)
74:   BIRO_dframe(dataname="data4_1_1",monitor=TRUE,filelog="")
75:
76:        BIRO_report(ind="4_1_1",
77:                    title="4.1.1 Total Population",
78:                    var=NULL,
79:                    var_cat=NULL,
80:                    lev_var_cat=NULL,
81:                    lev_class=NULL,
82:                    lab_var=NULL,
83:                    lab_var_cat=NULL,
84:                    lab_class=NULL,
85:                    tab=0,
86:                    box=0,
87:                    bar=0,
88:                    lines=0,
89:                    trellis=0,
```

```
                    pie=0,
                    texfile=texfile,
                    dirtables=dirtables,
                    dircsv=dirdataout,
                    dirgraph=dirgraph,
                    dirgraphlatex=dirgraphlatex,
                    cex=cex,
                    catcol=1
                    )

  rm(data4_1_1)
 }
} else BIRO_report_toc(title="4.1.1 Total Population")

# 4.1.2. Life expectancy


if (dim(pop)[1]>0) {

   for (thisyear in anayears) {

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

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

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

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

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

   assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)

  exm<-BIRO_explife(pop=pop[pop$year==thisyear,],
                    mor=pop[pop$year==thisyear,],
                    agewidth=5,
                    sex="M",
                    morname="mor",
                    popname="pop",
                    labsex='Males')
```

```
135:
136:   exf<-BIRO_explife(pop=pop[pop$year==thisyear,],
137:                     mor=pop[pop$year==thisyear,],
138:                     agewidth=5,
139:                     sex="F",
140:                     morname="mor",
141:                     popname="pop",
142:                     labsex='Females')
143:   data4_1_2<-as.data.frame(merge(exm,exf))
144:   data4_1_2$Males<-round(as.numeric(as.character(data4_1_2$Males,2)))
145:   data4_1_2$Females<-round(as.numeric(as.character(data4_1_2$Females,2)))
146:
147:
148:   BIRO_dframe(dataname="data4_1_2",monitor=TRUE,filelog=logfile)
149:   BIRO_dframe(dataname="data4_1_2",monitor=TRUE,filelog="")
150:
151:        BIRO_report(ind="4_1_2",
152:                    title="4.1.2 Life expectancy",
153:                    var=NULL,
154:                    var_cat=NULL,
155:                    lev_var_cat=NULL,
156:                    lev_class=NULL,
157:                    lab_var=NULL,
158:                    lab_var_cat=NULL,
159:                    lab_class=NULL,
160:                    tab=0,
161:                    box=0,
162:                    bar=0,
163:                    lines=0,
164:                    trellis=0,
165:                    pie=0,
166:                    texfile=texfile,
167:                    dirtables=dirtables,
168:                    dircsv=dirdataout,
169:                    dirgraph=dirgraph,
170:                    dirgraphlatex=dirgraphlatex,
171:                    cex=cex,
172:                    catcol=1
173:                    )
174:
175:   rm(list=c("exm","exf","data4_1_2"))
176:  }
177: } else BIRO_report_toc(title="4.1.2 Life expenctancy")
178:
179: # 4.1.3. Mortality data
```

```
180:
181: if (dim(pop)[1]>0) {
182:
183:  for (thisyear in anayears) {
184:
185:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
186:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
187:
188:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
189:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
190:
191:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
192:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
193:
194:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
195:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
196:
197:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
198:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
199:
200:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
201:
202:   data4_1_3<-pop[pop$year==thisyear,pmatch(c("age_band","morM","morF"),names(pop))]
203:   names(data4_1_3)<-c('Age in class','Males','Females')
204:   data4_1_3[,pmatch('Age in class',names(data4_1_3))]<-classlabel(seq(5,85,5))
205:
206:   BIRO_dframe(dataname="data4_1_3",monitor=TRUE,filelog=logfile)
207:   BIRO_dframe(dataname="data4_1_3",monitor=TRUE,filelog="")
208:
209:        BIRO_report(ind="4_1_3",
210:                    title="4.1.3 Mortality Data",
211:                    var=NULL,
212:                    var_cat=NULL,
213:                    lev_var_cat=NULL,
214:                    lev_class=NULL,
215:                    lab_var=NULL,
216:                    lab_var_cat=NULL,
217:                    lab_class=NULL,
218:                    tab=0,
219:                    box=0,
220:                    bar=0,
221:                    lines=0,
222:                    trellis=0,
223:                    pie=0,
224:                    texfile=texfile,
```

```
225:                     dirtables=dirtables,
226:                     dircsv=dirdataout,
227:                     dirgraph=dirgraph,
228:                     dirgraphlatex=dirgraphlatex,
229:                     cex=cex,
230:                     catcol=1
231:                     )
232:
233:   rm(data4_1_3)
234:  }
235: }  else BIRO_report_toc(title="4.1.3 Mortality Data")
236:
```

```
  1: # ############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: BIRO_se_indicator_risk_adjusted.r                                    #
  5: #                                                                            #
#
  6: # ############################################################################
  7: # ------------------------
  8: # Authors:
  9: # Luca Rossi       <redsluke@gmail.com)
 10: # Fabrizio Carinci <research@fabcarinci.net)
 11: # Institutions: Serectrix snc
 12: # Created: 2007-11-24
 13: # Version: 2008-11-24
 14: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 15: # Programming Language: R 2.8.0
 16: # ------------------------
 17: # COPYRIGHT INFORMATION
 18: #
 19: # This file is free software; you can redistribute it and/or modify
 20: # it under the terms of the GNU General Public License as published by
 21: # the Free Software Foundation; either version 2, or (at your option)
 22: # any later version.
 23: #
 24: # This file is distributed in the hope that it will be useful,
 25: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 26: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 27: # GNU General Public License for more details.
 28: #
 29: # You should have received a copy of the GNU General Public License
 30: # along with this file; see the file COPYING.  If not, write to
 31: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 32: #
 33: # In short: you may use this file any way you like, as long as you
 34: # don't charge money for it, remove this notice, or hold anyone liable
 35: # for its results.
 36: #
 37: # BIRO_se_indicator_risk_adjusted.r is part of WP Statistical Engine of the
 38: # BIRO Project
 39: #
 40: # GPL Copyright, The BIRO Project
 41: #
 42: # ------------------------
 43: #
 44: # ############################################################################
```

```
45:
46:  # 5.1 Epidemiology
47:
48: new_section(title="5.1. Epidemiology",texfile=texfile)
49: new_section_html(title="5.1. Epidemiology",htmlfile=htmlfile)
50:
51:  # 5.1.1 Prevalence of diabetes mellitus per 1,000
52:
53: if (dim(pop)[1]>0) {
54:
55:    for (thisyear in anayears) {
56:
57:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
58:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
59:
60:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
61:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
62:
63:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
64:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
65:
66:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
67:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
68:
69:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
70:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
71:
72:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
73:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
74:
75:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
76:
77:   pop$pop=pop$popM+pop$popF
78:   popyear<-pop[pop$year==thisyear,]
79:   if (is.null(pop$type)==TRUE) {
80:    poptype_dm<-ifelse(popyear$age_band<=7,1,2)
81:   }
82:   poptype<-aggregate(popyear,by=list(type_dm=poptype_dm),FUN=sum)
83:   rm(poptype_dm)
84:   poptype<-poptype[,c("type_dm","pop")]
85:
86:   BIRO_dframe(dataname="poptype",monitor=TRUE,filelog=logfile)
87:   BIRO_dframe(dataname="poptype",monitor=TRUE,filelog="")
88:
89:   checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
```

```
 90:
 91:   data5_1_1<-patient[patient$end_date>checkdate | is.na(patient$end_date),]
 92:
 93:   data5_1_1$age=round((as.Date(checkdate,format="%d/%m/%Y")-data5_1_1$dob)/365)
 94:   data5_1_1$age_c<-varclass(data5_1_1$age,ageth5)
 95:   data5_1_1$diabetes=1
 96:   data5_1_1<-data5_1_1[,c("diabetes","type_dm")]
 97:
 98:   BIRO_dframe(dataname="data5_1_1",monitor=TRUE,filelog=logfile)
 99:   BIRO_dframe(dataname="data5_1_1",monitor=TRUE,filelog="")
100:
101:
102:
103:        BIRO_report(ind="5_1_1",
104:                    title="5.1.1 Prevalence of diabetes mellitus per 1,000",
105:                    var=NULL,
106:                    var_cat=NULL,
107:                    lev_var_cat=NULL,
108:                    lev_class=NULL,
109:                    lab_var=NULL,
110:                    lab_var_cat=NULL,
111:                    lab_class=NULL,
112:                    tab=0,
113:                    box=0,
114:                    bar=0,
115:                    lines=0,
116:                    trellis=0,
117:                    pie=0,
118:                    texfile=texfile,
119:                    dirtables=dirtables,
120:                    dircsv=dirdataout,
121:                    dirgraph=dirgraph,
122:                    dirgraphlatex=dirgraphlatex,
123:                    cex=cex,
124:                    catcol=1,
125:                    stand=1,
126:                    outcome="diabetes",
127:                    cov=c("type_dm"),
128:                    factcov=c(TRUE),
129:                    strata=NULL,
130:                    levstd=list(a=list("Diabetic"=1),b=levtype_dm),
131:                    sucvalue=1,
132:                    per=1000,
133:                    pop=poptype,
134:                    caption="5_1_1",
```

```
135:                     chisq=NULL)
136:
137:   rm(data5_1_1)
138:    }
139: } else BIRO_report_toc(title="5.1.1 Prevalence of diabetes mellitus per 1,000")
140:
141:
142: #
143:
144:   if (length(na.omit(patient$dob))>0 & length(na.omit(patient$dt_diag))>0 &
145:       length(na.omit(patient$sex))>0 & length(na.omit(patient$type_dm))>0 ) {
146:
147:   new_sub_section("5.1.2. Age at diagnosis by 10 year age bands (incidence)",texfile=texfile)
148:
149:    for (thisyear in anayears) {
150:
151:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
152:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
153:
154:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
155:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
156:
157:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
158:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
159:
160:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
161:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
162:
163:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
164:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
165:
166:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
167:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
168:
169:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
170:
171:   data5_1_2<-unique(patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is.na(patient$end_date),
172:                            c("patient_id","dob","dt_diag","type_dm","sex")])
173:   data5_1_2$age=round((data5_1_2$dt_diag-data5_1_2$dob)/365)
174:   data5_1_2$age_c<-varclass(data5_1_2$age,ageth5)
175:   data5_1_2$diabetes=1
176:   data5_1_2<-data5_1_2[,c("diabetes","type_dm","age_c")]
177:
178:   BIRO_dframe(dataname="data5_1_2",monitor=TRUE,filelog=logfile)
179:   BIRO_dframe(dataname="data5_1_2",monitor=TRUE,filelog="")
```

```
180:
181:
182:
183:        BIRO_report(ind="5_1_2",
184:                    title="5.1.2. Age at diagnosis by 10 year age bands (incidence)",
185:                    var=NULL,
186:                    var_cat=NULL,
187:                    lev_var_cat=NULL,
188:                    lev_class=NULL,
189:                    lab_var=NULL,
190:                    lab_var_cat=NULL,
191:                    lab_class=NULL,
192:                    tab=0,
193:                    box=0,
194:                    bar=0,
195:                    lines=0,
196:                    trellis=0,
197:                    pie=0,
198:                    texfile=texfile,
199:                    dirtables=dirtables,
200:                    dircsv=dirdataout,
201:                    dirgraphlatex=dirgraphlatex,
202:                    dirgraph=dirgraph,
203:                    cex=cex,
204:                    catcol=c(2,3),
205:                    stand=1,
206:                    outcome="diabetes",
207:                    cov=c("type_dm","age_c"),
208:                    factcov=c(TRUE,TRUE),
209:                    strata=NULL,
210:                    levstd=list(a=list("Diabetic"=1),b=levtype_dm,c=classlabellist(ageth5)),
211:                    sucvalue=1,
212:                    per=1000,
213:                    caption="Age at diagnosis by 10 year age bands (incidence)",
214:                    printrate=FALSE,
215:                    chisq=NULL)
216:
217:   rm(data5_1_2)
218:    }
219: } else BIRO_report_toc(title="5.1.2. Age at diagnosis by 10 year age bands (incidence)")
220: #
221:
222: new_section(title="5.2. Process Quality",texfile=texfile)
223: new_section_html(title="5.2. Process Quality",htmlfile=htmlfile)
224:
```

```
  225:
  226:   #5.2.1.
  227:   #Percentage of patients with one or more HbA1c tests
  228:   #during the last 12 months
  229:
  230: if (is.null(episode$hba1c)==F) {
  231: if (length(episode$hba1c)>0) {
  232:
  233:    for (thisyear in anayears) {
  234:
  235:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  236:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  237:
  238:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  239:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  240:
  241:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  242:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  243:
  244:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  245:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  246:
  247:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  248:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  249:
  250:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  251:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  252:
  253:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  254:
  255:     noNAepirefyear<-episode[is.na(episode$hba1c)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
  256:     noNAepirefyear$hba1c_done=1
  257:
  258:   data5_2_1<-as.numeric(as.character(noNAepirefyear$hba1c_done))
  259:
  260:   data5_2_1<-as.data.frame(aggregate(data5_2_1,list(noNAepirefyear$patient_id),FUN="min"))
  261:   names(data5_2_1)<-c("patient_id","hba1c_done12")
  262:
  263:   data5_2_1<-merge(data5_2_1,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  264:                    by="patient_id",all=TRUE)
  265:
  266:   data5_2_1$age=(checkdate-data5_2_1$dob)/365.25
  267:   data5_2_1$age_c=varclass(data5_2_1$age,ageth)
```

```
  data5_2_1$hba1c_done12[is.na(data5_2_1$hba1c_done12)]<-2

  data5_2_1<-data5_2_1[,c("hba1c_done12","age_c","sex","type_dm")]

  BIRO_dframe(dataname="data5_2_1",monitor=TRUE,filelog=logfile)
  BIRO_dframe(dataname="data5_2_1",monitor=TRUE,filelog="")

       BIRO_report(ind="5_2_1",
                   title="5.2.1 Percentage of patients with one or more HbA1c tests during the last 12 months",
                   var_cat="hba1c_done12",
                   class=c("age_c","sex","type_dm"),
                   lev_var_cat=names(levhba1c_done),
                   lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
                   lab_var_cat="HbA1c done",lab_class=c("Age Classes","Gender","Type of Diabetes"),
                   tab=0,
                   bar=1,
                   barbeside=TRUE,
                   barnumber=TRUE,
                   barperc=TRUE,
                   box=0,
                   lines=0,
                   trellis=0,
                   pie=0,
                   texfile=texfile,
                   dirtables=dirtables,
                   dircsv=dirdataout,
                   dirgraph=dirgraph,
                   dirgraphlatex=dirgraphlatex,
                   cex=cex,
                   stand=1,
                   catcol=c(2,3,4),
                   outcome="hba1c_done12",
                   cov=c("age_c","sex","type_dm"),
                   factcov=c(TRUE,TRUE,TRUE),
                   strata=NULL,
                   levstd=list(a=levhba1c_done,b=classlabellist(ageth),c=levsex,d=levtype_dm),
                   sucvalue=1,
                   per=100,
                   caption="Percentage of patients with one or more HbA1c tests during the last 12 months")

  rm(data5_2_1)
  rm(noNAepirefyear)
   }
 }
 } else BIRO_report_toc(title="5.2.1 Percentage of patients with one or more HbA1c tests during the last 12 months"
```

```
)
  313:
  314:   #5.2.2.
  315:   #5.2.2 Percentage of patients with at least one test for microalbuminuria during the last 12 months
  316:
  317: if (is.null(episode$ma_test)==F) {
  318: if (length(na.omit(episode$ma_test))>0) {
  319:
  320:    for (thisyear in anayears) {
  321:
  322:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  323:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  324:
  325:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  326:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  327:
  328:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  329:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  330:
  331:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  332:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  333:
  334:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  335:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  336:
  337:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  338:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  339:
  340:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  341:
  342:     noNAepirefyear<-episode[is.na(episode$ma_test)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
  343:
  344:   data5_2_2<-noNAepirefyear$ma_test
  345:   data5_2_2<-aggregate(data5_2_2,list(noNAepirefyear$patient_id),FUN="min")
  346:   data5_2_2[is.na(data5_2_2$x)==TRUE]<-2
  347:
  348:   names(data5_2_2)<-c("patient_id","ma_done12")
  349:
  350:
  351:   data5_2_2<-merge(data5_2_2,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  352:                    by="patient_id",all=TRUE)
  353:
  354:   data5_2_2$age=(checkdate-data5_2_2$dob)/365.25
```

```
  355:   data5_2_2$age_c=varclass(data5_2_2$age,ageth)
  356:   data5_2_2$ma_done12[is.na(data5_2_2$ma_done12)]<-2
  357:
  358:   BIRO_dframe(dataname="data5_2_2",monitor=TRUE,filelog=logfile)
  359:   BIRO_dframe(dataname="data5_2_2",monitor=TRUE,filelog="")
  360:
  361:    BIRO_report(ind="5_2_2",
  362:                    title="5.2.2 Percentage of patients with at least one test for microalbuminuria during the last
12 months",
  363:                    var_cat="ma_done12",
  364:                    class=c("age_c","sex","type_dm"),
  365:                    lev_var_cat=names(levma_done),
  366:                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
  367:                    lab_var_cat="Microalbumin done",
  368:                    lab_class=c("Age Classes","Gender","Type of Diabetes"),
  369:                    tab=0,
  370:                    bar=1,
  371:                    barbeside=TRUE,
  372:                    barnumber=TRUE,
  373:                    barperc=TRUE,
  374:                    box=0,
  375:                    lines=0,
  376:                    trellis=0,
  377:                    pie=0,
  378:                    texfile=texfile,
  379:                    dirtables=dirtables,
  380:                    dircsv=dirdataout,
  381:                    dirgraph=dirgraph,
  382:                    dirgraphlatex=dirgraphlatex,
  383:                    cex=cex,
  384:                    stand=1,
  385:                    catcol=c(2,3,4),
  386:                    outcome="ma_done12",
  387:                    cov=c("age_c","sex","type_dm"),
  388:                    factcov=c(TRUE,TRUE,TRUE),
  389:                    strata=NULL,
  390:                    levstd=list(a=levma_done,b=classlabellist(ageth),c=levsex,d=levtype_dm),
  391:                    sucvalue=1,
  392:                    per=100,
  393:                    caption="Percentage of patients with at least one test for microalbuminuria during the last 12
months")
  394:
  395:   rm(data5_2_2)
  396:    }
  397: }
```

```
  398: } else BIRO_report_toc(title="5.2.2 Percentage of patients with at least one test for microalbuminuria during the last 12 months")
  399:
  400:  #5.2.3 Percentage of diabetes patients who received a dilated eye examination or evaluation of retinal photography by a trained caregiver within the last 12 months
  401:
  402:  if (is.null(episode$eye_exam)==F) {
  403:  if (length(na.omit(episode$eye_exam))>0) {
  404:
  405:    for (thisyear in anayears) {
  406:
  407:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  408:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  409:
  410:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  411:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  412:
  413:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  414:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  415:
  416:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  417:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  418:
  419:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  420:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  421:
  422:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  423:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  424:
  425:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  426:
  427:     noNAepisode<-episode[is.na(episode$eye_exam)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <= checkdate,]
  428:
  429:   data5_2_3<-noNAepisode$eye_exam
  430:   data5_2_3<-aggregate(data5_2_3,list(noNAepisode$patient_id),FUN="min")
  431:   data5_2_3[is.na(data5_2_3$x)==TRUE]=2
  432:   names(data5_2_3)<-c("patient_id","eye_done12")
  433:
  434:   data5_2_3<-merge(data5_2_3,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is.na(patient$end_date),],
  435:                    by="patient_id",all=TRUE)
  436:
  437:   data5_2_3$age=(checkdate-data5_2_3$dob)/365.25
  438:   data5_2_3$age_c=varclass(data5_2_3$age,ageth)
```

```
  439:   data5_2_3$eye_done12[is.na(data5_2_3$eye_done12)]<-2
  440:
  441:   data5_2_3<-data5_2_3[,c("eye_done12","age_c","sex","type_dm")]
  442:
  443:   BIRO_dframe(dataname="data5_2_3",monitor=TRUE,filelog=logfile)
  444:   BIRO_dframe(dataname="data5_2_3",monitor=TRUE,filelog="")
  445:
  446:        BIRO_report(ind="5_2_3",
  447:                    title="5.2.3 Percentage of diabetes patients who received a dilated eye examination or
evaluation of retinal photography by a trained caregiver within the last 12 months",
  448:                    var_cat="eye_done12",
  449:                    class=c("age_c","sex","type_dm"),
  450:                    lev_var_cat=names(leveye_done),
  451:                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
  452:                    lab_var_cat="Eye examination",
  453:                    lab_class=c("Age Classes","Gender","Type of Diabetes"),
  454:                    tab=0,
  455:                    bar=1,
  456:                    barbeside=TRUE,
  457:                    barnumber=TRUE,
  458:                    barperc=TRUE,
  459:                    box=0,
  460:                    lines=0,
  461:                    trellis=0,
  462:                    pie=0,
  463:                    texfile=texfile,
  464:                    dirtables=dirtables,
  465:                    dircsv=dirdataout,
  466:                    dirgraph=dirgraph,
  467:                    dirgraphlatex=dirgraphlatex,
  468:                    cex=cex,
  469:                    stand=1,
  470:                    catcol=c(2,3,4),
  471:                    outcome="eye_done12",
  472:                    cov=c("age_c","sex","type_dm"),
  473:                    factcov=c(TRUE,TRUE,TRUE),
  474:                    strata=NULL,
  475:                    levstd=list(a=leveye_done,b=classlabellist(ageth),c=levsex,d=levtype_dm),
  476:                    sucvalue=1,
  477:                    per=100,
  478:                    caption="Percentage of diabetes patients who received a dilated eye examination or evaluation
of retinal photography by a trained caregiver within the last 12 months")
  479:
  480:   rm(data5_2_3)
  481:
```

```
  482:
  483:     }
  484:   }
  485:   } else 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 within the last 12 months")
  486:
  487:    #5.2.4 Percentage of diabetes patients receiving at least one foot examination within the last 12 months
  488:
  489:    if (is.null(episode$foot_exam)==F) {
  490:   if (length(na.omit(episode$foot_exam))>0) {
  491:
  492:     for (thisyear in anayears) {
  493:
  494:      checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  495:      oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  496:
  497:     assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  498:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  499:
  500:     assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  501:           "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  502:
  503:     assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  504:           "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  505:
  506:     assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  507:            "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  508:
  509:     assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  510:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  511:
  512:     assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  513:
  514:     noNAepisode<-episode[is.na(episode$foot_exam)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
  515:
  516:    data5_2_4<-noNAepisode$foot_exam
  517:    data5_2_4<-aggregate(data5_2_4,list(noNAepisode$patient_id),FUN="min")
  518:    data5_2_4[is.na(data5_2_4$x)==TRUE]=2
  519:    names(data5_2_4)<-c("patient_id","foot_done12")
  520:
  521:    data5_2_4<-merge(data5_2_4,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  522:                     by="patient_id",all=TRUE)
  523:
```

```
  524:   data5_2_4$age=(checkdate-data5_2_4$dob)/365.25
  525:   data5_2_4$age_c=varclass(data5_2_4$age,ageth)
  526:   data5_2_4$foot_done12[is.na(data5_2_4$foot_done12)]<-2
  527:
  528:   data5_2_4<-data5_2_4[,c("foot_done12","age_c","sex","type_dm")]
  529:
  530:   BIRO_dframe(dataname="data5_2_4",monitor=TRUE,filelog=logfile)
  531:   BIRO_dframe(dataname="data5_2_4",monitor=TRUE,filelog="")
  532:
  533:        BIRO_report(ind="5_2_4",
  534:                    title="5.2.4 Percentage of diabetes patients receiving at least one foot examination within the
last 12 months",
  535:                    var_cat="foot_done12",
  536:                    class=c("age_c","sex","type_dm"),
  537:                    lev_var_cat=names(levfoot_done),
  538:                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
  539:                    lab_var_cat="Foot examination",
  540:                    lab_class=c("Age Classes","Gender","Type of Diabetes"),
  541:                    tab=0,
  542:                    bar=1,
  543:                    barbeside=TRUE,
  544:                    barnumber=TRUE,
  545:                    barperc=TRUE,
  546:                    box=0,
  547:                    lines=0,
  548:                    trellis=0,
  549:                    pie=0,
  550:                    texfile=texfile,
  551:                    dirtables=dirtables,
  552:                    dircsv=dirdataout,
  553:                    dirgraph=dirgraph,
  554:                    dirgraphlatex=dirgraphlatex,
  555:                    cex=cex,
  556:                    stand=1,
  557:                    catcol=c(2,3,4),
  558:                    outcome="foot_done12",
  559:                    cov=c("age_c","sex","type_dm"),
  560:                    factcov=c(TRUE,TRUE,TRUE),
  561:                    strata=NULL,
  562:                    levstd=list(a=levfoot_done,b=classlabellist(ageth),c=levsex,d=levtype_dm),
  563:                    sucvalue=1,
  564:                    per=100,
  565:                    caption="Percentage of diabetes patients receiving at least one foot examination within the
last 12 months")
  566:
```

```
  567:    rm(data5_2_4)
  568:     }
  569:   }
  570:   } else BIRO_report_toc(title="5.2.4 Percentage of diabetes patients receiving at least one foot examination
within the last 12 months")
  571:
  572:    #5.2.5 Percentage of diabetes patients whose smoking status was ascertained and documented within the last 12
months
  573:
  574:   if (is.null(episode$smok_stat)==F) {
  575:   if (length(na.omit(episode$smok_stat))>0) {
  576:
  577:     for (thisyear in anayears) {
  578:
  579:      checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  580:      oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  581:
  582:     assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  583:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  584:
  585:     assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  586:           "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  587:
  588:     assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  589:           "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  590:
  591:     assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  592:            "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  593:
  594:     assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  595:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  596:
  597:     assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  598:
  599:      noNAepisode<-episode[is.na(episode$smok_stat)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
  600:
  601:    data5_2_5<-noNAepisode[,c("patient_id","smok_stat")]
  602:    data5_2_5$smoke_doc12<-1
  603:    data5_2_5<-data5_2_5[,-pmatch("smok_stat",names(data5_2_5))]
  604:    data5_2_5<-unique(data5_2_5)
  605:
  606:    data5_2_5<-merge(data5_2_5,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  607:                     by="patient_id",all=TRUE)
```

```
608:
609:   data5_2_5$age=(checkdate-data5_2_5$dob)/365.25
610:   data5_2_5$age_c=varclass(data5_2_5$age,ageth)
611:   data5_2_5$smoke_doc12[is.na(data5_2_5$smoke_doc12)]<-2
612:
613:   data5_2_5<-data5_2_5[,c("smoke_doc12","age_c","sex","type_dm")]
614:
615:   BIRO_dframe(dataname="data5_2_5",monitor=TRUE,filelog=logfile)
616:   BIRO_dframe(dataname="data5_2_5",monitor=TRUE,filelog="")
617:
618:        BIRO_report(ind="5_2_5",
619:                    title="5.2.5 Percentage of diabetes patients whose smoking status was ascertained and
documented within the last 12 months",
620:                    var_cat="smoke_doc12",
621:                    class=c("age_c","sex","type_dm"),
622:                    lev_var_cat=names(levsmoke_doc),
623:                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
624:                    lab_var_cat="Smoke Status",
625:                    lab_class=c("Age Classes","Gender","Type of Diabetes"),
626:                    tab=0,
627:                    bar=1,
628:                    barbeside=TRUE,
629:                    barnumber=TRUE,
630:                    barperc=TRUE,
631:                    box=0,
632:                    lines=0,
633:                    trellis=0,
634:                    pie=0,
635:                    texfile=texfile,
636:                    dirtables=dirtables,
637:                    dircsv=dirdataout,
638:                    dirgraph=dirgraph,
639:                    dirgraphlatex=dirgraphlatex,
640:                    cex=cex,
641:                    stand=1,
642:                    catcol=c(2,3,4),
643:                    outcome="smoke_doc12",
644:                    cov=c("age_c","sex","type_dm"),
645:                    factcov=c(TRUE,TRUE,TRUE),
646:                    strata=NULL,
647:                    levstd=list(a=levsmoke_doc,b=classlabellist(ageth),c=levsex,d=levtype_dm),
648:                    sucvalue=1,
649:                    per=100,
650:                    caption="Percentage of diabetes patients whose smoking status was ascertained and documented
within the last 12 months")
```

```
  651:
  652:   rm(data5_2_5)
  653:    }
  654:  }
  655:  } else BIRO_report_toc(title="5.2.5 Percentage of diabetes patients whose smoking status was ascertained and
documented within the last 12 months")
  656:
  657:   #5.2.6 Percent with serum creatinine tested in last 12 months
  658:
  659:
  660:
  661:  if (is.null(episode$creat)==F) {
  662:  if (na.omit(length(episode$creat))>0) {
  663:
  664:    for (thisyear in anayears) {
  665:
  666:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  667:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  668:
  669:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  670:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  671:
  672:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  673:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  674:
  675:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  676:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  677:
  678:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  679:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  680:
  681:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  682:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  683:
  684:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  685:
  686:     noNAepisode<-episode[is.na(episode$creat)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
  687:
  688:   data5_2_6<-noNAepisode$creat
  689:   data5_2_6<-aggregate(data5_2_6,list(noNAepisode$patient_id),FUN="min")
  690:   data5_2_6$x<-ifelse(is.na(data5_2_6$x)==FALSE,1,0)
  691:   names(data5_2_6)<-c("patient_id","creat_done12")
  692:   data5_2_6<-merge(data5_2_6,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
```

```
693:                    by="patient_id",all=TRUE)
694:
695:   data5_2_6$age=(checkdate-data5_2_6$dob)/365.25
696:   data5_2_6$age_c=varclass(data5_2_6$age,ageth)
697:   data5_2_6$creat_done12[is.na(data5_2_6$creat_done12)]<-2
698:   data5_2_6<-data5_2_6[,c("creat_done12","age_c","sex","type_dm")]
699:
700:   BIRO_dframe(dataname="data5_2_6",monitor=TRUE,filelog=logfile)
701:   BIRO_dframe(dataname="data5_2_6",monitor=TRUE,filelog="")
702:
703:        BIRO_report(ind="5_2_6",
704:                    title="5.2.6 Percent with serum creatinine tested in last 12 months",
705:                    var_cat="creat_done12",
706:                    class=c("age_c","sex","type_dm"),
707:                    lev_var_cat=names(levcreat_done),
708:                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
709:                    lab_var_cat="Smoke Status",
710:                    lab_class=c("Age Classes","Gender","Type of Diabetes"),
711:                    tab=0,
712:                    bar=1,
713:                    barbeside=TRUE,
714:                    barnumber=TRUE,
715:                    barperc=TRUE,
716:                    box=0,
717:                    lines=0,
718:                    trellis=0,
719:                    pie=0,
720:                    texfile=texfile,
721:                    dirtables=dirtables,
722:                    dircsv=dirdataout,
723:                    dirgraph=dirgraph,
724:                    dirgraphlatex=dirgraphlatex,
725:                    cex=cex,
726:                    stand=1,
727:                    catcol=c(2,3,4),
728:                    outcome="creat_done12",
729:                    cov=c("age_c","sex","type_dm"),
730:                    factcov=c(TRUE,TRUE,TRUE),
731:                    strata=NULL,
732:                    levstd=list(a=levcreat_done,b=classlabellist(ageth),c=levsex,d=levtype_dm),
733:                    sucvalue=1,
734:                    per=100,
735:                    caption="Percent with serum creatinine tested in last 12 months")
736:
737:   rm(data5_2_6)
```

```
  738:     }
  739:   }
  740:   } else BIRO_report_toc(title="5.2.6 Percent with serum creatinine tested in last 12 months")
  741:
  742:    #5.2.7 Percentage of patients with diabetes and one or more blood pressure measurements within the last 12 months
  743:    #
  744:
  745:
  746:
  747:   if (is.null(episode$bp)==F) {
  748:   if (na.omit(length(episode$bp))>0) {
  749:
  750:     for (thisyear in anayears) {
  751:
  752:      checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  753:      oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  754:
  755:     assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  756:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  757:
  758:     assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  759:           "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  760:
  761:     assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  762:           "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  763:
  764:     assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  765:            "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  766:
  767:     assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  768:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  769:
  770:     assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  771:
  772:      noNAepisode<-episode[is.na(episode$bp)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <= checkdate,]
  773:
  774:    data5_2_7<-noNAepirefyear$bp
  775:    data5_2_7<-aggregate(data5_2_7,list(noNAepirefyear$patient_id),FUN="min")
  776:    data5_2_7$x<-ifelse(is.na(data5_2_7$x)==FALSE,1,2)
  777:    names(data5_2_7)<-c("patient_id","bp_done12")
  778:    data5_2_7<-merge(data5_2_7,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  779:                     by="patient_id",all=TRUE)
  780:
  781:    data5_2_7$age=(checkdate-data5_2_7$dob)/365.25
```

```
  782:   data5_2_7$age_c=varclass(data5_2_7$age,ageth)
  783:   data5_2_7$bp_done12[is.na(data5_2_7$bp_done12)]<-2
  784:   data5_2_7<-data5_2_7[,c("bp_done12","age_c","sex","type_dm")]
  785:
  786:   BIRO_dframe(dataname="data5_2_7",monitor=TRUE,filelog=logfile)
  787:   BIRO_dframe(dataname="data5_2_7",monitor=TRUE,filelog="")
  788:
  789:        BIRO_report(ind="5_2_7",
  790:                    title="5.2.7 Percentage of patients with diabetes and one or more blood pressure measurements
within the last 12 months",
  791:                    var_cat="bp_done12",
  792:                    class=c("age_c","sex","type_dm"),
  793:                    lev_var_cat=names(levbp12),
  794:                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
  795:                    lab_var_cat="BP measurements",
  796:                    lab_class=c("Age Classes","Gender","Type of Diabetes"),
  797:                    tab=0,
  798:                    bar=1,
  799:                    barbeside=TRUE,
  800:                    barnumber=TRUE,
  801:                    barperc=TRUE,
  802:                    box=0,
  803:                    lines=0,
  804:                    trellis=0,
  805:                    pie=0,
  806:                    texfile=texfile,
  807:                    dirtables=dirtables,
  808:                    dircsv=dirdataout,
  809:                    dirgraph=dirgraph,
  810:                    dirgraphlatex=dirgraphlatex,
  811:                    cex=cex,
  812:                    stand=1,
  813:                    catcol=c(2,3,4),
  814:                    outcome="bp_done12",
  815:                    cov=c("age_c","sex","type_dm"),
  816:                    factcov=c(TRUE,TRUE,TRUE),
  817:                    strata=NULL,
  818:                    levstd=list(a=levbp_done,b=classlabellist(ageth),c=levsex,d=levtype_dm),
  819:                    sucvalue=1,
  820:                    per=100,
  821:                    caption="Percentage of patients with diabetes and one or more blood pressure measurements
within the last 12 months")
  822:
  823:    rm(data5_2_7)
  824:    }
```

```
  825:  }
  826:  } else BIRO_report_toc(title="5.2.7 Percentage of patients with diabetes and one or more blood pressure
measurements within the last 12 months")
  827:
  828:   #5.2.8 Percentage of patients with hypertension who receive antihypertensive medication
  829:   #
  830:
  831:  if ((is.null(episode$hypert_med)==FALSE) & (is.null(episode$hypertension)==FALSE)) {
  832:  if ((length(na.omit(episode$hypert_med))>0) & (length(na.omit(episode$hypertension))>0)){
  833:
  834:   episode$hypert_med<-as.numeric(as.character(episode$hypert_med))
  835:
  836:    for (thisyear in anayears) {
  837:
  838:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
  839:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
  840:
  841:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
  842:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  843:
  844:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
  845:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
  846:
  847:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
  848:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
  849:
  850:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
  851:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
  852:
  853:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
  854:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
  855:
  856:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
  857:
  858:     noNAepirefyear<-episode[is.na(episode$hypert_med)==FALSE & is.na(episode$hypertension)==FALSE &
episode$epi_date >= oneyearbf & episode$epi_date <= checkdate,]
  859:
  860:   data5_2_8<-noNAepirefyear$hypert_med[noNAepirefyear$hypertension==1]
  861:   data5_2_8<-aggregate(data5_2_8,list(noNAepirefyear$patient_id[noNAepirefyear$hypertension==1]),FUN="min")
  862:   data5_2_8$x<-ifelse(data5_2_8$x==1,1,2)
  863:   data5_2_8$x<-ifelse(is.na(data5_2_8$x)==FALSE,1,2)
  864:   names(data5_2_8)<-c("patient_id","hypert_done12")
  865:   data5_2_8<-merge(data5_2_8,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
  866:                    by="patient_id",all=TRUE)
```

```
  867:
  868:   data5_2_8$age=(checkdate-data5_2_8$dob)/365.25
  869:   data5_2_8$age_c=varclass(data5_2_8$age,ageth)
  870:   data5_2_8$hypert_done12[is.na(data5_2_8$hypert_done12)]<-2
  871:   data5_2_8<-data5_2_8[,c("hypert_done12","age_c","sex","type_dm")]
  872:
  873:   BIRO_dframe(dataname="data5_2_8",monitor=TRUE,filelog=logfile)
  874:   BIRO_dframe(dataname="data5_2_8",monitor=TRUE,filelog="")
  875:
  876:        BIRO_report(ind="5_2_8",
  877:                    title="5.2.8 Percentage of patients with hypertension who receive antihypertensive medication",
  878:                    var_cat="hypert_done12",class=c("age_c","sex","type_dm"),
  879:                    lev_var_cat=names(levhypert),lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
  880:                    lab_var_cat="Hypertension",lab_class=c("Age Classes","Gender","Type of Diabetes"),
  881:                    tab=0,
  882:                    bar=1,
  883:                    barbeside=TRUE,
  884:                    barnumber=TRUE,
  885:                    barperc=TRUE,
  886:                    box=0,
  887:                    lines=0,
  888:                    trellis=0,
  889:                    pie=0,
  890:                    texfile=texfile,
  891:                    dirtables=dirtables,
  892:                    dircsv=dirdataout,
  893:                    dirgraph=dirgraph,
  894:                    dirgraphlatex=dirgraphlatex,
  895:                    cex=cex,
  896:                    stand=1,
  897:                    catcol=c(2,3,4),
  898:                    outcome="hypert_done12",
  899:                    cov=c("age_c","sex","type_dm"),
  900:                    factcov=c(TRUE,TRUE,TRUE),
  901:                    strata=NULL,
  902:                    levstd=list(a=levhypert,b=classlabellist(ageth),c=levsex,d=levtype_dm),
  903:                    sucvalue=1,
  904:                    per=100,
  905:                    caption="Percentage of patients with hypertension who receive antihypertensive medication")
  906:
  907:    rm(data5_2_8)
  908:     }
  909:   }
  910:   } else BIRO_report_toc(title="5.2.8 Percentage of patients with hypertension who receive antihypertensive
medication")
```

```
911:
912:   #5.2.9 Type of oral therapy (distribution of agents) in patients with diabetes type 2
913:
914:   if (is.null(episode$oral_therapy)==F) {
915:   if (length(na.omit(episode$oral_therapy))>0) {
916:
917:    for (thisyear in anayears) {
918:
919:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
920:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
921:
922:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
923:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
924:
925:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
926:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
927:
928:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
929:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
930:
931:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
932:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
933:
934:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
935:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
936:
937:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
938:
939:     noNAepirefyear<-episode[is.na(episode$oral_therapy)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date
<= checkdate,]
940:
941:
942:   data5_2_9<-noNAepirefyear[noNAepirefyear$oral_therapy %in% (1:5),]
943:   data5_2_9<-data5_2_9[,c("patient_id","oral_therapy")]
944:   names(data5_2_9)<-c("patient_id","oral_th12")
945:   data5_2_9$oral_th12<-1
946:
947:   data5_2_9<-merge(data5_2_9,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
948:                    by="patient_id",all=TRUE)
949:
950:   data5_2_9$age=(checkdate-data5_2_9$dob)/365.25
951:   data5_2_9$age_c=varclass(data5_2_9$age,ageth)
952:   data5_2_9$oral_th12[is.na(data5_2_9$oral_th12)]<-2
953:   data5_2_9<-data5_2_9[,c("oral_th12","age_c","sex")]
```

```
954:
955:
956:   BIRO_dframe(dataname="data5_2_9",monitor=TRUE,filelog=logfile)
957:   BIRO_dframe(dataname="data5_2_9",monitor=TRUE,filelog="")
958:
959:        BIRO_report(ind="5_2_9",
960:                    title="5.2.9 Type of oral therapy (distribution of agents) in patients with diabetes type 2",
961:                    var_cat="oral_th12",
962:                    class=c("age_c","sex"),
963:                    lev_var_cat=names(levoral_th12),
964:                    lev_class=list(a=classlabellist(ageth),b=levsex),
965:                    lab_var_cat="Oral Therapy",
966:                    lab_class=c("Age Classes","Gender"),
967:                    tab=0,
968:                    bar=1,
969:                    barbeside=TRUE,
970:                    barnumber=TRUE,
971:                    barperc=TRUE,
972:                    box=0,
973:                    lines=0,
974:                    trellis=0,
975:                    pie=0,
976:                    texfile=texfile,
977:                    dirtables=dirtables,
978:                    dircsv=dirdataout,
979:                    dirgraph=dirgraph,
980:                    dirgraphlatex=dirgraphlatex,
981:                    cex=cex,
982:                    stand=1,
983:                    catcol=c(2,3,4),
984:                    outcome="oral_th12",
985:                    cov=c("age_c","sex"),
986:                    factcov=c(TRUE,TRUE,TRUE),
987:                    strata=NULL,
988:                    levstd=list(a=levhypert,b=classlabellist(ageth),c=levsex),
989:                    sucvalue=1,
990:                    per=100,
991:                    caption="Type of oral therapy (distribution of agents) in patients with diabetes type 2")
992:
993:    rm(data5_2_9)
994:     }
995:   }
996:   } else BIRO_report_toc(title="5.2.9 Type of oral therapy (distribution of agents) in patients with diabetes type
2")
997:
```

```
  998:   #5.2.10 Portion of patients treated with insulin among patients with diabetes
  999:
 1000:   if (is.null(episode$drug_therapy)==F) {
 1001:   if (length(na.omit(episode$drug_therapy))>0) {
 1002:
 1003:    for (thisyear in anayears) {
 1004:
 1005:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 1006:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 1007:
 1008:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
 1009:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1010:
 1011:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
 1012:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 1013:
 1014:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 1015:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 1016:
 1017:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1018:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1019:
 1020:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1021:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1022:
 1023:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1024:
 1025:     noNAepirefyear<-episode[is.na(episode$drug_therapy)==FALSE  & episode$epi_date >= oneyearbf & episode$epi_date
<= checkdate,]
 1026:
 1027:   data5_2_10<-noNAepirefyear$drug_therapy
 1028:   data5_2_10<-ifelse(data5_2_10 %in% c(1,3),1,0)
 1029:   data5_2_10<-aggregate(data5_2_10,list(noNAepirefyear$patient_id),FUN=sum)
 1030:   data5_2_10$x<-ifelse(data5_2_10$x>1,1,0)
 1031:   data5_2_10$x<-ifelse(is.na(data5_2_10$x)==TRUE,0,1)
 1032:   names(data5_2_10)<-c("patient_id","insulin12")
 1033:   data5_2_10<-merge(data5_2_10,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1034:                     by="patient_id",all=TRUE)
 1035:
 1036:   data5_2_10$age=(checkdate-data5_2_10$dob)/365.25
 1037:   data5_2_10$age_c=varclass(data5_2_10$age,ageth)
 1038:   data5_2_10$insulin12[is.na(data5_2_10$insulin12)]<-2
 1039:   data5_2_10[,c("insulin12","age_c","sex","type_dm")]
 1040:   data5_2_10<-data5_2_10[data5_2_10$type_dm %in% c(1,2),]
```

```
1041:
1042:   BIRO_dframe(dataname="data5_2_10",monitor=TRUE,filelog=logfile)
1043:   BIRO_dframe(dataname="data5_2_10",monitor=TRUE,filelog="")
1044:
1045:        BIRO_report(ind="5_2_10",
1046:                    title="5.2.10 Portion of patients treated with insulin among patients with diabetes",
1047:                    var_cat="insulin12",
1048:                    class=c("age_c","sex","type_dm"),
1049:                    lev_var_cat=names(levinsulin12),
1050:                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
1051:                    lab_var_cat="Insulin",
1052:                    lab_class=c("Age Classes","Gender","Type of Diabetes"),
1053:                    tab=0,
1054:                    bar=1,
1055:                    barbeside=TRUE,
1056:                    barnumber=TRUE,
1057:                    barperc=TRUE,
1058:                    box=0,
1059:                    lines=0,
1060:                    trellis=0,
1061:                    pie=0,
1062:                    texfile=texfile,
1063:                    dirtables=dirtables,
1064:                    dircsv=dirdataout,
1065:                    dirgraph=dirgraph,
1066:                    dirgraphlatex=dirgraphlatex,
1067:                    cex=cex,
1068:                    stand=1,
1069:                    catcol=c(2,3,4),
1070:                    outcome="insulin12",
1071:                    cov=c("age_c","sex","type_dm"),
1072:                    factcov=c(TRUE,TRUE,TRUE),
1073:                    strata=NULL,
1074:                    levstd=list(a=levinsulin12,b=classlabellist(ageth),c=levsex,d=levtype_dm),
1075:                    sucvalue=1,
1076:                    per=100,
1077:                    caption="Portion of patients treated with insulin among patients with diabetes")
1078:
1079:    rm(data5_2_10)
1080:     }
1081:  }
1082:  } else BIRO_report_toc(title="5.2.10 Portion of patients treated with insulin among patients with diabetes")
1083:
1084:   #5.2.11 Portion of patients treated with insulin in combination with OADs among patients with diabetes
1085:
```

```
 1086:    if ((is.null(episode$drug_therapy)==F) | (is.null(episode$oral_therapy)==F)) {
 1087:    if ((length(na.omit(episode$drug_therapy))>0) & (length(na.omit(episode$oral_therapy))>0)) {
 1088:
 1089:     for (thisyear in anayears) {
 1090:
 1091:      checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 1092:      oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 1093:
 1094:     assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
 1095:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1096:
 1097:     assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
 1098:           "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 1099:
 1100:     assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 1101:           "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 1102:
 1103:     assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1104:            "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1105:
 1106:     assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1107:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1108:
 1109:     assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1110:
 1111:      noNAepirefyear<-episode[is.na(episode$oral_therapy)==FALSE  & episode$epi_date >= oneyearbf & episode$epi_date
<= checkdate,]
 1112:
 1113:       noNAepirefyear$insOAD<-ifelse((noNAepirefyear$drug_therapy==1 | noNAepirefyear$drug_therapy==3) &
noNAepirefyear$oral_therapy==1,1,2)
 1114:
 1115:    noNAepirefyear$insOAD[is.na(noNAepirefyear$insOAD)]<-2
 1116:    data5_2_11<-aggregate(noNAepirefyear$insOAD,list(noNAepirefyear$patient_id),FUN="min")
 1117:    names(data5_2_11)<-c("patient_id","insOAD")
 1118:    data5_2_11<-merge(data5_2_11,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1119:                      by="patient_id",all=TRUE)
 1120:
 1121:    data5_2_11$age=(checkdate-data5_2_11$dob)/365.25
 1122:    data5_2_11$age_c=varclass(data5_2_11$age,ageth)
 1123:    data5_2_11$insulin12[is.na(data5_2_11$insulin12)]<-2
 1124:    data5_2_11[,c("insOAD","age_c","sex","type_dm")]
 1125:    data5_2_11<-data5_2_11[data5_2_11$type_dm %in% c(1,2),]
 1126:
 1127:    BIRO_dframe(dataname="data5_2_11",monitor=TRUE,filelog=logfile)
```

```
 1128:   BIRO_dframe(dataname="data5_2_11",monitor=TRUE,filelog="")
 1129:
 1130:        BIRO_report(ind="5_2_11",
 1131:                    title="5.2.11 Portion of patients treated with insulin in combination with OADs among patients
with diabetes",
 1132:                    var_cat="insOAD",
 1133:                    class=c("age_c","sex","type_dm"),
 1134:                    lev_var_cat=names(levinsOAD),
 1135:                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
 1136:                    lab_var_cat="Insulin and OAD",
 1137:                    lab_class=c("Age Classes","Gender","Type of Diabetes"),
 1138:                    tab=0,
 1139:                    bar=1,
 1140:                    barbeside=TRUE,
 1141:                    barnumber=TRUE,
 1142:                    barperc=TRUE,
 1143:                    box=0,
 1144:                    lines=0,
 1145:                    trellis=0,
 1146:                    pie=0,
 1147:                    texfile=texfile,
 1148:                    dirtables=dirtables,
 1149:                    dircsv=dirdataout,
 1150:                    dirgraph=dirgraph,
 1151:                    dirgraphlatex=dirgraphlatex,
 1152:                    cex=cex,
 1153:                    stand=1,
 1154:                    catcol=c(2,3,4),
 1155:                    outcome="insOAD",
 1156:                    cov=c("age_c","sex","type_dm"),
 1157:                    factcov=c(TRUE,TRUE,TRUE),
 1158:                    strata=NULL,
 1159:                    levstd=list(a=levinsOAD,b=classlabellist(ageth),c=levsex,d=levtype_dm),
 1160:                    sucvalue=1,
 1161:                    per=100,
 1162:                    caption="Portion of patients treated with insulin in combination with OADs among patients with
diabetes")
 1163:
 1164:   rm(data5_2_11)
 1165:
 1166:    }
 1167: }
 1168: } else BIRO_report_toc(title="5.2.11 Portion of patients treated with insulin in combination with OADs among
patients with diabetes")
 1169:
```

```
1170:   #5.2.12 Percentage of insulin treated patients with pump therapy
1171:
1172:  if ((is.null(episode$drug_therapy)==F) | (is.null(episode$oral_therapy)==F)) {
1173:  if ((length(na.omit(episode$drug_therapy))>0) & (length(na.omit(episode$pump_therapy))>0)) {
1174:
1175:
1176:      episode$inspump<-ifelse((episode$drug_therapy==1 | episode$drug_therapy==3) & episode$pump_therapy==1,1,2)
1177:
1178:    for (thisyear in anayears) {
1179:
1180:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1181:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1182:
1183:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1184:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1185:
1186:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1187:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1188:
1189:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1190:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1191:
1192:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1193:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
1194:
1195:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
1196:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1197:
1198:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
1199:
1200:     noNAepirefyear<-episode[is.na(episode$oral_therapy)==FALSE  & episode$epi_date >= oneyearbf & episode$epi_date
<= checkdate,]
1201:
1202:   noNAepirefyear$inspump[is.na(noNAepirefyear$inspump)]<-2
1203:   data5_2_12<-aggregate(noNAepirefyear$inspump,list(noNAepirefyear$patient_id),FUN="min")
1204:   names(data5_2_12)<-c("patient_id","inspump")
1205:   data5_2_12<-merge(data5_2_12,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
1206:                     by="patient_id",all=TRUE)
1207:
1208:   data5_2_12$age=(checkdate-data5_2_12$dob)/365.25
1209:   data5_2_12$age_c=varclass(data5_2_12$age,ageth)
1210:   data5_2_12$inspump[is.na(data5_2_12$inspump)]<-2
1211:   data5_2_12<-data5_2_12[,c("inspump","age_c","sex","type_dm")]
1212:   data5_2_12<-data5_2_12[data5_2_12$type_dm %in% c(1,2),]
```

```
1213:
1214:   BIRO_dframe(dataname="data5_2_12",monitor=TRUE,filelog=logfile)
1215:   BIRO_dframe(dataname="data5_2_12",monitor=TRUE,filelog="")
1216:
1217:       BIRO_report(ind="5_2_12",
1218:                   title="5.2.12 Percentage of insulin treated patients with pump therapy",
1219:                   var_cat="inspump",class=c("age_c","sex","type_dm"),
1220:                   lev_var_cat=names(levinsOAD),lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
1221:                   lab_var_cat="Insulin",lab_class=c("Age Classes","Gender","Type of Diabetes"),
1222:                   tab=0,
1223:                   bar=1,
1224:                   barbeside=TRUE,
1225:                   barnumber=TRUE,
1226:                   barperc=TRUE,
1227:                   box=0,
1228:                   lines=0,
1229:                   trellis=0,
1230:                   pie=0,
1231:                   texfile=texfile,
1232:                   dirtables=dirtables,
1233:                   dircsv=dirdataout,
1234:                   dirgraph=dirgraph,
1235:                   dirgraphlatex=dirgraphlatex,
1236:                   cex=cex,
1237:                   stand=1,
1238:                   catcol=c(2,3,4),
1239:                   outcome="inspump",
1240:                   cov=c("age_c","sex","type_dm"),
1241:                   factcov=c(TRUE,TRUE,TRUE),
1242:                   strata=NULL,
1243:                   levstd=list(a=levinspump,b=classlabellist(ageth),c=levsex,d=levtype_dm),
1244:                   sucvalue=1,
1245:                   per=100,
1246:                   caption="Percentage of insulin treated patients with pump therapy")
1247:
1248:   rm(data5_2_12)
1249:    }
1250: }
1251: } else BIRO_report_toc(title="5.2.12 Percentage of insulin treated patients with pump therapy")
1252:
1253:   #5.2.13 Portion of diabetes patients with anti hypertensive treatment
1254:
1255:  if ((is.null(episode$hypert_med)==FALSE) & (is.null(episode$hypertension)==FALSE)) {
1256:  if ((length(na.omit(episode$hypert_med))>0) & (length(na.omit(episode$hypertension))>0)){
1257:
```

```
 1258:    for (thisyear in anayears) {
 1259:
 1260:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 1261:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 1262:
 1263:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
 1264:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1265:
 1266:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
 1267:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 1268:
 1269:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 1270:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 1271:
 1272:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1273:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1274:
 1275:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1276:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1277:
 1278:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1279:
 1280:   noNAepirefyear<-episode[is.na(episode$hypert_med)==FALSE  & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
 1281:
 1282:   noNAepirefyear$hypert_med[is.na(noNAepirefyear$hypert_med)]<-2
 1283:   data5_2_13<-aggregate(noNAepirefyear$hypert_med,list(noNAepirefyear$patient_id),FUN="min")
 1284:   names(data5_2_13)<-c("patient_id","hypert12")
 1285:   data5_2_13<-merge(data5_2_13,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1286:                     by="patient_id",all=TRUE)
 1287:
 1288:   data5_2_13$age=(checkdate-data5_2_13$dob)/365.25
 1289:   data5_2_13$age_c=varclass(data5_2_13$age,ageth)
 1290:   data5_2_13$hypert12[is.na(data5_2_13$hypert12)]<-2
 1291:   data5_2_13<-data5_2_13[,c("hypert12","age_c","sex","type_dm")]
 1292:   data5_2_13<-data5_2_13[data5_2_13$type_dm %in% c(1,2),]
 1293:
 1294:   BIRO_dframe(dataname="data5_2_13",monitor=TRUE,filelog=logfile)
 1295:   BIRO_dframe(dataname="data5_2_13",monitor=TRUE,filelog="")
 1296:
 1297:        BIRO_report(ind="5_2_13",
 1298:                    title="5.2.13 Portion of diabetes patients with anti hypertensive treatment",
 1299:                    var_cat="hypert12",
 1300:                    class=c("age_c","sex","type_dm"),
```

```
1301:                     lev_var_cat=names(levhypert12),
1302:                     lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
1303:                     lab_var_cat="Hypertension",
1304:                     lab_class=c("Age Classes","Gender","Type of Diabetes"),
1305:                     tab=0,
1306:                     bar=1,
1307:                     barbeside=TRUE,
1308:                     barnumber=TRUE,
1309:                     barperc=TRUE,
1310:                     box=0,
1311:                     lines=0,
1312:                     trellis=0,
1313:                     pie=0,
1314:                     texfile=texfile,
1315:                     dirtables=dirtables,
1316:                     dircsv=dirdataout,
1317:                     dirgraph=dirgraph,
1318:                     dirgraphlatex=dirgraphlatex,
1319:                     cex=cex,
1320:                     stand=1,
1321:                     catcol=c(2,3,4),
1322:                     outcome="hypert12",
1323:                     cov=c("age_c","sex","type_dm"),
1324:                     factcov=c(TRUE,TRUE,TRUE),
1325:                     strata=NULL,
1326:                     levstd=list(a=levhypert,b=classlabellist(ageth),c=levsex,d=levtype_dm),
1327:                     sucvalue=1,
1328:                     per=100,
1329:                     caption="Portion of diabetes patients with anti hypertensive treatment")
1330:
1331:   rm(data5_2_13)
1332:    }
1333:  }
1334:  } else BIRO_report_toc(title="5.2.13 Portion of diabetes patients with anti hypertensive treatment")
1335:
1336:   # 5.2.14 Portion of diabetes patients with lipid lowering treatment
1337:   #No variable in XML Schema
1338:
1339: if (is.null(episode$lipid_med)==F) {
1340:   if (length(na.omit(episode$lipid_med))>0) {
1341:
1342:    for (thisyear in anayears) {
1343:
1344:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1345:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
```

```
 1346:
 1347:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
 1348:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1349:
 1350:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
 1351:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 1352:
 1353:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 1354:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 1355:
 1356:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1357:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1358:
 1359:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1360:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1361:
 1362:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1363:
 1364:   noNAepirefyear<-episode[is.na(episode$lipid_med)==FALSE  & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
 1365:
 1366:   noNAepirefyear$hypert_med[is.na(noNAepirefyear$lipid_med)]<-2
 1367:   data5_2_14<-aggregate(noNAepirefyear$lipid_med,list(noNAepirefyear$patient_id),FUN="min")
 1368:   names(data5_2_14)<-c("patient_id","lipid12")
 1369:   data5_2_14<-merge(data5_2_14,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1370:                     by="patient_id",all=TRUE)
 1371:
 1372:   data5_2_14$age=(checkdate-data5_2_14$dob)/365.25
 1373:   data5_2_14$age_c=varclass(data5_2_14$age,ageth)
 1374:   data5_2_14$lipid12[is.na(data5_2_14$lipid12)]<-2
 1375:   data5_2_14<-data5_2_14[,c("lipid12","age_c","sex","type_dm")]
 1376:   data5_2_14<-data5_2_14[data5_2_14$type_dm %in% c(1,2),]
 1377:
 1378:   BIRO_dframe(dataname="data5_2_14",monitor=TRUE,filelog=logfile)
 1379:   BIRO_dframe(dataname="data5_2_14",monitor=TRUE,filelog="")
 1380:
 1381:        BIRO_report(ind="5_2_14",
 1382:                    title="5.2.14 Portion of diabetes patients with lipid lowering treatment",
 1383:                    var_cat="lipid12",
 1384:                    class=c("age_c","sex","type_dm"),
 1385:                    lev_var_cat=names(levlipid_med),
 1386:                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
 1387:                    lab_var_cat="Lipid lowering treatment",
 1388:                    lab_class=c("Age Classes","Gender","Type of Diabetes"),
```

```
1389:                     tab=0,
1390:                     bar=1,
1391:                     barbeside=TRUE,
1392:                     barnumber=TRUE,
1393:                     barperc=TRUE,
1394:                     box=0,
1395:                     lines=0,
1396:                     trellis=0,
1397:                     pie=0,
1398:                     texfile=texfile,
1399:                     dirtables=dirtables,
1400:                     dircsv=dirdataout,
1401:                     dirgraph=dirgraph,
1402:                     dirgraphlatex=dirgraphlatex,
1403:                     cex=cex,
1404:                     stand=1,
1405:                     catcol=c(2,3,4),
1406:                     outcome="lipid12",
1407:                     cov=c("age_c","sex","type_dm"),
1408:                     factcov=c(TRUE,TRUE,TRUE),
1409:                     strata=NULL,
1410:                     levstd=list(a=levhypert,b=classlabellist(ageth),c=levsex,d=levtype_dm),
1411:                     sucvalue=1,
1412:                     per=100,
1413:                     caption="Portion of diabetes patients with anti hypertensive treatment")
1414:
1415:   rm(data5_2_14)
1416:    }
1417:  }
1418:  } else BIRO_report_toc(title="5.2.14 Portion of diabetes patients with lipid lowering treatment")
1419:
1420:   # 5.2.15 Portion of diabetes patients with ASA treatment
1421:   #No variable in XML Schema
1422:
1423: if (is.null(episode$asa_med)==F) {
1424:   if (length(na.omit(episode$asa_med))>0) {
1425:
1426:    for (thisyear in anayears) {
1427:
1428:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1429:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1430:
1431:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1432:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1433:
```

```
 1434:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
 1435:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 1436:
 1437:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 1438:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 1439:
 1440:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1441:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1442:
 1443:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1444:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1445:
 1446:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1447:
 1448:   noNAepirefyear<-episode[is.na(episode$asa_med)==FALSE  & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
 1449:
 1450:   noNAepirefyear$asa_med[is.na(noNAepirefyear$asa_med)]<-2
 1451:   data5_2_15<-aggregate(noNAepirefyear$asa_med,list(noNAepirefyear$patient_id),FUN="min")
 1452:   names(data5_2_15)<-c("patient_id","asa_med12")
 1453:   data5_2_15<-merge(data5_2_15,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1454:                     by="patient_id",all=TRUE)
 1455:
 1456:   data5_2_15$age=(checkdate-data5_2_15$dob)/365.25
 1457:   data5_2_15$age_c=varclass(data5_2_15$age,ageth)
 1458:   data5_2_15$lipid12[is.na(data5_2_15$asa_med12)]<-2
 1459:   data5_2_15<-data5_2_15[,c("asa_med12","age_c","sex","type_dm")]
 1460:   data5_2_15<-data5_2_15[data5_2_15$type_dm %in% c(1,2),]
 1461:
 1462:   BIRO_dframe(dataname="data5_2_15",monitor=TRUE,filelog=logfile)
 1463:   BIRO_dframe(dataname="data5_2_15",monitor=TRUE,filelog="")
 1464:
 1465:        BIRO_report(ind="5_2_15",
 1466:                    title="5.2.15 Portion of diabetes patients with ASA treatment",
 1467:                    var_cat="asa_med12",
 1468:                    class=c("age_c","sex","type_dm"),
 1469:                    lev_var_cat=names(levasa_med),
 1470:                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
 1471:                    lab_var_cat="ASA treatment",
 1472:                    lab_class=c("Age Classes","Gender","Type of Diabetes"),
 1473:                    tab=0,
 1474:                    bar=1,
 1475:                    barbeside=TRUE,
 1476:                    barnumber=TRUE,
```

```
1477:                     barperc=TRUE,
1478:                     box=0,
1479:                     lines=0,
1480:                     trellis=0,
1481:                     pie=0,
1482:                     texfile=texfile,
1483:                     dirtables=dirtables,
1484:                     dircsv=dirdataout,
1485:                     dirgraph=dirgraph,
1486:                     dirgraphlatex=dirgraphlatex,
1487:                     cex=cex,
1488:                     stand=1,
1489:                     catcol=c(2,3,4),
1490:                     outcome="asa_med12",
1491:                     cov=c("age_c","sex","type_dm"),
1492:                     factcov=c(TRUE,TRUE,TRUE),
1493:                     strata=NULL,
1494:                     levstd=list(a=levhypert,b=classlabellist(ageth),c=levsex,d=levtype_dm),
1495:                     sucvalue=1,
1496:                     per=100,
1497:                     caption="Portion of diabetes patients with anti hypertensive treatment")
1498:
1499:   rm(data5_2_15)
1500:    }
1501:  }
1502:  } else BIRO_report_toc(title="5.2.15 Portion of diabetes patients with ASA treatment")
1503:
1504:   # 5.2.16 Percent of patients with diabetes performing selfmonitoring
1505:   # of blood glucose/ urine testing** Svein questions the
1506:   # urine at least in some countries
1507:
1508: if (is.null(episode$self_mon)==F) {
1509:   if (length(na.omit(episode$self_mon))>0) {
1510:
1511:    for (thisyear in anayears) {
1512:
1513:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1514:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1515:
1516:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1517:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1518:
1519:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1520:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1521:
```

```
 1522:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 1523:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 1524:
 1525:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1526:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1527:
 1528:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1529:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1530:
 1531:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1532:
 1533:   noNAepirefyear<-episode[is.na(episode$self_mon)==FALSE  & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
 1534:
 1535:   noNAepirefyear$self_mon[is.na(noNAepirefyear$self_mon)]<-2
 1536:   data5_2_16<-aggregate(noNAepirefyear$self_mon,list(noNAepirefyear$patient_id),FUN="min")
 1537:   names(data5_2_16)<-c("patient_id","self_mon12")
 1538:   data5_2_16<-merge(data5_2_16,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1539:                     by="patient_id",all=TRUE)
 1540:
 1541:   data5_2_16$age=(checkdate-data5_2_16$dob)/365.25
 1542:   data5_2_16$age_c=varclass(data5_2_16$age,ageth)
 1543:   data5_2_16$self_mon12[is.na(data5_2_16$self_mon12)]<-2
 1544:   data5_2_16<-data5_2_16[,c("self_mon12","age_c","sex","type_dm")]
 1545:   data5_2_16<-data5_2_16[data5_2_16$type_dm %in% c(1,2),]
 1546:
 1547:   BIRO_dframe(dataname="data5_2_16",monitor=TRUE,filelog=logfile)
 1548:   BIRO_dframe(dataname="data5_2_16",monitor=TRUE,filelog="")
 1549:
 1550:        BIRO_report(ind="5_2_16",
 1551:                    title="5.2.16 Percent of patients with diabetes performing selfmonitoring of blood glucose/
urine testing",
 1552:                    var_cat="self_mon12",
 1553:                    class=c("age_c","sex","type_dm"),
 1554:                    lev_var_cat=names(levself_mon),
 1555:                    lev_class=list(a=classlabellist(ageth),b=levsex,c=levtype_dm),
 1556:                    lab_var_cat="selfmon. blood glucose/ urine",
 1557:                    lab_class=c("Age Classes","Gender","Type of Diabetes"),
 1558:                    tab=0,
 1559:                    bar=1,
 1560:                    barbeside=TRUE,
 1561:                    barnumber=TRUE,
 1562:                    barperc=TRUE,
 1563:                    box=0,
```

```
1564:                     lines=0,
1565:                     trellis=0,
1566:                     pie=0,
1567:                     texfile=texfile,
1568:                     dirtables=dirtables,
1569:                     dircsv=dirdataout,
1570:                     dirgraph=dirgraph,
1571:                     dirgraphlatex=dirgraphlatex,
1572:                     cex=cex,
1573:                     stand=1,
1574:                     catcol=c(2,3,4),
1575:                     outcome="self_mon12",
1576:                     cov=c("age_c","sex","type_dm"),
1577:                     factcov=c(TRUE,TRUE,TRUE),
1578:                     strata=NULL,
1579:                     levstd=list(a=levhypert,b=classlabellist(ageth),c=levsex,d=levtype_dm),
1580:                     sucvalue=1,
1581:                     per=100,
1582:                     caption="Portion of diabetes patients with anti hypertensive treatment")
1583:
1584:    rm(data5_2_16)
1585:     }
1586:   }
1587:   } else BIRO_report_toc(title="5.2.16 Percent of patients with diabetes performing selfmonitoring of blood
glucose/ urine testing")
1588:
1589:    #5.3 Outcome quality-intermediate outcomes
1590:
1591:    new_section("5.3. Outcome quality - intermediate outcomes",texfile=texfile)
1592:    new_section_html("5.3. Outcome quality - intermediate outcomes",htmlfile=htmlfile)
1593:
1594:    #5.3.1 Percentage of patients with most recent HbA1c level >9.0% (poor control)
1595:
1596:   if (is.null(episode$hba1c)==F) {
1597:   if (length(na.omit(episode$hba1c))>0) {
1598:
1599:     for (thisyear in anayears) {
1600:
1601:      checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1602:      oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1603:
1604:     assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1605:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1606:
1607:     assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
```

```
1608:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1609:
1610:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1611:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1612:
1613:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1614:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
1615:
1616:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
1617:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1618:
1619:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
1620:
1621:   hba1cth9<-9
1622:   episode$hba1c_c9<-varclass(as.numeric(as.character(episode$hba1c)),hba1cth9)
1623:   noNAepirefyear<-episode[is.na(episode$hba1c_c9)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
1624:   data5_3_1<-aggregate(noNAepirefyear$hba1c_c9,list(noNAepirefyear$patient_id),FUN="max")
1625:   names(data5_3_1)<-c("patient_id","hba1c_c9")
1626:   data5_3_1<-data5_3_1[is.na(data5_3_1$hba1c_c9)==FALSE,]
1627:   data5_3_1<-merge(data5_3_1,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
1628:                    by="patient_id",all=TRUE)
1629:
1630:   data5_3_1$age=(checkdate-data5_3_1$dob)/365.25
1631:   data5_3_1$age_c=varclass(data5_3_1$age,ageth)
1632:   data5_3_1<-data5_3_1[,c("hba1c_c9","age_c","sex","type_dm")]
1633:
1634:   BIRO_dframe(dataname="data5_3_1",monitor=TRUE,filelog=logfile)
1635:   BIRO_dframe(dataname="data5_3_1",monitor=TRUE,filelog="")
1636:
1637:
1638:        BIRO_report(ind="5_3_1",
1639:                    title="5.3.1 Percentage of patients with most recent HbA1c level greater than 9.0 pct (poor
control)",
1640:                    var=NULL,
1641:                    var_cat=NULL,
1642:                    lev_var_cat=NULL,
1643:                    lev_class=NULL,
1644:                    lab_var=NULL,
1645:                    lab_var_cat=NULL,
1646:                    lab_class=NULL,
1647:                    tab=0,
1648:                    box=0,
1649:                    bar=0,
```

```
1650:                     lines=0,
1651:                     trellis=0,
1652:                     pie=0,
1653:                     texfile=texfile,
1654:                     dirtables=dirtables,
1655:                     dircsv=dirdataout,
1656:                     dirgraph=dirgraph,
1657:                     dirgraphlatex=dirgraphlatex,
1658:                     cex=cex,
1659:                     stand=1,
1660:                     catcol=c(2,3,4),
1661:                     outcome="hba1c_c9",
1662:                     cov=c("age_c","sex","type_dm"),
1663:                     factcov=c(TRUE,TRUE,TRUE),
1664:                     strata=NULL,
1665:                     levstd=list(a=classlabellist(hba1cth9),b=classlabellist(ageth),c=levsex,d=levtype_dm),
1666:                     sucvalue=1,
1667:                     per=100,
1668:                     caption="Percentage of patients with most recent HbA1c level greater than 9.0 pct (poor
control)")
1669:
1670:    rm(data5_3_1)
1671:    rm(noNAepirefyear)
1672:     }
1673: }
1674: } else BIRO_report_toc(title="5.3.1 Percentage of patients with most recent HbA1c level greater than 9.0 pct (poor
control)")
1675:
1676:    #5.3.2 Percentage of patients with most recent HbA1c level >7,5%
1677:
1678:   if (is.null(episode$hba1c)==F) {
1679:   if (length(na.omit(episode$hba1c))>0) {
1680:
1681:    hba1cth75<-7.5
1682:    episode$hba1c_c75<-varclass(episode$hba1c,hba1cth75)
1683:
1684:     for (thisyear in anayears) {
1685:
1686:      checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1687:      oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1688:
1689:     assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1690:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1691:
1692:     assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
```

```
 1693:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 1694:
 1695:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 1696:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 1697:
 1698:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1699:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1700:
 1701:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1702:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1703:
 1704:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1705:
 1706:   noNAepirefyear<-episode[is.na(episode$hba1c_c75)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
 1707:   data5_3_2<-aggregate(noNAepirefyear$hba1c_c75,list(noNAepirefyear$patient_id),FUN="max")
 1708:   names(data5_3_2)<-c("patient_id","hba1c_c75")
 1709:   data5_3_2<-merge(data5_3_2,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1710:                    by="patient_id",all=TRUE)
 1711:
 1712:   data5_3_2$age=(checkdate-data5_3_2$dob)/365.25
 1713:   data5_3_2$age_c=varclass(data5_3_2$age,ageth)
 1714:   data5_3_2<-data5_3_2[,c("hba1c_c75","age_c","sex","type_dm")]
 1715:
 1716:   BIRO_dframe(dataname="data5_3_2",monitor=TRUE,filelog=logfile)
 1717:   BIRO_dframe(dataname="data5_3_2",monitor=TRUE,filelog="")
 1718:
 1719:
 1720:        BIRO_report(ind="5_3_2",
 1721:                    title="5.3.2 Percentage of patients with most recent HbA1c level greater than 7,5 pct",
 1722:                    var=NULL,
 1723:                    var_cat=NULL,
 1724:                    lev_var_cat=NULL,
 1725:                    lev_class=NULL,
 1726:                    lab_var=NULL,
 1727:                    lab_var_cat=NULL,
 1728:                    lab_class=NULL,
 1729:                    tab=0,
 1730:                    box=0,
 1731:                    bar=0,
 1732:                    lines=0,
 1733:                    trellis=0,
 1734:                    pie=0,
 1735:                    texfile=texfile,
```

```
1736:                     dirtables=dirtables,
1737:                     dircsv=dirdataout,
1738:                     dirgraph=dirgraph,
1739:                     dirgraphlatex=dirgraphlatex,
1740:                     cex=cex,
1741:                     stand=1,
1742:                     catcol=c(2,3,4),
1743:                     outcome="hba1c_c75",
1744:                     cov=c("age_c","sex","type_dm"),
1745:                     factcov=c(TRUE,TRUE,TRUE),
1746:                     strata=NULL,
1747:                     levstd=list(a=classlabellist(hba1cth75),b=classlabellist(ageth),c=levsex,d=levtype_dm),
1748:                     sucvalue=1,
1749:                     per=100,
1750:                     caption="Percentage of patients with most recent HbA1c level greater than 7,5 pct ")
1751:
1752:
1753:   rm(data5_3_2)
1754:   rm(noNAepirefyear)
1755:
1756:    }
1757:  }
1758:  } else BIRO_report_toc(title="5.3.2 Percentage of patients with most recent HbA1c level greater than 7,5 pct")
1759:
1760:   #5.3.3 Percentage of patients with most recent blood pressure <140/90 mmHg
1761:
1762:  if (is.null(episode$sbp)==F) {
1763:  if (length(na.omit(episode$sbp))>0) {
1764:
1765:   sbpth140<-140
1766:   episode$sbp_c140<-varclass(as.numeric(as.character(episode$sbp)),sbpth140)
1767:
1768:    for (thisyear in anayears) {
1769:
1770:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1771:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1772:
1773:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1774:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1775:
1776:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1777:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1778:
1779:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1780:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
```

```
 1781:
 1782:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 1783:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 1784:
 1785:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 1786:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 1787:
 1788:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1789:
 1790:   noNAepirefyear<-episode[is.na(episode$sbp_c140)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
 1791:   data5_3_3<-aggregate(noNAepirefyear$sbp_c140,list(noNAepirefyear$patient_id),FUN="max")
 1792:   names(data5_3_3)<-c("patient_id","sbp_c140")
 1793:   data5_3_3<-merge(data5_3_3,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1794:                    by="patient_id",all=TRUE)
 1795:
 1796:   data5_3_3$age=(checkdate-data5_3_3$dob)/365.25
 1797:   data5_3_3$age_c=varclass(data5_3_3$age,ageth)
 1798:   data5_3_3<-data5_3_3[,c("sbp_c140","age_c","sex","type_dm")]
 1799:
 1800:   BIRO_dframe(dataname="data5_3_3",monitor=TRUE,filelog=logfile)
 1801:   BIRO_dframe(dataname="data5_3_3",monitor=TRUE,filelog="")
 1802:
 1803:
 1804:       BIRO_report(ind="5_3_3",
 1805:                   title="5.3.3 Percentage of patients with most recent blood pressure less than 140/90 mmHg",
 1806:                   var=NULL,
 1807:                   var_cat=NULL,
 1808:                   lev_var_cat=NULL,
 1809:                   lev_class=NULL,
 1810:                   lab_var=NULL,
 1811:                   lab_var_cat=NULL,
 1812:                   lab_class=NULL,
 1813:                   tab=0,
 1814:                   box=0,
 1815:                   bar=0,
 1816:                   lines=0,
 1817:                   trellis=0,
 1818:                   pie=0,
 1819:                   texfile=texfile,
 1820:                   dirtables=dirtables,
 1821:                   dircsv=dirdataout,
 1822:                   dirgraph=dirgraph,
 1823:                   dirgraphlatex=dirgraphlatex,
```

```
1824:                     cex=cex,
1825:                     stand=1,
1826:                     catcol=c(2,3,4),
1827:                     outcome="sbp_c140",
1828:                     cov=c("age_c","sex","type_dm"),
1829:                     factcov=c(TRUE,TRUE,TRUE),
1830:                     strata=NULL,
1831:                     levstd=list(a=classlabellist(sbpth140),b=classlabellist(ageth),c=levsex,d=levtype_dm),
1832:                     sucvalue=1,
1833:                     per=100,
1834:                     caption="Percentage of patients with most recent blood pressure less than 140/90 mmHg")
1835:
1836:   rm(data5_3_3)
1837:   rm(noNAepirefyear)
1838:    }
1839: }
1840: } else BIRO_report_toc(title="5.3.3 Percentage of patients with most recent blood pressure less than 140/90 mmHg")
1841:
1842:   #5.3.4 Percentage of patients with BMI > 30
1843:
1844: if (is.null(episode$bmi)==F) {
1845: if (length(na.omit(episode$bmi))>0) {
1846:
1847:   bmith30<-30
1848:   episode$bmi_c30<-varclass(episode$bmi,bmith30)
1849:
1850:    for (thisyear in anayears) {
1851:
1852:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1853:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1854:
1855:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1856:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1857:
1858:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1859:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1860:
1861:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1862:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1863:
1864:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1865:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
1866:
1867:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
1868:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
```

```
 1869:
 1870:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 1871:
 1872:   noNAepirefyear<-episode[is.na(episode$bmi_c30)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
 1873:   data5_3_4<-aggregate(noNAepirefyear$bmi_c30,list(noNAepirefyear$patient_id),FUN="max")
 1874:   names(data5_3_4)<-c("patient_id","bmi_c30")
 1875:   data5_3_4<-merge(data5_3_4,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1876:                    by="patient_id",all=TRUE)
 1877:
 1878:   data5_3_4$age=(checkdate-data5_3_4$dob)/365.25
 1879:   data5_3_4$age_c=varclass(data5_3_4$age,ageth)
 1880:   data5_3_4<-data5_3_4[,c("bmi_c30","age_c","sex","type_dm")]
 1881:
 1882:   BIRO_dframe(dataname="data5_3_4",monitor=TRUE,filelog=logfile)
 1883:   BIRO_dframe(dataname="data5_3_4",monitor=TRUE,filelog="")
 1884:
 1885:        BIRO_report(ind="5_3_4",
 1886:                    title="5.3.4 Percentage of patients with BMI greater than 30",
 1887:                    var=NULL,
 1888:                    var_cat=NULL,
 1889:                    lev_var_cat=NULL,
 1890:                    lev_class=NULL,
 1891:                    lab_var=NULL,
 1892:                    lab_var_cat=NULL,
 1893:                    lab_class=NULL,
 1894:                    tab=0,
 1895:                    box=0,
 1896:                    bar=0,
 1897:                    lines=0,
 1898:                    trellis=0,
 1899:                    pie=0,
 1900:                    texfile=texfile,
 1901:                    dirtables=dirtables,
 1902:                    dircsv=dirdataout,
 1903:                    dirgraph=dirgraph,
 1904:                    dirgraphlatex=dirgraphlatex,
 1905:                    cex=cex,
 1906:                    stand=1,
 1907:                    catcol=c(2,3,4),
 1908:                    outcome="bmi_c30",
 1909:                    cov=c("age_c","sex","type_dm"),
 1910:                    factcov=c(TRUE,TRUE,TRUE),
 1911:                    strata=NULL,
```

```
1912:                     levstd=list(a=classlabellist(bmith30),b=classlabellist(ageth),c=levsex,d=levtype_dm),
1913:                     sucvalue=1,
1914:                     per=100,
1915:                     caption="Percentage of patients with BMI greater than 30")
1916:
1917:    rm(data5_3_4)
1918:    rm(noNAepirefyear)
1919:     }
1920: }
1921: } else BIRO_report_toc(title="5.3.4 Percentage of patients with BMI greater than 30")
1922:
1923:    #5.3.5 Percentage with microalbuminuria in last 12 months (among those who have been tested)
1924:
1925:
1926: if (is.null(episode$microal)==F) {
1927: if (length(na.omit(episode$microal))>0) {
1928:
1929:    microalth30<-30
1930:    episode$microal_c30<-varclass(episode$microal,microalth30)
1931:
1932:     for (thisyear in anayears) {
1933:
1934:      checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
1935:      oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
1936:
1937:     assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
1938:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1939:
1940:     assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
1941:           "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
1942:
1943:     assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
1944:           "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
1945:
1946:     assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
1947:            "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
1948:
1949:     assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
1950:            "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
1951:
1952:     assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
1953:
1954:    noNAepirefyear<-episode[is.na(episode$microal_c30)==FALSE & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,]
1955:    data5_3_5<-aggregate(noNAepirefyear$microal_c30,list(noNAepirefyear$patient_id),FUN="max")
```

```
 1956:   names(data5_3_5)<-c("patient_id","microal_c30")
 1957:   data5_3_5<-merge(data5_3_5,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 1958:                   by="patient_id",all=TRUE)
 1959:
 1960:   data5_3_5$age=(checkdate-data5_3_5$dob)/365.25
 1961:   data5_3_5$age_c=varclass(data5_3_5$age,ageth)
 1962:   data5_3_5<-data5_3_5[,c("microal_c30","age_c","sex","type_dm")]
 1963:
 1964:   BIRO_dframe(dataname="data5_3_5",monitor=TRUE,filelog=logfile)
 1965:   BIRO_dframe(dataname="data5_3_5",monitor=TRUE,filelog="")
 1966:
 1967:        BIRO_report(ind="5_3_5",
 1968:                   title="5.3.5 Percentage with microalbuminuria in last 12 months (among those who have been
tested)",
 1969:                   var=NULL,
 1970:                   var_cat=NULL,
 1971:                   lev_var_cat=NULL,
 1972:                   lev_class=NULL,
 1973:                   lab_var=NULL,
 1974:                   lab_var_cat=NULL,
 1975:                   lab_class=NULL,
 1976:                   tab=0,
 1977:                   box=0,
 1978:                   bar=0,
 1979:                   lines=0,
 1980:                   trellis=0,
 1981:                   pie=0,
 1982:                   texfile=texfile,
 1983:                   dirtables=dirtables,
 1984:                   dircsv=dirdataout,
 1985:                   dirgraph=dirgraph,
 1986:                   dirgraphlatex=dirgraphlatex,
 1987:                   cex=cex,
 1988:                   stand=1,
 1989:                   catcol=c(2,3,4),
 1990:                   outcome="microal_c30",
 1991:                   cov=c("age_c","sex","type_dm"),
 1992:                   factcov=c(TRUE,TRUE,TRUE),
 1993:                   strata=NULL,
 1994:                   levstd=list(a=classlabellist(bmith30),b=classlabellist(ageth),c=levsex,d=levtype_dm),
 1995:                   sucvalue=1,
 1996:                   per=100,
 1997:                   caption="Percentage with microalbuminuria in last 12 months (among those who have been tested)")
 1998:
```

```
 1999:   rm(data5_3_5)
 2000:   rm(noNAepirefyear)
 2001:    }
 2002: }
 2003: } else BIRO_report_toc(title="5.3.5 Percentage with microalbuminuria in last 12 months (among those who have been
tested)")
 2004:
 2005:   #5.3.6 Rate of current smokers among diabetes patients
 2006:
 2007: if (is.null(episode$smok_stat)==F) {
 2008: if (length(na.omit(episode$smok_stat))>0) {
 2009:
 2010:    for (thisyear in anayears) {
 2011:
 2012:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 2013:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 2014:
 2015:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
 2016:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 2017:
 2018:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
 2019:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 2020:
 2021:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 2022:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 2023:
 2024:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 2025:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 2026:
 2027:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 2028:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 2029:
 2030:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 2031:
 2032:   epirefyear<-episode[episode$epi_date >= oneyearbf & episode$epi_date <= checkdate,]
 2033:
 2034:   current_smoker<-as.data.frame(epirefyear[epirefyear$smok_stat==1,c("patient_id")])
 2035:   current_smoker$current_smoker=1
 2036:   names(current_smoker)<-c("patient_id","current_smoker")
 2037:   data5_3_6<-merge(current_smoker,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
 2038:                    by="patient_id",all=TRUE)
 2039:
 2040:   data5_3_6$age=(checkdate-data5_3_6$dob)/365.25
 2041:   data5_3_6$age_c=varclass(data5_3_6$age,ageth)
```

```
2042:   data5_3_6$current_smoker[is.na(data5_3_6$current_smoker)]<-2
2043:   rm(current_smoker)
2044:   data5_3_6<-data5_3_6[,c("current_smoker","age_c","sex","type_dm")]
2045:   data5_3_6<-data5_3_6[data5_3_6$type_dm %in% c(1,2),]
2046:
2047:   BIRO_dframe(dataname="data5_3_6",monitor=TRUE,filelog=logfile)
2048:   BIRO_dframe(dataname="data5_3_6",monitor=TRUE,filelog="")
2049:
2050:        BIRO_report(ind="5_3_6",
2051:                    title="5.3.6 Rate of current smokers among diabetes patients",
2052:                    var=NULL,
2053:                    var_cat=NULL,
2054:                    lev_var_cat=NULL,
2055:                    lev_class=NULL,
2056:                    lab_var=NULL,
2057:                    lab_var_cat=NULL,
2058:                    lab_class=NULL,
2059:                    tab=0,
2060:                    box=0,
2061:                    bar=0,
2062:                    lines=0,
2063:                    trellis=0,
2064:                    pie=0,
2065:                    texfile=texfile,
2066:                    dirtables=dirtables,
2067:                    dircsv=dirdataout,
2068:                    dirgraph=dirgraph,
2069:                    dirgraphlatex=dirgraphlatex,
2070:                    cex=cex,
2071:                    stand=1,
2072:                    catcol=c(2,3,4),
2073:                    outcome="current_smoker",
2074:                    cov=c("age_c","sex","type_dm"),
2075:                    factcov=c(TRUE,TRUE,TRUE),
2076:                    strata=NULL,
2077:                    levstd=list(a=levcurrentsmoker,b=classlabellist(ageth),c=levsex,d=levtype_dm),
2078:                    sucvalue=1,
2079:                    per=100,
2080:                    caption="Rate of current smokers among diabetes patients")
2081:
2082:   rm(data5_3_6)
2083:    }
2084: }
2085: } else BIRO_report_toc(title="5.3.6 Rate of current smokers among diabetes patients")
2086:
```

```
2087:   #5.3.7 Former or current foot ulceration
2088:
2089: if (length(na.omit(episode$ulcer_l))>0 | length(na.omit(episode$ulcer_r))>0) {
2090:
2091:    for (thisyear in anayears) {
2092:
2093:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
2094:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
2095:
2096:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
2097:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
2098:
2099:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
2100:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
2101:
2102:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
2103:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
2104:
2105:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
2106:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
2107:
2108:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
2109:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
2110:
2111:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
2112:
2113:   epirefyear<-episode[ episode$epi_date >= oneyearbf & episode$epi_date <= checkdate,]
2114:
2115:   data5_3_7<-ifelse(epirefyear$ulcer_l==1 | epirefyear$ulcer_r==1,1,0)
2116:   data5_3_7<-aggregate(data5_3_7,list(epirefyear$patient_id),FUN="min")
2117:   names(data5_3_7)<-c("patient_id","ulcer")
2118:   data5_3_7<-merge(data5_3_7,patient[patient$end_date>checkdate | patient$end_date>oneyearbf | is
.na(patient$end_date),],
2119:                    by="patient_id",all=TRUE)
2120:
2121:   data5_3_7$age=(checkdate-data5_3_7$dob)/365.25
2122:   data5_3_7$age_c=varclass(data5_3_7$age,ageth)
2123:   data5_3_7<-data5_3_7[,c("ulcer","age_c","sex","type_dm")]
2124:
2125:   BIRO_dframe(dataname="data5_3_7",monitor=TRUE,filelog=logfile)
2126:   BIRO_dframe(dataname="data5_3_7",monitor=TRUE,filelog="")
2127:
2128:        BIRO_report(ind="5_3_7",
2129:                    title="5.3.7 Former or current foot ulceration",
2130:                    var=NULL,
```

```
2131:                     var_cat=NULL,
2132:                     lev_var_cat=NULL,
2133:                     lev_class=NULL,
2134:                     lab_var=NULL,
2135:                     lab_var_cat=NULL,
2136:                     lab_class=NULL,
2137:                     tab=0,
2138:                     box=0,
2139:                     bar=0,
2140:                     lines=0,
2141:                     trellis=0,
2142:                     pie=0,
2143:                     texfile=texfile,
2144:                     dirtables=dirtables,
2145:                     dircsv=dirdataout,
2146:                     dirgraph=dirgraph,
2147:                     dirgraphlatex=dirgraphlatex,
2148:                     cex=cex,
2149:                     stand=1,
2150:                     catcol=c(2,3,4),
2151:                     outcome="ulcer",
2152:                     cov=c("age_c","sex","type_dm"),
2153:                     factcov=c(TRUE,TRUE,TRUE),
2154:                     strata=NULL,
2155:                     levstd=list(a=levulcer,b=classlabellist(ageth),c=levsex,d=levtype_dm),
2156:                     sucvalue=1,
2157:                     per=100,
2158:                     caption="Former or current foot ulceration")
2159:
2160:   rm(data5_3_7)
2161:    }
2162: } else BIRO_report_toc(title="5.3.7 Former or current foot ulceration")
2163:
2164:   #5.3 Outcome quality-terminal outcomes
2165:
2166:   new_section("5.4. Outcome quality - terminal outcomes",texfile=texfile)
2167:   new_section_html("5.4. Outcome quality - terminal outcomes",htmlfile=htmlfile)
2168:
2169: #5.4.1 Annual incidence of dialysis and/or transplantation (renal
2170: #      replacement therapy in patients with diabetes
2171:
2172:
2173: if (length(na.omit(episode$dialysis==1))>0 | length(na.omit(episode$transplant==1))>0) {
2174:
2175:    for (thisyear in anayears) {
```

```
 2176:
 2177:     checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
 2178:     oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
 2179:
 2180:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
 2181:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 2182:
 2183:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
 2184:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
 2185:
 2186:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
 2187:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
 2188:
 2189:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
 2190:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
 2191:
 2192:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
 2193:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
 2194:
 2195:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
 2196:
 2197:   data5_4_1<-as.data.frame(episode[(episode$dialysis==1 | episode$transplant==1) & episode$epi_date >= oneyearbf &
episode$epi_date <= checkdate,"patient_id"])
 2198:   names(data5_4_1)<-"patient_id"
 2199:   if (dim(data5_4_1)[1]>0) {
 2200:    data5_4_1$diatrans=1
 2201:
 2202:    if (is.null(dim(data5_4_1))==FALSE) {
 2203:     data5_4_1<-merge(data5_4_1,patient[patient$end_date>checkdate | (format(patient$end_date,format="%Y") ==
format(checkdate,format="%Y")) | is.na(patient$end_date),],
 2204:                      by="patient_id",all=TRUE)
 2205:
 2206:     data5_4_1$age=(checkdate-data5_4_1$dob)/365.25
 2207:     data5_4_1$age_c=varclass(data5_4_1$age,ageth)
 2208:     data5_4_1<-data5_4_1[,c("patient_id","type_dm","sex","age_c","diatrans")]
 2209:
 2210:
 2211:     data5_4_1$diatrans[is.na(data5_4_1$diatrans)]<-2
 2212:
 2213:        BIRO_report(ind="5_4_1",
 2214:                    title="5.4.1 Annual incidence of dialysis and/or transplantation (renal replacement therapy in
patients with diabetes",
 2215:                    var=NULL,
 2216:                    var_cat=NULL,
 2217:                    lev_var_cat=NULL,
```

```
 2218:                     lev_class=NULL,
 2219:                     lab_var=NULL,
 2220:                     lab_var_cat=NULL,
 2221:                     lab_class=NULL,
 2222:                     tab=0,
 2223:                     box=0,
 2224:                     bar=0,
 2225:                     lines=0,
 2226:                     trellis=0,
 2227:                     pie=0,
 2228:                     texfile=texfile,
 2229:                     dirtables=dirtables,
 2230:                     dircsv=dirdataout,
 2231:                     dirgraph=dirgraph,
 2232:                     dirgraphlatex=dirgraphlatex,
 2233:                     cex=cex,
 2234:                     stand=1,
 2235:                     catcol=c(2,3,4),
 2236:                     outcome="diatrans",
 2237:                     cov=c("age_c","sex","type_dm"),
 2238:                     factcov=c(TRUE,TRUE,TRUE),
 2239:                     strata=NULL,
 2240:                     levstd=list(a="Yes",b=classlabellist(ageth),c=levsex,d=levtype_dm),
 2241:                     sucvalue=1,
 2242:                     per=100000,
 2243:                     caption="Annual incidence of dialysis and/or transplantation (renal replacement therapy in
patients with diabetes")
 2244:
 2245:
 2246:
 2247:    rm(data5_4_1)
 2248:    }
 2249:   }
 2250:  }
 2251: }  else BIRO_report_toc(title="5.4.1 Annual incidence of dialysis and/or transplantation (renal replacement
therapy in patients with diabetes")
 2252:
 2253:
 2254: #5.4.2 ESRF in Persons with Diabetes
 2255:
 2256:  if (length(na.omit(episode$esrf==1))>0) {
 2257:
 2258:  for (thisyear in anayears) {
 2259:
 2260:    assign("dirdataout",paste(dirse,"/output/data/", "#",launchtime,
```

```
2261:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
2262:
2263:    assign("dirgraph",paste(dirse,"/output/reports/", "#",launchtime,
2264:          "/",thisyear,"/",centre_id,"/graphs",sep=""),envir=.GlobalEnv)
2265:
2266:    assign("dirtables",paste(dirse,"/output/reports/", "#",launchtime,
2267:          "/",thisyear,"/",centre_id,"/tables",sep=""),envir=.GlobalEnv)
2268:
2269:    assign("dirhtml",paste(dirse,"/output/reports/", "#",launchtime,
2270:           "/",thisyear,"/",centre_id,"/html",sep=""),envir=.GlobalEnv)
2271:
2272:    assign("dirreport",paste(dirse,"/output/reports/", "#",launchtime,
2273:           "/",thisyear,"/",centre_id,sep=""),envir=.GlobalEnv)
2274:
2275:    assign("texfile",paste(dirreport,"/report.tex",sep=""),envir=.GlobalEnv)
2276:
2277:    checkdate<-as.Date(paste(refanadate,"/",thisyear,sep=""),"%d/%m/%Y")
2278:    oneyearbf<-as.Date(paste(refanadate,"/",thisyear-1,sep=""),"%d/%m/%Y")
2279:
2280:   data5_4_2<-as.data.frame(na.omit(episode[episode$esrf==1  & episode$epi_date >= oneyearbf & episode$epi_date <=
checkdate,"patient_id"]))
2281:   names(data5_4_2)<-"patient_id"
2282:   data5_4_2$esrf=1
2283:
2284:   if (is.null(dim(data5_4_2))==FALSE) {
2285:   data5_4_2<-merge(data5_4_2,patient[patient$end_date>checkdate | (format(patient$end_date,format="%Y") ==
format(checkdate,format="%Y")) | is.na(patient$end_date),],
2286:                    by="patient_id",all=TRUE)
2287:
2288:   data5_4_2$age=(checkdate-data5_4_2$dob)/365.25
2289:   data5_4_2$age_c=varclass(data5_4_2$age,ageth)
2290:   data5_4_2<-data5_4_2[,c("patient_id","type_dm","sex","age_c","esrf")]
2291:
2292:
2293:   data5_4_2$esrf[is.na(data5_4_2$esrf)]<-2
2294:
2295:        BIRO_report(ind="5_4_2",
2296:                    title="5.4.2 ESRF in Persons with Diabetes",
2297:                    var=NULL,
2298:                    var_cat=NULL,
2299:                    lev_var_cat=NULL,
2300:                    lev_class=NULL,
2301:                    lab_var=NULL,
2302:                    lab_var_cat=NULL,
2303:                    lab_class=NULL,
```

```
2304:                     tab=0,
2305:                     box=0,
2306:                     bar=0,
2307:                     lines=0,
2308:                     trellis=0,
2309:                     pie=0,
2310:                     texfile=texfile,
2311:                     dirtables=dirtables,
2312:                     dircsv=dirdataout,
2313:                     dirgraph=dirgraph,
2314:                     dirgraphlatex=dirgraphlatex,
2315:                     cex=cex,
2316:                     stand=1,
2317:                     catcol=c(2,3,4),
2318:                     outcome="esrf",
2319:                     cov=c("age_c","sex","type_dm"),
2320:                     factcov=c(TRUE,TRUE,TRUE),
2321:                     strata=NULL,
2322:                     levstd=list(a="Yes",b=classlabellist(ageth),c=levsex,d=levtype_dm),
2323:                     sucvalue=1,
2324:                     per=100000,
2325:                     caption="ESRD in Persons with Diabetes")
2326:
2327:   }
2328:
2329:  rm(data5_4_2)
2330:  }
2331: } else BIRO_report_toc(title="5.4.2 ESRF in Persons with Diabetes")
2332:
2333:
2334:
```

```
    1: # ##############################################################################
    2: #                                                                            #
    3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
    4: # File: BIRO_boxplots.r                                                      #
    5: #                                                                            #
#
    6: # ##############################################################################
    7: # ------------------------
    8: # Authors:
    9: # Luca Rossi       <redsluke@gmail.com>
   10: # Fabrizio Carinci <research@fabcarinci.net>
   11: # Institutions: Serectrix snc
   12: # Created: 2007-11-21
   13: # Version: 2008-11-23
   14: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
   15: # Programming Language: R 2.8.0
   16: # ------------------------
   17: # COPYRIGHT INFORMATION
   18: #
   19: # This file is free software; you can redistribute it and/or modify
   20: # it under the terms of the GNU General Public License as published by
   21: # the Free Software Foundation; either version 2, or (at your option)
   22: # any later version.
   23: #
   24: # This file is distributed in the hope that it will be useful,
   25: # but WITHOUT ANY WARRANTY; without even the implied warranty of
   26: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   27: # GNU General Public License for more details.
   28: #
   29: # You should have received a copy of the GNU General Public License
   30: # along with this file; see the file COPYING.  If not, write to
   31: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
   32: #
   33: # In short: you may use this file any way you like, as long as you
   34: # don't charge money for it, remove this notice, or hold anyone liable
   35: # for its results.
   36: #
   37: # biro_location.r is part of WP Statistical Engine of the BIRO Project
   38: # GPL Copyright, The BIRO Project
   39: #
   40: # ------------------------
   41: #
   42: # CONTENT
   43: #
   44: #   BIRO_boxplot
```

```
45: #   BIRO_plotbox
46: #
47: # ##########################################################################
48:
49: # ##########################################################################
50: #  BIRO_boxplot                                                            #
51: # ##########################################################################
52: # ------------------------
53: # Authors:
54: # Luca Rossi       <redsluke@gmail.com>
55: # Fabrizio Carinci <research@fabcarinci.net>
56: # Institutions: Serectrix snc
57: # Created: 2007-11-21
58: # Version: 2008-11-23
59: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
60: # Programming Language: R 2.8.0
61: # ------------------------
62: # DESCRIPTION
63: #
64: # BIRO_boxplot draws boxplot for a conditionated distribution
65: #
66: # DEPENDENCIES
67: #
68: #  BIRO_out     (BIRO_dispersion.r)
69: #  BIRO_plotbox
70: #
71: # ##########################################################################
72: # PARAMETERS(!=required)
73: #
74: # data     !               = dataset containing data
75: # var      !               = the name of the target variable
76: # lev      !               = a list containing the vactor with che levels of
77: #                            the categorical variable (list)
78: # stratum                  = the name of the stratum variables
79: # out     ["out"]          = the name of the outliers csv files
80: # notch    !               = logical. if notch is TRUE, a notch is drawn in each
81: #                            side of the boxes. If the notches of two plots do
82: #                            not overlap this is \221strong evidence\222 that the two
83: #                            means differ (Chambers et al., 1983, p. 62).
84: #                            See boxplot.stats for the calculations used.
85: # ext     [csv]            = the extension of the output files
86: # date    [System date]    = date of analysis
87: # dirout     [current]     = the path of the output text files
88: # namegraph !              = the coded name of the output files
89: # dirgraph   [current]     = the path of the output (jpg,eps,png files)
```

```
 90: # ##############################################################################
 91:
 92: BIRO_boxplot<-function(data,
 93:                        var,
 94:                        lev=as.list(),
 95:                        stratum=NULL,
 96:                        out="out",
 97:                        notch,
 98:                        dirout=getwd(),
 99:                        ext="csv",
100:                        date=Sys.Date(),
101:                        namegraph,
102:                        dirgraph) {
103:
104:
105:
106:  depend=""
107:  if (length(stratum)>0) {
108:   depend="~"
109:  }
110:
111:  write("Preparing to do boxp",file="")
112:  if (nchar(logfile)>0) {
113:   write("Preparing to do boxp",file=logfile,append=TRUE)
114:  }
115:  boxp<-boxplot(as.formula(paste(var,depend,paste(stratum,collapse="+"))),
116:                notch = notch,
117:                boxfill= rainbow(5),
118:                frame= FALSE,
119:                outl= TRUE,
120:                pch=4,
121:                outline=TRUE,
122:                plot=FALSE,
123:                data=data)
124:
125:  write("boxp done",file="")
126:  if (nchar(logfile)>0) {
127:   write("boxp done",file=logfile,append=TRUE)
128:  }
129:
130:  longnames<-unlist(strsplit(as.character(boxp$names), ".", fixed = TRUE))
131:  stats<-t(as.data.frame((boxp$stats)))
132:  conf<-t(as.data.frame((boxp$conf)))
133:  n<-as.data.frame((boxp$n))
134:  longnames<-as.data.frame(matrix(longnames,ncol=length(stratum),byrow=TRUE))
```

```
135:  names(longnames)<-stratum
136:  boxp<-cbind(stats,longnames)
137:
138:  names(boxp)<-c(c("l_wisk","pcl_25x","median","pcl_75x","u_wisk"),stratum)
139:
140:  conf<-cbind(conf,longnames)
141:  names(conf)<-c(c("l_notch","u_notch"),stratum)
142:
143:  n<-cbind(n,longnames)
144:  names(n)<-c("n",stratum)
145:  outliers<-BIRO_out(data=data,
146:                     var=var,
147:                     stratum=stratum,
148:                     out="temp",
149:                     dirout=getwd(),
150:                     ext="csv",
151:                     date=Sys.Date())
152:
153:  if (is.null(outliers)==FALSE) {
154:  # file.remove(paste(getwd(),"//","out.csv",sep=""))
155:   names(outliers)[pmatch("value",names(outliers))]<-"outlie_x"
156:   write("Boxplot statistics calculated",file="")
157:   if (nchar(logfile)>0) {
158:    write("Boxplot statistics calculated",file=logfile,append=TRUE)
159:   }
160:  }
161:  INDEX<-list()
162:  for (i in 1:length(stratum)) {
163:   INDEX[[i]]<-data[,pmatch(stratum[i],names(data))]
164:  }
165:  means<-aggregate(data[,pmatch(var,names(data))],INDEX,FUN="mean")
166:  names(means)<-c(stratum,"mean")
167:  if (is.null(outliers)==FALSE) {
168:   bxptab<-merge(means,outliers,by=stratum,all.x=TRUE)
169:  } else {
170:   bxptab<-means
171:   bxptab$outlie_x<-NA #these are no outliers so we puts NAs instead of nothing
172:   bxptab$variable<-var
173:   bxptab$Freq<-0
174:  }
175:  bxptab<-merge(bxptab,boxp,by=stratum,all.x=TRUE)
176:  bxptab<-merge(bxptab,conf,by=stratum,all.x=TRUE)
177:  bxptab<-merge(bxptab,n,by=stratum,all.x=TRUE)
178:  bxp<-as.data.frame(bxptab)
179:  bxptab$variable=var
```

```
180:  bxptab$codist=centre_id
181:  write.csv(bxptab,paste(dirdataout,"/i",namegraph,"d4_3a.csv",sep=""),
182:            row.names = FALSE)
183:
184:  write("Boxplot statistics stored in csv file",file="")
185:  if (nchar(logfile)>0) {
186:   write("Boxplot statistics stored in csv file",file=logfile,append=TRUE)
187:  }
188:
189:  BIRO_plotbox(bxpdata=bxptab,
190:               lev=lev,
191:               notch=notch,
192:               namegraph=namegraph,
193:               dirgraph=dirgraph,
194:               out=out,
195:               n=n,
196:               variable=variable,
197:               stratum=stratum)
198:
199: }
200:
```

```
  1: # ############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: BIRO_dispersion.r                                                    #
  5: #                                                                            #
#
  6: # ############################################################################
  7:
  8: # ------------------------
  9: # Authors:
 10: # Luca Rossi       <redsluke@gmail.com)
 11: # Fabrizio Carinci <research@fabcarinci.net)
 12: # Institutions: Serectrix snc
 13: # Created: 2007-11-21
 14: # Version: 2008-10-23
 15: # OS: Windows XP/Vista
 16: # Programming Language: R 2.8.0
 17: # ------------------------
 18: # COPYRIGHT INFORMATION
 19: #
 20: # This file is free software; you can redistribute it and/or modify
 21: # it under the terms of the GNU General Public License as published by
 22: # the Free Software Foundation; either version 2, or (at your option)
 23: # any later version.
 24: #
 25: # This file is distributed in the hope that it will be useful,
 26: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 27: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 28: # GNU General Public License for more details.
 29: #
 30: # You should have received a copy of the GNU General Public License
 31: # along with this file; see the file COPYING.  If not, write to
 32: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 33: #
 34: # In short: you may use this file any way you like, as long as you
 35: # don't charge money for it, remove this notice, or hold anyone liable
 36: # for its results.
 37: #
 38: # BIRO_dispersion.r is part of WP Statistical Engine of the BIRO Project
 39: # GPL Copyright, The BIRO Project
 40: #
 41: # ------------------------
 42: #
 43: # CONTENT
 44: #
```

```
45: #   BIRO_range
46: #   BIRO_out
47: #   BIRO_variance
48: #
49: ##############################################################################
50:
51:
52: # ############################################################################
53: #  BIRO_range                                                                #
54: # ############################################################################
55: # ------------------------
56: # Authors:
57: # Luca Rossi       <redsluke@gmail.com>
58: # Fabrizio Carinci <research@fabcarinci.net>
59: # Institutions: Serectrix snc
60: # Created: 2007-11-21
61: # Version: 2008-11-23
62: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
63: # Programming Language: R 2.8.0
64: # ------------------------
65: # DESCRIPTION
66: #
67: # BIRO_range calculates range for Conditionated Frequency Distribution
68: #
69: # DEPENDENCIES
70: #
71: #  R package: base
72: #
73: # ############################################################################
74: # PARAMETERS(!=required)
75: #
76: # idcentro    !             = the id of the regional repository
77: # date        [System date] = date of analysis
78: # data        !             = the dataset processed containing data
79: # var         !             = the name of the stratum variable
80: # stratum     !             = the name of the stratum variable
81: # codtab      !             =
82: # dircsv      !             = the path of the output (csv files)
83: # ############################################################################
84:
85: BIRO_range<-function(idcentro,
86:                      date=Sys.Date(),
87:                      data,
88:                      var,
89:                      stratum=NULL,
```

```
 90:                     codtab,
 91:                     dircsv) {
 92:
 93:  onlythese<-c(var,stratum)
 94:  data<-na.omit(data[,pmatch(onlythese,names(data))])
 95:
 96:  retdata<-NULL
 97:  for (k in 1:length(var)) {
 98:   spltdata<-
 99:   split(data[,pmatch(var,names(data))],data[,pmatch(stratum,names(data))])
100:
101:   newdata<-NULL
102:   for (i in 1:(length(names(spltdata)))) {
103:    temp<-as.data.frame(range(as.vector(spltdata[[i]])))
104:    temp$stat=c("min","max")
105:    names(temp)<-"value"
106:    temp$variable=var
107:    temp$stratum<-names(spltdata)[i]
108:    names(temp)[pmatch("stratum",names(temp))]<-stratum
109:    temp$idcentro=idcentro
110:    temp$date=date
111:
112:    newdata<-rbind(newdata,temp)
113:   }
114:
115:   retdata<-rbind(retdata,newdata)
116:  }
117:  write.csv(as.data.frame(retdata),paste(dircsv,"/i",codtab,"d2_3a.csv",sep=""),
118:            row.names = FALSE)
119:  return(retdata)
120: }
121:
122: # ##############################################################################
123: #  BIRO_out                                                                    #
124: # ##############################################################################
125: # ------------------------
126: # Authors:
127: # Luca Rossi       <redsluke@gmail.com>
128: # Fabrizio Carinci <research@fabcarinci.net>
129: # Institutions: Serectrix snc
130: # Created: 2007-11-21
131: # Version: 2008-11-23
132: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
133: # Programming Language: R 2.8.0
134: # ------------------------
```

```
135: # DESCRIPTION
136: #
137: # BIRO_out calculates outliers for Conditionated Frequency Distribution
138: #
139: # DEPENDENCIES
140: #
141: #  R package: base
142: #
143: # ###########################################################################
144: # PARAMETERS(!=required)
145: #
146: # data      !             = input dataframe
147: # var       !             = one or more target variable corresponding to frame
148: #                           column names
149: # stratum   !             = one or more strata variable corresponding to frame
150: #                           column names (including centre id)
151: #                           all variables preceding n are strata variables
152: # dirout    [getwd()]     = output location
153: # out       [out]         = name of output csv dataset
154: # ext       [csv]         = output file format
155: # date      [Sys.Date()]  = date variable or value
156: # ###########################################################################
157:
158: BIRO_out<-function(data,
159:                   var,
160:                   stratum=NULL,
161:                   out="out",
162:                   dirout=getwd(),
163:                   ext="csv",
164:                   date=Sys.Date()) {
165:
166:  # initializes the return data.frame
167:  retdata<-NULL
168:
169:  # for each variables (this con do more then a variable each time)
170:  #
171:  # IMPORTANT: they must be analyzed with the same stratum
172:  for (k in 1:length(var)) {
173:   # for each variables (this con do more then a variable each time)
174:
175:   # is stratum are spedified
176:   if (is.null(stratum)==FALSE) {
177:    boxp<-
178:    boxplot(as.formula(paste(var[k],"~",paste(stratum,collapse="+"))),
179:            plot=FALSE,data=data)
```

```
180:    longnames<-unlist(strsplit(as.character(boxp$names), ".", fixed = TRUE))
181:    group<-1:length(boxp$names)
182:    #if there aren't outliers
183:    if (length(boxp$out)>0 && length(boxp$out)>0) {
184:     boxp<-as.data.frame(table(as.vector(boxp$out),as.vector(boxp$group)))
185:    }
186:    #if there aren't outliers
187:    if (length(boxp$out)==0) {
188:     boxp<-as.data.frame(cbind(NA,NA,NA))
189:     names(boxp)<-c("Var1","Var2","Freq")
190:     boxp<-na.omit(boxp)
191:    }
192:    st<-NULL
193:    for (i in 1:length(stratum)) {
194:     st[[i]]<-longnames[seq(from=i,to=length(longnames),by=length(stratum))]
195:    }
196:    ns<-as.data.frame(st)
197:    names(ns)<-stratum
198:    nsd<-cbind(group,ns)
199:   }
200:   # if stratum are not spedified
201:   # do simply the boxplot of the variable
202:   if (is.null(stratum)==TRUE) {
203:    boxp<-boxplot(as.vector(data[,pmatch(var[k],names(data))]))
204:    boxp<-as.data.frame(table(as.vector(boxp$out)))
205:   }
206:   # if there aren't outliers
207:   if (dim(boxp)[1]>0) {
208:    boxp<-boxp[boxp$Freq>0,]
209:   }
210:   # if stratum are specified
211:   # prepare data.frame for the merge in this way
212:   if (is.null(stratum)==FALSE) {
213:    names(boxp)<-c("value","group","Freq")
214:    boxp<-merge(boxp,nsd,by="group")
215:    boxp<-boxp[,pmatch(c("value","Freq",stratum),names(boxp))]
216:   }
217:   # if stratum are not specified
218:   # prepare data.frame for the merge in this way
219:   if (is.null(stratum)==TRUE) {
220:    names(boxp)<-c("value","Freq")
221:   }
222:   # if there aren't outliers
223:   if (dim(boxp)[1]>0) {
224:    boxp$variable<-var[k]
```

```
225:   }
226:   #append the data.frame to the return data.frame
227:   retdata<-rbind(retdata,as.data.frame(boxp))
228:  }
229:  # if there are outliers for some variables
230:  if (dim(retdata)[1]>0) {
231:   retdata$date=date
232:   write.csv(as.data.frame(retdata),paste(dirout,"/i",out,"d1_2a.",ext,sep=""),
233:             row.names = FALSE)
234:   return(retdata)
235:  }
236:
237:  #if there aren't outliers for noone variables
238:  if (dim(retdata)[1]==0) {
239:    return(NULL)
240:  }
241:
242: }
243:
244: # ##############################################################################
245: #  BIRO_out                                                                    #
246: # ##############################################################################
247: # ------------------------
248: # Authors:
249: # Luca Rossi       <redsluke@gmail.com>
250: # Fabrizio Carinci <research@fabcarinci.net>
251: # Institutions: Serectrix snc
252: # Created: 2007-11-21
253: # Version: 2008-11-23
254: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
255: # Programming Language: R 2.8.0
256: # ------------------------
257: # DESCRIPTION
258: #
259: # BIRO_out calculates outliers for Conditionated Frequency Distribution
260: #
261: # DEPENDENCIES
262: #
263: #  R package: base
264: #
265: # ##############################################################################
266: # PARAMETERS(!=required)
267: #
268: # data     !               = input dataframe
269: # var      !               = one or more target variable corresponding to frame
```

```
270: #                           column names
271: #                           stats for more variables are appended, a variable
272: #                           name column is automatically produced
273: # stratum  !               = one or more strata variable corresponding to frame
274: #                           column names (including centre id)
275: #                           all variables preceding n are strata variables
276: # dirout    [getwd()]      = output location
277: # out       [var]         <= name of output csv dataset
278: # ext       [csv]          = output file format
279: # date      [Sys.Date()]   = date variable or value
280: # ###########################################################################
281:
282: BIRO_variance<-function(data,
283:                         var,
284:                         stratum=NULL,
285:                         out="out",
286:                         dirout=getwd(),
287:                         ext="csv",
288:                         date=Sys.Date()) {
289:
290:  myvar=var
291:  rm(var)
292:  retdata<-NULL
293:  for (k in 1:length(myvar)) {
294:   data<-data[is.na(data[,pmatch(c(myvar[k],stratum),names(data))])==FALSE,]
295:   tab<-NULL
296:   if (is.null(stratum)==FALSE) {
297:    data<-na.omit(data[,pmatch(c(myvar,stratum),names(data))])
298:    INDEX<-list()
299:    for (i in 1:length(stratum)) {
300:     INDEX[[i]]<-data[,pmatch(stratum[i],names(data))]
301:    }
302:    tabn<-
303:      as.data.frame(aggregate(data[,pmatch(myvar[k],names(data))],INDEX,length))
304:    tabsum<-
305:      as.data.frame(aggregate(data[,pmatch(myvar[k],names(data))],INDEX,sum))
306:    tabvar<-
307:      as.data.frame(aggregate(data[,pmatch(myvar[k],names(data))],INDEX,var))
308:    for (j in 1:length(stratum)) {
309:     names(tabsum)[j]<-stratum[j]
310:     names(tabn)[j]<-stratum[j]
311:     names(tabvar)[j]<-stratum[j]
312:    }
313:    names(tabsum)[length(stratum)+1]<-"sum"
314:    names(tabn)[length(stratum)+1]<-"n"
```

```
315:    names(tabvar)[length(stratum)+1]<-"var"
316:    tab<-merge(tabn,tabsum,by=stratum)
317:    tab<-merge(tab,tabvar,by=stratum)
318:   }
319:   if (is.null(stratum)==TRUE) {
320:    tab$mean<-mean(na.omit(data[,pmatch(myvar,names(data))]))
321:    tab$n<-length(na.omit(data[,pmatch(myvar,names(data))]))
322:    tab$var<-var(na.omit(data[,pmatch(myvar,names(data))]))
323:   }
324:   tab$variable=myvar[k]
325:   retdata<-rbind(retdata,as.data.frame(tab))
326:  }
327:  retdata$date=date
328:  write("Variance Data Frame Calculated",file="",sep="\n")
329:  write.csv(as.data.frame(retdata),paste(dirout,"/i",out,".3_1a",ext,sep=""),
330:            row.names = FALSE)
331:  write("Variance Data Frame Stored",file="",sep="\n")
332:  return(retdata)
333: }
334:
```

```
# ##############################################################################
#                                                                              #
# 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: 2008-11-23
# OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
# 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_histograms.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#   BIRO_barplot
```

```
45: #   BIRO_drawbars
46: #   BIRO_pie
47: #
48: #
49: # ##############################################################################
50:
51: # ##############################################################################
52: #  BIRO_barplot                                                                #
53: # ##############################################################################
54: # ------------------------
55: # Authors:
56: # Luca Rossi       <redsluke@gmail.com>
57: # Fabrizio Carinci <research@fabcarinci.net>
58: # Institutions: Serectrix snc
59: # Created: 2007-11-21
60: # Version: 2008-11-23
61: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
62: # Programming Language: R 2.8.0
63: # ------------------------
64: # DESCRIPTION
65: #
66: # BIRO_barplot prepares data for BIRO_drawbars
67: #
68: # DEPENDENCIES
69: #
70: #  R package: base
71: #
72: # ##############################################################################
73: # PARAMETERS(!=required)
74: #
75: # data      !              = dataset containing data
76: # var       !              = two or more vector with the values of the target
77: #                            variables, passed as a list (from the second there
78: #                            are stratum variables)
79: # labelvar  !              = the labels of the variable
80: # lev       !              = a list containing the vactor with che levels of the
81: #                            categorical variable (list)
82: # beside     [TRUE]        = a logical value. If FALSE, the columns of height
83: #                            are portrayed as stacked bars,
84: #                            and if TRUE the columns are portrayed as juxtaposed
85: #                            bars.
86: # number     [FALSE]       = a logical value. If TRUE values are printed over
87: #                            each bar, FALSE otherwise.
88: # perc       [TRUE]        = a logical value. If TRUE percentages are calculated.
89: # what       ["patients"]  = the subject of the representation (this is for the
```

```
  90: #                               label)
  91: # dirgraph  !                 = the path of the output (jpg,eps,png files)
  92: # namegraph !                 = the coded name of the output files
  93: # cex       !                 = A numerical value giving the amount by which
  94: #                               plotting text and symbols should be magnified
  95: #                               relative to the default. Note that some graphics
  96: #                               functions such as plot.default have an argument of
  97: #                               this name which multiplies this graphical parameter,
  98: #                               and some functions such as points accept a vector
  99: #                               of values which are recycled. Other uses will take
 100: #                               just the first value if a vector of length greater
 101: #                               than one is supplied.
 102: # ##############################################################################
 103:
 104:
 105: BIRO_barplot<-function(data,
 106:                        var,
 107:                        labelvar,
 108:                        lev=list(),
 109:                        beside=TRUE,
 110:                        number=FALSE,
 111:                        perc=TRUE,
 112:                        what="patients",
 113:                        namegraph,
 114:                        dirgraph,
 115:                        cex) {
 116:
 117:
 118: # calculates the number of each group of bars (conbination of stratum variable)
 119:
 120:  count=0
 121:  nrows=1
 122:  varlist<-list()
 123:  for (i in 1:length(var)) {
 124:   count=count+1
 125:   varlist[[count]]<-as.factor(data[,pmatch(var[count],names(data))])
 126:   levels(varlist[[count]])<-lev[[count]]
 127:    if (count>1) nrows<-nrows*nlevels(varlist[[count]])
 128:  }
 129:  # calculates the contingency table (for each combination)
 130:
 131:  tab<-table(varlist)
 132:  tab<-as.data.frame(tab)
 133:
 134:  names(tab)<-c(var,"Freq")
```

```
135:
136:  # create the splitvar (a variable that contains a different
137:  # value for each group of bars)
138:
139:  splitvar<-rep("",dim(tab)[1])
140:  for (i in 1:(dim(tab)[1])) {
141:   for (j in 2:(dim(tab)[2]-1)) {
142:    splitvar[i]<-paste(splitvar[i],as.character(tab[i,j]),sep=":")
143:   }
144:   splitvar[i]<-substr(splitvar[i],2,nchar(splitvar[i]))
145:  }
146:
147:  # Calculates sums of frequency for each group of bars (useful for calculating
148:  # percentages)
149:  sums<-aggregate(tab$Freq,
150:                  list(splitvar = splitvar),
151:                  sum)
152:
153:  names(sums)[2]<-"sum"
154:  sums$splitvar<-as.character(sums$splitvar)
155:  tab<-cbind(tab,splitvar)
156:
157:  #puts the data in the same data.frame
158:  tab<-merge(tab,sums,by.x = "splitvar", by.y = "splitvar")
159:  tab<-as.data.frame(tab)
160:  tab$id=centre_id
161:
162:  tab<-tab[,2:dim(tab)[2]]
163:  names(tab)[pmatch("Freq",names(tab))]<-"n"
164:
165:  #writecsv
166:  write.csv(tab,paste(dirdataout,"/i",namegraph,"d4_1a.csv",sep=""),
167:            row.names = FALSE)
168:
169:  write("Barplot data stored in csv file",file="")
170:  if (nchar(logfile)>0) {
171:   write("Barplot data stored in csv file",file=logfile,append=TRUE)
172:  }
173:
174:  #Draw graph
175:  BIRO_drawbars(tab=tab,
176:                labelvar=labelvar,
177:                lev=lev,
178:                beside=beside,
179:                number=number,
```

```
180:                perc=perc,
181:                what=what,
182:                namegraph,
183:                dirgraph,
184:                cex=cex)
185:
186: }
187:
188: # ##############################################################################
189: # BIRO_pie
190: # ##############################################################################
191: # ------------------------
192: # Authors:
193: # Luca Rossi       <redsluke@gmail.com>
194: # Fabrizio Carinci <research@fabcarinci.net>
195: # Institutions: Serectrix snc
196: # Created: 2007-11-21
197: # Version: 2008-11-23
198: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
199: # Programming Language: R 2.8.0
200: # ------------------------
201: # DESCRIPTION
202: #
203: # BIRO_pie draws Pie Plots
204: #
205: # DEPENDENCIES
206: #
207: #  R packages: grDevice, graphics, Cairo
208: #
209: # ##############################################################################
210: # PARAMETERS(!=required)
211: #
212: # formula        !          = a formula with <contitionated variable> ~
213: #                             <contitionating variables> (separated by +)
214: # data           !          = a data frame containing data
215: # labelvar       !          = the label of the variable
216: # labelstratum   !          = a vector containing string values (the labels
217: #                            of the conditionating variables)
218: # lev_var_cat   [NULL]      = a list containing levels of the conditionated
219: #                             variable (list)
220: # levclass       !          = a list containing the levels of the contitionating
221: #                             variables (list)
222: # dirgraph       !          = the path of the graphical output (jpg,eps,png files)
223: # namegraph      !          = the coded name of the output files
224: # dirdataout     !          = the path of the output csv file
```

```
225: # cex            !          = A numerical value giving the amount by which
226: #                              plotting text and symbols should be magnified
227: #                              relative to the default. Note that some graphics
228: #                              functions such as plot.default have an argument of
229: #                              this name which multiplies this graphical
230: #                              parameter, and some functions such as points accept
231: #                              a vector of values which are recycled.
232: #                              Other uses will take just the first value if a
233: #                              vector of length greater than one is supplied.
234: # ##############################################################################
235:
236: BIRO_pie<-function(formula=as.formula(),
237:                    data,
238:                    labelvar,
239:                    labelstratum,
240:                    lev_var_cat=NULL,
241:                    lev_class,
242:                    dirgraph,
243:                    namegraph,
244:                    dirdataout=dirdataout,
245:                    cex) {
246:
247:  temp<-data[,pmatch(all.vars(formula),names(data))]
248:  names(temp)<-all.vars(formula)
249:
250:  for (i in 1:(length(all.vars(formula))-1)) {
251:   Freqs<-as.data.frame(table(temp[,1],temp[,1+i]))
252:   nlevels<-nlevels(as.factor(Freqs[,2]))
253:   levels(Freqs[,2])<-lev_class
254:   write("Preparing to do SVG file",file="")
255:   if (nchar(logfile)>0) {
256:    write("Preparing to do SVG file",file=logfile,append=TRUE)
257:   }
258:
259:   file = paste(dirgraph,"/",namegraph,".svg",sep="")
260:   CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
261:   par("mfrow"=c(1,nlevels))
262:   par(cex=cex)
263:
264:   split.data<-split(Freqs,Freqs[2])
265:   for (i in (1:length(split.data))) {
266:    pie(split.data[i][[1]]$Freq,col=rainbow(length(lev_var_cat)),
267:        cex=cex,labels="")
268:    mtext(names(lev_class[i]),cex=cex)
269:   }
```

```
270:   legend("topright",col=rainbow(length(lev_var_cat)),pch=1,cex=cex,
271:          title=labelvar,legend=lev_var_cat)
272:   dev.off()
273:
274:   write("SVG file: Done",file="")
275:   if (nchar(logfile)>0) {
276:    write("SVG file: Done",file=logfile,append=TRUE)
277:   }
278:
279:   write("Preparing to do JPEG file",file="")
280:   if (nchar(logfile)>0) {
281:    write("Preparing to do JPEG file",file=logfile,append=TRUE)
282:   }
283:
284:   file = paste(dirgraph,"/",namegraph,".jpeg",sep="")
285:   jpeg(file, width = 11.33, height =4.84 , units="in",pointsize = 9,
286:        bg = "transparent",res=9)
287:
288:   par("mfrow"=c(1,nlevels))
289:   par(cex=cex)
290:
291:   split.data<-split(Freqs,Freqs[2])
292:
293:   for (i in (1:length(split.data))) {
294:    pie(split.data[i][[1]]$Freq,col=rainbow(length(lev_var_cat)),
295:        cex=cex,labels="")
296:    mtext(names(lev_class[i]),cex=cex)
297:
298:   }
299:
300:   legend("topright",col=rainbow(length(lev_var_cat)),pch=1,cex=cex,
301:          title=labelvar,legend=lev_var_cat)
302:   dev.off()
303:
304:   write("JPEG file: Done",file="")
305:   if (nchar(logfile)>0) {
306:    write("JPEG file: Done",file=logfile,append=TRUE)
307:   }
308:
309:   write("Preparing to do PNG file",file="")
310:   if (nchar(logfile)>0) {
311:    write("Preparing to do PNG file",file=logfile,append=TRUE)
312:   }
313:
314:   file = paste(dirgraph,"/",namegraph,".png",sep="")
```

```
315:   CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
316:   par("mfrow"=c(1,nlevels))
317:   par(cex=cex)
318:
319:   split.data<-split(Freqs,Freqs[2])
320:
321:   for (i in (1:length(split.data))) {
322:    pie(split.data[i][[1]]$Freq,col=rainbow(length(lev_var_cat)),
323:        cex=cex,labels="")
324:    mtext(names(lev_class[i]),cex=cex)
325:
326:   }
327:
328:   legend("topright",col=rainbow(length(lev_var_cat)),pch=1,cex=cex,
329:          title=labelvar,legend=lev_var_cat)
330:
331:   dev.off()
332:
333:   write("PNG file: Done",file="")
334:   if (nchar(logfile)>0) {
335:    write("PNG file: Done",file=logfile,append=TRUE)
336:   }
337:
338:   write("Preparing to do PDF file",file="")
339:   if (nchar(logfile)>0) {
340:    write("Preparing to do PDF file",file=logfile,append=TRUE)
341:   }
342:
343:   file = paste(dirgraph,"/",namegraph,".pdf",sep="")
344:
345:   pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
346:   par("mfrow"=c(1,nlevels))
347:   par(cex=cex)
348:
349:   split.data<-split(Freqs,Freqs[2])
350:
351:   for (i in (1:length(split.data))) {
352:    pie(split.data[i][[1]]$Freq,col=rainbow(length(lev_var_cat)),
353:        cex=cex,labels="")
354:    mtext(names(lev_class[i]),cex=cex)
355:
356:   }
357:
358:   legend("topright",col=rainbow(length(lev_var_cat)),pch=1,cex=cex,
359:          title=labelvar,legend=lev_var_cat)
```

```
  dev.off()

  write("PDF file: Done",file="")
  if (nchar(logfile)>0) {
   write("PDF file: Done",file=logfile,append=TRUE)
  }

 }


 names(split.data)<-c(all.vars(formula))
 write.csv(split.data,paste(dirdataout,"/",namegraph,".csv",sep=""),
           row.names = FALSE)

 write("Pieplot data stored in csv file",file="")
 if (nchar(logfile)>0) {
  write("Pieplot data stored in csv file",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)
 }

 return(split.data)

}
```

```
  1: # ############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: BIRO_location.r                                                      #
  5: #                                                                            #
#
  6: # ############################################################################
  7:
  8: # ------------------------
  9: # Authors:
 10: # Luca Rossi       <redsluke@gmail.com>
 11: # Fabrizio Carinci <research@fabcarinci.net>
 12: # Institutions: Serectrix snc
 13: # Created: 2007-11-21
 14: # Version: 2008-11-23
 15: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 16: # Programming Language: R 2.8.0
 17: # ------------------------
 18: # COPYRIGHT INFORMATION
 19: #
 20: # This file is free software; you can redistribute it and/or modify
 21: # it under the terms of the GNU General Public License as published by
 22: # the Free Software Foundation; either version 2, or (at your option)
 23: # any later version.
 24: #
 25: # This file is distributed in the hope that it will be useful,
 26: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 27: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 28: # GNU General Public License for more details.
 29: #
 30: # You should have received a copy of the GNU General Public License
 31: # along with this file; see the file COPYING.  If not, write to
 32: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 33: #
 34: # In short: you may use this file any way you like, as long as you
 35: # don't charge money for it, remove this notice, or hold anyone liable
 36: # for its results.
 37: #
 38: # BIRO_location.r is part of WP Statistical Engine of the BIRO Project
 39: # GPL Copyright, The BIRO Project
 40: #
 41: # ------------------------
 42: #
 43: # CONTENT
 44: #
```

```
45: #   BIRO_mean
46: #
47: # ##############################################################################
48:
49: # ##############################################################################
50: #  BIRO_mean                                                                   #
51: # ##############################################################################
52: # ------------------------
53: # Authors:
54: # Luca Rossi       <redsluke@gmail.com>
55: # Fabrizio Carinci <research@fabcarinci.net>
56: # Institutions: Serectrix snc
57: # Created: 2007-11-21
58: # Version: 2008-11-23
59: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
60: # Programming Language: R 2.8.0
61: # ------------------------
62: # DESCRIPTION
63: #
64: # BIRO_mean calculates conditionated mean
65: #
66: # DEPENDENCIES
67: #
68: #  R package: base
69: #
70: # ##############################################################################
71: # PARAMETERS(!=required)
72: #
73: # data    !               = input dataframe
74: # var     !               = one or more target variable corresponding to frame
75: #                           column names
76: #                           stats for more variables are appended, a variable
77: #                           name column is automatically produced
78: # stratum   [NULL]        = one or more strata variable corresponding to frame
79: #                           column names (including centre id)
80: #                           all variables preceding n are strata variables
81: # dirout    [getwd()]     = output location
82: # out       [mean]        = name of output csv dataset
83: # ext       [csv]         = output file format
84: # date      [Sys.Date()]  = date variable or value
85: # ##############################################################################
86:
87: BIRO_mean<-function(data,
88:                     var,
89:                     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")
 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))
}
```

```
  1: # ############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: biro_se_report.r                                                         #
  5: #                                                                            #
#
  6: # ############################################################################
  7:
  8: # ------------------------
  9: # Authors:
 10: # Luca Rossi       <redsluke@gmail.com>
 11: # Fabrizio Carinci <research@fabcarinci.net>
 12: # Institutions: Serectrix snc
 13: # Created: 2008-11-21
 14: # Version: 2008-10-23
 15: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 16: # Programming Language: R 2.8.0
 17: # ------------------------
 18: # COPYRIGHT INFORMATION
 19: #
 20: # This file is free software; you can redistribute it and/or modify
 21: # it under the terms of the GNU General Public License as published by
 22: # the Free Software Foundation; either version 2, or (at your option)
 23: # any later version.
 24: #
 25: # This file is distributed in the hope that it will be useful,
 26: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 27: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 28: # GNU General Public License for more details.
 29: #
 30: # You should have received a copy of the GNU General Public License
 31: # along with this file; see the file COPYING.  If not, write to
 32: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 33: #
 34: # In short: you may use this file any way you like, as long as you
 35: # don't charge money for it, remove this notice, or hold anyone liable
 36: # for its results.
 37: #
 38: # biro_se_report.r is part of WP Statistical Engine of the BIRO Project
 39: # GPL Copyright, The BIRO Project
 40: #
 41: # ------------------------
 42: #
 43: # CONTENT
 44: #
```

```
45: #   BIRO_report
46: #   open_tex
47: #   new_chapter
48: #   new_section
49: #   new_sub_section
50: #   import_png
51: #   close_tex
52: #
53: # ##############################################################################
54:
55:
56: # ##############################################################################
57: #  BIRO_report                                                                 #
58: # ##############################################################################
59: # ------------------------
60: # Authors:
61: # Luca Rossi       <redsluke@gmail.com>
62: # Fabrizio Carinci <research@fabcarinci.net>
63: # Institutions: Serectrix snc
64: # Created: 2008-11-21
65: # Version: 2008-10-23
66: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
67: # Programming Language: R 2.8.0
68: # ------------------------
69: # DESCRIPTION
70: #
71: # BIRO_report manages the graphical outputs and tables for each indicator
72: #
73: # DEPENDENCIES
74: #
75: #  R package: base
76: #
77: # ##############################################################################
78: # PARAMETERS(!=required)
79: #
80: # ind    !                = string. the index of the indicator
81: # var                     = string. the name of the countinous conditionated
82: #                           variable
83: # var_cat     !           = string. the name of the categorical conditionated
84: #                           variable
85: # class                   = a vector containing the names of the conditionating
86: #                           variables
87: # lev_var_cat             = levels of the categorical conditionated variable
88: # lev_class               = levels of the categorical conditionating variable
89: # lab_var                 = a list for the labels of the categorical conditionated
```

```
 90: #                                variable
 91: # lab_var_cat                  = a list for the labels of the categorical conditionating
 92: #                                variable
 93: # tab                          = a boolean (0,1) 1 if output table is required
 94: # chisq                        = a vector containing the name of variable to calculate
 95: #                                Cochran-Mantel-Haenszel Chi-Squared Test for Count
 96: #                                Data or Pearson Chi-Squared Test
 97: # tabside                      = "H" for orizontal, "V" for vertical
 98: # tabperc                      = logical. TRUE if you want percentages in the tables
 99: # tabwidth                     = the width of table in the PDF document
100: # bar                          = a boolean (0,1) 1 if output barplot is required
101: # barbeside     [TRUE]          = a logical value. If FALSE, the columns of height
102: #                                are portrayed as stacked bars,
103: #                                and if TRUE the columns are portrayed as juxtaposed
104: #                                bars.
105: # barnumber     [FALSE]         = a logical value. If TRUE values are printed over
106: #                                each bar, FALSE otherwise.
107: #
108: # barperc       [TRUE]         = a logical value. If TRUE percentages are calculated
109: #                                in the barplot
110: # box                          = a boolean (0,1) 1 if output boxplot is required
111: # notch      !                 = logical. if notch is TRUE, a notch is drawn in
112: #                                each side of the boxes. If the notches of two
113: #                                plots do not overlap this is \221strong evidence\222 that
114: #                                the two means differ (Chambers et al., 1983, p. 62).
115: #                                See boxplot.stats for the calculations used.
116: # lines                        = a boolean (0,1) 1 if output trend is required
117: # printlines                   = logical. TRUE if you want to print
118: # varlines                    = the name of
119: # timelines
120: # pie                          = a boolean (0,1) 1 if output pieplot is required
121: # cex                          = expansion factor for words and number printed
122: # struct                       = if you want to create the structure of HTML pages
123: # catcol                       = the number of categorical variables in the data.frame
124: #                                 that you want to print
125: # stand                        = if you want to call BIRO_standardize function
126: # outcome                      = the name of outcome variable (for BIRO_standardize)
127: # cov                          = a vector containing the names of covariates
128: # factcov
129: # levstd                       = a list contaning levels of depvar and indepvar
130: # strata                       = the name of the variable of stratification
131: # factindep                    = a vector containig logistic values TRUE if the i-th independent
132: #                                variable is a factor, FALSE otherwise
133: # per                          = the moltiplication factor of rates
134: # pop                          = the data.frame containing the population data
```

```
135: # caption                 =  caption to print after the table (BIRO_standardize)
136: # printrate               = logical. TRUE if you want to print rate
137: # patmap                  = a boolean (0,1) 1 if output map is required
138: # shapefile               = the name of the SpatialPolygonsDataFrame
139: # mapvar                  = the variable that must be represented
140: # maplev                  = the levels of mapvar
141: # maplab                  = the label of mapvar
142: # mapstat                 = only 2 values "freq" for Absolute Frequencies
143: # shapearea_id            = the name of area_id variable in the SpatialPolygonsDataFrame
144: # valuesarea_id           = the name of area_id variable in the data data.frame
145: # texfile                 = the path of the texfile
146: # dirtables               = the path of HTML tables directory
147: # dircsv                  = the path of output .csv directory
148: # dirgraph                = the path of output graphs directory
149: # dirgraphlatex           = the relative path of output graphs directory (for Latex)
150:
151: # ##############################################################################
152:
153: BIRO_report<-function(ind,
154:                       title="title",
155:                       var=NULL,
156:                       var_cat=NULL,
157:                       class=NULL,
158:                       lev_var_cat=NULL,
159:                       lev_class=NULL,
160:                       lab_var=NULL,
161:                       lab_var_cat=NULL,
162:                       lab_class=NULL,
163:                       tab=1,
164:                       chisq=class,
165:                       tabside="H",
166:                       tabperc=TRUE,
167:                       tabwidth=0,
168:                       barbeside=TRUE,
169:                       barnumber=TRUE,
170:                       barperc=TRUE,
171:                       box=0,
172:                       bar=0,
173:                       lines=0,
174:                       printlines=1,
175:                       trellis=0,
176:                       pie=0,
177:                       varlines=0,
178:                       timelines=0,
179:                       notch=FALSE,
```

```
180:                        texfile=texfile,
181:                        dirtables=dirtables,
182:                        dircsv=dirdataout,
183:                        dirgraph=dirgraph,
184:                        dirgraphlatex=dirgraphlatex,
185:                        cex=1,
186:                        struc=TRUE,
187:                        catcol=1,
188:                        stand=0,
189:                        outcome=NULL,
190:                        cov=NULL,
191:                        factcov=NULL,
192:                        levstd,
193:                        strata=NULL,
194:                        sucvalue=1,
195:                        per=100,
196:                        pop=NULL,
197:                        caption="",
198:                        printrate=TRUE,
199:                        patmap=0,
200:                        shapefile,
201:                        mapvar,
202:                        maplev,
203:                        maplab,
204:                        mapstat="freq",
205:                        shapearea_id,
206:                        valuesarea_id) {
207:
208:  if (is.null(chisq)==FALSE){
209:   chisq<-na.omit(class[1:2])
210:  }
211:
212:
213:  #print title in .tex file
214:  new_sub_section(title=title,texfile=texfile)
215:
216:  #create html indicator
217:  file.copy(paste(dirse,"/source/html/","layout_open.html",sep=""),
218:            paste(dirhtml,"/",ind,".html",sep=""),
219:            overwrite=TRUE)
220:
221:  cat("<body>",
222:      file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
223:  cat("<table class=\"reportheader\" border=\"0\">",
224:      file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
```

```
  225:  cat("<tbody>",
  226:      file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  227:  cat(" <tr>",
  228:      file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  229:  cat("  <td><img src=\"../images/biro-logo01.jpg\"></td>",
  230:       file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  231:  cat(" </tr>",
  232:      file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  233:  cat("</tbody>",
  234:      file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  235:  cat("</table>",
  236:      file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  237:
  238:  cat("<table class=\"reportheader\" border=\"0\">",file=paste(dirhtml,"/",ind,".html",sep=""),fill =
TRUE,append=TRUE)
  239:  cat("<tbody>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  240:  cat(" <tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  241:  cat(" <td class=\"reportheader\"><a href=\"../report.html\"> BIRO Report </a> site: ",centre_id,", ",
  242:      format(as.Date(launchtime,format="%d%m%y%H%M%S"),format="%d %B %Y"),
  243:      "</td>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  244:  cat(" </tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  245:  cat(" <tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  246:  cat("  <td class=\"reportheader\">Reference date: ",format(checkdate,format="%d/%m/%y"),"</td>"
,file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  247:  cat(" </tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  248:  cat(" <tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  249:  cat("  <td class=\"reportheader\">Parameter: ",title,"</td>",file=paste(dirhtml,"/",ind,".html",sep=""),fill =
TRUE,append=TRUE)
  250:  cat(" </tr>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  251:  cat("</tbody>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  252:  cat("</table>",file=paste(dirhtml,"/",ind,".html",sep=""),fill = TRUE,append=TRUE)
  253:
  254:
  255:  # insert link in report.html
  256:  cat("<a href=\" html/",ind,".html\">",title,"</a><br>",sep="",
  257:      file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
  258:  if (struc==TRUE) {
  259:   dirstruc<-paste(dirse,"/output/reports/structure",sep="")
  260:   dir.create(dirstruc,showWarnings = FALSE)
  261:   sthtml<-paste(dirstruc,"/",ind,".html",sep="")
  262:
  263:   cat("",file=sthtml,fill = FALSE,append=FALSE)
  264:
  265:   assign("struc",struc,envir=.GlobalEnv)
  266:   assign("sthtml",sthtml,envir=.GlobalEnv)
```

```
267:  }
268:
269:  # Load Coorte data
270:
271:  namedata<-paste("data",ind,sep="")
272:
273:  if (lines==1) {
274:   #data for line plot
275:   namedatal<-paste("data",ind,"l",sep="")
276:   datal<-get(namedatal)
277:  }
278:
279:  data<-get(namedata)
280:   if (stand==1) {
281:    datastd<-BIRO_standardize(data=data,
282:                              outcome=outcome,
283:                              cov=cov,
284:                              factcov=factcov,
285:                              strata=strata,
286:                              lev=levstd,
287:                              sucvalue=sucvalue,
288:                              per=per,
289:                              pop=pop,
290:                              csvindex=ind,
291:                              dirdataout=dircsv,
292:                              filename=ind,
293:                              dirgraph=dirgraph,
294:                              dirgraphlatex=dirgraphlatex,
295:                              dirtables=dirtables,
296:                              texfile=texfile,
297:                              caption=caption,
298:                              chisq=chisq)
299:
300:   if (printrate==FALSE) {
301:    datastd<-datastd[,-pmatch("Denom",names(datastd))]
302:    datastd<-datastd[-dim(datastd)[2]]
303:   }
304:   mycov<-NULL
305:   for (i in 1:length(cov)) {
306:    mycov[i]<-paste(unlist(strsplit(cov[i],"_")),collapse="")
307:   }
308:
309:   catcol<-pmatch(mycov,names(datastd))
310:   BIRO_df2html(data=datastd,
311:                htmlfile=paste(dirtables,"/i",ind,"1_3a.html",sep=""),
```

```
312:                catcol=na.omit(catcol))
313:   file.append(paste(dirhtml,"/",ind,".html",sep=""),
314:              paste(dirtables,"/i",ind,"1_3a.html",sep=""))
315:
316:   for (i in 1:length(names(datastd))) {
317:    names(datastd)[i]<-paste(unlist(strsplit(names(datastd)[i],"\\_")),collapse="")
318:   }
319:   if (is.null(chisq)==TRUE) {
320:    latex(datastd,file=texfile,append=TRUE,table.env=FALSE)
321:   }
322:  }
323:
324:  if (is.null(var_cat)==FALSE) {
325:   if (tab==1) {
326:   # creating table
327:   # creating input parameters
328:   firstvar<-substr(var_cat,1,nchar(var_cat)-2)
329:   tabvars<-c(firstvar,class)
330:   lev<- vector("list",(length(class)+1))
331:   for (i in 1:(length(class)+1)) {
332:    if (i==1) {
333:     lev[[1]]<-lev_var_cat
334:    }
335:    if (i>1) {
336:     lev[[i]]<-lev_class[[i-1]]
337:    }
338:   }
339:   names(lev)<-letters[1:(length(class)+1)]
340:   tabcaption=paste(lab_var_cat," (by ",paste(lab_class,collapse=","),")",
341:                    sep="")
342:   if (tabside=="H") side=0
343:   if (tabside=="V") side=1
344:   BIRO_table_compute(data=data,
345:                      var=c(var_cat,class),
346:                      FirstLab=lab_var_cat,
347:                      lev=lev,
348:                      texfile=texfile,
349:                      codtab=ind,
350:                      dirtables=dirtables,
351:                      dircsv=dirdataout,
352:                      caption=tabcaption,
353:                      side=side,
354:                      width=tabwidth,
355:                      perc=tabperc,
356:                      ind=ind,
```

```
357:                       chisq=chisq)
358:  }
359:
360:  if (typeof(lev_class)!="list") {
361:   levlist<-list(lev_class)
362:  }
363:  if (typeof(lev_class)=="list") {
364:   levlist<-lev_class
365:  }
366:  if (bar==1) {
367:   # barplots
368:   write(paste("Preparing To Do Barplots...Data Frame: ",ind,sep=""),
369:         file="",sep="\n")
370:   noNAclass<-na.omit(class)
371:   noNAlev_class<-list()
372:   j=0
373:   for (i in 1:length(lev_class)) {
374:    if (is.na(lev_class[[i]])==FALSE) {
375:     j=j+1
376:     noNAlev_class[[j]]<-lev_class[[i]]
377:    }
378:   }
379:   for (i in 1:length(noNAclass)) {
380:    namegraph<-paste(ind,letters[i],sep="")
381:    if (length(class)==1) {
382:     BIRO_barplot(data=data,
383:                  var=c(var_cat,na.omit(noNAclass)),
384:                  labelvar=c(lab_var_cat,lab_class),
385:                  lev=list(a=as.vector(lev_var_cat),b=noNAlev_class[[1]]),
386:                  beside=barbeside,
387:                  number=TRUE,
388:                  perc=TRUE,
389:                  namegraph=paste(ind,letters[i],sep=""),
390:                  dirgraph=dirgraph,
391:                  cex=cex)
392:    }
393:    if (length(noNAclass)>1) {
394:     BIRO_barplot(data=data,
395:                  var=c(var_cat,na.omit(noNAclass[i])),
396:                  labelvar=c(lab_var_cat,lab_class[i]),
397:                  lev=list(a=as.vector(lev_var_cat),b=noNAlev_class[[i]]),
398:                  beside=barbeside,
399:                  number=TRUE,
400:                  perc=TRUE,
401:                  namegraph=paste(ind,letters[i],sep=""),
```

```
402:                   dirgraph=dirgraph,
403:                   cex=cex)
404:    }
405:    import_png(namegraph=paste("i",namegraph,"g4_1a.png",sep=""),
406:               caption=paste("Barplot: ",lab_var_cat,
407:               " (by ",paste(lab_class[i],collapse=","),")",
408:               sep=""),
409:               width=1,
410:               dirgraph=dirgraphlatex,
411:               texfile=texfile)
412:    import_png_html(image=paste("../graphs/","i",namegraph,"g4_1a.png",sep=""),
413:                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
414:                    caption=paste("Barplot: ",lab_var_cat,
415:                    " (by ",paste(lab_class[i],collapse=","),")",
416:                    sep=""))
417:   }
418:   write("Barplots Done",file="",sep="\n")
419:  }
420:  # pie charts
421:  noNAlev_class<-list()
422:  j=0
423:  for (i in 1:length(lev_class)) {
424:   if (is.na(lev_class[[i]])==FALSE) {
425:    j=j+1
426:    noNAlev_class[[j]]<-lev_class[[i]]
427:   }
428:  }
429:  if (pie==1) {
430:   write(paste("Preparing To Do Pieplots...Data Frame: ",ind,sep=""),
431:         file="",sep="\n")
432:   noNAclass<-na.omit(class)
433:   for (i in 1:length(noNAclass)) {
434:    formula<-as.formula(paste(var_cat,"~",noNAclass[i],sep=""))
435:    namegraph<-paste("pie",ind,letters[i],sep="")
436:    BIRO_pie(formula=formula,
437:             data=data,
438:             labelvar=lab_var_cat,
439:             lev_var_cat=lev_var_cat,
440:             lev_class=noNAlev_class[[i]],
441:             labelstratum=lab_class[i],
442:             dirdataout=dirdataout,
443:             namegraph=namegraph,
444:             dirgraph=dirgraph,
445:             cex=cex)
446:    import_png(namegraph=paste(namegraph,".png",sep=""),
```

```
447:                caption=paste("Pieplot: ",lab_var_cat,
448:                " (by ",paste(lab_class[i],collapse=","),")",
449:                sep=""),
450:                width=1,
451:                dirgraph=dirgraphlatex,
452:                texfile=texfile)
453:     import_png_html(image=paste("../graphs/",namegraph,".png",sep=""),
454:                     htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
455:                     caption=paste("Pieplot: ",lab_var_cat,
456:                     " (by ",paste(lab_class[i],collapse=","),")",
457:                     sep=""))
458:     }
459:     write("Pieplots Done",file="",sep="\n")
460:    }
461:    if (patmap==1) {
462:     if (maploaded!=shapefile) {
463:      maploaded=shapefile
464:      assign(maploaded,shapefile,envir=.GlobalEnv)
465:      mapfile<-readShapePoly(paste(dirmap,"/",shapefile,".shp",sep=""))
466:      mapfile<-mapfile[mapfile$ID %in% unique(patient$are),]
467:      assign("mapfile",mapfile,,envir=.GlobalEnv)
468:     }
469:     BIRO_patmap(ind=ind,
470:                 shape=mapfile,
471:                 data=data,
472:                 var=mapvar,
473:                 lev=lev_class,
474:                 stat=mapstat,
475:                 labvar=maplab,
476:                 legendpos="topright",
477:                 dirgraph=dirgraph,
478:                 namegraph=ind,
479:                 shapearea_id=shapearea_id,
480:                 valuesarea_id=valuesarea_id)
481:    }
482:   }
483:   # continuous var
484:   if (is.null(var)==FALSE) {
485:    if (lines==1) {
486:     write(paste("Preparing To Do Lines plots...Data Frame: ",ind,sep=""),
487:           file="",sep="\n")
488:     for (i in 1:length(class)) {
489:      namegraph<-paste("i",ind,letters[i],sep="")
490:      myvar=list(y=as.numeric(as.character(
491:                 datal[,pmatch(varlines,names(datal))])),
```

```
492:                 times=as.numeric(as.character(
493:                 datal[,pmatch(timelines,names(datal))])),
494:                 strata=as.factor(datal[,pmatch(class[i],names(datal))]))
495:     BIRO_lines(list(y=as.numeric(as.character(
496:                      datal[,pmatch(varlines,names(datal))])),
497:                 times=as.numeric(as.character(
498:                      datal[,pmatch(timelines,names(datal))])),
499:                 strata=as.numeric(as.character(
500:                      datal[,pmatch(class[i],names(datal))]))),
501:                 descvar=lab_class[i],
502:                 lev=names(lev_class[[i]]),
503:                 namegraph=paste(ind,letters[i],sep=""),
504:                 dirgraph=dirgraph)
505:
506:     if (printlines==1) {
507:      import_png(namegraph=paste("i",ind,letters[i],"g4_5a",sep=""),
508:                  caption=paste("Linesplot: ",
509:                  lab_var_cat,
510:                  " (by ",paste(lab_class[i],collapse=","),")",sep=""),
511:                  width=1,
512:                  dirgraph=dirgraphlatex,
513:                  texfile=texfile)
514:
515:      import_png_html(image=paste("../graphs/i",ind,letters[i],"g4_5a.png",sep=""),
516:                      htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
517:                      caption=paste("Linesplot: ",
518:                      lab_var_cat,
519:                      " (by ",paste(lab_class[i],collapse=","),")",
520:                      sep=""))
521:
522:     }
523:    }
524:    write("Lines plots Done",file="",sep="\n")
525:
526:   }
527:   # boxplots
528:   if (box==1) {
529:    write(paste("Preparing To Do Boxplots...Data Frame: ",ind,sep=""),
530:          file="",sep="\n")
531:    for (i in 1:length(class)) {
532:     namegraph<-paste(ind,letters[i],sep="")
533:     BIRO_fd(date=Sys.Date(),
534:             data=data,
535:             var=var[1],
536:             stratum=class[[i]],
```

```
537:             typeround=c("integer"),
538:             codtab=paste(ind,letters[i],sep=""),
539:             dircsv=dircsv)
540:
541:     BIRO_boxplot(data=data,
542:                  var=var,
543:                  lev=levlist[[i]],
544:                  notch=notch,
545:                  stratum=class[[i]],
546:                  namegraph=namegraph,
547:                  dirgraph=dirgraph)
548:     import_png(namegraph=paste("i",ind,letters[i],"g4_3a.png",sep=""),
549:                caption=paste("Boxplot: ",
550:                        lab_var_cat,
551:                        " (by ",paste(lab_class[i],collapse=","),")",sep=""),
552:                width=1,
553:                dirgraph=dirgraphlatex,
554:                texfile=texfile)
555:     import_png_html(image=paste("../graphs/i",ind,letters[i],"g4_3a.png",sep=""),
556:                     htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
557:                     caption=paste("Boxplot: ",
558:                             lab_var_cat,
559:                             " (by ",paste(lab_class[i],collapse=","),")",
560:                             sep=""))
561:    }
562:    write("Boxplots Done",file="",sep="\n")
563:   }
564:   if (trellis==1) {
565:    BIRO_fd(date=Sys.Date(),
566:            data=data,
567:            var=var[1],
568:            stratum=class,
569:            typeround=c("integer"),
570:            codtab=paste(ind,sep=""),
571:            dircsv=dircsv)
572:    write(paste("Preparing To Do Histogram Trellis...Data Frame: ",ind,sep=""),
573:          file="",sep="\n")
574:    BIRO_histtrellis(data=data,
575:                     var=var,
576:                     strata=class,
577:                     labvar=lab_var,
578:                     labclass=lab_class,
579:                     lev_class=lev_class,
580:                     dirout=dircsv,
581:                     ext="csv",
```

```
582:                     date=Sys.Date(),
583:                     namegraph=ind,
584:                     dirgraph=dirgraph)
585:
586:    import_large_png(namegraph=paste("i",ind,"g4_2a.png",sep=""),
587:                     caption=paste("Trellis density plot: ",lab_var_cat,
588:                             " (by ",paste(lab_class,collapse=","),")",sep=""),
589:                     dirgraph=dirgraphlatex,
590:                     texfile=texfile)
591:    import_png_html(image=paste("../graphs/i",ind,"g4_2a.png",sep=""),
592:                    htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
593:                    caption=paste("Trellis density plot: ",lab_var_cat,
594:                            " (by ",paste(lab_class,collapse=","),")",sep=""))
595:    write("Histogram Trellis Done",file="",sep="\n")
596:    write(paste("Preparing To Do Boxplot Trellis...Data Frame: ",ind,sep=""),
597:          file="",sep="\n")
598:    BIRO_boxtrellis(data=data,
599:                    var=var,
600:                    strata=class,
601:                    labvar=lab_var,
602:                    labclass=lab_class,
603:                    lev_class=lev_class,
604:                    dirout=dircsv,
605:                    ext="csv",
606:                    date=Sys.Date(),
607:                    namegraph=ind,
608:                    dirgraph=dirgraph)
609:
610:    import_large_png(namegraph=paste("i",ind,"g4_3a.png",sep=""),
611:                     caption=paste("Trellis boxplot: ",lab_var_cat," (by ",
612:                                   paste(lab_class,collapse=","),")",sep=""),
613:                     dirgraph=dirgraphlatex,
614:                     texfile=texfile)
615:     import_png_html(image=paste("../graphs/i",ind,"g4_3a.png",sep=""),
616:                     htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
617:                     caption=paste("Trellis boxplot: ",lab_var_cat," (by ",
618:                                   paste(lab_class,collapse=","),")",sep=""))
619:
620:    write("Boxplot Trellis Done",file="",sep="\n")
621: #    namegraph<-paste("dt",ind,sep="")
622: #
623: #    write(paste("Preparing To Do Density Trellis...Data Frame: ",ind,sep=""),
624: #          file="",sep="\n")
625: #
626: #    BIRO_densitytrellis(data=data,
```

```
627: #                         var=var,
628: #                         strata=class,
629: #                         labvar=lab_var,
630: #                         labclass=lab_class,
631: #                         lev_class=lev_class,
632: #                         dirout=dircsv,
633: #                         ext="csv",
634: #                         date=Sys.Date(),
635: #                         namegraph=namegraph,
636: #                         dirgraph=dirgraph)
637: #
638: #
639: #    import_large_png(namegraph=paste(namegraph,".png",sep=""),
640: #                       caption=paste("Trellis density plot: ",lab_var_cat," (by ",
641: #                                     paste(lab_class,collapse=","),")",sep=""),
642: #                       dirgraph=dirgraphlatex,
643: #                       texfile=texfile)
644: #
645: #    import_png_html(image=paste("../graphs/",namegraph,".png",sep=""),
646: #                      htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
647: #                       caption=paste("Trellis density plot: ",lab_var_cat," (by ",
648: #                                     paste(lab_class,collapse=","),")",sep=""))
649: #
650: #
651: #    write("Density Trellis Done",file="",sep="\n")
652:
653:   }
654:  }
655:
656:  if (is.null(var)==TRUE & is.null(var_cat)==TRUE & stand==0) {
657:   BIRO_df2html(data=data,
658:                htmlfile=paste(dirtables,"/i",ind,"1_3a.html",sep=""),
659:                catcol=catcol)
660:   file.append(paste(dirhtml,"/",ind,".html",sep=""),
661:               paste(dirtables,"/i",ind,"1_3a.html",sep=""))
662:
663:   for (i in 1:length(names(data))) {
664:    names(data)[i]<-paste(unlist(strsplit(names(data)[i],"_")),collapse="\\_")
665:   }
666:
667:   if (is.na(pmatch("centre\\_id",names(data)))==FALSE) {
668:    for (i in 1:dim(data)[1]) {
669:     indexcentre<-pmatch("centre\\_id",names(data))
670:     data[i,indexcentre]<-paste(unlist(strsplit(as.character(data[i,indexcentre]),"_")),collapse="\\_")
671:    }
```

```
672:   }
673:   latex(data,file=texfile,append=TRUE,table.env=FALSE)
674:  }
675:  write(paste("Indicator:",title,"Done"),file="")
676:  if (nchar(logfile)>0) {
677:   write(paste("Indicator:",title,"Done"),file=logfile,append=TRUE)
678:  }
679:  file.append(paste(dirhtml,"/",ind,".html",sep=""),
680:              paste(dirse,"/source/html/","layout_close.html",sep=""))
681: }
682: ##############################################################################
683: #                                                                            #
684: ##############################################################################
685:
686: BIRO_report_toc<-function(title="") {
687:  cat(title,"<br>",sep="",
688:  file=paste(dirreport,"/","report.html",sep=""),fill = TRUE,append=TRUE)
689:  write(paste("Indicator:",title," Not Done"),file="")
690:  if (nchar(logfile)>0) {
691:   write(paste("Indicator:",title,"Not Done"),file=logfile,append=TRUE)
692:  }
693: }
694:
695:
696: ##############################################################################
697: #  open_tex                                                                  #
698: ##############################################################################
699: # ------------------------
700: # Authors:
701: # Luca Rossi       <redsluke@gmail.com>
702: # Fabrizio Carinci <research@fabcarinci.net>
703: # Institutions: Serectrix snc
704: # Created: 2008-11-21
705: # Version: 2008-10-23
706: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
707: # Programming Language: R 2.8.0
708: # ------------------------
709: # DESCRIPTION
710: #
711: # open_tex manages the creation and style of the tex file
712: #
713: # DEPENDENCIES
714: #
715: #  R package: base
716: #
```

```
717: ##############################################################################
718: # PARAMETERS(!=required)
719: #
720: #title     [""] = a string containing the title of the report
721: #authors   [""] = a vector of strings containing the authors of the report
722: #
723: #footnote  [""] = a string containing the footnote of the report
724: #
725: #time      [""] = a string containing the date of the report
726: #
727: #logo      [""] = a string containing the path of the logo (PNG format)
728: #layout     !   = a string containing the name of the layout .tex file
729: #final      !   = a string containing the name of the report .tex file
730: ##############################################################################
731:
732: open_tex<-function(title="",
733:                    authors="",
734:                    footnote="",
735:                    time="",
736:                    logo="",
737:                    layout,
738:                    final) {
739:
740:  x<-FALSE
741:  if (is.null(layout)) {
742:   print(layout)
743:   stop("layout not recognized")
744:  }
745:
746:  if (is.null(final)) {
747:   print(final)
748:   stop("final not recognized")
749:  }
750:  if (file.exists(layout)==FALSE) {
751:   print(layout)
752:   stop("layout not exists")
753:  }
754:
755:  if (file.exists(layout)==TRUE) {
756:   x<-  file.copy(layout,final,overwrite=TRUE)
757:  }
758:  invisible(x)
759: }
760:
761: ##############################################################################
```

```
762: #  new_chapter                                                              #
763: ###############################################################################
764: # ------------------------
765: # Authors:
766: # Luca Rossi       <redsluke@gmail.com>
767: # Fabrizio Carinci <research@fabcarinci.net>
768: # Institutions: Serectrix snc
769: # Created: 2008-11-21
770: # Version: 2008-10-23
771: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
772: # Programming Language: R 2.8.0
773: # ------------------------
774: # DESCRIPTION
775: #
776: # new_chapter creates a new chapter in the report PDF file
777: #
778: # DEPENDENCIES
779: #
780: #  R package: base
781: #
782: ###############################################################################
783: # PARAMETERS(!=required)
784: #
785: #title        !   = a string containing the title of the chapter
786: #texfile      !   = a string containing the name of the report .tex file
787: ###############################################################################
788:
789: new_chapter<-function(title,
790:                       texfile) {
791:
792:  title<-paste(strsplit(title,"<"),collapse="\\<")
793:  title<-paste(strsplit(title,"%"),collapse="\\%")
794:  title<-paste(strsplit(title,">"),collapse="\\>")
795:  cat("\\chapter{",title,"}",file=texfile,fill = TRUE,append=TRUE)
796: }
797:
798: ###############################################################################
799: #  new_section                                                              #
800: ###############################################################################
801: # ------------------------
802: # Authors:
803: # Luca Rossi       <redsluke@gmail.com>
804: # Fabrizio Carinci <research@fabcarinci.net>
805: # Institutions: Serectrix snc
806: # Created: 2008-11-21
```

```
807: # Version: 2008-10-23
808: # OS: Windows XP/Vista
809: # Programming Language: R 2.8.0, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
810: # ------------------------
811: # DESCRIPTION
812: #
813: # new_section creates a new section in the report PDF file
814: #
815: # DEPENDENCIES
816: #
817: #  R package: base
818: #
819: ##############################################################################
820: # PARAMETERS(!=required)
821: #
822: #title        !   = a string containing the title of the section
823: #texfile      !   = a string containing the name of the report .tex file
824: ##############################################################################
825:
826: new_section<-function(title,
827:                       texfile) {
828:
829:  title<-paste(strsplit(title,"<"),collapse="\\<")
830:  title<-paste(strsplit(title,"%"),collapse="\\%")
831:  title<-paste(strsplit(title,">"),collapse="\\>")
832:  cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
833:  cat("\\section*{",title,"}",file=texfile,fill = TRUE,append=TRUE)
834: }
835:
836: ##############################################################################
837: #  new_sub_section                                                           #
838: ##############################################################################
839: # ------------------------
840: # Authors:
841: # Luca Rossi       <redsluke@gmail.com>
842: # Fabrizio Carinci <research@fabcarinci.net>
843: # Institutions: Serectrix snc
844: # Created: 2008-11-21
845: # Version: 2008-10-23
846: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
847: # Programming Language: R 2.8.0
848: # ------------------------
849: # DESCRIPTION
850: #
851: # new_sub_section creates a new sub section in the report PDF file
```

```
852: #
853: # DEPENDENCIES
854: #
855: #  R package: base
856: #
857: ##############################################################################
858: # PARAMETERS(!=required)
859: #
860: #title        !   = a string containing the title of the sub section
861: #texfile      !   = a string containing the name of the report .tex file
862: ##############################################################################
863:
864: new_sub_section<-function(title,texfile) {
865:
866:  title<-paste(strsplit(title,"<"),collapse="\\<")
867:  title<-paste(strsplit(title,"%"),collapse="\\%")
868:  title<-paste(strsplit(title,">"),collapse="\\>")
869:
870:
871:  cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
872:  cat("\\subsection*{",title,"}",file=texfile,fill = TRUE,append=TRUE)
873:  cat("Reference Date: ",format(checkdate,format="%d/%m/%Y"),file=texfile,fill = TRUE,append=TRUE)
874:
875: }
876:
877: ##############################################################################
878: #  include_tex                                                               #
879: ##############################################################################
880: # ------------------------
881: # Authors:
882: # Luca Rossi       <redsluke@gmail.com>
883: # Fabrizio Carinci <research@fabcarinci.net>
884: # Institutions: Serectrix snc
885: # Created: 2008-11-21
886: # Version: 2008-10-23
887: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
888: # Programming Language: R 2.8.0
889: # ------------------------
890: # DESCRIPTION
891: #
892: # include_tex include a tex file into another
893: #
894: # DEPENDENCIES
895: #
896: #  R package: base
```

```
897: #
898: ##############################################################################
899: # PARAMETERS(!=required)
900: #
901: #incfile      !   = a string containing the name of the report .tex file
902: #texfile      !   = a string containing the name of the report .tex file
903: #caption      !   = a string containing the caption
904: ##############################################################################
905:
906: include_tex<-function(incfile,
907:                       texfile,
908:                       caption) {
909:
910:  cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
911:  cat("\\parbox[2]{",1,"\\textwidth}{",file=texfile,fill = TRUE,append=TRUE)
912:  cat("\\include{",incfile,"}",file=texfile,fill = TRUE,append=TRUE)
913:  cat("}",file=texfile,fill = TRUE,append=TRUE)
914:
915: }
916:
917: ##############################################################################
918: #  import_png                                                                #
919: ##############################################################################
920: # ------------------------
921: # Authors:
922: # Luca Rossi       <redsluke@gmail.com>
923: # Fabrizio Carinci <research@fabcarinci.net>
924: # Institutions: Serectrix snc
925: # Created: 2008-11-21
926: # Version: 2008-10-23
927: # OS: Windows XP/Vista
928: # Programming Language: R 2.8.0
929: # ------------------------
930: # DESCRIPTION
931: #
932: # import_png include a PNG file into a te
933: #
934: # DEPENDENCIES
935: #
936: #  R package: base
937: #
938: ##############################################################################
939: # PARAMETERS(!=required)
940: #
941: #namegraph    !   = a string containing the name of the PNG file
```

```
942: #caption      !   = a string containing the caption
943: #width        !   = a numeric value between 0 and 1 indicating the width in
944: #                  the report
945: #dirgraph     !   = the directory where is the PGN file (latex format)
946: #texfile      !   = a string containing the name of the report .tex file
947: ##############################################################################
948:
949: import_png<-function(namegraph,
950:                      caption,
951:                      width,
952:                      dirgraph,
953:                      texfile) {
954:
955:
956:  for (i in 1:nchar(dirgraph)) {
957:   if (substr(dirgraph,i,i)=="\\") {
958:    substr(dirgraph,i,i)<-"/"
959:   }
960:  }
961:
962:  cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
963:  cat("\\parbox[2]{",width,"\\textwidth}{",file=texfile,fill = TRUE,append=TRUE)
964:  cat("\\fbox{",file=texfile,fill = TRUE,append=TRUE)
965:  cat("\\includegraphics[width=",width,"\\textwidth]{",file=texfile,fill = FALSE,append=TRUE,sep="")
966:  cat(dirgraph,"/",namegraph,"}",file=texfile,fill = TRUE,append=TRUE,sep="")
967:  cat("}",file=texfile,fill = TRUE,append=TRUE)
968:  cat("\\scriptsize",file=texfile,fill = TRUE,append=TRUE)
969:  cat(caption,file=texfile,fill = TRUE,append=TRUE)
970:  cat("}",file=texfile,fill = TRUE,append=TRUE)
971:  cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)
972:
973: }
974:
975: ##############################################################################
976: #  import_large_png                                                          #
977: ##############################################################################
978: # ------------------------
979: # Authors:
980: # Luca Rossi       <redsluke@gmail.com>
981: # Fabrizio Carinci <research@fabcarinci.net>
982: # Institutions: Serectrix snc
983: # Created: 2008-11-21
984: # Version: 2008-10-23
985: # OS: Windows XP/Vista
986: # Programming Language: R 2.8.0
```

```
 987: # ------------------------
 988: # DESCRIPTION
 989: #
 990: # import_png include a PNG file into a te, layout landscape
 991: #
 992: # DEPENDENCIES
 993: #
 994: #  R package: base
 995: #
 996: ################################################################################
 997: # PARAMETERS(!=required)
 998: #
 999: #namegraph    !   = a string containing the name of the PNG file
1000: #caption      !   = a string containing the caption
1001: #dirgraph     !   = the directory where is the PGN file (latex format)
1002: #texfile      !   = a string containing the name of the report .tex file
1003: ################################################################################
1004:
1005: import_large_png<-function(namegraph,
1006:                           caption,
1007:                           dirgraph,
1008:                           texfile) {
1009:
1010:
1011:  for (i in 1:nchar(dirgraph)) {
1012:   if (substr(dirgraph,i,i)=="\\") {
1013:    substr(dirgraph,i,i)<-"/"
1014:   }
1015:  }
1016:  cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
1017:  cat("\\includegraphics[width=1\\textheight,height=0.85\\textwidth,angle=90]",
1018:      file=texfile,fill = TRUE,append=TRUE,sep="")
1019:  cat("{",paste(dirgraph,"/",namegraph,sep=""),"}",file=texfile,fill = TRUE,
1020:      append=TRUE,sep="")
1021:  cat("\\",file=texfile,fill = TRUE,append=TRUE)
1022:  cat("\\begin{rotate}{90}",file=texfile,fill = TRUE,append=TRUE)
1023:  cat("\\parbox[2]{1\\textheight}{",file=texfile,fill = TRUE,append=TRUE)
1024:  cat("\\scriptsize",file=texfile,fill = TRUE,append=TRUE)
1025:  cat(caption,file=texfile,fill = TRUE,append=TRUE)
1026:  cat("}",file=texfile,fill = TRUE,append=TRUE)
1027:  cat("\\end{rotate}",file=texfile,fill = TRUE,append=TRUE)
1028:  cat("\\newpage",file=texfile,fill = TRUE,append=TRUE)
1029: }
1030:
1031: ################################################################################
```

```
1032: #  close_tex                                                              #
1033: ###############################################################################
1034: # ------------------------
1035: # Authors:
1036: # Luca Rossi       <redsluke@gmail.com>
1037: # Fabrizio Carinci <research@fabcarinci.net>
1038: # Institutions: Serectrix snc
1039: # Created: 2008-11-21
1040: # Version: 2008-10-23
1041: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
1042: # Programming Language: R 2.8.0
1043: ## ------------------------
1044: # DESCRIPTION
1045: #
1046: # BIRO_df2html writes the last rows of a .tex file (closes comands like
1047: #  \document,...)
1048: #
1049: # DEPENDENCIES
1050: #
1051: # R package: base
1052: #
1053: ###############################################################################
1054: # PARAMETERS(!=required)
1055: #
1056: #file       !              =  string containig the name of the file that must
1057: #                             be closed
1058: #
1059: ###############################################################################
1060:
1061: close_tex<-function(file) {
1062:
1063:  x<-FALSE
1064:  if (is.null(file)) {
1065:   print(file)
1066:   stop("file not recognized")
1067:  }
1068:
1069:  if (file.exists(file)==TRUE) {
1070:   cat("\\end{document}",file=file,fill = TRUE,append=TRUE)
1071:  }
1072:  invisible(x)
1073: }
1074:
1075: # ##############################################################################
1076: #  import_png_html                                                           #
```

```
1077: # ##############################################################################
1078: # ------------------------
1079: # Authors:
1080: # Luca Rossi       <redsluke@gmail.com>
1081: # Fabrizio Carinci <research@fabcarinci.net>
1082: # Institutions: Serectrix snc
1083: # Created: 2008-11-21
1084: # Version: 2008-10-23
1085: # OS: Windows XP/Vista
1086: # Programming Language: R 2.8.0
1087: # ------------------------
1088: # DESCRIPTION
1089: #
1090: # import_png include a PNG file into an HTML file
1091: #
1092: # DEPENDENCIES
1093: #
1094: #  R package: base
1095: #
1096: # ##############################################################################
1097: # PARAMETERS(!=required)
1098: #
1099: # image        !   = a string containing the path of the PNG file
1100: # caption      !   = a string containing the caption
1101: # htmlfile     !   = a string containing the name of an HTML file
1102: # ##############################################################################
1103:
1104: import_png_html<-function(image,
1105:                           htmlfile,caption) {
1106:
1107:  cat("<BR><BR><BR><BR><BR><BR>",
1108:      file=htmlfile,fill=TRUE,append=TRUE)
1109:  cat("<table style=\"width: 100%;\" border=\"1\">",
1110:      file=htmlfile,fill=TRUE,append=TRUE)
1111:  cat("<tbody>",
1112:      file=htmlfile,fill=TRUE,append=TRUE)
1113:  cat("<tr>",
1114:      file=htmlfile,fill=TRUE,append=TRUE)
1115:  cat("<td class=\"graph\">",
1116:      file=htmlfile,fill=TRUE,append=TRUE)
1117:  cat(paste("<img src=\"",image,"\">",sep=""),
1118:      file=htmlfile,fill=TRUE,append=TRUE)
1119:  cat("</td>",
1120:      file=htmlfile,fill=TRUE,append=TRUE)
1121:  cat("</tr>",
```

```
1122:       file=htmlfile,fill=TRUE,append=TRUE)
1123:   cat("</tbody>",
1124:       file=htmlfile,fill=TRUE,append=TRUE)
1125:   cat("<caption>",caption,"</caption>",
1126:       file=htmlfile,fill = TRUE,append=TRUE)
1127:   cat("</table>",
1128:      file=htmlfile,fill=TRUE,append=TRUE)
1129: }
1130:
1131: # ##############################################################################
1132: #  new_section_html                                                            #
1133: # ##############################################################################
1134: # ------------------------
1135: # Authors:
1136: # Luca Rossi       <redsluke@gmail.com>
1137: # Fabrizio Carinci <research@fabcarinci.net>
1138: # Institutions: Serectrix snc
1139: # Created: 2008-11-21
1140: # Version: 2008-10-23
1141: # OS: Windows XP/Vista
1142: # Programming Language: R 2.8.0
1143: # ------------------------
1144: # DESCRIPTION
1145: #
1146: # new_section_html create a new section into a HTML file
1147: #
1148: # DEPENDENCIES
1149: #
1150: #  R package: base
1151: #
1152: # ##############################################################################
1153: # PARAMETERS(!=required)
1154: #
1155: # title        !   = a string containing the title of the sub section
1156: # htmlfile      !   = a string containing the path of the HTML file
1157: # ##############################################################################
1158:
1159: new_section_html<-function(title,htmlfile) {
1160:
1161:  title<-paste(strsplit(title,"<"),collapse="&lt;")
1162:  title<-paste(strsplit(title,"%"),collapse="\\%")
1163:  title<-paste(strsplit(title,">"),collapse="&gt;")
1164:
1165:  cat("<table >",
1166:      file=htmlfile,fill=TRUE,append=TRUE)
```

```
1167:  cat("<tbody>",
1168:      file=htmlfile,fill=TRUE,append=TRUE)
1169:  cat("<tr>",
1170:      file=htmlfile,fill=TRUE,append=TRUE)
1171:  cat("<td class=\"section\">",title,
1172:      file=htmlfile,fill=TRUE,append=TRUE)
1173:  cat("</td>",
1174:      file=htmlfile,fill=TRUE,append=TRUE)
1175:  cat("</tr>",
1176:      file=htmlfile,fill=TRUE,append=TRUE)
1177:  cat("</tbody>",
1178:      file=htmlfile,fill=TRUE,append=TRUE)
1179:  cat("</table>",
1180:      file=htmlfile,fill=TRUE,append=TRUE)
1181: }
1182: # ##############################################################################
1183: #  new_chapter_html                                                            #
1184: # ##############################################################################
1185: # ------------------------
1186: # Authors:
1187: # Luca Rossi       <redsluke@gmail.com>
1188: # Fabrizio Carinci <research@fabcarinci.net>
1189: # Institutions: Serectrix snc
1190: # Created: 2008-11-21
1191: # Version: 2008-10-23
1192: # OS: Windows XP/Vista
1193: # Programming Language: R 2.8.0
1194: # ------------------------
1195: # DESCRIPTION
1196: #
1197: # new_chapter_html create a new chapter into a HTML file
1198: #
1199: # DEPENDENCIES
1200: #
1201: #  R package: base
1202: #
1203: # ##############################################################################
1204: # PARAMETERS(!=required)
1205: #
1206: # title        !   = a string containing the title of the sub section
1207: # htmlfile      !   = a string containing the path of the HTML file
1208: # ##############################################################################
1209:
1210: new_chapter_html<-function(title,
1211:                            htmlfile) {
```

```
1212:
1213:  title<-paste(strsplit(title,"<"),collapse="&lt;")
1214:  title<-paste(strsplit(title,"%"),collapse="\\%")
1215:  title<-paste(strsplit(title,">"),collapse="&gt;")
1216:
1217:  cat("<table >",
1218:      file=htmlfile,fill=TRUE,append=TRUE)
1219:  cat("<tbody>",
1220:      file=htmlfile,fill=TRUE,append=TRUE)
1221:  cat("<tr>",
1222:      file=htmlfile,fill=TRUE,append=TRUE)
1223:  cat("<td class=\"chapter\">",title,
1224:      file=htmlfile,fill=TRUE,append=TRUE)
1225:  cat("</td>",
1226:      file=htmlfile,fill=TRUE,append=TRUE)
1227:  cat("</tr>",
1228:      file=htmlfile,fill=TRUE,append=TRUE)
1229:  cat("</tbody>",
1230:      file=htmlfile,fill=TRUE,append=TRUE)
1231:  cat("</table>",
1232:      file=htmlfile,fill=TRUE,append=TRUE)
1233: }
1234: # ##############################################################################
1235: #  new_sub_section_html                                                        #
1236: # ##############################################################################
1237: # ------------------------
1238: # Authors:
1239: # Luca Rossi       <redsluke@gmail.com>
1240: # Fabrizio Carinci <research@fabcarinci.net>
1241: # Institutions: Serectrix snc
1242: # Created: 2008-11-21
1243: # Version: 2008-10-23
1244: # OS: Windows XP/Vista
1245: # Programming Language: R 2.8.0
1246: # ------------------------
1247: # DESCRIPTION
1248: #
1249: # new_sub_section_html create a new section into a HTML file
1250: #
1251: # DEPENDENCIES
1252: #
1253: #  R package: base
1254: #
1255: # ##############################################################################
1256: # PARAMETERS(!=required)
```

```
1257: #
1258: # title        !   = a string containing the title of the sub section
1259: # htmlfile      !   = a string containing the path of the HTML file
1260: # ##############################################################################
1261:
1262: new_sub_section_html<-function(title,
1263:                                htmlfile) {
1264:
1265:  title<-paste(strsplit(title,"<"),collapse="&lt;")
1266:  title<-paste(strsplit(title,"%"),collapse="\\%")
1267:  title<-paste(strsplit(title,">"),collapse="&gt;")
1268:  cat("<table >",
1269:      file=htmlfile,fill=TRUE,append=TRUE)
1270:  cat("<tbody>",
1271:      file=htmlfile,fill=TRUE,append=TRUE)
1272:  cat("<tr>",
1273:      file=htmlfile,fill=TRUE,append=TRUE)
1274:  cat("<td class=\"subsection\">",title,
1275:      file=htmlfile,fill=TRUE,append=TRUE)
1276:  cat("</td>",
1277:      file=htmlfile,fill=TRUE,append=TRUE)
1278:  cat("</tr>",
1279:      file=htmlfile,fill=TRUE,append=TRUE)
1280:  cat("</tbody>",
1281:      file=htmlfile,fill=TRUE,append=TRUE)
1282:  cat("</table>",
1283:      file=htmlfile,fill=TRUE,append=TRUE)
1284: }
1285:
1286:
```

```
  1: # ############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: biro_se_tables.r                                                     #
  5: #                                                                            #
#
  6: # ############################################################################
  7:
  8: # ------------------------
  9: # Authors:
 10: # Luca Rossi       <redsluke@gmail.com>
 11: # Fabrizio Carinci <research@fabcarinci.net>
 12: # Institutions: Serectrix snc
 13: # Created: 2008-11-21
 14: # Version: 2008-10-23
 15: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 16: # Programming Language: R 2.8.0
 17: # ------------------------
 18: # COPYRIGHT INFORMATION
 19: #
 20: # This file is free software; you can redistribute it and/or modify
 21: # it under the terms of the GNU General Public License as published by
 22: # the Free Software Foundation; either version 2, or (at your option)
 23: # any later version.
 24: #
 25: # This file is distributed in the hope that it will be useful,
 26: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 27: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 28: # GNU General Public License for more details.
 29: #
 30: # You should have received a copy of the GNU General Public License
 31: # along with this file; see the file COPYING.  If not, write to
 32: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 33: #
 34: # In short: you may use this file any way you like, as long as you
 35: # don't charge money for it, remove this notice, or hold anyone liable
 36: # for its results.
 37: #
 38: # biro_se_tables.r is part of WP Statistical Engine of the BIRO Project
 39: # GPL Copyright, The BIRO Project
 40: #
 41: # ------------------------
 42: #
 43: # CONTENT
 44: #
```

```
45: #   BIRO_fd
46: #   BIRO_table_format
47: #   BIRO_table_compute
48: #   BIRO_table_print
49: #   BIRO_df2html
50: #
51: # ##############################################################################
52:
53: # ##############################################################################
54: #  BIRO_fd                                                                     #
55: # ##############################################################################
56: # ------------------------
57: # Authors:
58: # Luca Rossi       <redsluke@gmail.com>
59: # Fabrizio Carinci <research@fabcarinci.net>
60: # Institutions: Serectrix snc
61: # Created: 2008-11-21
62: # Version: 2008-10-23
63: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
64: # Programming Language: R 2.8.0
65: ## ------------------------
66: # DESCRIPTION
67: #
68: # BIRO_fd for Conditional Frequency Distribution
69: #
70: # DEPENDENCIES
71: #
72: # R package: base
73: #
74: # ##############################################################################
75: # PARAMETERS(!=required)
76: #
77: # date       [System date] = date of analysis
78: # data                     = a data.frame containing the data
79: # var                      = the name of the target variable
80: # stratum                  = the name of the stratum variable
81: # typeround                = the type of rounding:
82: #                            option values:
83: #                            - "integer" , for integer rounding
84: #                            - a vector containig the thresholds
85: # codtab                   = the code table according the BIRO indicators list
86: # dircsv                   = the path of the output (csv files)
87: #
88: # ##############################################################################
89:
```

```
 90: BIRO_fd<-function(date,data,var,stratum,typeround,codtab,dircsv) {
 91:
 92:  for (i in 1:length(var)) {
 93:    temp<-as.vector(as.numeric(as.character(data[,pmatch(var[i],names(data))])))
 94:
 95:  if ((typeround[[i]][1])=="integer") {
 96:   temp<-as.data.frame(round(temp))
 97:   names(temp)<-paste(var[i],"_i",sep="")
 98:  }
 99:
100:
101:  if (typeof(typeround[[i]][1])=="double") {
102:    temp<-as.data.frame(varclass(temp,typeround[[i]]))
103:    names(temp)<-paste(var[i],"_c",sep="")
104:
105:  }
106:
107:  if (i==1) {
108:   newdata=temp
109:  }
110:
111:  if (i>1) {
112:  newdata<-as.data.frame(cbind(newdata,temp))
113:  }
114:
115:  }
116:
117:  strata<-as.data.frame(data[,pmatch(stratum,names(data))])
118:  names(strata)<-stratum
119:
120:  newdata<-cbind(newdata,strata)
121:
122:
123:  tab<-as.data.frame(table(newdata))
124:  tab$id=centre_id
125:  tab$date=date
126:  #Delete 0-Frequency cell
127:  tab<-tab[tab$Freq>0,]
128:  names(tab)[pmatch("Freq",names(tab))]<-"n"
129:
130:  write.csv(tab,paste(dircsv,"/i",codtab,"d1_1a.csv",sep=""),row.names = FALSE)
131:  #return(tab)
132: }
133:
134:
```

```
135: # ##############################################################################
136: #  BIRO_table_format                                                           #
137: # ##############################################################################
138: # ------------------------
139: # Authors:
140: # Luca Rossi       <redsluke@gmail.com>
141: # Fabrizio Carinci <research@fabcarinci.net>
142: # Institutions: Serectrix snc
143: # Created: 2008-11-21
144: # Version: 2008-10-23
145: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
146: # Programming Language: R 2.8.0
147: # ------------------------
148: # DESCRIPTION
149: #
150: # BIRO_table_format manages the condition: table with 4 variables, according
151: # the BIRO layout
152: #
153: # DEPENDENCIES
154: #
155: # R package: base
156: #
157: # ##############################################################################
158: # PARAMETERS(!=required)
159: #
160: # var       !              = two or more vector with the values of the target
161: #                            variables, Passed as a list
162: # FirstLab  !              = the name of the first variable, the first variable
163: #                            of var is the vertical variable in the table
164: # lev       !              = a list containing the vactor with che levels of
165: #                            the categorical variable (list)
166: # texfile   !              = the path of the output texfile
167: # codtab    !              = the coded name of the output csv file
168: # dirtables !              = directory for the output html table file
169: # dircsv    !              = directory for the output csv file
170: # caption   !              = string for the caption
171: # side      [0]            = if side is egual to 1 to table in the pdf file is
172: #                            printed in portrait layout
173: # width     [1]            = width for the pdf (see latex for details)
174: # perc      [TRUE]         = TRUE for percentage
175: # dirout    [getwd()]      = output location
176: # out       [out]          = name of output csv dataset
177: # ext       [csv]          = output file format
178: # date      [Sys.Date()]   = date variable or value
179: # ##############################################################################
```

```
180:
181: BIRO_table_format<-function(data,
182:                             var,
183:                             FirstLab,
184:                             lev=as.list(),
185:                             texfile,
186:                             codtab,
187:                             dirtables,
188:                             dircsv,
189:                             caption,
190:                             side=0,
191:                             width=1,
192:                             perc=TRUE) {
193:
194:
195:  temp<-na.omit(data[,pmatch(var,names(data))])
196:  if (dim(temp)[2]==4) {
197:   for (k in 1:length(sort(unique(temp[,4])))) {
198:    funcdata<-rep(NA,dim(temp)[2])
199:    for (h in 1:length(var)) {
200:     funcdata<-temp[temp[,4]==sort(unique(temp[4]))[k],]
201:    }
202:    if (length(var) > 0 & length(lev)>0) {
203:     funcdata[,pmatch(var[1],names(funcdata))]<-
204:     as.factor(funcdata[,pmatch(var[1],names(funcdata))])
205:     levels(funcdata[,pmatch(var[1],names(funcdata))])<-lev[[1]]
206:    }
207:    if (length(var) > 0 & length(lev)==0) {
208:     levels(funcdata[,pmatch(var[1],names(funcdata))])<-
209:     levels(as.factor(funcdata[,pmatch(var[1],names(funcdata))]))
210:    }
211:    if (length(var) > 1 & length(lev)>1) {
212:     funcdata[,pmatch(var[2],names(funcdata))]<-
213:     as.factor(funcdata[,pmatch(var[2],names(funcdata))])
214:     levels(funcdata[,pmatch(var[2],names(funcdata))])<-lev[[2]]
215:    }
216:    if (length(var) > 1 & length(lev)==1) {
217:     levels(funcdata[,pmatch(var[2],names(funcdata))])<-
218:     levels(as.factor(funcdata[,pmatch(var[2],names(funcdata))]))
219:    }
220:    if (length(var) > 2 & length(lev)>2)  {
221:     funcdata[,pmatch(var[3],names(funcdata))]<-
222:     as.factor(funcdata[,pmatch(var[3],names(funcdata))])
223:     levels(funcdata[,pmatch(var[3],names(funcdata))])<-lev[[3]]
224:    }
```

```
225:    if (length(var) > 2 & length(lev)==2)  {
226:     levels(funcdata[,pmatch(var[3],names(funcdata))])<-
227:     levels(as.factor(funcdata[,pmatch(var[3],names(funcdata))]))
228:    }
229:    if (length(var) > 3 & length(lev)>3)   {
230:     funcdata[,pmatch(var[4],names(funcdata))]<-
231:     factor(funcdata[,pmatch(var[4],names(funcdata))],levels=lev[[4]][k])
232:    }
233:    if (length(var) > 3 & length(lev)==3)   {
234:     funcdata[,pmatch(var[4],names(funcdata))]<-
235:     factor(funcdata[,pmatch(var[4],names(funcdata))],
236:     levels=sort(levels(as.factor(funcdata[,pmatch(var[4],
237:                                           names(funcdata))])))[k])
238:    }
239:
240:    newlev<-list()
241:    for (i in 1:length(var)) {
242:     newlev[[i]]<-levels(var[[i]])
243:    }
244:
245:    codtabv4<-paste(codtab,letters[k],sep="")
246:    BIRO_table_compute(data=funcdata,
247:                       var=var,
248:                       FirstLab=FirstLab,
249:                       lev=newlev,
250:                       texfile=texfile,
251:                       codtab=codtabv4,
252:                       dirtables=dirtables,
253:                       dircsv=dircsv,
254:                       caption=newcaption,
255:                       side=side,
256:                       width=width,
257:                       perc=perc,
258:                       ind=ind,
259:                       chisq=chisq[1:2])
260:   }
261:  }
262:  if (length(var)<4) {
263:   BIRO_table_compute(data=funcdata,
264:                      var=var,
265:                      FirstLab=FirstLab,
266:                      lev=lev,
267:                      texfile=texfile,
268:                      codtab=codtab,
269:                      dirtables=dirtables,
```

```
270:                     dircsv=dircsv,
271:                     caption=caption,
272:                     side=side,
273:                     width=width,
274:                     perc=perc,
275:                     ind=ind,
276:                     chisq=chisq)
277:  }
278: }
279:
280:
281:
282: # ##############################################################################
283: #  BIRO_table_compute                                                          #
284: # ##############################################################################
285: # ------------------------
286: # Authors:
287: # Luca Rossi       <redsluke@gmail.com>
288: # Fabrizio Carinci <research@fabcarinci.net>
289: # Institutions: Serectrix snc
290: # Created: 2008-11-21
291: # Version: 2008-10-23
292: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
293: # Programming Language: R 2.8.0
294: # ------------------------
295: # DESCRIPTION
296: #
297: # BIRO_table_compute calculates contingency table
298: #
299: # DEPENDENCIES
300: #
301: # R package: base
302: #
303: # ##############################################################################
304: # PARAMETERS(!=required)
305: #
306: # var         !            = two or more vector with the values of the target
307: #                            variables, Passed as a list
308: # FirstLab    !            = the name of the first variable, the first variable
309: #                            of var is the vertical variable in the table
310: # lev         !            = a list containing the vactor with che levels of
311: #                            the categorical variable (list)
312: # texfile     !            = the path of the output texfile
313: # codtab      !            = the coded name of the output csv file
314: # dirtables   !            = directory for the output html table file
```

```
315: # dircsv      !             = directory for the output csv file
316: # caption     !             = string for the caption
317: # side      [0]             = if side is egual to 1 to table in the pdf file is
318: #                             printed in portrait layout
319: # width     [1]             = width for the pdf (see latex for details)
320: # perc      [TRUE]          = TRUE for percentage
321: # dirout    [getwd()]       = output location
322: # out       [out]           = name of output csv dataset
323: # ext       [csv]           = output file format
324: # date      [Sys.Date()]    = date variable or value
325: # ind       !               = a string containing the index of indicator
326: # ##############################################################################
327:
328: BIRO_table_compute<-function(data,
329:                              var,
330:                              FirstLab,
331:                              lev=list(),
332:                              texfile,
333:                              codtab,
334:                              dirtables,
335:                              dircsv,
336:                              caption,
337:                              side=0,
338:                              width=1,
339:                              perc=TRUE,
340:                              ind=ind,
341:                              chisq=chisq) {
342:
343:  # Creates association between levels of a categorical variable and its values
344:
345:  funcdata<-data[,pmatch(na.omit(var),names(data))]
346:
347:  if (length(var) > 0 & length(lev)>0) {
348:   if (is.na(var[1])==FALSE) {
349:    funcdata[,pmatch(var[1],names(funcdata))]<-
350:    as.factor(funcdata[,pmatch(var[1],names(funcdata))])
351:    levels(funcdata[,pmatch(var[1],names(funcdata))])<-lev[[1]]
352:   }
353:  }
354:
355:  if (length(var) > 0 & length(lev)==0) {
356:   if (is.na(var[1])==FALSE) {
357:    levels(funcdata[,pmatch(var[1],names(funcdata))])<-
358:    levels(as.factor(funcdata[,pmatch(var[1],names(funcdata))]))
359:   }
```

```
360:  }
361:
362:  if (length(var) > 1 & length(lev)>1) {
363:   if (is.na(var[2])==FALSE) {
364:    funcdata[,pmatch(var[2],names(funcdata))]<-
365:    as.factor(funcdata[,pmatch(var[2],names(funcdata))])
366:    levels(funcdata[,pmatch(var[2],names(funcdata))])<-lev[[2]]
367:   }
368:  }
369:
370:  if (length(var) > 1 & length(lev)==1) {
371:   if (is.na(var[2])==FALSE) {
372:    levels(funcdata[,pmatch(var[2],names(funcdata))])<-
373:    levels(as.factor(funcdata[,pmatch(var[2],names(funcdata))]))
374:   }
375:  }
376:
377:  if (length(var) > 2 & length(lev)>2)  {
378:   if (is.na(var[3])==FALSE) {
379:    funcdata[,pmatch(var[3],names(funcdata))]<-
380:    as.factor(funcdata[,pmatch(var[3],names(funcdata))])
381:    levels(funcdata[,pmatch(var[3],names(funcdata))])<-lev[[3]]
382:   }
383:  }
384:
385:  if (length(var) > 2 & length(lev)==2)  {
386:   if (is.na(var[3])==FALSE) {
387:    levels(funcdata[,pmatch(var[3],names(funcdata))])<-
388:    levels(as.factor(funcdata[,pmatch(var[3],names(funcdata))]))
389:   }
390:  }
391:
392:  if (length(var) > 3 & length(lev)>3)   {
393:   if (is.na(var[4])==FALSE) {
394:    funcdata[,pmatch(var[4],names(funcdata))]<-
395:    factor(funcdata[,pmatch(var[4],names(funcdata))],levels=lev[[4]])
396:    levels(funcdata[,pmatch(var[4],names(funcdata))])<-lev[[4]]
397:   }
398:  }
399:
400:  if (length(var) > 3 & length(lev)==3)   {
401:   if (is.na(var[4])==FALSE) {
402:    funcdata[,pmatch(var[4],names(funcdata))]<-
403:    factor(funcdata[,pmatch(var[4],names(funcdata))],
404:           levels=sort(levels(as.factor(funcdata[,pmatch(var[4],
```

```
405:           names(funcdata))])))
406:   }
407:  }
408:
409:  for (i in 1:length(var)) {
410:   if (typeof(lev[[i]])=="list") {
411:    lev[[i]]<-names(unlist(lev[[i]]))
412:   }
413:  }
414:
415:  # creates a vector in which each element is then number of
416:  # the levels of the corrisponding variable
417:  nvar=length(var)
418:  n<-rep(NA,4)
419:  for (i in 1:nvar) {
420:   if (is.na(var[i])==FALSE) {
421:    n[i]<-nlevels(funcdata[,pmatch(var[i],names(funcdata))])
422:   }
423:  }
424:
425:  NAn<-is.na(n)
426:  n[is.na(n)]<-1
427:  varlist<-as.list(funcdata)
428:  # calculates contingency table
429:  tab<-as.data.frame(table(varlist))
430:
431:  # creates a schema of the output data.frame
432:  names(lev)<-names(varlist)
433:  schema<-NULL
434:  if (length(names(lev))>0) {
435:   times1<-rep(prod(n[2:nvar]),length(lev[[1]]))
436:   v1<-rep(lev[[1]],times1)
437:   schema<-cbind(schema,v1)
438:  }
439:
440:  if (length(names(lev))>1) {
441:   times2<-rep(prod(n[3:nvar]),length(lev[[2]]))
442:   if (nvar==2) {times2<-1}
443:   v2<-rep(lev[[2]],times2)
444:   schema<-cbind(schema,v2)
445:  }
446:
447:  if (length(names(lev))>2) {
448:   times3<-rep(n[4],length(lev[[3]]))
449:   if (nvar==3) {times3<-1}
```

```
450:    v3<-rep(lev[[3]],times3)
451:    schema<-cbind(schema,v3)
452:   }
453:
454:   if (length(names(lev))==4) {
455:    v4<-rep(lev[[4]],prod(n)/length(lev[[4]]))
456:    schema<-cbind(schema,v4)
457:   }
458:   schema<-as.data.frame(schema)
459:   names(schema)<-var
460:
461:   # merges the contingency table with the schema and puts 0
462:   # where there are not observations
463:
464:   tab<-merge(tab,schema,by=na.omit(var))
465:   tab$Freq[is.na(tab$Freq)]<-0
466:   tab$id=centre_id
467:   # calls BIRO_table_print for printing table in tex and html formats
468:
469:   htmlfile=paste(dirtables,"/",codtab,".html",sep="")
470:   paste(htmlfile)
471:   cat("<table border=\"1\">",file=htmlfile,fill = TRUE,append=TRUE)
472:   cat("",file=htmlfile,fill = TRUE,append=TRUE)
473:   cat(" <tbody>",file=htmlfile,fill = TRUE,append=TRUE)
474:   cat("",file=htmlfile,fill = TRUE,append=TRUE)
475:   cat("  <tr>",file=htmlfile,fill = TRUE,append=TRUE)
476:   cat("",file=htmlfile,fill = TRUE,append=TRUE)
477:   cat("   <td>",file=htmlfile,fill = TRUE,append=TRUE)
478:   if (struc==TRUE) {
479:    paste(sthtml)
480:    cat("",file=sthtml,fill = TRUE,append=TRUE)
481:    cat("<table border=\"1\">",file=sthtml,fill = TRUE,append=TRUE)
482:    cat("",file=sthtml,fill = TRUE,append=TRUE)
483:    cat(" <tbody>",file=sthtml,fill = TRUE,append=TRUE)
484:    cat("",file=sthtml,fill = TRUE,append=TRUE)
485:    cat("  <tr>",file=sthtml,fill = FALSE,append=TRUE)
486:    cat("",file=sthtml,fill = TRUE,append=TRUE)
487:    cat("   <td>",file=sthtml,fill = TRUE,append=TRUE)
488:   }
489:
490:   if (length(var)==4) {
491:    for (k in 1:length(unique(tab[,pmatch(var[4],names(tab))]))) {
492:     # values of the 4th categorical variable
493:     values<-unique(tab[,pmatch(var[4],names(tab))])
494:     temptab<-tab[tab[,pmatch(var[4],names(tab))]==values[k],]
```

```
495:    newlev<-list()
496:    for (i in 1:3) {
497:     newlev[[i]]<-lev[[i]]
498:    }
499:    newlev[[4]]<-as.character(values[k])
500:    n[4]<-1
501:    noNAnewlev<-list()
502:    j=0
503:    for (i in 1:4) {
504:     if (is.na(newlev[[i]])==FALSE) {
505:      j=j+1
506:      noNAnewlev[[j]]<-newlev[[i]]
507:     }
508:    }
509:    names(noNAnewlev)<-na.omit(var)
510:    if (k==length(unique(tab[,pmatch(var[4],names(tab))]))) {
511:     newcaption=caption
512:    }
513:    if (k<length(unique(tab[,pmatch(var[4],names(tab))]))) {
514:     newcaption=""
515:    }
516:    nnew<-c(n[NAn==FALSE],rep(1,length(n[NAn==TRUE])))
517:    temptab<-temptab[,pmatch(c(na.omit(var),"Freq","id"),names(temptab))]
518:    newcodtab<-paste(codtab,letters[k],sep="")
519:    BIRO_table_print(tab=temptab,
520:                     var=na.omit(var),
521:                     n=nnew,
522:                     FirstLab=FirstLab,
523:                     lev=noNAnewlev,
524:                     texfile=texfile,
525:                     codtab=paste(ind,letters[k],sep=""),
526:                     dirtables=dirtables,
527:                     dircsv=dircsv,
528:                     caption=newcaption,
529:                     side=side,
530:                     width=width,
531:                     perc=perc,
532:                     ind=ind,
533:                     chisq=chisq[1:2])
534:
535:    cat("<BR>",file=htmlfile,fill = TRUE,append=TRUE)
536:    cat("<BR>",file=sthtml,fill = TRUE,append=TRUE)
537:   }
538:  }
539:  if (length(var)<4) {
```

```
  BIRO_table_print(tab=tab,
                   var=na.omit(var),
                   n=na.omit(n),
                   FirstLab=FirstLab,
                   lev=lev,
                   texfile=texfile,
                   codtab=codtab,
                   dirtables=dirtables,
                   dircsv=dircsv,
                   caption=caption,
                   side=side,
                   width=width,
                   perc=perc,
                   ind=ind,
                   chisq=chisq)
 }

 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)
 }
}
```

```
    1: # ##############################################################################
    2: #                                                                            #
    3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
    4: # File: biro_se_trend.r                                                      #
    5: #                                                                            #
#
    6: # ##############################################################################
    7:
    8: # ------------------------
    9: # Authors:
   10: # Luca Rossi       <redsluke@gmail.com>
   11: # Fabrizio Carinci <research@fabcarinci.net>
   12: # Institutions: Serectrix snc
   13: # Created: 2007-11-21
   14: # Version: 2008-10-23
   15: # OS: Windows XP/Vista
   16: # Programming Language: R 2.8.0
   17: # ------------------------
   18: # COPYRIGHT INFORMATION
   19: #
   20: # This file is free software; you can redistribute it and/or modify
   21: # it under the terms of the GNU General Public License as published by
   22: # the Free Software Foundation; either version 2, or (at your option)
   23: # any later version.
   24: #
   25: # This file is distributed in the hope that it will be useful,
   26: # but WITHOUT ANY WARRANTY; without even the implied warranty of
   27: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   28: # GNU General Public License for more details.
   29: #
   30: # You should have received a copy of the GNU General Public License
   31: # along with this file; see the file COPYING.  If not, write to
   32: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
   33: #
   34: # In short: you may use this file any way you like, as long as you
   35: # don't charge money for it, remove this notice, or hold anyone liable
   36: # for its results.
   37: #
   38: # biro_se_trend.r is part of WP Statistical Engine of the BIRO Project
   39: # GPL Copyright, The BIRO Project
   40: #
   41: # ------------------------
   42: #
   43: # CONTENT
   44: #
```

```
45: #   BIRO_lines
46: #   BIRO_plotlines
47: #
48: # ##############################################################################
49:
50: # ##############################################################################
51: #  BIRO_lines                                                                  #
52: # ##############################################################################
53: # ------------------------
54: # Authors:
55: # Luca Rossi       <redsluke@gmail.com>
56: # Fabrizio Carinci <research@fabcarinci.net>
57: # Institutions: Serectrix snc
58: # Created: 2008-11-21
59: # Version: 2008-10-23
60: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
61: # Programming Language: R 2.8.0
62: # ------------------------
63: # DESCRIPTION
64: #
65: # BIRO_lines calculates mean of patient means for a particular and prepare data
66: # for BIRO_plotlines
67: #
68: # DEPENDENCIES
69: #
70: # BIRO_plotlines
71: #
72: # ##############################################################################
73: # PARAMETERS(!=required)
74: #
75: # var        !             = two or more vector with the values of the target
76: #                            variables, passed as a list (from the second there
77: #                            are stratum variables)
78: # descvar    !             = the labels of the first variable
79: # lev        !             = a list containing the vactor with che levels of the
80: #                            categorical variable (list)
81: # namegraph  !             = the coded name of the output files
82: # dirgraph   !             = the path of the output (jpg,eps,png files)
83: # stat       ["mean"]      = statistic applied to patient data
84: #
85: # ##############################################################################
86:
87: BIRO_lines<-function(var=list(),
88:                      descvar,
89:                      lev,
```

```
 90:                     namegraph,
 91:                     dirgraph,
 92:                     stat="mean") {
 93:
 94:  if (is.null(lev)==FALSE) {
 95:   levels(var$strata)=lev
 96:  }
 97:
 98:
 99:  if (tolower(stat)=="mean") {
100:   new_data<-as.data.frame(var)
101:   new_data$y=as.numeric(new_data$y)
102:   new_data=na.omit(new_data)
103:   new_data$strata=as.numeric(as.character(new_data$strata))
104:   new_data<-new_data[is.na(new_data$y)==FALSE,]
105:   splitvar<-interaction(new_data$times,new_data$strata,sep = ":")
106:   spltdata<-split(new_data, as.factor(splitvar), drop = FALSE)
107:   for (i in 1:length(spltdata)) {
108:    spltdata[i]<-na.omit(spltdata[i])
109:   }
110:   means<-as.data.frame(t(mapply(mean,spltdata)))
111:   for (i in 1:length(spltdata)) {
112:    if (dim(spltdata[[i]])[1]==0) spltdata[[i]]<-
113:        as.data.frame(matrix(rep(999,6),2))
114:   }
115:   sds<-t(lapply(spltdata,sd))
116:   sds<-(unlist(sds[1,]))
117:   sds<-as.vector(sds[seq(1,length(sds),3)])
118:   sds[is.na(sds)]<-0
119:   num<-aggregate(new_data$y,list(splitvar),"length")
120:   names(num)<-c("byvar","n")
121:  }
122:  means2<-as.data.frame(cbind(means[1],means$times,means$strata))
123:  means2$byvar=row.names(means2)
124:  names(means2)<-c("y","times","strata","byvar")
125:  means2<-merge(means2,num,by="byvar",all.x=TRUE)
126:  means2<-means2[,-1]
127:  names(means2)<-c("y","times","strata","n")
128:  means2$stat=stat
129:  means2$sd<-sds
130:  means2$codist=centre_id
131:  means2<-na.omit(means2)
132:  write.csv(means2,
133:            paste(dirdataout,"/i",namegraph,"d4_5a.csv",sep=""),
134:             row.names = FALSE)
```

```
135:
136:  write("Lines Data Stored",file="")
137:  if (nchar(logfile)>0) {
138:   write("Lines Data Stored",file=logfile,append=TRUE)
139:  }
140:
141:  BIRO_plotlines(data=means2,
142:                 lev=lev,
143:                 descvar=descvar,
144:                 namegraph=namegraph,
145:                 dirgraph=dirgraph)
146:
147: }
```

```
  1: ##############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: BIRO_se_recode.r                                                       #
  5: #                                                                            #
#
  6: ##############################################################################
  7:
  8: # ------------------------
  9: # Authors:
 10: # Luca Rossi       <redsluke@gmail.com>
 11: # Fabrizio Carinci <research@fabcarinci.net>
 12: # Institutions: Serectrix snc
 13: # Created: 2007-11-21
 14: # Version: 2008-10-23
 15: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 16: # Programming Language: R 2.8.0
 17: # ------------------------
 18: # COPYRIGHT INFORMATION
 19: #
 20: # This file is free software; you can redistribute it and/or modify
 21: # it under the terms of the GNU General Public License as published by
 22: # the Free Software Foundation; either version 2, or (at your option)
 23: # any later version.
 24: #
 25: # This file is distributed in the hope that it will be useful,
 26: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 27: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 28: # GNU General Public License for more details.
 29: #
 30: # You should have received a copy of the GNU General Public License
 31: # along with this file; see the file COPYING.  If not, write to
 32: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 33: #
 34: # In short: you may use this file any way you like, as long as you
 35: # don't charge money for it, remove this notice, or hold anyone liable
 36: # for its results.
 37: #
 38: # BIRO_se_recode.r is part of WP Statistical Engine of the BIRO Project
 39: # GPL Copyright, The BIRO Project
 40: #
 41: # ------------------------
 42: #
 43: # CONTENT
 44: #
```

```
45: ##############################################################################
46:
47:   #Thresholds
48:
49:   ageth<-c(35,55,75)
50:   ageth5<-seq(5,85,5)
51:   durdiabth<-c(10,20)
52:   weightth<-c(50,70,90,110,130)
53:   bmith<-c(25,27,30,40)
54:   sbpth<-c(130,160)
55:   dbpth<-c(70,100,120)
56:   cholth<-c(100,200,300)
57:   hdlth<-c(50)
58:   ldlth<-c(100,200)
59:   creatth<-c(0.90,1.20)
60:   hba1cth<-c(7,8)
61:
62:   #Limits
63:
64:   bmilimit<-c(4,44)
65:   weightlimit<-c(25,200)
66:   heightlimit<-c(60,250)
67:   chollimit<-c(70,400)
68:   hdllimit<-c(10,100)
69:   ldllimit<-c(30,300)
70:   creatlimit<-c(0.1,10)
71:   hba1climit<-c(2.5,20)
72:   ma_testlimit<-c(0,300)
73:   dbplimit<-c(30,150)
74:   sbplimit<-c(60,250)
75:   triglimit<-c(10,1000)
76:   trigl12limit<-c(10,1000)
77:
78:   #Levels of Categorical Variables
79:
80:   levtype_dm<-list("Type 1"="1","Type 2"="2")
81:   levsex<-list("Female"=2,"Male"=1)
82:   levma_test<-list("MA Test Recorded"=1,"No MA Test Recorded"=2)
83:   levsmok_stat<-list("Current smoker"=1,"Non-Smoker"=2,"Ex-Smoker"=3)
84:   levretin<-list("No Retinopathy"=1,"Background Retinopathy"=2,"Referable Retinopathy"=3)
85:   levmacul<-list("No Maculopathy"=1,"Maculopathy"=2)
86:   levfoot_exam<-list("Yes"=1,"No"=2)
87:   levesrf<-list("Yes"=1,"No"=2)
88:   levulcer<-list("Yes"=1,"No"=2)
89:   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("Yes"=1,"No"=2)
levbp<-list("Yes"=1,"No"=2)
levma_test<-list("Yes"=1,"No"=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("Yes"=1,"No"=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("Yes"=1,"No"=2)
levinsOAD<-list("Yes"=1,"No"=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("High"=1,"Low"=2,"No one"=1)
levhypert<-list("Yes"=1,"No"=2)
levbp_done<-list("Yes"=1,"No"=2)
```

```
 1: # ------------------------
 2: # Authors:
 3: # Luca Rossi       <redsluke@gmail.com)
 4: # Fabrizio Carinci <research@fabcarinci.net)
 5: # Institutions: Serectrix snc
 6: # Created: 2008-02-12
 7: # Version: 2008-10-23
 8: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 9: # Programming Language: R 2.8.0
10: # ------------------------
11: # COPYRIGHT INFORMATION
12: #
13: # This file is free software; you can redistribute it and/or modify
14: # it under the terms of the GNU General Public License as published by
15: # the Free Software Foundation; either version 2, or (at your option)
16: # any later version.
17: #
18: # This file is distributed in the hope that it will be useful,
19: # but WITHOUT ANY WARRANTY; without even the implied warranty of
20: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
21: # GNU General Public License for more details.
22: #
23: # You should have received a copy of the GNU General Public License
24: # along with this file; see the file COPYING.  If not, write to
25: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
26: #
27: # In short: you may use this file any way you like, as long as you
28: # don't charge money for it, remove this notice, or hold anyone liable
29: # for its results.
30: #
31: # BIRO_aggregate.r is part of WP Statistical Engine of the BIRO Project
32: # GPL Copyright, The BIRO Project
33: #
34: # ------------------------
35: #
36: # CONTENT
37: #
38: #  BIRO_aggregate
39: #
40: #
41: ##############################################################################
42:
43: ##############################################################################
44: #  BIRO_aggregate                                                            #
45: ##############################################################################
```

```
# ------------------------
# Authors:
# Luca Rossi       <research@fabcarinci.net)
# Fabrizio Carinci <research@fabcarinci.net)
# Institutions: Serectrix snc
# Created: 2008-02-12
# Version: 2008-10-23
# OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
# 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"))
#
```

```
 91: #data<-as.data.frame(cbind(patient_id,episode_date,
 92: #                          episode_field_value,year_episode))
 93: #Checks to do before
 94: #
 95: #data$patient_id=as.numeric(as.character(as.factor(data$patient_id)))
 96: #data$episode_date=as.Date(as.character(data$episode_date))
 97: #data$year_episode=as.numeric(as.character(data$year_episode))
 98: #data$episode_field_value=as.numeric(as.character(data$episode_field_value))
 99: #
100: #BIRO_aggregate(data=data,
101: #               request=c("patient_id"),
102: #               fun="mean",
103: #               vard="episode_field_value",
104: #               isdate=TRUE)
105: #
106: ###############################################################################
107:
108: BIRO_aggregate<-function(data,
109:                          request,
110:                          fun,
111:                          vard,
112:                          isdate=FALSE,
113:                          year_epi,
114:                          create_year=FALSE) {
115:
116:
117:   write("BIRO_aggregate called",file="")
118:   if (nchar(logfile)>0) {
119:    write("BIRO_aggregate called",file=logfile,append=TRUE)
120:   }
121:
122: data<-data[is.na(data[,pmatch(vard,names(data))])==FALSE,]
123:
124: if (create_year==TRUE) {
125:   data$year_epi=as.numeric(format(as.Date(as.character(data[,pmatch("epi_date",
126:                            names(data))]),
127:                            format="%Y-%m-%d"),format="%Y"))
128:
129: }
130:
131:
132: if (fun!="mean") {
133:
134:  if (is.na(pmatch("year_epi",request))==FALSE) {
135:   year="year_"
```

```
136:  } else {
137:   year=""
138:  }
139:
140:
141:  temp<-data[,c(pmatch(request,names(data)),pmatch(vard,names(data)))]
142:  if (fun=="max") {
143:
144:  names(temp)[pmatch(vard,names(temp))]<-paste("last_",year,vard,sep="")
145:
146:  vard<-paste("last_",year,vard,sep="")
147:
148:  }
149:  if (fun=="min") {
150:  names(temp)[pmatch(vard,names(temp))]<-paste("first_",year,vard,sep="")
151:
152:  vard<-paste("first_",year,vard,sep="")
153:
154:  }
155: }
156:
157: if (fun=="mean") {
158:  temp<-data[,c(pmatch(request,names(data)),pmatch(vard,names(data)))]
159:  names(temp)[pmatch(vard,names(temp))]<-paste("mean_",vard,sep="")
160:  vard<-paste("mean_",vard,sep="")
161: }
162:
163: request2<-list()
164: for (i in 1:length(request)) {
165: request2[[i]]<-as.numeric(as.character(temp[,pmatch(request[i],names(temp))]))
166:
167: }
168:
169: names(request2)<-request
170:
171:
172: temp<-aggregate(temp,request2,fun)
173:
174: temp<-as.data.frame(cbind(temp[,1:length(request2)],
175:                           temp[,pmatch(vard,names(temp))]))
176: names(temp)<-c(names(request2),vard)
177:
178: if (isdate) {
179:
180:  temp[,pmatch(vard,names(temp))]<-as.Date("1970-01-01")+
```

```
181:                                    temp[,pmatch(vard,names(temp))]
182:
183: }
184:
185:   write("BIRO_aggregate finished",file="")
186:   if (nchar(logfile)>0) {
187:    write("BIRO_aggregate finished",file=logfile,append=TRUE)
188:   }
189:
190:
191: return(temp)
192:
193:
194: }
195:
196:
```

```
#################################################################################
#                                                                               #
# 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: 2008-10-23
# OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
# 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
```

```
45: #
46: #
47: ##############################################################################
48:
49:
50: ##############################################################################
51: #  BIRO_demographic                                                          #
52: ##############################################################################
53: # ------------------------
54: # Authors:
55: # Luca Rossi       <redsluke@gmail.com)
56: # Fabrizio Carinci <research@fabcarinci.net)
57: # Institutions: Serectrix snc
58: # Created: 2008-02-14
59: # Version: 2008-10-23
60: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
61: # Programming Language: R 2.8.0
62: # ------------------------
63: # DESCRIPTION
64: #
65: # BIRO_demographic calculates the value of target variables, at a specified
66: # date, for each patient
67: #
68: # DEPENDENCIES
69: #
70: #  BIRO_aggregate
71: #
72: ##############################################################################
73: # PARAMETERS(!=required)
74: #
75: # ! patient   : the dataset processed containing patients data
76: # ! episode   : the dataset processed containing episodes data
77: # ! var       : the target variables names, which
78: #               can be applied to all data subsets
79: # ! date      : date of measurement
80: #               valid date or vector (for separate date for each patient)
81: #                 in the format DD/MM/YYYY [e.g. c("31/12/2006","31/12/2007")]
82: #               "last episode date"
83: #               "analysis date"
84: #               "last episode date for year"
85: #               "first episode date"
86: # ! startvar  : the initial values of a specific time intervals
87: #               for example:dob for age)
88: # ! th        : thresholds for a numeric variable
89: # ! hold      : logical. TRUE if you want to keep in the return dataset
```

```
  90: #                var and date
  91: # ! varname   : the name of the variable created by this function
  92: #
  93: ##############################################################################
  94: #
  95: # EXAMPLE
  96: #
  97: # This is the cohort for the BIRO indicator 1.1
  98: #
  99: # BIRO_demographic(patient=patient,episode=episode,
 100: # var=c("sex","age_c"),date=refdate,startvar="dob",
 101: # th=ageth,hold=FALSE,varname="age")
 102: ##############################################################################
 103:
 104:
 105: BIRO_demographic<-function(patient,episode,var,date,startvar,th,hold,varname) {
 106:
 107:
 108:   write("BIRO_demographic called",file="")
 109:   if (nchar(logfile)>0) {
 110:    write("BIRO_demographic called",file=logfile,append=TRUE)
 111:   }
 112:
 113: no_last_epi=0
 114: if ((as.character(date)=="analysis date") || (nchar(date)<=10)) {
 115:
 116:  if (nchar(date)<=10) {
 117:    date<-as.Date(date,"%d/%m/%Y")
 118:  }
 119:
 120:  if (as.character(date)=="analysis date") {
 121:   date<-as.Date(Sys.Date())
 122:  }
 123:
 124:  patient$myvar<-as.numeric(difftime(date,as.Date(as.character(patient[,pmatch(startvar,names(patient))])),units=
"days"))/365.25
 125:  patient$myvar_c<-as.factor(varclass(patient$myvar,th))
 126:
 127:  no_last_epi=1
 128:
 129: }
 130:
 131: if (as.character(date)=="episode date") {
 132:
 133:  patient<-merge(patient,episode,by="patient_id",all=FALSE)
```

```
  134:
patient$myvar<-as.numeric(difftime(as.Date(as.character(patient$epi_date)),as.Date(as.character(patient[,pmatch(startvar,n
ames(patient))])),units="days"))/365.25
  135:  patient$myvar_c<-as.factor(varclass(patient$myvar,th))
  136:
  137:   no_last_epi=1
  138:
  139: }
  140:
  141:
  142: if (as.character(date)=="last episode date") {
  143:
  144:  last_epi<-BIRO_aggregate(data=episode,request=c("patient_id"),fun="max",vard="epi_date",isdate=TRUE)
  145:
  146:  patient<-merge(patient,last_epi,by="patient_id",all=FALSE)
  147:
patient$myvar<-as.numeric(difftime(as.Date(as.character(patient$last_epi_date)),as.Date(as.character(patient[,pmatch(start
var,names(patient))])),units="days"))/365.25
  148:  patient$myvar_c<-as.factor(varclass(patient$myvar,th))
  149: }
  150:
  151: if (as.character(date)=="first episode date") {
  152:
  153:  last_epi<-BIRO_aggregate(data=episode,request=c("patient_id"),fun="min",vard="epi_date",isdate=TRUE)
  154:
  155:  patient<-merge(patient,last_epi,by="patient_id",all=FALSE)
  156:
patient$myvar<-as.numeric(difftime(as.Date(as.character(patient$first_epi_date)),as.Date(as.character(patient[,pmatch(star
tvar,names(patient))])),units="days"))/365.25
  157:  patient$myvar_c<-as.factor(varclass(patient$myvar,th))
  158: }
  159:
  160:
  161: if (as.character(date)=="last episode date for year") {
  162:
  163:
  164:
  165:  episode$year_epi=as.numeric(format(as.Date(as.character(episode[,pmatch("epi_date",names(episode))]),format=
"%Y-%m-%d"),format="%Y"))
  166:  last_epi<-BIRO_aggregate(data=episode,request=c("patient_id","year_epi"),fun="max",vard="epi_date"
,isdate=TRUE,create_year=TRUE)
  167:
  168:  patient<-merge(patient,last_epi,by="patient_id",all=FALSE)
  169:
patient$myvar<-as.numeric(difftime(as.Date(as.character(patient$last_year_epi_date)),as.Date(as.character(patient[,pmatch(
```

```
startvar,names(patient))])),units="days"))/365.25
  170:  patient$myvar_c<-as.factor(varclass(patient$myvar,th))
  171:
  172: }
  173:  names(patient)[pmatch("myvar",names(patient))]<-varname
  174:  names(patient)[pmatch("myvar_c",names(patient))]<-paste(varname,"_c",sep="")
  175:
  176: #Select target variables
  177:
  178:
  179: if (hold==TRUE) {
  180:  if (no_last_epi==0) {
  181:   var<-c(var,names(last_epi))
  182:  }
  183: }
  184:  write("BIRO_demographic finished",file="")
  185:   if (nchar(logfile)>0) {
  186:    write("BIRO_demographic finished",file=logfile,append=TRUE)
  187:   }
  188:
  189:
  190: return(patient[,pmatch(var,names(patient))])
  191:
  192: }
  193:
  194:
```

```
  1: # ############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: BIRO_explife.r                                                       #
  5: #                                                                            #
#
  6: # ############################################################################
  7: # ------------------------
  8: # Authors:
  9: # Luca Rossi       <redsluke@gmail.com)
 10: # Fabrizio Carinci <research@fabcarinci.net)
 11: # Institutions: Serectrix snc
 12: # Created: 2008-08-30
 13: # Version: 2008-10-23
 14: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 15: # Programming Language: R 2.8.0
 16: # ------------------------
 17: # COPYRIGHT INFORMATION
 18: #
 19: # This file is free software; you can redistribute it and/or modify
 20: # it under the terms of the GNU General Public License as published by
 21: # the Free Software Foundation; either version 2, or (at your option)
 22: # any later version.
 23: #
 24: # This file is distributed in the hope that it will be useful,
 25: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 26: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 27: # GNU General Public License for more details.
 28: #
 29: # You should have received a copy of the GNU General Public License
 30: # along with this file; see the file COPYING.  If not, write to
 31: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 32: #
 33: # In short: you may use this file any way you like, as long as you
 34: # don't charge money for it, remove this notice, or hold anyone liable
 35: # for its results.
 36: #
 37: # BIRO_explife.r is part of WP Statistical Engine of the BIRO Project
 38: # GPL Copyright, The BIRO Project
 39: #
 40: # ------------------------
 41: #
 42: # CONTENT
 43: #
 44: #  BIRO_explife
```

```
45: #
46: #
47: # ##############################################################################
48:
49:
50: # ##############################################################################
51: #  BIRO_explife                                                                 #
52: # ##############################################################################
53: # ------------------------
54: # Authors:
55: # Luca Rossi       <redsluke@gmail.com)
56: # Fabrizio Carinci <research@fabcarinci.net)
57: # Institutions: Serectrix snc
58: # Created: 2008-08-30
59: # Version: 2008-10-23
60: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
61: # Programming Language: R 2.8.0
62: # ------------------------
63: # DESCRIPTION
64: #
65: # BIRO_explife calculates the life expectancy (age in classes)
66: #
67: # DEPENDENCIES
68: #
69: # R package: base
70: #
71: # ##############################################################################
72: # PARAMETERS(!=required)
73: #
74: # pop       !     = the dataset processed containing population data
75: # mor       !     = the dataset processed containing mortality data
76: # agewidth  !     = the width of age classes
77: # sex       !     = the last part of sex variable indicating if Female or Male
78: # morname   !     = the initial part of sex variable containing mortality data
79: # popname   !     = the initial part of sex variable containing population data
80: # labsex    !     = a string value for the sex label
81: #
82: #
83: # ##############################################################################
84: #
85: # EXAMPLE
86: #
87: #    exm<-BIRO_explife(pop=pop,
88: #                      mor=pop,
89: #                      agewidth=5,
```

```
  90: #                     sex="M",
  91: #                     morname="mor",
  92: #                     popname="pop",
  93: #                     labsex='Males')
  94: #
  95: #    data.frame(pop):
  96: #
  97: #     age_band    tot   popM   popF  morM  morF
  98: #            1  37578  19356  18289     8    14
  99: #            2  36865  18623  17240     3     0
 100: #            3  43135  18641  17562     5     1
 101: #            .      .      .      .     .     .
 102: #            .      .      .      .     .     .
 103: #            .      .      .      .     .     .
 104: #
 105: #
 106: # ##############################################################################
 107:
 108: BIRO_explife<-function(pop,mor,agewidth,sex,morname,popname,labsex) {
 109:
 110:  write("BIRO_explife called",file="")
 111:  if (nchar(logfile)>0) {
 112:   write("BIRO_explife called",file=logfile,append=TRUE)
 113:  }
 114:
 115:  mor[,pmatch(paste(morname,sex,sep=""),names(mor))]<-100000*mor[,pmatch(paste(morname,sex,sep=""
),names(mor))]/sum(mor[,pmatch(paste(morname,sex,sep=""),names(mor))])
 116:
 117:  lx<-rep(NA,length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))]))
 118:  for (i in 1:(length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])[1])) {
 119:   if (i==1) lx[1]<-100000
 120:   if (i>1)  lx[i]= lx[i-1]-mor[,pmatch(paste(morname,sex,sep=""),names(mor))][i-1]
 121:  }
 122:
 123:  qx<-rep(NA,length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))]))
 124:  for (i in 1:length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])) {
 125:   qx[i]= mor[,pmatch(paste(morname,sex,sep=""),names(mor))][i]/lx[i]
 126:  }
 127:  Lx<-rep(NA,length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))]))
 128:  for (i in 1:length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])) {
 129:   if (i<length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])) Lx[i]=(lx[i]+lx[i+1])/2
 130:   if (i==length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])) Lx[i]<-NA
 131:  }
 132:  Tx<-rep(NA,length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))]))
 133:  for (i in 1:length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])) {
```

```
134:   Tx[i]=sum(Lx[i:(length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])-1)],na.rm=TRUE)
135:  }
136:
137:  ex<-rep(length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))]))
138:  for (i in 1:length(pop[,pmatch(paste(popname,sex,sep=""),names(pop))])) {
139:   ex[i]=(Tx[i]/lx[i])*agewidth
140:  }
141:
142:  agelab<-classlabel(seq(agewidth,85,5))
143:  retdata<-as.data.frame(cbind(agelab,ex))
144:  names(retdata)<-c('Age in classes',labsex)
145:  return(retdata)
146:
147:  write("BIRO_explife finished",file="")
148:  if (nchar(logfile)>0) {
149:   write("BIRO_explife finished",file=logfile,append=TRUE)
150:  }
151:
152: }
153:
154:
```

```
  1: # ############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: BIRO_forest.r                                                        #
  5: #                                                                            #
#
  6: # ############################################################################
  7: # ------------------------
  8: # Authors:
  9: # Luca Rossi       <redsluke@gmail.com)
 10: # Fabrizio Carinci <research@fabcarinci.net)
 11: # Institutions: Serectrix snc
 12: # Created: 2008-00-00
 13: # Version: 2008-10-23
 14: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 15: # Programming Language: R 2.8.0
 16: # ------------------------
 17: # COPYRIGHT INFORMATION
 18: #
 19: # This file is free software; you can redistribute it and/or modify
 20: # it under the terms of the GNU General Public License as published by
 21: # the Free Software Foundation; either version 2, or (at your option)
 22: # any later version.
 23: #
 24: # This file is distributed in the hope that it will be useful,
 25: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 26: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 27: # GNU General Public License for more details.
 28: #
 29: # You should have received a copy of the GNU General Public License
 30: # along with this file; see the file COPYING.  If not, write to
 31: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 32: #
 33: # In short: you may use this file any way you like, as long as you
 34: # don't charge money for it, remove this notice, or hold anyone liable
 35: # for its results.
 36: #
 37: # BIRO_forest.r is part of WP Statistical Engine of the BIRO Project
 38: # GPL Copyright, The BIRO Project
 39: #
 40: # ------------------------
 41: #
 42: # CONTENT
 43: #
 44: #  BIRO_forest
```

```
45: #
46: #
47: # ######################################################################################
48:
49:
50: # ######################################################################################
51: #  BIRO_forest                                                                         #
52: # ######################################################################################
53: # ------------------------
54: # Authors:
55: # Luca Rossi       <research@fabcarinci.net)
56: # Fabrizio Carinci <research@fabcarinci.net)
57: # Institutions: Serectrix snc
58: # Created: 2002-00-00
59: # Version: 2003-01-20
60: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
61: # Programming Language: R 2.8.0
62: # ------------------------
63: # DESCRIPTION
64: #
65: # Draw forest plot
66:
67: # DEPENDENCIES
68: #
69: #  R packages: grDevice, rmeta, Cairo
70: #
71: # ######################################################################################
72: # PARAMETERS(!=required)
73: # tabletext            !  = a table with labels
74: # values               !  = a vector with central values
75: # lbound               !  = a vector with lower values
76: # ubound               !  = a vector with upper values
77: # limits      [c(0,200)]  = a vector with axis limits
78: # xlab                 !  = a string containing the X label
79: # col                  !  = See meta.colors
80: # zero                 !  = x-axis coordinate for zero line
81: # csvindex             !  = the coded name fo the csv file
82: #
83: # ######################################################################################
84:
85: BIRO_forest<-function(tabletext,
86:                       values,
87:                       lbound,
88:                       ubound,
89:                       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(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("SVG file: Done",file="")
  if (nchar(logfile)>0) {
   write("SVG file: Done",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 = 11.33, height =4.84 , units="in",pointsize = 9,
       bg = "transparent",res=9)
  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("JPEG file: Done",file="")
  if (nchar(logfile)>0) {
   write("JPEG file: Done",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=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("PNG file: Done",file="")
  if (nchar(logfile)>0) {
   write("PNG file: Done",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("PDF file: Done",file="")
  if (nchar(logfile)>0) {
   write("PDF file: Done",file=logfile,append=TRUE)
  }
  data<-cbind(tabletext,values,lbound,ubound)
  write.csv(data,
            paste(dirdataout,"/for",csvindex,".csv",sep=""),
            row.names = FALSE)
  write("Forest plot data stored in csv file",file="")
  if (nchar(logfile)>0) {
   write("Forest plot data stored in csv file",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)
  }

 }

```

```
    1: ##############################################################################
    2: #                                                                            #
    3: # Project: BIRO-Project (Funded by European Commission 2005-2008)           #
    4: # File: BIRO_maps.r                                                  #
    5: #                                                                            #
#
    6: ##############################################################################
    7: # ------------------------
    8: # Authors:
    9: # Luca Rossi       <redsluke@gmail.com)
   10: # Fabrizio Carinci <research@fabcarinci.net)
   11: # Institutions: Serectrix snc
   12: # Created: 2008-08-10
   13: # Version: 2008-11-23
   14: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
   15: # Programming Language: R 2.8.0
   16: # ------------------------
   17: # COPYRIGHT INFORMATION
   18: #
   19: # This file is free software; you can redistribute it and/or modify
   20: # it under the terms of the GNU General Public License as published by
   21: # the Free Software Foundation; either version 2, or (at your option)
   22: # any later version.
   23: #
   24: # This file is distributed in the hope that it will be useful,
   25: # but WITHOUT ANY WARRANTY; without even the implied warranty of
   26: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   27: # GNU General Public License for more details.
   28: #
   29: # You should have received a copy of the GNU General Public License
   30: # along with this file; see the file COPYING.  If not, write to
   31: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
   32: #
   33: # In short: you may use this file any way you like, as long as you
   34: # don't charge money for it, remove this notice, or hold anyone liable
   35: # for its results.
   36: #
   37: # BIRO_maps.r is part of WP Statistical Engine of the BIRO Project
   38: # GPL Copyright, The BIRO Project
   39: #
   40: # ------------------------
   41: #
   42: # CONTENT
   43: #
   44: #  BIRO_map
```

```
45: #  BIRO_patmap
46: #
47: ###############################################################################
48:
49:
50: ###############################################################################
51: #  BIRO_map                                                                   #
52: ###############################################################################
53: # ------------------------
54: # Authors:
55: # Luca Rossi       <redsluke@gmail.com)
56: # Fabrizio Carinci <research@fabcarinci.net)
57: # Institutions: Serectrix snc
58: # Created: 2008-08-10
59: # Version: 2008-11-23
60: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
61: # Programming Language: R 2.8.0
62: # ------------------------
63: # DESCRIPTION
64: #
65: # To plot a map with desidered values
66: #
67: # DEPENDENCIES
68: #
69: #  R packages: maptools,sp,lattice,Cairo
70: #
71: ###############################################################################
72: # PARAMETERS(!=required)
73: #
74: # shape            ! = the name the the S4 object (shape file loaded)
75: # datavalues       ! = the data.frame containing the value that you want to represent
76: # frqvar           ! = the variable the variable that you want to represent
77: # shapearea_id     ! = the id in the S4 of the areas
78: # valuesarea_id    ! = the id in the values data.frame of the areas
79: # labvar           ! = label written in the legend
80: # levels           ! = a vector containing the threshold for the map
81: # legendpos        ! = the position of the legend
82: # dirgraph               !  = a string containing the path of the output graph
83: # namegraph              !  = a string containing the name of the graph
84: #
85: ###############################################################################
86:
87: BIRO_map<-function(shape,
88:                   datavalues,
89:                   frqvar=region$POP2001,
```

```
 90:                     shapearea_id,
 91:                     valuesarea_id,
 92:                     labvar="Population",
 93:                     breaks=c(5000,10000),
 94:                     legendpos="topright",
 95:                     dirgraph,
 96:                     namegraph) {
 97:
 98:  myshapefile=shape
 99:  shape$myid=1:length(shape$ID)
100:  print("map1")
101:  print(names(shape))
102:  print(names(datavalues))
103:  shapedata<-merge(shape,datavalues,by.x=shapearea_id,by.y=valuesarea_id,all.x=TRUE,all.y=FALSE)
104:  print("map2")
105:  shapedata$myarea<-varclass(shapedata[,pmatch(frqvar,names(shapedata))],breaks)
106:  cols<-rainbow(length(breaks)+1,start = 0.4)
107:  datacol<-as.data.frame(cols)
108:  names(datacol)<-"col"
109:  datacol$col<-as.character(datacol$col)
110:  datacol$lab<-classlabel(breaks)
111:  datacol$idc<-1:(length(breaks)+1)
112:  datacol$idcolor<-1:(length(breaks)+1)
113:
114:  datacol<-merge(shapedata,datacol,by.x="myarea",by.y="idc",all.x=TRUE,all.y=FALSE)
115:
116: datacol<-sort.data.frame(datacol,by=~myid)
117:  file = paste(dirgraph,"/i",namegraph,"g4_8a.png",sep="")
118:  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
119:  plot(myshapefile,col=datacol$col)
120:  datacol<-sort.data.frame(datacol,by=~idcolor)
121:  legend(legendpos,legend=datacol$lab,cex=cex,fill=datacol$col,title=labvar)
122:
123:  dev.off()
124:
125:  file = paste(dirgraph,"/i",namegraph,"g4_8a.pdf",sep="")
126:  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
127:  plot(myshapefile,col=datacol$col)
128:  datacol<-sort.data.frame(datacol,by=~idcolor)
129:  legend(legendpos,legend=datacol$lab,cex=cex,fill=datacol$col,title=labvar)
130:  dev.off()
131:
132:  file = paste(dirgraph,"/i",namegraph,"g4_8a.svg",sep="")
133:  CairoSVG(file, width = 11.33, height =4.84, pointsize=9,  bg = "transparent")
134:  plot(myshapefile,col=datacol$col)
```

```
135:  datacol<-sort.data.frame(datacol,by=~idcolor)
136:  legend(legendpos,legend=datacol$lab,cex=cex,fill=datacol$col,title=labvar)
137:  dev.off()
138:
139:  file = paste(dirgraph,"/i",namegraph,"g4_8a.jpeg",sep="")
140:  jpeg(file, width= 813.6, height= 348.48, pointsize=9, units="px",
141:       bg = "transparent",res=9)
142:  plot(myshapefile,col=datacol$col)
143:  datacol<-sort.data.frame(datacol,by=~idcolor)
144:  legend(legendpos,legend=datacol$lab,cex=cex,fill=datacol$col,title=labvar)
145:  dev.off()
146:
147:
148:
149: }
150:
151: ##############################################################################
152: #  BIRO_patmap                                                               #
153: ##############################################################################
154: # ------------------------
155: # Authors:
156: # Luca Rossi       <redsluke@gmail.com)
157: # Fabrizio Carinci <research@fabcarinci.net)
158: # Institutions: Serectrix snc
159: # Created: 2008-08-10
160: # Version: 2008-11-23
161: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
162: # Programming Language: R 2.8.0
163: # ------------------------
164: # DESCRIPTION
165: #
166: # Manage the plotting of maps (absolute frequencies or percentages)
167: #
168: # DEPENDENCIES
169: #
170: #  R packages: maptools,sp,lattice,Cairo
171: #
172: ##############################################################################
173: # PARAMETERS(!=required)
174: #
175: # ind              ! = the code of the BIRO indicator
176: # shape            ! = the name the the S4 object (shape file loaded)
177: # data             ! = the data.frame containing the value that you want to represent
178: # lev              !  = a list contaning levels of variables specified in var
179: # var              ! = a vector containing the names of variables
```

```
180: # frqvar           ! = the variable the variable that you want to represent
181: # shapearea_id     ! = the id in the S4 of the areas
182: # valuesarea_id    ! = the id in the values data.frame of the areas
183: # labvar           ! = label written in the legend
184: # stat             ! = a sting containing "freq" or "perc"
185: # legendpos        ! = the position of the legend
186: # dirgraph               !  = a string containing the path of the output graph
187: # namegraph              !  = a string containing the name of the graph
188: #
189: ###############################################################################
190:
191: BIRO_patmap<-function(ind,
192:                       shape,
193:                       data,
194:                       lev,
195:                       var,
196:                       labvar,
197:                       stat,
198:                       legendpos="topright",
199:                       dirgraph,
200:                       namegraph,
201:                       frqvar=region$POP2001,
202:                       shapearea_id,
203:                       valuesarea_id) {
204:
205:
206:  myFreqs<-list()
207:  for (i in 1:length(stat)) {
208:   if (stat=="freq") {
209:    for (j in 1:length(var)) {
210:     splitdata<-split(data[,pmatch(valuesarea_id,names(data))],list(data[,pmatch(var[i],names(data))]))
211:     for (k in 1:length(unique(data[,pmatch(var[i],names(data))]))) {
212:      Freqs<-as.data.frame(table(splitdata[[k]]))
213:      names(Freqs)[pmatch("Freq",names(Freqs))]<-"n"
214:      names(Freqs)[pmatch("Var1",names(Freqs))]<-"id_geo"
215:      Freqs$centre_id=centre_id
216:
217:      myFreqs[[k]]<-Freqs
218:
219:      if (length(unique(Freqs$n))<=4) {
220:       breaks<-mean(unique(Freqs$n))
221:      }
222:      if (length(unique(Freqs$n))>4) {
223:       step<-(min(unique(Freqs$n))-max(unique(Freqs$n)))/5
224:       breaks<-round(seq(from=min(unique(Freqs$n))+step,to=max(unique(Freqs$n))-step,by=step))
```

```
225:       }
226:       namegraph<-paste(namegraph,letters[k],sep="")
227:       BIRO_map(shape=shape,
228:                datavalues=Freqs,
229:                frqvar="n",
230:                shapearea_id=shapearea_id,
231:                valuesarea_id="id_geo",
232:                labvar=labvar,
233:                breaks=breaks,
234:                legendpos="topright",
235:                dirgraph=dirgraph,
236:                namegraph=namegraph)
237:
238:
239:        import_png(namegraph=paste("i",namegraph,"g4_8a.png",sep=""),
240:                   caption=paste("Map:",labvar[j],lev[[j]][k],sep=""),
241:                   width=1,
242:                   dirgraph=dirgraphlatex,
243:                   texfile=texfile)
244:
245:        import_png_html(image=paste("../graphs/i",namegraph,"g4_8a.png",sep=""),
246:                        htmlfile=paste(dirhtml,"/",ind,".html",sep=""),
247:                        caption=paste("Map: ",labvar,sep=""))
248:
249:     }
250:
251:     allFreqs<-Freqs[[1]]
252:     for (k in 2:length(unique(data[,pmatch(var[i],names(data))]))) {
253:      allFreqs<-rbind(allFreqs,Freqs[[k]])
254:     }
255:     write.csv(allFreqs,paste(dirdataout,"/i",ind,letters[j],"d4_8a.csv",sep=""),row.names=FALSE)
256:
257:    }
258:   }
259:   }
260: }
261:
262:
263:
264:
```

```
#############################################################################
#                                                                           #
# 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: 2008-10-23
# 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
#
```

```
45: #   BIRO_plotlines
46: #   BIRO_table_print
47: #   BIRO_plotbox
48: #
49: ##############################################################################
50:
51: ##############################################################################
52: #  BIRO_plotlines                                                            #
53: ##############################################################################
54: # ------------------------
55: # Authors:
56: # Luca Rossi       <redsluke@gmail.com>
57: # Fabrizio Carinci <research@fabcarinci.net>
58: # Institutions: Serectrix snc
59: # Created: 2008-11-21
60: # Version: 2008-10-23
61: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
62: # Programming Language: R 2.8.0
63: ## ------------------------
64: # DESCRIPTION
65: #
66: #BIRO_plotlines drawing lines into a graph
67: #
68: # DEPENDENCIES
69: #
70: #  R packages: grDevice, graphics, Cairo
71: #
72: ##############################################################################
73: # PARAMETERS(!=required)
74: #
75: #data       !              = dataset containing data prepared by BIRO_lines
76: #lev        !              = a list containing the vactor with che levels of the
77: #                           categorical variable (list)
78: #descvar    !              = string for the x axe
79: #dirgraph   !              = the path of the output (jpg,eps,png files)
80: #namegraph  !              = the coded name of the output files
81: #
82: ##############################################################################
83:
84:
85: BIRO_plotlines<-function(data,
86:                          lev,
87:                          descvar,
88:                          namegraph,
89:                          dirgraph) {
```

```
 90:
 91:
 92: #X11(width = 21, height =9 , pointsize=9)
 93:
 94: #Confidence Intervals
 95:   data$Lower=as.numeric(as.character(data$y))-
 96:             1.65*as.numeric(as.character(data$sd))
 97:   data$Upper=as.numeric(as.character(data$y))+
 98:             1.65*as.numeric(as.character(data$sd))
 99:
100:   write("Preparing to do SVG file",file="")
101:   if (nchar(logfile)>0) {
102:    write("Preparing to do SVG file",file=logfile,append=TRUE)
103:   }
104:
105:   file = paste(dirgraph,"/","/i",namegraph,"g4_5a.svg",sep="")
106:
107:   CairoSVG(file, width = 11.33, height =4.84, pointsize=9,  bg = "transparent")
108: #First line
109:   plot(data$times[data$strata==levels(as.factor(data$strata))[1]],
110:        data$y[data$strata==levels(as.factor(data$strata))[1]],
111:        type="l",
112:        col=rainbow(nlevels(as.factor(data$strata)))[1],
113:        ylim=c(min(data$Lower,na.rm=TRUE)-sd(data$y,na.rm=TRUE),
114:               max(data$Upper,na.rm=TRUE)+3*nlevels(as.factor(data$strata))),
115:        xlab=descvar[2],
116:        ylab="",
117:        lty=2,
118:        cex=cex)
119:
120: #Confidence boundary for the first line
121:   for (j in 1:dim(data)[1]) {
122:
123:    if (data$strata[j]==levels(as.factor(data$strata))[1]) {
124:      text(data$time[j],
125:           data[j,pmatch("Lower",names(data))],"|-----|",
126:           col=rainbow(nlevels(as.factor(data$strata)))[1],
127:           lty=2,
128:           lwd=2)
129:      text(data$time[j],
130:           data[j,pmatch("Upper",names(data))],"|-----|",
131:           col=rainbow(nlevels(as.factor(data$strata)))[1],
132:           lty=2,
133:           lwd=2)
134:    }
```

```
135:
136: }
137:
138:
139: #Further lines
140:   for (i in 2:nlevels(as.factor(data$strata))) {
141:    lines(data$time[data$strata==levels(as.factor(data$strata))[i]],
142:          data$y[data$strata==levels(as.factor(data$strata))[i]],
143:          col=rainbow(length(lev))[i],
144:          lty=2,
145:          lwd=2,
146:          cex=cex)
147:
148:    for (j in 1:dim(data)[1]) {
149:     if (data$strata[j]%in%levels(as.factor(data$strata))[i]) {
150:          text(data$time[j],
151:               data[j,pmatch("Lower",names(data))],"|-----|",
152:               col=rainbow(length(lev))[i],
153:               lty=2,
154:               lwd=2)
155:          text(data$time[j],
156:               data[j,pmatch("Upper",names(data))],"|-----|",
157:               col=rainbow(length(lev))[i],
158:               lty=2,
159:               lwd=2)
160:     }
161:
162:    }
163:
164:
165:   }
166:
167:   axis(1,levels(data$times))
168:   leg<-rep(NA,nlevels(as.factor(data$strata)))
169:   for (i in 1:nlevels(as.factor(data$strata))) {
170:    leg[i]<-lev[i]
171:   }
172:   x<-legend("topright", leg,
173:             col = rainbow(nlevels(as.factor(data$strata))),
174:             lty=2, lwd=2,pt.bg="white",cex=cex,title=descvar)
175:
176:   dev.off()
177:
178:   write("SVG file: Done",file="")
179:   if (nchar(logfile)>0) {
```

```
180:    write("SVG file: done",file=logfile,append=TRUE)
181:   }
182:
183:   write("Preparing to do JPEG file",file="")
184:   if (nchar(logfile)>0) {
185:    write("Preparing to do JPEG file",file=logfile,append=TRUE)
186:   }
187:
188:   file = paste(dirgraph,"/","/i",namegraph,"g4_5a.jpeg",sep="")
189:   jpeg(file, width= 11.33, height= 4.84, pointsize=9,
190:        units="in", bg = "transparent",res=9)
191: #First line
192:   plot(data$times[data$strata==levels(as.factor(data$strata))[1]],
193:        data$y[data$strata==levels(as.factor(data$strata))[1]],
194:        type="l",
195:        col=rainbow(nlevels(as.factor(data$strata)))[1],
196:        ylim=c(min(data$Lower,na.rm=TRUE)-sd(data$y,na.rm=TRUE),
197:               max(data$Upper,na.rm=TRUE)+3*nlevels(as.factor(data$strata))),
198:        xlab=descvar[2],
199:        ylab="",
200:        lty=2,
201:       cex=cex)
202:
203:   #Confidence boundary for the first line
204:   for (j in 1:dim(data)[1]) {
205:
206:    if (data$strata[j]==levels(as.factor(data$strata))[1]) {
207:      text(data$time[j],
208:           data[j,pmatch("Lower",names(data))],"|-----|",
209:           col=rainbow(nlevels(as.factor(data$strata)))[1],
210:           lty=2,
211:           lwd=2)
212:      text(data$time[j],
213:           data[j,pmatch("Upper",names(data))],"|-----|",
214:           col=rainbow(nlevels(as.factor(data$strata)))[1],
215:           lty=2,
216:           lwd=2)
217:    }
218:
219:   }
220:
221:
222: #Further lines
223:   for (i in 2:nlevels(as.factor(data$strata))) {
224:    lines(data$time[data$strata==levels(as.factor(data$strata))[i]],
```

```
225:        data$y[data$strata==levels(as.factor(data$strata))[i]],
226:        col=rainbow(length(lev))[i],
227:        lty=2,
228:        lwd=2,
229:        cex=cex)
230:
231:    for (j in 1:dim(data)[1]) {
232:     if (data$strata[j]%in%levels(as.factor(data$strata))[i]) {
233:      text(data$time[j],
234:           data[j,pmatch("Lower",names(data))],"|-----|",
235:           col=rainbow(length(lev))[i],
236:           lty=2,
237:           lwd=2)
238:      text(data$time[j],
239:           data[j,pmatch("Upper",names(data))],"|-----|",
240:           col=rainbow(length(lev))[i],
241:           lty=2,
242:           lwd=2)
243:     }
244:    }
245:
246:
247:   }
248:   axis(1,levels(data$times))
249:   leg<-rep(NA,nlevels(as.factor(data$strata)))
250:   for (i in 1:nlevels(as.factor(data$strata))) {
251:    leg[i]<-lev[i]
252:   }
253:   x<-legend("topright", leg, col = rainbow(nlevels(as.factor(data$strata))),
254:             lty=2, lwd=2,pt.bg="white",cex=cex,title=descvar)
255:   dev.off()
256:
257:   write("JPEG file: Done",file="")
258:   if (nchar(logfile)>0) {
259:    write("JPEG file: done",file=logfile,append=TRUE)
260:   }
261:
262:   write("Preparing to do PNG file",file="")
263:   if (nchar(logfile)>0) {
264:    write("Preparing to do PNG file",file=logfile,append=TRUE)
265:   }
266:
267:   file = paste(dirgraph,"/","/i",namegraph,"g4_5a.png",sep="")
268:   CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
269: #First line
```

```
270:   plot(data$times[data$strata==levels(as.factor(data$strata))[1]],
271:        data$y[data$strata==levels(as.factor(data$strata))[1]],
272:        type="l",
273:        col=rainbow(nlevels(as.factor(data$strata)))[1],
274:        ylim=c(min(data$Lower,na.rm=TRUE)-sd(data$y,na.rm=TRUE),
275:               max(data$Upper,na.rm=TRUE)+3*nlevels(as.factor(data$strata))),
276:        xlab=descvar[2],
277:        ylab="",
278:        lty=2,
279:        cex=cex)
280:
281: #Confidence boundary for the first line
282:   for (j in 1:dim(data)[1]) {
283:
284:    if (data$strata[j]==levels(as.factor(data$strata))[1]) {
285:      text(data$time[j],
286:           data[j,pmatch("Lower",names(data))],"|-----|",
287:           col=rainbow(nlevels(as.factor(data$strata)))[1],
288:           lty=2,
289:           lwd=2)
290:      text(data$time[j],
291:           data[j,pmatch("Upper",names(data))],"|-----|",
292:           col=rainbow(nlevels(as.factor(data$strata)))[1],
293:           lty=2,
294:           lwd=2)
295:    }
296:
297:   }
298:
299: #Further lines
300:   for (i in 2:nlevels(as.factor(data$strata))) {
301:    lines(data$time[data$strata==levels(as.factor(data$strata))[i]],
302:          data$y[data$strata==levels(as.factor(data$strata))[i]],
303:          col=rainbow(length(lev))[i],
304:          lty=2,
305:          lwd=2,
306:          cex=cex)
307:
308:    for (j in 1:dim(data)[1]) {
309:     if (data$strata[j]%in%levels(as.factor(data$strata))[i]) {
310:      text(data$time[j],
311:           data[j,pmatch("Lower",names(data))],"|-----|",
312:           col=rainbow(length(lev))[i],
313:           lty=2,
314:           lwd=2)
```

```
315:       text(data$time[j],
316:            data[j,pmatch("Upper",names(data))],"|-----|",
317:            col=rainbow(length(lev))[i],
318:            lty=2,
319:            lwd=2)
320:      }
321:
322:     }
323:
324:
325:    }
326:    axis(1,levels(data$times))
327:    leg<-rep(NA,nlevels(as.factor(data$strata)))
328:    for (i in 1:nlevels(as.factor(data$strata))) {
329:     leg[i]<-lev[i]
330:    }
331:    x<-legend("topright", leg, col = rainbow(nlevels(as.factor(data$strata))),
332:              lty=2, lwd=2,pt.bg="white",cex=cex,title=descvar)
333:    dev.off()
334:
335:    write("PNG file: Done",file="")
336:    if (nchar(logfile)>0) {
337:     write("PNG file: done",file=logfile,append=TRUE)
338:    }
339:
340:    write("Preparing to do PDF file",file="")
341:    if (nchar(logfile)>0) {
342:     write("Preparing to do PDF file",file=logfile,append=TRUE)
343:    }
344:
345:    file = paste(dirgraph,"/","/i",namegraph,"g4_5a.pdf",sep="")
346:    pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
347:    #First line
348:    plot(data$times[data$strata==levels(as.factor(data$strata))[1]],
349:         data$y[data$strata==levels(as.factor(data$strata))[1]],
350:         type="l",
351:         col=rainbow(nlevels(as.factor(data$strata)))[1],
352:         ylim=c(min(data$Lower,na.rm=TRUE)-sd(data$y,na.rm=TRUE),
353:                max(data$Upper,na.rm=TRUE)+3*nlevels(as.factor(data$strata))),
354:         xlab=descvar[2],
355:         ylab="",
356:         lty=2,
357:         cex=cex)
358:
359: #Confidence boundary for the first line
```

```
360:   for (j in 1:dim(data)[1]) {
361:    if (data$strata[j]==levels(as.factor(data$strata))[1]) {
362:     text(data$time[j],
363:          data[j,pmatch("Lower",names(data))],"|-----|",
364:          col=rainbow(nlevels(as.factor(data$strata)))[1],
365:          lty=2,
366:          lwd=2)
367:     text(data$time[j],
368:          data[j,pmatch("Upper",names(data))],"|-----|",
369:          col=rainbow(nlevels(as.factor(data$strata)))[1],
370:          lty=2,
371:          lwd=2)
372:    }
373:
374:   }
375:
376:
377: #Further lines
378:   for (i in 2:nlevels(as.factor(data$strata))) {
379:    lines(data$time[data$strata==levels(as.factor(data$strata))[i]],
380:          data$y[data$strata==levels(as.factor(data$strata))[i]],
381:          col=rainbow(length(lev))[i],
382:          lty=2,
383:          lwd=2,
384:          cex=cex)
385:
386:    for (j in 1:dim(data)[1]) {
387:     if (data$strata[j]%in%levels(as.factor(data$strata))[i]) {
388:       text(data$time[j],
389:            data[j,pmatch("Lower",names(data))],"|-----|",
390:            col=rainbow(length(lev))[i],
391:            lty=2,
392:            lwd=2)
393:       text(data$time[j],
394:            data[j,pmatch("Upper",names(data))],"|-----|",
395:            col=rainbow(length(lev))[i],
396:            lty=2,
397:            lwd=2)
398:     }
399:
400:
401:    }
402:
403:
404:   }
```

```
  405:
  406:   axis(1,levels(data$times))
  407:   leg<-rep(NA,nlevels(as.factor(data$strata)))
  408:   for (i in 1:nlevels(as.factor(data$strata))) {
  409:    leg[i]<-lev[i]
  410:   }
  411:   x<-legend("topright", leg,
  412:             col = rainbow(nlevels(as.factor(data$strata))),lty=2, lwd=2,
  413:             pt.bg="white",cex=cex,title=descvar)
  414:   dev.off()
  415:
  416:   write("PDF file: Done",file="")
  417:   if (nchar(logfile)>0) {
  418:    write("PDF file: done",file=logfile,append=TRUE)
  419:   }
  420:
  421:   if (struc==TRUE) {
  422:     cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
  423:     cat("<tbody>",file=sthtml,fill=TRUE,append=TRUE)
  424:     cat("<tr>",file=sthtml,fill=FALSE,append=TRUE)
  425:     cat(paste("<td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/lines",namegraph,".png\">","</td>",sep=""
),file=sthtml,fill=FALSE,append=TRUE)
  426:     cat("</tr>",file=sthtml,fill=TRUE,append=TRUE)
  427:     cat("</tbody>",file=sthtml,fill=TRUE,append=TRUE)
  428:     cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
  429:   }
  430:
  431:
  432: }
  433:
  434:
  435: ##############################################################################
  436: #  BIRO_table_print                                                           #
  437: ##############################################################################
  438: # ------------------------
  439: # Authors:
  440: # Luca Rossi       <research@fabcarinci.net)
  441: # Fabrizio Carinci <research@fabcarinci.net)
  442: # Institutions: Serectrix snc
  443: # Created: 2007-11-21
  444: # Version: 2008-11-24
  445: # OS: Windows XP/Vista
  446: # Programming Language: R 2.8.0
  447: # ------------------------
  448: # DESCRIPTION
```

```
449: #
450: # BIRO_table_print print table in html/latex2e format
451: #
452: # DEPENDENCIES
453: #
454: # R package: base
455: #
456: ##############################################################################
457: # PARAMETERS(!=required)
458: #
459: #tab       !              = dataset as the type of the result dataset producted
460: #                          by the BIRO_tab
461: #var       !              = two or more vector with the values of the target
462: #                          variables, Passed as a list
463: #n         !              = a vector which each element is the number of levels
464: #                          of each categorical variables
465: #FirstLab  !              = the name of the first variable, the first variable
466: #                          of var is the vertical variable in the table
467: #lev       !              = a list containing the vactor with che levels of the
468: #                          categorical variable (list)
469: #texfile   !              = the path of the output texfile
470: #codtab    !              = the coded name of the output csv file
471: #dirtables !              = directory for the output html table file
472: #dircsv    !              = directory for the output csv file
473: #caption   !              = string for the caption
474: #side      [0]            = if side is egual to 1 to table in the pdf file is
475: #                          printed in portrait layout
476: #width     [1]            = width for the pdf (see latex for details)
477: # perc     [TRUE]         = TRUE for percentage
478: #dirout    [getwd()]      = output location
479: #out       [out]          = name of output csv dataset
480: #ext       [csv]          = output file format
481: #date      [Sys.Date()]   = date variable or value
482: #ind       !              = a string containing the index of indicator
483: #chisq                    = a vector containing the name of variables to
484: #                          calculate the Cochran-Mantel-Haenszel Chi-Squared
485: #                          Test for Count Data
486: ##############################################################################
487:
488:
489: BIRO_table_print<-function(tab,
490:                           var,
491:                           n,
492:                           FirstLab,
493:                           lev=list(),
```

```
494:                            texfile,
495:                            codtab,
496:                            dirtables,
497:                            dircsv,
498:                            caption,
499:                            side=0,
500:                            width=1,
501:                            perc=TRUE,
502:                            ind=ind,
503:                            chisq=chisq) {
504:
505:
506:  sperc=""
507:  hsperc=""
508:  tab<-as.data.frame(tab)
509:  nvar=length(var)
510:
511:  INDEX<-list()
512:  for (i in 1:(nvar-1)) {
513:   if (i==1) {j=i-1}
514:    if (i==2) {
515:     j=3
516:    } else {j=j+1}
517:   INDEX[[i]]<-tab[,j]
518:  }
519:
520:
521:  sums<-as.data.frame(aggregate(tab$Freq,INDEX,FUN="sum"))
522:  x<-1:nvar
523:  names(sums)[-nvar]<-var[-2]
524:  names(sums)[nvar]<-"sum"
525:  tab<-merge(sums,tab,by=var[-2])
526:  tab$perc<-round((tab$Freq/tab$sum)*100,2)
527:  sperc=" \\% "
528:  hsperc="(%)"
529:  tab<-as.data.frame(tab)
530:  tab$Freq[is.na(tab$Freq)]<-0
531:  tab$perc[is.nan(tab$perc)]<-0
532:  tab$sum[is.na(tab$sum)]<-0
533:
534:  #latex
535:  tab<-sort(tab, by = as.formula(paste("~",paste(var[c(1,nvar:2)],collapse="+"))))
536:  l<-list()
537:  p<-list()
538:  s<-list()
```

```
539:  k=0
540:  for (i in 1:(dim(tab)[1]/n[1])) {
541:   l[[i]]<-tab$Freq[seq(i,dim(tab)[1],(dim(tab)[1]/n[1]))]
542:   p[[i]]<-tab$perc[seq(i,dim(tab)[1],(dim(tab)[1]/n[1]))]
543:   s[[i]]<-tab$sum[seq(i,dim(tab)[1],(dim(tab)[1]/n[1]))]
544:  }
545:  t<-unlist(l)
546:  t<-matrix(t,n[1],length(l))
547:  rowsum<-apply(t,1, sum)
548:  colsum<-apply(t,2, sum)
549:  for (i in 1:4) {
550:   if (is.na(n[i]==TRUE)) n[i]<-1
551:  }
552:  cat("\\begin{center}",file=texfile,fill = TRUE,append=TRUE)
553:  if (side==1) {
554:   cat("\\begin{sidewaystable}",file=texfile,fill = TRUE,append=TRUE)
555:  }
556:  cat("\\parbox[2]{",width,"\\textwidth}{",file=texfile,fill = TRUE,append=TRUE)
557:  cat("\\resizebox{",width,"\\textwidth}{!}{",file=texfile,fill = TRUE,
558:      append=TRUE)
559:  cat("\\begin{tabular}{|c",rep("|r",prod(n[2:4])+(prod(n[2:4])/n[2])+1),"|}",
560:      file=texfile,fill = TRUE,append=TRUE)
561:  cat("\\hline",file=texfile,fill = TRUE,append=TRUE)
562:  cat("\\multirow{",nvar-1,"}{*}{",FirstLab,"}  ",file=texfile,fill = FALSE,
563:      append=TRUE)
564:  if (n[4]>=1 & nvar==4) {
565:   cat("& \\multicolumn{",prod(n[2:3])+(prod(n[2:4])/n[2])+1,"}{|c|}{",
566:       lev[[4]][1],"} \\\\",file=texfile,fill = TRUE,append=TRUE)
567:   cat("\\cline{2-",prod(n[2:4])+(prod(n[2:4])/n[2])+2,"}",file=texfile,
568:       fill = TRUE,append=TRUE)
569:  }
570:
571: #Terza variabile
572:
573: if (nvar>=3) {
574: second_head<-rep(lev[[3]],prod(n[4]))
575: cat(" & ",file=texfile,fill = FALSE,append=TRUE)
576: if (n[3]>1) {
577:  for(i in 1:(length(second_head)-1)) {
578:   cat("\\multicolumn{",n[2]+1,"}{|c|}{",second_head[i],"} &",file=texfile,
579:       fill = FALSE,append=TRUE)
580:  }
581: }
582:  cat("\\multicolumn{",n[2]+1,"}{|c|}{",second_head[length(second_head)],
583:      "}  & \\\\",file=texfile,fill = TRUE,append=TRUE)
```

```
584:
585:  cat("\\cline{2-",prod(n[2:4])+(prod(n[2:4])/n[2])+2,"}",file=texfile,
586:      fill = TRUE,append=TRUE)
587: }
588:
589: if (n[2]>1) {
590: third_head<-rep(lev[[2]],prod(n[3:4]))
591:
592: for(i in 1:(length(third_head)-1)) {
593: cat(" & ",third_head[i],"(",sperc,")",file=texfile,fill = FALSE,append=TRUE)
594: if (i %in% seq(0,prod(n[2:nvar])-1,by=n[2])) {
595: cat("& PT ",file=texfile,fill = FALSE,append=TRUE)
596:  }
597: }
598: cat(" & ",third_head[length(third_head)],"(",sperc,") & PT &\\\\",file=texfile,
599:     fill = TRUE,append=TRUE)
600:
601: }
602: cat("\\hline",file=texfile,fill = TRUE,append=TRUE)
603:
604:
605: for (i in 1:n[1]) {  #righe
606:
607: cat(lev[[1]][i],"  &",file=texfile,fill=FALSE,append=TRUE)
608: for(j in 1:(length(l)-1)) {
609:
610:
611: cat(l[[j]][i]," (",p[[j]][i],") & ",file=texfile,fill=FALSE,append=TRUE)
612:
613: if (j %in% seq(0,prod(n[2:nvar])-1,by=n[2])) {
614:  cat(s[[j]][i], " & ",file=texfile,fill=FALSE,append=TRUE)
615: }
616:
617: }
618: cat(l[[length(l)]][i],"(",p[[length(l)]][i],") & ",s[[length(l)]][i], " & ",
619:     rowsum[i],"  \\\\",file=texfile,fill=TRUE,append=TRUE)
620: }
621:
622: cat("\\hline",file=texfile,fill = TRUE,append=TRUE)
623: cat(" & ",file=texfile,fill = FALSE,append=TRUE)
624:
625: for (i in 1:length(colsum)) {  #totale in fondo
626:
627:  cat(colsum[i],"&",file=texfile,fill = FALSE,append=TRUE)
628:
```

```
629:   if (i %in% seq(0,prod(n[2:nvar])-1,by=n[2])) {
630:
631:  cat(" & ",file=texfile,fill=FALSE,append=TRUE)
632:
633: }
634:
635: }
636: cat("&",round(sum(colsum)),"\\\\",file=texfile,fill = TRUE,append=TRUE)
637: cat("\\hline",file=texfile,fill = TRUE,append=TRUE)
638: cat("\\end{tabular}",file=texfile,fill = TRUE,append=TRUE)
639: cat("}",file=texfile,fill = TRUE,append=TRUE)
640: cat("\\scriptsize",file=texfile,fill = TRUE,append=TRUE)
641: cat(caption,file=texfile,fill = TRUE,append=TRUE)
642: cat("\\vspace{5mm}",file=texfile,fill = TRUE,append=TRUE)
643: cat("}",file=texfile,fill = TRUE,append=TRUE)
644: if (side==1) {
645: cat("\\end{sidewaystable}",file=texfile,fill = TRUE,append=TRUE)
646: }
647: cat("\\end{center}",file=texfile,fill = TRUE,append=TRUE)
648:
649: # HTML table
650:
651: htmlfile=paste(dirtables,"/i",codtab,"d1_3a.html",sep="")
652: htmlfilep=paste(dirtables,"/i",codtab,"d1_3b.html",sep="")
653:
654:
655: write("Writing HTML file",file="")
656:
657: cat("",file=htmlfile,fill = TRUE,append=TRUE)
658: cat("     <table border=\"1\">",file=htmlfile,fill = TRUE,append=TRUE)
659: cat("",file=htmlfile,fill = TRUE,append=TRUE)
660: cat("      <tbody>",file=htmlfile,fill = TRUE,append=TRUE)
661: cat("",file=htmlfile,fill = TRUE,append=TRUE)
662: cat("       <thead>",file=htmlfile,fill = TRUE,append=TRUE)
663: cat("",file=htmlfile,fill = TRUE,append=TRUE)
664:
665: cat("        <tr>",file=htmlfile,fill = TRUE,append=TRUE)
666: cat("         <th rowspan=\"",nvar-1,"\">",FirstLab,"</th>",file=htmlfile,
667:     fill = TRUE,append=TRUE)
668:
669: if (n[4]>1) {
670:
671: for(i in 1:(n[4]-1)) {
672:
673: cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][i],"</th>",file=htmlfile,
```

```
674:     fill = TRUE,append=TRUE)
675:
676: }
677:
678: cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][n[4]]," </th>",
679:     file=htmlfile,fill = TRUE,append=TRUE)
680: cat("         <th></th>",file=htmlfile,fill = TRUE,append=TRUE)
681: }
682:
683: if ((length(lev)==4) & n[4]==1) {
684: cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][1],"</th>",
685:     file=htmlfile,fill = TRUE,append=TRUE)
686: cat("         <th></th>",file=htmlfile,fill = TRUE,append=TRUE)
687:
688:
689: }
690:
691: if (nvar>=3) {
692: second_head<-rep(lev[[3]],prod(n[4]))
693: if (nvar==4) {
694:  cat("         <tr>",file=htmlfile,fill = TRUE,append=TRUE)
695: }
696: if (n[3]>1) {
697:  for(i in 1:(length(second_head)-1)) {
698:  cat("         <th colspan=\"",n[2],"\">",second_head[i],"</th>",file=htmlfile,
699:      fill = TRUE,append=TRUE)
700:  }
701: }
702:  cat("         <th colspan=\"",n[2],"\">",second_head[length(second_head)],
703:      "</th>",file=htmlfile,fill = TRUE,append=TRUE)
704: cat("         <th></th>",file=htmlfile,fill = TRUE,append=TRUE)
705: cat("         </tr>",file=htmlfile,fill = TRUE,append=TRUE)
706: }
707:
708: if (n[2]>1) {
709: third_head<-rep(lev[[2]],prod(n[3:4]))
710: if (nvar==3) {
711:
712:  cat("",file=htmlfile,fill = TRUE,append=TRUE)
713:  cat("         <tr>",file=htmlfile,fill = FALSE,append=TRUE)
714: }
715: if (n[3]>=1) {
716:  for(i in 1:(length(third_head)-1)) {
717:   cat("          <th>",third_head[i],"</th>",file=htmlfile,fill = TRUE,append=TRUE)
718:  }
```

```
  719: }
  720: cat("          <th>",third_head[length(third_head)]," </th>",
  721:     file=htmlfile,fill = TRUE,append=TRUE)
  722: cat("          <th></th>",file=htmlfile,fill = TRUE,append=TRUE)
  723: cat("         </tr>",file=htmlfile,fill = TRUE,append=TRUE)
  724: }
  725:
  726: for (i in 1:n[1]) {  #rows
  727:
  728: cat("",file=htmlfile,fill = TRUE,append=TRUE)
  729: cat("        <tr>",file=htmlfile,fill=TRUE,append=TRUE)
  730: cat("         <td class=\"catcol\">",lev[[1]][i],"</td>",file=htmlfile,fill=TRUE,append=TRUE)
  731: cat("         <td>",file=htmlfile,fill=FALSE,append=TRUE)
  732:
  733: for(j in 1:(length(l)-1)) {
  734:
  735:
  736: cat(l[[j]][i]," (",format(round(p[[j]][i],digits=2),nsmall=2,width=6),"%) </td> "
,file=htmlfile,fill=TRUE,append=TRUE)
  737: cat("         <td>",file=htmlfile,fill=FALSE,append=TRUE)
  738: }
  739: cat(l[[length(l)]][i]," (",format(round(p[[length(l)]][i],digits=2),nsmall=2,width=6),"%) <td class=\"rowsn\" > "
,rowsum[i],"( ",round(100*rowsum[i]/round(sum(colsum)),digits=2),"%)  </td>",
  740:     file=htmlfile,fill=TRUE,append=TRUE)
  741: cat("        </tr>",file=htmlfile,fill=TRUE,append=TRUE)
  742: }
  743:
  744: cat("",file=htmlfile,fill = TRUE,append=TRUE)
  745: cat("        <tr>",file=htmlfile,fill = TRUE,append=TRUE)
  746: cat("         <td></td>",file=htmlfile,fill = TRUE,append=TRUE)
  747: cat("         <td class=\"colsn\">",file=htmlfile,fill = FALSE,append=TRUE)
  748: for (i in 1:length(colsum)) {  #total last row
  749:  cat(colsum[i]," (",round(100*(colsum[i]/round(sum(colsum))),digits=2),"%) </td>",file=htmlfile,fill =
TRUE,append=TRUE)
  750:  if (i<length(colsum)) {
  751:   cat("         <td class=\"colsn\">",file=htmlfile,fill = FALSE,append=TRUE)
  752:  }
  753:  if (i==length(colsum)) {
  754:   cat("         <td class=\"grandn\">",file=htmlfile,fill = FALSE,append=TRUE)
  755:  }
  756:
  757: }
  758: cat(round(sum(colsum)),"</td>",file=htmlfile,fill = TRUE,append=TRUE)
  759: cat("        </tr>",file=htmlfile,fill = TRUE,append=TRUE)
  760: cat("",file=htmlfile,fill = TRUE,append=TRUE)
```

```
761: cat("        </tbody>",file=htmlfile,fill = TRUE,append=TRUE)
762: cat("",file=htmlfile,fill = TRUE,append=TRUE)
763: cat("       </table>",file=htmlfile,fill = TRUE,append=TRUE)
764: cat("",file=htmlfile,fill = TRUE,append=TRUE)
765: cat("       <br>",file=htmlfile,fill = TRUE,append=TRUE)
766: cat("",file=htmlfile,fill = TRUE,append=TRUE)
767:
768:  if (struc==TRUE) {
769:
770:   cat("",file=sthtml,fill = TRUE,append=TRUE)
771:   cat("     <table border=\"1\">",file=sthtml,fill = TRUE,append=TRUE)
772:   cat("",file=sthtml,fill = TRUE,append=TRUE)
773:   cat("      <tbody>",file=sthtml,fill = TRUE,append=TRUE)
774:   cat("",file=sthtml,fill = TRUE,append=TRUE)
775:   cat("       <thead>",file=sthtml,fill = TRUE,append=TRUE)
776:   cat("",file=sthtml,fill = TRUE,append=TRUE)
777:
778:   cat("        <tr>",file=sthtml,fill = TRUE,append=TRUE)
779:   cat("         <th rowspan=\"",nvar-1,"\">",FirstLab,"</th>",file=sthtml,
780:       fill = TRUE,append=TRUE)
781:
782:   if (n[4]>1) {
783:
784:
785:    for(i in 1:(n[4]-1)) {
786:
787:     cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][i],"</th>",file=sthtml,
788:         fill = TRUE,append=TRUE)
789:
790:    }
791:
792:    cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][n[4]]," </th>",
793:        file=sthtml,fill = TRUE,append=TRUE)
794:    cat("         <th></th>",file=sthtml,fill = TRUE,append=TRUE)
795:   }
796:
797:   if ((length(lev)==4) & n[4]==1) {
798:    cat("         <th colspan=\"",prod(n[2:3]),"\">",lev[[4]][1],"</th>",
799:        file=sthtml,fill = TRUE,append=TRUE)
800:    cat("         <th></th>",file=sthtml,fill = TRUE,append=TRUE)
801:    cat("        </tr>",file=sthtml,fill = TRUE,append=TRUE)
802:   }
803:
804:   if (nvar>=3) {
805:    second_head<-rep(lev[[3]],prod(n[4]))
```

```
806:     if (nvar==4) {
807:      cat("",file=sthtml,fill = TRUE,append=TRUE)
808:      cat("        <tr>",file=sthtml,fill = TRUE,append=TRUE)
809:     }
810:     for(i in 1:(length(second_head)-1)) {
811:      cat("         <th colspan=\"",n[2],"\">",second_head[i],"</th>",file=sthtml,
812:          fill = TRUE,append=TRUE)
813:     }
814:     cat("         <th colspan=\"",n[2],"\">",second_head[length(second_head)],
815:         "</th>",file=sthtml,fill = TRUE,append=TRUE)
816:     cat("         <th></th>",file=sthtml,fill = TRUE,append=TRUE)
817:     cat("        </tr>",file=sthtml,fill = TRUE,append=TRUE)
818:    }
819:
820:    if (n[2]>1) {
821:     third_head<-rep(lev[[2]],prod(n[3:4]))
822:     if (nvar>=3) {
823:      cat("",file=sthtml,fill = TRUE,append=TRUE)
824:      cat("        <tr>",file=sthtml,fill = TRUE,append=TRUE)
825:     }
826:     for(i in 1:(length(third_head)-1)) {
827:      cat("         <th>",third_head[i],"</th>",file=sthtml,fill = TRUE,append=TRUE)
828:     }
829:     if (nvar>=3) {
830:      cat("         <th>",third_head[length(third_head)]," </th>",
831:          file=sthtml,fill = TRUE,append=TRUE)
832:     }
833:     cat("         <th></th>",file=sthtml,fill = TRUE,append=TRUE)
834:     cat("        </tr>",file=sthtml,fill = TRUE,append=TRUE)
835:    }
836:    cat("       </thead>",file=sthtml,fill = TRUE,append=TRUE)
837:    for (i in 1:n[1]) {  #rows
838:
839:     cat("",file=sthtml,fill = TRUE,append=TRUE)
840:     cat("       <tr>",file=sthtml,fill=TRUE,append=TRUE)
841:     cat("        <td class=\"catcol\">",lev[[1]][i],"</td>",file=sthtml,fill=TRUE,append=TRUE)
842:     cat("        <td>",file=sthtml,fill=FALSE,append=TRUE)
843:
844:     for(j in 1:(length(l)-1)) {
845:
846:      cat("n"," (","p","%) </td> ",file=sthtml,fill=TRUE,append=TRUE)
847:      cat("        <td>",file=sthtml,fill=FALSE,append=TRUE)
848:     }
849:      cat("n"," (","p","%) <td class=\"rowsn\" > ","N","  </td>",
850:          file=sthtml,fill=TRUE,append=TRUE)
```

```
851:     cat("       </tr>",file=sthtml,fill=TRUE,append=TRUE)
852:   }
853:
854:   cat("",file=sthtml,fill = TRUE,append=TRUE)
855:   cat("       <tr>",file=sthtml,fill = TRUE,append=TRUE)
856:   cat("        <td></td>",file=sthtml,fill = TRUE,append=TRUE)
857:   cat("        <td class=\"colsn\">",file=sthtml,fill = FALSE,append=TRUE)
858:   for (i in 1:length(colsum)) {  #totale in fondo
859:    cat("N","</td>",file=sthtml,fill = TRUE,append=TRUE)
860:    if (i<length(colsum)) {
861:     cat("        <td class=\"colsn\">",file=sthtml,fill = FALSE,append=TRUE)
862:    }
863:    if (i==length(colsum)) {
864:     cat("        <td class=\"grandn\">",file=sthtml,fill = FALSE,append=TRUE)
865:    }
866:
867:   }
868:   cat("N","</td>",file=sthtml,fill = TRUE,append=TRUE)
869:   cat("       </tr>",file=sthtml,fill = TRUE,append=TRUE)
870:   cat("",file=sthtml,fill = TRUE,append=TRUE)
871:   cat("       </tbody>",file=sthtml,fill = TRUE,append=TRUE)
872:   cat("",file=sthtml,fill = TRUE,append=TRUE)
873:   cat("      </table>",file=sthtml,fill = TRUE,append=TRUE)
874:   cat("",file=sthtml,fill = TRUE,append=TRUE)
875:   cat("      <br>",file=sthtml,fill = TRUE,append=TRUE)
876:   cat("",file=sthtml,fill = TRUE,append=TRUE)
877:   cat("       <caption>",caption,"</caption>",file=sthtml,fill = TRUE,append=TRUE)
878:   cat("",file=sthtml,fill = TRUE,append=TRUE)
879:  }
880: #write the csv file
881:
882:  names(tab)[pmatch("Freq",names(tab))]<-"n"
883:  write.csv(tab,paste(dircsv,"/i",codtab,"d1_3a.csv",sep=""),row.names = FALSE)
884:  file.append(paste(dirhtml,"/",ind,".html",sep=""),
885:              htmlfile)
886:
887: if (is.null(chisq)==FALSE) {
888:
889: if (length(na.omit(chisq))==2) {
890:  if (all(tab$n>1)) {
891:  formula<-as.formula(paste("n~",paste(c(var[1],na.omit(chisq[1:2])),collapse="+"),sep=""))
892:     mytable<-xtabs(formula=formula,data=tab)
893:
894:     #Cochran-Mantel-Haenszel Chi-Squared Test for Count Data
895:
```

```
896:
897:     mh<-mantelhaen.test(mytable)
898:     chitab<-NULL
899:     chitab<-as.data.frame(cbind(mh$statistic,format(mh$p.value,nsmall=4),mh$parameter))
900:     names(chitab)<-c("Chi-Squared","p.value","df")
901:
902:     row.names(chitab)="C-M-H Chi-Squared Test"
903:
904:     write.csv(chitab,paste(dircsv,"/i",codtab,"1_3b.csv",sep=""),row.names = FALSE)
905:
906:     BIRO_df2html(data=chitab,
907:                  htmlfile=htmlfilep,
908:                  catcol=NULL)
909:
910:  file.append(paste(dirhtml,"/",ind,".html",sep=""),
911:              htmlfilep)
912:
913:    for (j in 1:length(names(chitab))) {
914:     names(chitab)[j]<-paste(unlist(strsplit(names(chitab)[j],"_")),collapse="\\_")
915:    }
916:
917:    latex(chitab,file=texfile,append=TRUE,table.env=FALSE)
918:
919:   } else {
920:    chitab<-NULL
921:    chitab<-as.data.frame("One or more cells have less than 2 obs")
922:    names(chitab)<-"C-M-H Chi-Squared Test"
923:    latex(chitab,file=texfile,append=TRUE,table.env=FALSE)
924:    BIRO_df2html(data=chitab,
925:                 htmlfile=htmlfilep,
926:                 catcol=NULL)
927:
928:    file.append(paste(dirhtml,"/",ind,".html",sep=""),
929:                htmlfilep)
930:
931:    }
932: }
933: if (length(na.omit(chisq))==1) {
934:  formula<-as.formula(paste("n~",paste(c(var[1],na.omit(chisq)),collapse="+"),sep=""))
935:     mytable<-xtabs(formula=formula,data=tab)
936:     #Pearson Chi-Squared Test for Count Data
937:      mh<-chisq.test(mytable)
938:
939:     chitab<-as.data.frame(cbind(mh$statistic,format(mh$p.value,nsmall=4),mh$parameter))
940:     names(chitab)<-c("Chi-Squared","p.value","df")
```

```
941:     if (nvar==4) {
942:      row.names(chitab)="C-M-H Chi-Squared Test"
943:     }
944:
945:     write.csv(chitab,paste(dircsv,"/i",codtab,"1_3b.csv",sep=""),row.names = FALSE)
946:
947:    for (j in 1:length(names(chitab))) {
948:     names(chitab)[j]<-paste(unlist(strsplit(names(chitab)[j],"_")),collapse="\\_")
949:    }
950:   } else {
951:    chitab<-as.data.frame("One or more cells have less than 2 obs")
952:    names(chitab)<-"C-M-H Chi-Squared Test"
953:   }
954:  }
955:  cat("        <caption>",caption,"</caption>",file=htmlfile,fill = TRUE,append=TRUE)
956:  cat("",file=htmlfile,fill = TRUE,append=TRUE)
957: }
958:
959:
960: ##############################################################################
961: #  BIRO_df2html                                                              #
962: ##############################################################################
963: # ------------------------
964: # Authors:
965: # Luca Rossi       <redsluke@gmail.com>
966: # Fabrizio Carinci <research@fabcarinci.net>
967: # Institutions: Serectrix snc
968: # Created: 2008-11-21
969: # Version: 2008-10-23
970: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
971: # Programming Language: R 2.8.0
972: ## ------------------------
973: # DESCRIPTION
974: #
975: # BIRO_df2html writes a data.frame into an html table
976: #
977: # DEPENDENCIES
978: #
979: # R package: base
980: #
981: ##############################################################################
982: # PARAMETERS(!=required)
983: #
984: #data       !              =  a string containing the name of a t data.frame
985: #
```

```
 986: #htmlfile   !              =  a string containing the path and the name of the
 987: #                             html file
 988: #catcol     !              =  the position (into the data.frame) of the
 989: #                             category variables
 990: #
 991: ###############################################################################
 992:
 993:
 994: BIRO_df2html<-function(data,htmlfile,catcol=1) {
 995:  cat("       <table border=\"1\">",file=htmlfile,fill = TRUE,append=TRUE)
 996:  cat("        <tbody>",file=htmlfile,fill = TRUE,append=TRUE)
 997:  cat("         <thead>",file=htmlfile,fill = TRUE,append=TRUE)
 998:  headerpos<-1:length(names(data))
 999:  if (is.null(catcol)==FALSE) {
1000:   headerpos<-headerpos[-catcol]
1001:  }
1002:  cat("          <tr>",file=htmlfile,fill = TRUE,append=TRUE)
1003:  for (j in catcol) {
1004:   cat("          <th>",file=htmlfile,fill = FALSE,append=TRUE)
1005:   cat(names(data)[j],file=htmlfile,fill = FALSE,append=TRUE)
1006:   cat("          </th>",file=htmlfile,fill = TRUE,append=TRUE)
1007:  }
1008:  for (i in headerpos) {
1009:   cat("          <th>",file=htmlfile,fill = FALSE,append=TRUE)
1010:   cat(names(data)[i],file=htmlfile,fill = FALSE,append=TRUE)
1011:   cat("          </th>",file=htmlfile,fill = TRUE,append=TRUE)
1012:  }
1013:  cat("          </tr>",file=htmlfile,fill = TRUE,append=TRUE)
1014:  cat("         </thead>",file=htmlfile,fill = TRUE,append=TRUE)
1015:  for (k in 1:dim(data)[1]) {
1016:  cat("         <tr>",file=htmlfile,fill = TRUE,append=TRUE)
1017:  for (j in catcol) {
1018:   cat("          <td class=\"catcol\">",file=htmlfile,fill = FALSE,append=TRUE)
1019:   cat(as.character(data[k,j]),file=htmlfile,fill = FALSE,append=TRUE)
1020:   cat("          </td>",file=htmlfile,fill = TRUE,append=TRUE)
1021:  }
1022:  for (i in headerpos) {
1023:    cat("         <td>",file=htmlfile,fill = FALSE,append=TRUE)
1024:    cat(as.character(data[k,i]),file=htmlfile,fill = FALSE,append=TRUE)
1025:    cat("         </td>",file=htmlfile,fill = TRUE,append=TRUE)
1026:   }
1027:  cat("         </tr>",file=htmlfile,fill = TRUE,append=TRUE)
1028:  }
1029:  cat("        </tbody>",file=htmlfile,fill = TRUE,append=TRUE)
1030:  cat("       </table>",file=htmlfile,fill = TRUE,append=TRUE)
```

```
1031:  cat("       <br>",file=sthtml,fill = TRUE,append=TRUE)
1032:  cat("       <br>",file=sthtml,fill = TRUE,append=TRUE)
1033:
1034:  if (struc==1) {
1035:
1036:   cat("       <table border=\"1\">",file=sthtml,fill = TRUE,append=TRUE)
1037:   cat("        <tbody>",file=sthtml,fill = TRUE,append=TRUE)
1038:   cat("         <thead>",file=sthtml,fill = TRUE,append=TRUE)
1039:   headerpos<-1:length(names(data))
1040:   if (is.null(catcol)==FALSE) {
1041:    headerpos<-headerpos[-catcol]
1042:   }
1043:   cat("          <tr>",file=sthtml,fill = TRUE,append=TRUE)
1044:   for (j in catcol) {
1045:    cat("           <th>",file=sthtml,fill = FALSE,append=TRUE)
1046:    cat(names(data)[j],file=sthtml,fill = FALSE,append=TRUE)
1047:    cat("           </th>",file=sthtml,fill = TRUE,append=TRUE)
1048:   }
1049:   for (i in headerpos) {
1050:    cat("           <th>",file=sthtml,fill = FALSE,append=TRUE)
1051:    cat(names(data)[i],file=sthtml,fill = FALSE,append=TRUE)
1052:    cat("           </th>",file=sthtml,fill = TRUE,append=TRUE)
1053:   }
1054:   cat("          </tr>",file=sthtml,fill = TRUE,append=TRUE)
1055:   cat("         </thead>",file=sthtml,fill = TRUE,append=TRUE)
1056:   for (k in 1:dim(data)[1]) {
1057:   cat("          <tr>",file=sthtml,fill = TRUE,append=TRUE)
1058:   for (j in catcol) {
1059:    cat("           <td class=\"catcol\">",file=sthtml,fill = FALSE,append=TRUE)
1060:    cat(as.character(data[k,j]),file=sthtml,fill = FALSE,append=TRUE)
1061:    cat("           </td>",file=sthtml,fill = TRUE,append=TRUE)
1062:   }
1063:   for (i in headerpos) {
1064:    cat("           <td>",file=sthtml,fill = FALSE,append=TRUE)
1065:    cat("n",file=sthtml,fill = FALSE,append=TRUE)
1066:    cat("           </td>",file=sthtml,fill = TRUE,append=TRUE)
1067:   }
1068:   cat("          </tr>",file=sthtml,fill = TRUE,append=TRUE)
1069:   }
1070:   cat("        </tbody>",file=sthtml,fill = TRUE,append=TRUE)
1071:   cat("       </table>",file=sthtml,fill = TRUE,append=TRUE)
1072:   cat("       <br>",file=sthtml,fill = TRUE,append=TRUE)
1073:   cat("       <br>",file=sthtml,fill = TRUE,append=TRUE)
1074:  }
1075: }
```

```
1076:
1077:
1078: ##############################################################################
1079: # BIRO_drawbars
1080: ##############################################################################
1081: # ------------------------
1082: # Authors:
1083: # Luca Rossi       <redsluke@gmail.com>
1084: # Fabrizio Carinci <research@fabcarinci.net>
1085: # Institutions: Serectrix snc
1086: # Created: 2007-11-21
1087: # Version: 2008-11-23
1088: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
1089: # Programming Language: R 2.8.0
1090: # ------------------------
1091: # DESCRIPTION
1092: #
1093: # BIRO_barplot prepares data for BIRO_drawbars
1094: #
1095: # DEPENDENCIES
1096: #
1097: #  R packages: grDevice, graphics, Cairo
1098: #
1099: ##############################################################################
1100: # PARAMETERS(!=required)
1101: #
1102: #tab        !             = a data.frame from BIRO_barplot
1103: #labelvar   !             = the label of the variable
1104: #lev        !             = a list containing the vactor with che levels of the
1105: #                           categorical variable (list)
1106: #beside     [TRUE]        = a logical value. If FALSE, the columns of height
1107: #                           are portrayed as stacked bars,
1108: #                           and if TRUE the columns are portrayed as juxtaposed
1109: #                           bars.
1110: #number     [FALSE]       = a logical value. If TRUE values are printed over
1111: #                           each bar, FALSE otherwise.
1112: #perc       [TRUE]        = a logical value. If TRUE percentages are calculated.
1113: #what       ["patients"]  = the subject of the representation (this is for the
1114: #                           label)
1115: #dirgraph   !             = the path of the output (jpg,eps,png files)
1116: #namegraph  !             = the coded name of the output files
1117: ##############################################################################
1118:
1119:  BIRO_drawbars<-function(tab,
1120:                          labelvar,
```

```
1121:                           lev=list(),
1122:                           beside=TRUE,
1123:                           number=FALSE,
1124:                           perc=TRUE,
1125:                           what="patients",
1126:                           namegraph,
1127:                           dirgraph,
1128:                           cex=cex) {
1129:
1130:   #correction to user's mistake
1131:   if (beside==FALSE) {
1132:    number=TRUE
1133:   }
1134:
1135:   tab$Perc<-round((tab$n/tab$sum)*100,2)
1136:
1137:   l<-list()
1138:   p<-list()
1139:   for (i in 1:(dim(tab)[1]/nlevels(as.factor(tab[,1])))) {
1140:   start<-(i-1)*nlevels(as.factor(tab[,1]))+1
1141:   l[[i]]<-tab$n[start:(start+nlevels(as.factor(tab[,1]))-1)]
1142:   p[[i]]<-tab$Perc[start:(start+nlevels(as.factor(tab[,1]))-1)]
1143:   }
1144:
1145:
1146:    #Create the summary dataset
1147:   sumdata<-NULL
1148:
1149:   columnnames<-as.data.frame(as.matrix(unique(tab[,1:(dim(tab)[2]-3)])))
1150:   if (perc==FALSE) {
1151:   for (i in 1:1:(dim(tab)[1]/nlevels(as.factor(tab[,1])))) {
1152:    sumdata<-cbind(sumdata,as.vector(l[[i]]))
1153:   }
1154:   }
1155:
1156:   if (perc==TRUE) {
1157:   for (i in 1:1:(dim(tab)[1]/nlevels(as.factor(tab[,1])))) {
1158:    sumdata<-cbind(sumdata,as.vector(p[[i]]))
1159:   }
1160:   }
1161:
1162:   nlevd<-1
1163:
1164:   nvar<-dim(tab)[2]-4
1165:
```

```
1166:   for (i in 2:nvar) {
1167:    nlevd<-nlevd*nlevels(as.factor(tab[,1]))
1168:   }
1169:
1170:   equal=""
1171:   if (length(labelvar)>1 ) {
1172:     if (labelvar[2]>"")
1173:       equal="= " }
1174:   columnnames2<-rep(NA,nlevd)
1175:   for (i in 1:dim(columnnames)[2]) {
1176:    for (k in 1:(length(var)-1)) {
1177:     if (is.na(columnnames2[i])) columnnames2[i]<-
1178:                                 paste(labelvar[k+1],equal,columnnames[i,k],
1179:                                       sep="")
1180:     else columnnames2[i]<-
1181:          paste(columnnames2[i],", ",labelvar[k+1],equal,columnnames[i,k],
1182:          sep="")
1183:    }
1184:   }
1185:
1186:   columnnames<-as.data.frame(columnnames2)
1187:   rm(columnnames2)
1188:   colnames(sumdata)<-as.vector(unique(tab[,2]))
1189:   row.names(sumdata)<-as.vector(unique(tab[,1]))
1190:   sumdata<-as.data.frame(sumdata)
1191:   sumdata$lev<-as.vector(unique(tab[,1]))
1192:   namevar<-
1193:   names(tab)[-c(dim(tab)[2]-3,dim(tab)[2]-2,dim(tab)[2]-1,dim(tab)[2])]
1194:   nrows=1
1195:   for (i in 1:nvar) {
1196:
1197:     tab[,pmatch(namevar[i],names(tab))]<-
1198:     as.factor(tab[,pmatch(namevar[i],names(tab))])
1199:     levels(tab[,pmatch(namevar[i],names(tab))])<-
1200:     lev[[i]]
1201:     if (i>1) nrows<-
1202:              nrows*nlevels(as.factor(tab[,pmatch(namevar[i],names(tab))]))
1203:   }
1204:
1205:   if ((beside==TRUE)) {uplim<-max(tab$n)+sd(tab$n)}
1206:
1207:    if (beside==FALSE) {
1208:     sums<-apply(sumdata[,1:dim(sumdata)[2]-1], 2, sum)
1209:     uplim<-max(as.numeric(as.character(sums)))
1210:    }
```

```
1211:
1212:   if (perc==TRUE) if (uplim>100) uplim=110
1213:   dwlim<-(-(nrows*4)-4)
1214:    #Drawing
1215:
1216:   if (perc==TRUE) {
1217:
1218:     ylab=paste("% of", what)
1219:     }
1220:   if (perc==FALSE) {
1221:     ylab=paste("Frequencies (",what,")",sep="")
1222:     }
1223:
1224:     height<-sumdata[,-dim(sumdata)[2]]
1225:
1226:   height<-as.matrix(height)
1227:   par(cex=cex)
1228:   write("Preparing to do SVG file",file="")
1229:   if (nchar(logfile)>0) {
1230:    write("Preparing to do SVG file",file=logfile,append=TRUE)
1231:   }
1232:   file = paste(dirgraph,"/","i",namegraph,"g4_1a.svg",sep="")
1233:   CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
1234:
1235:
1236:   r <- barplot(height,beside=beside,col=rainbow(length(sumdata$lev)),
1237:                ylim= c(dwlim,uplim),ylab=ylab,angle=90,axisnames=FALSE,
1238:                axes=FALSE,font=1,cex.names=cex,cex.axis=cex)
1239:   legend("topright",legend=lev[[1]],cex=cex,col=rainbow(length(sumdata$lev)),
1240:          pch=1,title=labelvar[1])
1241:
1242:
1243:   if (beside==TRUE) {x<-r[(round(nrow(r)/2))+1,]}
1244:
1245:   if (beside==FALSE) {x<-r}
1246:
1247:   for(i in 1:ncol(sumdata)) {
1248:    s<-strsplit(colnames(sumdata)[i],", ")
1249:    s<-as.vector(s[[1]])
1250:    for (j in 1:length(s)) {
1251:     text(x[i]-0.3,-par("cxy")[2]*j,s[j],font=1,cex=cex)
1252:    }
1253:   }
1254:     if ((beside==TRUE)  & (perc==FALSE)){uplim<-max(height)}
1255:
```

```
1256:     if (perc==TRUE) {uplim=100}
1257:   par(cex=cex)
1258:   axis(2,c(0:uplim),cex.axis=cex)
1259:
1260:
1261:   #write upper or lower to the upper line of a bar
1262:   halfheight<-uplim/2
1263:
1264:   signs<-rep(NA,length(height))
1265:
1266:   signs[height>halfheight]<--1
1267:   signs[height<halfheight]<-1
1268:
1269:   #write numbers
1270:
1271:   if (beside==TRUE & number==TRUE) {
1272:   text(r,height+signs*par("cxy")[2],height,font=1,cex=cex)
1273:   }
1274:
1275:   dev.off()
1276:
1277:   write("SVG file: Done",file="")
1278:   if (nchar(logfile)>0) {
1279:    write("SVG file: Done",file=logfile,append=TRUE)
1280:   }
1281:
1282:   write("Preparing to do JPEG file",file="")
1283:   if (nchar(logfile)>0) {
1284:    write("Preparing to do JPEG file",file=logfile,append=TRUE)
1285:   }
1286:    file = paste(dirgraph,"/","i",namegraph,"g4_1a.jpeg",sep="")
1287:   jpeg(file, width= 813.6, height= 348.48, bg = "transparent",res=9)
1288:   r <- barplot(height,beside=beside,col=rainbow(length(sumdata$lev)),
1289:                ylim= c(dwlim,uplim),ylab=ylab,angle=90,axisnames=FALSE,
1290:                axes=FALSE,font=1,cex.names=cex,cex.axis=cex)
1291:   legend("topright",legend=lev[[1]],cex=cex,col=rainbow(length(sumdata$lev)),
1292:          pch=1,title=labelvar[1])
1293:
1294:
1295:   if (beside==TRUE) {x<-r[(round(nrow(r)/2))+1,]}
1296:
1297:   if (beside==FALSE) {x<-r}
1298:
1299:   for(i in 1:ncol(sumdata)) {
1300:    s<-strsplit(colnames(sumdata)[i],", ")
```

```
1301:     s<-as.vector(s[[1]])
1302:     for (j in 1:length(s)) {
1303:      text(x[i]-0.3,-par("cxy")[2]*j,s[j],font=1,cex=cex)
1304:     }
1305:    }
1306:      if ((beside==TRUE)  & (perc==FALSE)){uplim<-max(height)}
1307:
1308:      if (perc==TRUE) {uplim=100}
1309:    par(cex=cex)
1310:    axis(2,c(0:uplim),cex.axis=cex)
1311:
1312:
1313:    #write upper or lower to the upper line of a bar
1314:    halfheight<-uplim/2
1315:
1316:    signs<-rep(NA,length(height))
1317:
1318:    signs[height>halfheight]<--1
1319:    signs[height<halfheight]<-1
1320:
1321:    #write numbers
1322:
1323:    if (beside==TRUE & number==TRUE) {
1324:    text(r,height+signs*par("cxy")[2],height,font=1,cex=cex)
1325:    }
1326:
1327:    dev.off()
1328:
1329:    write("JPEG file: Done",file="")
1330:    if (nchar(logfile)>0) {
1331:     write("JPEG file: Done",file=logfile,append=TRUE)
1332:    }
1333:
1334:    write("Preparing to do PNG file",file="")
1335:    if (nchar(logfile)>0) {
1336:     write("Preparing to do PNG file",file=logfile,append=TRUE)
1337:    }
1338:
1339:    file = paste(dirgraph,"/","i",namegraph,"g4_1a.png",sep="")
1340:    CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
1341:    r <- barplot(height,beside=beside,col=rainbow(length(sumdata$lev)),
1342:                 ylim= c(dwlim,uplim),ylab=ylab,angle=90,axisnames=FALSE,
1343:                 axes=FALSE,font=1,cex.names=cex,cex.axis=cex)
1344:    legend("topright",legend=lev[[1]],cex=cex,col=rainbow(length(sumdata$lev)),
1345:           pch=1,title=labelvar[1])
```

```
1346:
1347:
1348:   if (beside==TRUE) {x<-r[(round(nrow(r)/2))+1,]}
1349:
1350:   if (beside==FALSE) {x<-r}
1351:
1352:   for(i in 1:ncol(sumdata)) {
1353:    s<-strsplit(colnames(sumdata)[i],", ")
1354:    s<-as.vector(s[[1]])
1355:    for (j in 1:length(s)) {
1356:     text(x[i]-0.3,-par("cxy")[2]*j,s[j],font=1,cex=cex)
1357:    }
1358:   }
1359:     if ((beside==TRUE)  & (perc==FALSE)){uplim<-max(height)}
1360:
1361:     if (perc==TRUE) {uplim=100}
1362:   par(cex=cex)
1363:   axis(2,c(0:uplim),cex.axis=cex)
1364:
1365:
1366:   #write upper or lower to the upper line of a bar
1367:   halfheight<-uplim/2
1368:
1369:   signs<-rep(NA,length(height))
1370:
1371:   signs[height>halfheight]<--1
1372:   signs[height<halfheight]<-1
1373:
1374:   #write numbers
1375:
1376:   if (beside==TRUE & number==TRUE) {
1377:   text(r,height+signs*par("cxy")[2],height,font=1,cex=cex)
1378:   }
1379:
1380:   dev.off()
1381:
1382:   write("PNG file: Done",file="")
1383:   if (nchar(logfile)>0) {
1384:    write("PNG file: Done",file=logfile,append=TRUE)
1385:   }
1386:
1387:   write("Preparing to do PDF file",file="")
1388:   if (nchar(logfile)>0) {
1389:    write("Preparing to do PDF file",file=logfile,append=TRUE)
1390:   }
```

```
1391:
1392:   file = paste(dirgraph,"/","i",namegraph,"g4_1a.pdf",sep="")
1393:   pdf(file, width = 11.33, height =4.84 , pointsize = 7, bg = "transparent")
1394:   r <- barplot(height,beside=beside,col=rainbow(length(sumdata$lev)),
1395:                ylim= c(dwlim,uplim),ylab=ylab,angle=90,axisnames=FALSE,
1396:                axes=FALSE,font=1,cex.names=cex,cex.axis=cex)
1397:   legend("topright",legend=lev[[1]],cex=cex,col=rainbow(length(sumdata$lev)),
1398:          pch=1,title=labelvar[1])
1399:
1400:
1401:   if (beside==TRUE) {x<-r[(round(nrow(r)/2))+1,]}
1402:
1403:   if (beside==FALSE) {x<-r}
1404:
1405:   for(i in 1:ncol(sumdata)) {
1406:    s<-strsplit(colnames(sumdata)[i],", ")
1407:    s<-as.vector(s[[1]])
1408:    for (j in 1:length(s)) {
1409:     text(x[i]-0.3,-par("cxy")[2]*j,s[j],font=1,cex=cex)
1410:    }
1411:   }
1412:     if ((beside==TRUE)  & (perc==FALSE)){uplim<-max(height)}
1413:
1414:     if (perc==TRUE) {uplim=100}
1415:   par(cex=cex)
1416:   axis(2,c(0:uplim),cex.axis=cex)
1417:
1418:
1419:   #write upper or lower to the upper line of a bar
1420:   halfheight<-uplim/2
1421:
1422:   signs<-rep(NA,length(height))
1423:
1424:   signs[height>halfheight]<--1
1425:   signs[height<halfheight]<-1
1426:
1427:   #write numbers
1428:
1429:   if (beside==TRUE & number==TRUE) {
1430:   text(r,height+signs*par("cxy")[2],height,font=1,cex=cex)
1431:   }
1432:   dev.off()
1433:
1434:   write("PDF file: Done",file="")
1435:   if (nchar(logfile)>0) {
```

```
 1436:    write("PDF file: Done",file=logfile,append=TRUE)
 1437:   }
 1438:
 1439:   if (struc==TRUE) {
 1440:     cat("<table border=\"1\">",file=sthtml,fill=TRUE,append=TRUE)
 1441:     cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
 1442:     cat("  <tr>",file=sthtml,fill=FALSE,append=TRUE)
 1443:     cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/","i",namegraph,"g4_1a.png\">","</td>"
,sep=""),file=sthtml,fill=FALSE,append=TRUE)
 1444:     cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
 1445:     cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
 1446:     cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
 1447:   }
 1448:
 1449: }
 1450:
 1451:
 1452:
 1453:
 1454: ##############################################################################
 1455: #  BIRO_plotbox                                                              #
 1456: ##############################################################################
 1457: # ------------------------
 1458: # Authors:
 1459: # Luca Rossi       <redsluke@gmail.com>
 1460: # Fabrizio Carinci <research@fabcarinci.net>
 1461: # Institutions: Serectrix snc
 1462: # Created: 2007-11-21
 1463: # Version: 2008-11-23
 1464: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 1465: # Programming Language: R 2.8.0
 1466: # ------------------------
 1467: # DESCRIPTION
 1468: #
 1469: # BIRO_plotbox draws boxplot for a conditionated distribution
 1470: #
 1471: # DEPENDENCIES
 1472: #
 1473: #  R packages: grDevice, graphics, Cairo
 1474: #
 1475: ##############################################################################
 1476: # PARAMETERS(!=required)
 1477: #
 1478: # bxpdata    !            = dataset containing data
 1479: #
```

```
1480: # lev        !           = a list containing the vactor with che levels of
1481: #                          the categorical variable (list)
1482: # notch      !           = logical. if notch is TRUE, a notch is drawn in
1483: #                          each side of the boxes. If the notches of two
1484: #                          plots do not overlap this is \221strong evidence\222 that
1485: #                          the two means differ (Chambers et al., 1983, p. 62).
1486: #                          See boxplot.stats for the calculations used.
1487: # namegraph  !           = the coded name of the output files
1488: # dirgraph     [current]  = the path of the output (jpg,eps,png files)
1489: # out     [out"]          = the name of the outliers csv files
1490: # n      [n]              = a vector containint he number of levels of each
1491: #                          variable
1492: #variable    !           = the name of the target variable
1493: # stratum    !           = the names of the stratum variables
1494: ##############################################################################
1495:
1496: BIRO_plotbox<-function(bxpdata,
1497:                        lev,
1498:                        notch=TRUE,
1499:                        namegraph,
1500:                        dirgraph,
1501:                        out=out,
1502:                        n=n,
1503:                        variable=variable,
1504:                        stratum=stratum) {
1505:
1506:
1507:  bxpdata<-sort.data.frame(bxpdata,
1508:                           by=as.formula(paste("~",paste(stratum,collapse="+"))))
1509:
1510:   write("Boxplot statistics sorted",file="")
1511:   if (nchar(logfile)>0) {
1512:    write("Boxplot statistics sorted",file=logfile,append=TRUE)
1513:   }
1514:
1515:   for (j in 1:length(stratum)) {
1516:
1517:   bxpdata[,pmatch(stratum[j],names(bxpdata))]<-
1518:   as.factor(as.character(bxpdata[,pmatch(stratum[j],names(bxpdata))]))
1519:
1520:   levels(bxpdata[,pmatch(stratum[j],names(bxpdata))])<-lev
1521:
1522:   bxpdata$newvar<-as.character(bxpdata[,pmatch(stratum[j],names(bxpdata))])
1523:   names(bxpdata)[pmatch("newvar",names(bxpdata))]<-paste("names",j,sep="")
1524:  }
```

```
1525:
1526:   for (i in 1:dim(bxpdata)[1]) {
1527:   bxpdata$names[i]=
1528:   paste(bxpdata[i,pmatch(paste(rep("names",length(stratum)),
1529:                                1:length(stratum),sep=""),
1530:         names(bxpdata))],collapse=".")
1531:  }
1532:
1533:
1534:  statdata<-
1535:  bxpdata[pmatch(c("mean","l_wisk","pcl_25x","median","pcl_75x",
1536:                  "u_wisk","names",stratum),names(bxpdata))]
1537:
1538:
1539:
1540:
1541:
1542:  stats<-aggregate(statdata,by=list(statdata$names),FUN="unique")
1543:  stats$group<-1:dim(stats)[1]
1544:
1545:  groupdata<-stats[,c("names","group")]
1546:
1547:
1548: bxpdata<-merge(bxpdata,groupdata,by="names")
1549:
1550:
1551:
1552: bxpdata<-sort.data.frame(bxpdata,~group)
1553: stats<-sort.data.frame(stats,~group)
1554:
1555:
1556:
1557: #matrix with summary statistics
1558: statsm<-as.matrix(t(stats[c("l_wisk","pcl_25x","median","pcl_75x","u_wisk")]))
1559:
1560:
1561: n<-as.vector(as.numeric(as.character(bxpdata$n)))
1562:
1563:
1564:
1565: conf<-as.matrix(t(unique(bxpdata[,c("l_notch","u_notch")])))
1566:
1567: out<-as.vector(rep(bxpdata$outlie_x[is.na(bxpdata$outlie_x)==FALSE],
1568:                   bxpdata$Freq[is.na(bxpdata$outlie_x)==FALSE]))
1569:
```

```
1570: groupout<-as.vector(rep(bxpdata$group[is.na(bxpdata$outlie_x)==FALSE],
1571:                         bxpdata$Freq[is.na(bxpdata$outlie_x)==FALSE]))
1572:
1573: boxp<-list(stats=statsm,
1574:            n=n,
1575:            conf=conf,
1576:            out=as.numeric(out),
1577:            group=as.numeric(groupout),
1578:            names=as.vector(unique(bxpdata$names)))
1579:
1580: #X11(width = 21, height =9 , pointsize=9)
1581:
1582:   write("Preparing to do SVG file",file="")
1583:   if (nchar(logfile)>0) {
1584:    write("Preparing to do SVG file",file=logfile,append=TRUE)
1585:   }
1586:
1587:   file = paste(dirgraph,"/","/i",namegraph,"g4_3a.svg",sep="")
1588:   CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
1589:   bxp(boxp,notch=notch,boxfill=rainbow(dim(groupdata)[1]))
1590:   dev.off()
1591:
1592:   write("SVG file: Done",file="")
1593:   if (nchar(logfile)>0) {
1594:    write("SVG file: Done",file=logfile,append=TRUE)
1595:   }
1596:
1597:   write("Preparing to do JPEG file",file="")
1598:   if (nchar(logfile)>0) {
1599:    write("Preparing to do JPEG file",file=logfile,append=TRUE)
1600:   }
1601:
1602:   file = paste(dirgraph,"/","/i",namegraph,"g4_3a.jpeg",sep="")
1603:   jpeg(file, width=11.33, height=4.84, pointsize=9, units="in",
1604:        bg = "transparent",res=9)
1605:   bxp(boxp,notch=notch,boxfill=rainbow(dim(groupdata)[1]))
1606:   dev.off()
1607:
1608:   write("JPEG file: Done",file="")
1609:   if (nchar(logfile)>0) {
1610:    write("JPEG file: Done",file=logfile,append=TRUE)
1611:   }
1612:
1613:   write("Preparing to do PNG file",file="")
1614:   if (nchar(logfile)>0) {
```

```
 1615:    write("Preparing to do PNG file",file=logfile,append=TRUE)
 1616:   }
 1617:
 1618:   file = paste(dirgraph,"/","/i",namegraph,"g4_3a.png",sep="")
 1619:   CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
 1620:   bxp(boxp,notch=notch,boxfill=rainbow(dim(groupdata)[1]))
 1621:   dev.off()
 1622:
 1623:   write("PNG file: Done",file="")
 1624:   if (nchar(logfile)>0) {
 1625:    write("PNG file: Done",file=logfile,append=TRUE)
 1626:   }
 1627:
 1628:   write("Preparing to do PDF file",file="")
 1629:   if (nchar(logfile)>0) {
 1630:    write("Preparing to do PDF file",file=logfile,append=TRUE)
 1631:   }
 1632:
 1633:   file = paste(dirgraph,"/","/i",namegraph,"g4_3a.pdf",sep="")
 1634:   pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
 1635:   bxp(boxp,notch=notch,boxfill=rainbow(dim(groupdata)[1]))
 1636:   dev.off()
 1637:
 1638:   write("PDF file: Done",file="")
 1639:   if (nchar(logfile)>0) {
 1640:    write("PDF file: Done",file=logfile,append=TRUE)
 1641:   }
 1642:   if (struc==TRUE) {
 1643:     cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
 1644:     cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
 1645:     cat("  <tr>",file=sthtml,fill=TRUE,append=TRUE)
 1646:     cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/",namegraph,".png\">","</td>",sep=""
),file=sthtml,fill=TRUE,append=TRUE)
 1647:     cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
 1648:     cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
 1649:     cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
 1650:   }
 1651:
 1652:
 1653:
 1654: }
 1655:
 1656:
 1657:
```

```
  1: ##############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: BIRO_regression.r                                                   #
  5: #                                                                            #
#
  6: ##############################################################################
  7: # ------------------------
  8: # Authors:
  9: # Luca Rossi       <redsluke@gmail.com)
 10: # Fabrizio Carinci <research@fabcarinci.net)
 11: # Institutions: Serectrix snc
 12: # Created: 2008-08-30
 13: # Version: 2008-10-23
 14: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 15: # Programming Language: R 2.8.0
 16: ## ------------------------
 17: # COPYRIGHT INFORMATION
 18: #
 19: # This file is free software; you can redistribute it and/or modify
 20: # it under the terms of the GNU General Public License as published by
 21: # the Free Software Foundation; either version 2, or (at your option)
 22: # any later version.
 23: #
 24: # This file is distributed in the hope that it will be useful,
 25: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 26: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 27: # GNU General Public License for more details.
 28: #
 29: # You should have received a copy of the GNU General Public License
 30: # along with this file; see the file COPYING.  If not, write to
 31: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 32: #
 33: # In short: you may use this file any way you like, as long as you
 34: # don't charge money for it, remove this notice, or hold anyone liable
 35: # for its results.
 36: #
 37: # BIRO_regression.r is part of WP Statistical Engine of the BIRO Project
 38: # GPL Copyright, The BIRO Project
 39: #
 40: # ------------------------
 41: #
 42: # CONTENT
 43: #
 44: #  BIRO_regression
```

```
45: #
46: #
47: ##############################################################################
48:
49:
50: ##############################################################################
51: #  BIRO_regression                                                           #
52: ##############################################################################
53: # ------------------------
54: # Authors:
55: # Luca Rossi       <redsluke@gmail.com)
56: # Fabrizio Carinci <research@fabcarinci.net)
57: # Institutions: Serectrix snc
58: # Created: 2008-08-30
59: # Version: 2008-10-23
60: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
61: # Programming Language: R 2.8.0
62: ## ------------------------
63: # DESCRIPTION
64: #
65: # BIRO_regression is used to fit generalized linear models (linear models, logistic models and poisson models)
66: #
67: # DEPENDENCIES
68: #
69: #  R package: stats
70: #
71: ##############################################################################
72: # PARAMETERS(!=required)
73: #
74: # data      ! = a data frame containing data
75: # depvar    ! = the name of dependent variable
76: # indepvar  ! = a vector containig the names of independent variables
77: # lev       ! = a list contaning levels of depvar and indepvar
78: # factindep ! = a vector containig logistic values TRUE if the i-th independent
79: #               variable is a factor, FALSE otherwise
80: # type      ! = possible values: "linear","logistic","poisson"
81: # sucvalue  ! = the value of depvar indicating a success
82: # csvindex  ! = the string indicating the name of the output csv file
83: # dirdataout! = the path of the output directory
84: ##############################################################################
85:
86: regression<-function(data,
87:                      depvar,
88:                      indepvar,
89:                      lev=list(),
```

```
  90:                       factindep,
  91:                       type,
  92:                       sucvalue,
  93:                       csvindex,
  94:                       dirdataout) {
  95:
  96: if (length(indepvar)>=0) {
  97:
  98:   if (tipe!="poisson") {
  99:    funcdata<-data[,c(pmatch(depvar,names(data)),pmatch(indepvar,names(data)))]
 100:
 101:   if (length(depvar) > 0 & length(lev)>0) {
 102:     funcdata[,pmatch(depvar,names(funcdata))]<-as.factor(funcdata[,pmatch(depvar,names(funcdata))])
 103:     levels(funcdata[,pmatch(depvar,names(funcdata))])<-lev[[1]]
 104:      }
 105:
 106:   if (length(depvar) > 0 & length(lev)==0) {
 107:     levels(funcdata[,pmatch(depvar,names(funcdata))])<-levels(as.factor(funcdata[,pmatch(depvar,names(funcdata))]))
 108:      }
 109:   }
 110:   if (tipe=="poisson") {
 111:    funcdata<-data[,pmatch(indepvar,names(data))]
 112:   }
 113:
 114:
 115:   for (i in 1:length(indepvar)) {
 116:    if (length(var) > i & length(lev)>i) {
 117:      funcdata[,pmatch(indepvar[i],names(funcdata))]<-as.factor(funcdata[,pmatch(indepvar[i],names(funcdata))])
 118:      levels(funcdata[,pmatch(indepvar[i],names(funcdata))])<-lev[[i+1]]
 119:       }
 120:
 121:    if (length(var) > i & length(lev)==i) {
 122:
levels(funcdata[,pmatch(indepvar[i],names(funcdata))])<-levels(as.factor(funcdata[,pmatch(indepvar[i],names(funcdata))]))
 123:       }
 124:
 125:   }
 126:
 127:   for (i in 1:length(var)) {
 128:    if (typeof(lev[[i]])=="list") {
 129:    lev[[i]]<-names(unlist(lev[[i]]))
 130:    }
 131:   }
 132:
 133:   varlist<-as.list(funcdata)
```

```
134:
135:   #calculates contingency table (total frequency of non-missing values)
136:   tab<-as.data.frame(table(varlist))
137:
138:
139:   write.csv(tab,paste(dirdataout,"\\log",csvindex,".csv",sep=""),row.names = FALSE)
140:   }
141:
142:   #create the formula part with the covariates
143:   covformula<-""
144:   for (i in 1:length(indepvar)) {
145:    if (factindep[i]==TRUE) {
146:     covformula<-paste(covformula,"+as.factor(",indepvar[i],")",sep="")
147:    }
148:    if (factindep[i]==FALSE) {
149:     covformula<-paste(covformula,"+",indepvar[i],sep="")
150:    }
151:   }
152:
153:
154:   if (type=="logistic") {
155:   #calculates contingency table of successes
156:   tabs<-tab[tab[,1]==sucvalue,]
157:   names(tabs)[dim(tabs)[2]]<-"Successes"
158:   tabs<-tabs[,-1]
159:
160:
161:   #calculates contingency table of failures
162:   tabf<-tab[tab[,1]!=sucvalue,]
163:   names(tabf)[dim(tabf)[2]]<-"Insuccesses"
164:   tabf<-tabf[,-1]
165:
166:
167:   tab<-merge(tabs,tabf,by=var[2:length(var)],all=TRUE)
168:   tab$out=var[1]
169:
170:   #create formula
171:
172:   formula<-paste("cbind(Successes,Insuccesses)",covformula,sep="~")
173:
174:   model<-glm(formula=as.formula(formula),data=tab,family=binomial())
175:   }
176:
177:   if (tipe=="linear") {
178:      formula<-paste(depvar,paste(indepvar,collapse="+"),sep="~")
```

```
179:   }
180:
181:   if (tipe=="poisson") {
182:      formula<-paste("Freq",paste(indepvar,collapse="+"),sep="~")
183:   }
184:
185:       est.disp <- FALSE
186:     df.r <- model$df.residual
187:
188:         if (model$family$family %in% c("poisson",
189:             "binomial")) {
190:             dispersion <- 1
191:         } else if (df.r > 0) {
192:             est.disp <- TRUE
193:             if (any(model$weights == 0)) {
194:                 warning("observations with zero weight not used for calculating dispersion")
195:             dispersion <- sum((model$weights * model$residuals^2)[model$weights > 0])/df.r
196:         }
197:         } else {
198:             est.disp <- TRUE
199:             dispersion <- NaN
200:             NaN
201:         }
202:
203:
204:
205: p <- model$rank
206:
207: p1 <- 1:p
208: model$qr$qr
209:  coef.p <- model$coefficients[model$qr$pivot[p1]]
210:  covmat.unscaled <- chol2inv(model$qr$qr[p1, p1, drop = FALSE])
211:
212:  dimnames(covmat.unscaled) <- list(names(coef.p), names(coef.p))
213:  covmat <- dispersion * covmat.unscaled
214:  var.cf <- diag(covmat)
215:  s.err <- sqrt(var.cf)
216:
217:  confint<-confint(model)
218:  coeff<- names(coef.p)
219:
220:  estimates<-as.data.frame(cbind(coeff,coef.p,s.err,confint),row.names="")
221:  names(estimates)<-c("Coefficient","Value","Std. Err.","2.5 \\%","97.5 \\%")
222:
223:  row.names(estimates)<- NULL
```

```
224:
225:  write.csv(estimates,paste(dirdataout,"\\est",csvindex,".csv",sep=""),row.names = FALSE)
226:
227:  latex(estimates,file="pippo.tex",append=TRUE)
228:
229:  #Calculate test
230:
231:  test<-anova(model)
232:  ResDev=test[1,4]
233:  Df=test[1,3]
234:  Chisq=ResDev/Df
235:  pvalue=1-pchisq(Chisq,Df)
236:
237:  testcsv<-as.data.frame(cbind(ResDev,Df,Chisq,pvalue))
238:
239:  write.csv(test,paste(dirdataout,"\\tst",csvindex,".csv",sep=""),row.names = FALSE)
240:
241:  invisible(estimates)
242: }
243:
244:
```

```
# ##############################################################################
#                                                                              #
# 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: 2008-10-23
# OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
# 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_standardize.r is part of WP Statistical Engine of the BIRO Project
# GPL Copyright, The BIRO Project
#
# ------------------------
#
# CONTENT
#
#  BIRO_standardize
```

```
45: #
46: #
47: # ##############################################################################
48:
49:
50: # ##############################################################################
51: #  BIRO_standardize                                                            #
52: # ##############################################################################
53: # ------------------------
54: # Authors:
55: # Luca Rossi       <redsluke@gmail.com)
56: # Fabrizio Carinci <research@fabcarinci.net)
57: # Institutions: Serectrix snc
58: # Created: 2008-08-30
59: # Version: 2008-10-23
60: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
61: # Programming Language: R 2.8.0
62: # ------------------------
63: # DESCRIPTION
64: #
65: # BIRO_standardize is used to fit generalized linear models (linear models, logistic models and poisson models)
66: #
67: # DEPENDENCIES
68: #
69: #  R package: stats
70: #
71: # ##############################################################################
72: # PARAMETERS(!=required)
73: #
74: # data       = a data frame containing data
75: # outcome    = the name of outcome variable
76: # cov        = a vector containig the names of covariate variables
77: # strata     = the name of the variable of stratification
78: # lev        = a list contaning levels of depvar and indepvar
79: # factindep  = a vector containig logistic values TRUE if the i-th independent
80: #              variable is a factor, FALSE otherwise
81: # url        = the url/path of a csv file containing previous estimates or
82: #              estimates providing form the BIRO central analysis centre
83: # sucvalue   = the value of depvar indicating a success
84: # pop    [NULL] = the name of the data frame containing population data (area
85: #                 level)
86: # per    [NULL] = the moltiplication factor of rates
87: # csvindex   = the string indicating the name of the output csv file
88: # dirdataout = the path of the output directory
89: # ##############################################################################
```

```
 90: #
 91: # EXAMPLE
 92: # y<-rbinom(1000,1,0.5)
 93: # sex<-rbinom(1000,1,0.5)
 94: # sex<-sex+1
 95: # age<-rbinom(1000,3,0.5)
 96: # age<-age+1
 97: # levsex<-list("Female"=2,"Male"=1)
 98: # levage<-list("0-34"=1,"35-54"=2,"55-74"=3,"75+"=4)
 99: # mydata<-as.data.frame(cbind(y,sex,age))
100: # BIRO_standardize(data=mydata,
101: #                  outcome="y",
102: #                  cov=c("sex","age"),
103: #                  factcov=c(TRUE,TRUE),
104: #                  strata=NULL,
105: #                  lev=list(a=levsex,b=levage),
106: #                  sucvalue=1,
107: #                  per=100,
108: #                  csvindex="provastd",
109: #                  dirdataout=getwd())
110: #
111: # strata<-rep(c("101","102","103","104"),250)
112: # mydata2<-cbind(mydata,strata)
113: #
114: # BIRO_standardize(data=mydata2,
115: #                  outcome="y",
116: #                  cov=c("sex","age"),
117: #                  factcov=c(TRUE,TRUE),
118: #                  strata="strata",
119: #                  lev=list(a=levsex,b=levage),
120: #                  sucvalue=1,
121: #                  per=100000,
122: #                  csvindex="provastd",
123: #                  dirdataout=getwd(),
124: #                  strataname="Centres")
125: # ##############################################################################
126:
127: BIRO_standardize<-function(data=NULL,
128:                            datafreq=NULL,
129:                            title="",
130:                            outcome,
131:                            cov,
132:                            factcov,
133:                            strata=NULL,
134:                            std=0,
```

```
                              lev=list(),
                              sucvalue,
                              url=NULL,
                              pop=NULL,
                              per,
                              csvindex,
                              dirdataout,
                              strataname,
                              filename,
                              dirgraph,
                              dirgraphlatex,
                              dirtables,
                              texfile,
                              caption,
                              chisq,
                              map=0,
                              shapefile="eurnuts3",
                              mapvar,
                              valuesarea_id="area_id",
                              shapearea_id="ID",
                              frqvar="Adjusted Rate",
                              labvarmap="Adjusted Rate",
                              mapth=1) {
  pval=0
  write("BIRO_standardize called",file="")
  if (nchar(logfile)>0) {
   write("BIRO_standardize called",file=logfile,append=TRUE)
  }

  if (length(cov)>=0) {
   if (std==0) {
    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(funcdata))])
    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))]))
```

```
179:    }
180:
181:    for (i in 1:length(cov)) {
182:     if (length(var) > i & length(lev)>i) {
183:      funcdata[,pmatch(cov[i],names(funcdata))]<-as.factor(funcdata[,pmatch(cov[i],names(funcdata))])
184:      if (is.null(lev[[i+1]])==FALSE) {
185:       levels(funcdata[,pmatch(cov[i],names(funcdata))])<-lev[[i+1]]
186:      }
187:     }
188:    }
189:    # calculates contingency table (total frequency of non-missing values)
190:    tab<-as.data.frame(table(funcdata))
191:    if (is.null(pop)==FALSE) {
192:     wtab<-tab
193:     wtab$centre_id=centre_id
194:     write.csv(wtab,paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""),row.names = FALSE)
195:     rm(wtab)
196:    }
197:
198:    # calculates contingency table of successes
199:    tabs<-tab[tab[,1]==sucvalue,]
200:    names(tabs)[dim(tabs)[2]]<-"Successes"
201:    tabs<-tabs[,-1]
202:
203:    if (is.null(pop)==TRUE) {
204:     # calculates contingency table of failures
205:     tabf<-tab[tab[,1]!=sucvalue,]
206:     names(tabf)[dim(tabf)[2]]<-"Insuccesses"
207:     tabf<-tabf[,-1]
208:     if (dim(tabf)[1]>0 & dim(tabs)[1]>0) {
209:      tab<-merge(tabs,tabf,by=cov,all=TRUE)
210:     }
211:     if (dim(tabf)[1]==0) {
212:      tab<-tabs
213:      tab$Insuccesses<-0
214:     }
215:     if (dim(tabs)[1]==0) {
216:      tab<-tabf
217:      tab$Successes<-0
218:    }
219:
220:    if (is.null(strata)==FALSE & is.null(pop)==TRUE) {
221:     tab<-merge(tabs,tabf,by=c(cov,strata),all=TRUE)
222:    }
223:
```

```
224:     tab$out=outcome
225:     wtab<-tab
226:     wtab$centre_id=centre_id
227:     write.csv(wtab,paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""),row.names = FALSE)
228:     rm(wtab)
229:    }
230:    if (is.null(pop)==FALSE & is.null(strata)==FALSE) {
231:     tab<-tabs
232:     tab<-merge(pop,tab,by=c(cov,strata),all.x=TRUE,all.y=FALSE)
233:     tab$Insuccesses<-(tab$pop)-tab$Successes
234:     tab$out=outcome
235:     tab$Successes[is.na(tab$Successes)]<-0
236:     tab$Insuccesses<-ifelse(is.na(tab$Insuccesses),tab$pop,tab$Insuccesses)
237:     tab<-tab[,-pmatch("pop",names(tab))]
238:    }
239:    if (is.null(pop)==FALSE & is.null(strata)==TRUE) {
240:     tab<-tabs
241:     tab<-merge(pop,tab,by=c(cov),all.x=TRUE,all.y=TRUE)
242:     tab$Insuccesses<-(tab$pop)-tab$Successes
243:     tab$Successes[is.na(tab$Successes)]<-0
244:     tab$Insuccesses<-ifelse(is.na(tab$Insuccesses),tab$pop,tab$Insuccesses)
245:     tab<-tab[,-pmatch("pop",names(tab))]
246:     tab$out=outcome
247:     wtab<-tab
248:     wtab$centre_id=centre_id
249:     write.csv(wtab,paste(dirdataout,"/i",csvindex,"d6_1a.csv",sep=""),row.names = FALSE)
250:     rm(wtab)
251:    }
252:    datafreq<-tab
253:  }
254:  # calculates p-values
255:  tab<-datafreq
256:  ptabs<-tab[,c(cov,"Successes")]
257:  names(ptabs)<-c(cov,"Freq")
258:  ptabs$myout=1
259:  ptabi<-tab[,c(cov,"Insuccesses")]
260:  names(ptabi)<-c(cov,"Freq")
261:  ptabi$myout=0
262:  ptab<-rbind(ptabs,ptabi)
263:  nvar<-length(cov)+1
264:  var<-c("myout",cov)
265:  if (is.null(chisq)==FALSE) {
266:   if (length(cov)==3) {
267:    chilist<-list()
268:    for (k in 1:nlevels(factor(ptab[,pmatch(var[4],names(ptab))]))) {
```

```
269:     formula<-as.formula(paste("Freq~",paste(c("myout",chisq[1:2]),collapse="+"),sep=""))
270:     subptab<-ptab[ptab[,pmatch(var[4],names(ptab))]==sort(unique(ptab[,pmatch(var[4],names(ptab))]))[k],]
271:     if (all(subptab$Freq>1)) {
272:      # Cochran-Mantel-Haenszel Chi-Squared Test for Count Data
273:      mytable<-xtabs(formula=formula,data=subptab)
274:      mh<-mantelhaen.test(mytable)
275:      chitab<-as.data.frame(cbind(mh$statistic,format(mh$p.value,nsmall=4),mh$parameter))
276:      names(chitab)<-c("Chi-Squared","p.value","df")
277:      lab4<-paste(unlist(strsplit(var[4],"_")),collapse="")
278:      row.names(chitab)=paste("C-M-H Chi-Squared Test",lab4,sort(unique(ptab[,pmatch(var[4],names(ptab))]))[k])
279:      chitab$centre_id=centre_id
280:      write.csv(chitab,paste(dirdataout,"/i",csvindex,"d1_3b.csv",sep=""),row.names = FALSE)
281:
282:      BIRO_df2html(data=chitab,
283:                   htmlfile=paste(dirtables,"/i",csvindex,"d1_3b.html",sep=""),
284:                   catcol=NULL)
285:
286:      file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
287:                   paste(dirtables,"/i",csvindex,"d1_3b.html",sep=""))
288:
289:      for (j in 1:length(names(chitab))) {
290:       names(chitab)[j]<-paste(unlist(strsplit(names(chitab)[j],"_")),collapse="\\_")
291:      }
292:      chilist[[k]]<-chitab
293:     } else {
294:      chitab<-as.data.frame("One or more cells have less than 2 obs")
295:      names(chitab)<-"C-M-H Chi-Squared Test"
296:      chilist[[k]]<-chitab
297:     }
298:    }
299:   }
300:   if (length(cov)==2) {
301:    formula<-as.formula(paste("Freq~",paste(c("myout",chisq),collapse="+"),sep=""))
302:    if (all(ptab$Freq>1)) {
303:     mytable<-xtabs(formula=formula,data=ptab)
304:     # Cochran-Mantel-Haenszel Chi-Squared Test for Count Data
305:     mh<-mantelhaen.test(mytable)
306:     chitab<-as.data.frame(cbind(mh$statistic,format(mh$p.value,nsmall=4),mh$parameter))
307:     names(chitab)<-c("Chi-Squared","p.value","df")
308:     row.names(chitab)="C-M-H Chi-Squared Test"
309:     chitab$centre_id=centre_id
310:     write.csv(chitab,paste(dirdataout,"/i",csvindex,"d1_3b.csv",sep=""),row.names = FALSE)
311:     BIRO_df2html(data=chitab,
312:                   htmlfile=paste(dirtables,"/i",csvindex,"1_3b.html",sep=""),
313:                   catcol=NULL)
```

```
314:
315:     file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
316:                 paste(dirtables,"/i",csvindex,"1_3b.html",sep=""))
317:
318:     for (j in 1:length(names(chitab))) {
319:      names(chitab)[j]<-paste(unlist(strsplit(names(chitab)[j],"_")),collapse="\\_")
320:     }
321:    } else {
322:     chitab<-as.data.frame("One or more cells have less than 2 obs")
323:     names(chitab)<-"C-M-H Chi-Squared Test"
324:    }
325:   }
326:   if (length(na.omit(cov))==1) {
327:    formula<-as.formula(paste("Freq~",paste(c(var[1],cov),collapse="+"),sep=""))
328:    mytable<-xtabs(formula=formula,data=ptab)
329:    # Cochran-Mantel-Haenszel Chi-Squared Test for Count Data
330:    mh<-chisq.test(mytable)
331:    chitab<-as.data.frame(cbind(mh$statistic,mh$p.value,mh$parameter))
332:    names(chitab)<-c("Chi-Squared","p.value","df")
333:    row.names(chitab)="C-M-H Chi-Squared Test"
334:    chitab$centre_id=centre_id
335:    write.csv(chitab,paste(dirdataout,"/i",csvindex,"d1_3b.csv",sep=""),row.names = FALSE)
336:
337:    BIRO_df2html(data=chitab,
338:                 htmlfile=paste(dirtables,"/i",csvindex,"1_3b.html",sep=""),
339:                 catcol=NULL)
340:
341:    file.append(paste(dirhtml,"/",csvindex,".html",sep=""),
342:                paste(dirtables,"/i",csvindex,"1_3b.html",sep=""))
343:
344:    for (j in 1:length(names(chitab))) {
345:     names(chitab)[j]<-paste(unlist(strsplit(names(chitab)[j],"_")),collapse="\\_")
346:    }
347:   }
348:  }
349:  # read coefficients from the central repository
350:  if (is.null(url)==FALSE) {
351:   coeff=read.csv(url)
352:   # create dummy in the table (covariates)
353:   tabdummy<-tab
354:   for (i in 1:length(cov)) {
355:    if (factcov[i]==TRUE) {
356:     for (j in 2:nlevels(factor(tabdummy[,pmatch(cov[i],names(tabdummy))]))) {
357:
tabdummy[,dim(tabdummy)[2]+1]<-ifelse(tabdummy[,pmatch(cov[i],names(tabdummy))]==unique(tabdummy[,pmatch(cov[i],names(tabd
```

```
ummy))])[order(unique(tabdummy[,pmatch(cov[i],names(tabdummy))]))][j],1,0)
  358:
names(tabdummy)<-c(names(tabdummy)[1:dim(tabdummy)[2]-1],paste(cov[i],unique(tabdummy[,pmatch(cov[i],names(tabdummy))])[or
der(unique(tabdummy[,pmatch(cov[i],names(tabdummy))]))][j],sep=""))
  359:     }
  360:    }
  361:   }
  362:   oldcoeff<-as.data.frame(t(coeff[,c(2)]))
  363:
  364:   # fitted values
  365:   ncoeff<-dim(oldcoeff)[2]-1
  366:
  367:   for (i in 1:dim(tabdummy)[1]) {
  368:    sum=oldcoeff$Intercept
  369:    for (j in 2:ncoeff) {
  370:     newvar<-names(oldcoeff)[j]
  371:     sum=sum+tabdummy[i,pmatch(newvar,names(tabdummy))]*oldcoeff[1,j]
  372:    }
  373:    model$fitted.values[i]=exp(sum[i])/(1+exp(sum[i]))
  374:   }
  375:  }
  376:  if (is.null(url)==TRUE & is.null(strata)==FALSE & std==1) {
  377:   # logistic regression
  378:   if (is.null(data)==TRUE) {
  379:    tab<-datafreq
  380:   }
  381:   model<-glm(as.formula(paste("cbind(Successes,Insuccesses)~",paste(cov,collapse="+"),sep=""
)),data=tab,family=binomial())
  382:   est.disp <- FALSE
  383:   df.r <- model$df.residual
  384:   if (model$family$family %in% c("poisson","binomial")) {
  385:    dispersion <- 1
  386:   } else if (df.r > 0) {
  387:    est.disp <- TRUE
  388:    if (any(model$weights == 0)) {
  389:     warning("observations with zero weight not used for calculating dispersion")
  390:     dispersion <- sum((model$weights * model$residuals^2)[model$weights > 0])/df.r
  391:    }
  392:   } else {
  393:    est.disp <- TRUE
  394:    dispersion <- NaN
  395:   }
  396:   p <- model$rank
  397:   p1 <- 1:p
  398:   model$qr$qr
```

```
399:   coef.p <- model$coefficients[model$qr$pivot[p1]]
400:   covmat.unscaled <- chol2inv(model$qr$qr[p1, p1, drop = FALSE])
401:
402:   dimnames(covmat.unscaled) <- list(names(coef.p), names(coef.p))
403:   covmat <- dispersion * covmat.unscaled
404:   var.cf <- diag(covmat)
405:   s.err <- sqrt(var.cf)
406:   confint<-confint(model)
407:   coeff<- names(coef.p)
408:   estimates<-as.data.frame(cbind(coeff,coef.p,s.err,confint),row.names="")
409:   names(estimates)<-c("Coefficient","Value","Std. Err.","2.5 \\%","97.5 \\%")
410:   row.names(estimates)<- NULL
411:
412:   estimates$codist=centre_id
413:   write.csv(estimates,paste(dirdataout,"/i",csvindex,"d5_1a.csv",sep=""),row.names = FALSE)
414:
415:   write("Estimates written in csv file",file="")
416:   if (nchar(logfile)>0) {
417:    write("Estimates written in csv file",file=logfile,append=TRUE)
418:   }
419:  }
420:  if (is.null(strata)==TRUE) {
421:   tab$Denom <- as.numeric(as.character(tab$Successes+tab$Insuccesses))
422:   tab$Obs <- as.numeric(as.character(tab$Successes))
423:   tot_Obs <- sum(tab$Obs)
424:   tot_Denom <- sum(tab$Denom)
425:   tab<-tab[,pmatch(c("Obs","Denom",cov),names(tab))]
426:   tab$Obs<-as.numeric(as.character(tab$Obs))
427:   tab$Denom<-as.numeric(as.character(tab$Denom))
428:   tab$RawRate<-(tab$Obs/tab$Denom)*per
429:   # Total rate
430:   tab<-rbind(tab,NA)
431:   tab$RawRate[dim(tab)[1]]<-(tot_Obs/tot_Denom)*per
432:   tab$Obs[dim(tab)[1]]<-tot_Obs
433:   tab$Denom[dim(tab)[1]]<-tot_Denom
434:   if (per==100000) {
435:    tab$RawRate<-round(tab$RawRate,0)
436:   }
437:   if (per==100) {
438:    tab$RawRate<-round(tab$RawRate,2)
439:   }
440:   for (i in 1:length(cov)) {
441:    if (factcov[i]==TRUE) {
442:     tab[,pmatch(cov[i],names(tab))]<-as.factor(tab[,pmatch(cov[i],names(tab))])
443:         levels(tab[,pmatch(cov[i],names(tab))])<-lev[[i+1]]
```

```
444:     }
445:    }
446:   }
447:   if (per==100) {
448:    names(tab)[pmatch("RawRate",names(tab))]<-'Percentage'
449:   }
450:   if (per==100000) {
451:    names(tab)[pmatch("RawRate",names(tab))]<-paste('Rate x ',per,sep="")
452:   }
453:   if (is.null(pop)==FALSE) {
454:    names(tab)[pmatch("Denom",names(tab))]<-'Population'
455:   }
456:   # Expecteds (standardization)
457:   if (is.null(strata)==FALSE & std==1 ) {
458:    if (is.null(data)==TRUE) {
459:     tab<-datafreq
460:    }
461:    tab$Denom <- as.numeric(as.character(tab$Successes))+as.numeric(as.character(tab$Insuccesses))
462:    tab$Exp <- round(tab$Denom*model$fitted.value)
463:    tab$Obs <- as.numeric(as.character(tab$Successes))
464:    tot_Obs <- sum(tab$Obs)
465:    tot_Denom <- sum(tab$Denom)
466:    # Keep only Obs, Exp, Denom, strata
467:    tab<-tab[,pmatch(c("Obs","Exp","Denom",strata),names(tab))]
468:    tab$Obs<-as.numeric(as.character(tab$Obs))
469:    tab$Exp<-as.numeric(as.character(tab$Exp))
470:    tab$Denom<-as.numeric(as.character(tab$Denom))
471:    # Aggregate data (sums)
472:    mystrata=list(Strata=tab[,pmatch(strata,names(tab))])
473:    tab<-aggregate(tab[,c("Obs","Exp","Denom")],by=mystrata,FUN=sum)
474:    tab$RawRate<-(tab$Obs/tab$Denom)*per
475:    tab$AdjRate=(tot_Obs/tot_Denom)*(tab$Obs/tab$Exp)*per
476:    # Variance of Adjusted Rates
477:    tab$SEAdj=sqrt(((tot_Obs/tot_Denom)/(tab$Exp/tot_Denom))^2 * (1/tab$Denom)^2 *
(tab$Denom*(tab$Exp/tab$Denom)*(1-(tab$Exp/tab$Denom))))
478:    tab$LowAdj=((tab$AdjRate/per)-1.96*tab$SEAdj/sqrt(tab$Denom))*per
479:    tab$UppAdj=((tab$AdjRate/per)+1.96*tab$SEAdj/sqrt(tab$Denom))*per
480:    # Reference rate
481:    tab<-rbind(tab,NA)
482:    tab$RawRate[dim(tab)[1]]<-(tot_Obs/tot_Denom)*per
483:    tab$Obs[dim(tab)[1]]<-tot_Obs
484:    tab$Denom[dim(tab)[1]]<-tot_Denom
485:    if (per==100000) {
486:     tab$RawRate<-round(tab$RawRate,0)
487:     tab$AdjRate<-round(tab$AdjRate,0)
```

```
  488:    tab$LowAdj<-round(tab$LowAdj,0)
  489:    tab$UppAdj<-round(tab$UppAdj,0)
  490:   }
  491:   if (per==100) {
  492:    tab$RawRate<-round(tab$RawRate,2)
  493:    tab$AdjRate<-round(tab$AdjRate,2)
  494:    tab$LowAdj<-round(tab$LowAdj,2)
  495:    tab$UppAdj<-round(tab$UppAdj,2)
  496:   }
  497:   fortab<-tab
  498:   tab<-tab[,-pmatch("SEAdj",names(tab))]
  499:   tab$strataname=strataname
  500:   tab$codist=centre_id
  501:   write.csv(tab,paste(dirdataout,"/i",csvindex,"d6_1b.csv",sep=""),row.names = FALSE)
  502:   tab<-tab[,1:(dim(tab)[2])-1]
  503:   names(tab)<-c("Strata","Observeds","Expexteds","N",'Raw Rate','Adjusted Rate','Lower Bound','Upper Bound',
"Strataname")
  504:   if (map==1) {
  505:    maptab<-as.data.frame(tab)
  506:   }
  507:   if (per==100) {
  508:    names(tab)[pmatch("RawRate",names(tab))]<-'Percentage'
  509:    names(tab)[pmatch("AdjRate",names(tab))]<-'Adj. Percentage'
  510:   }
  511:   if (per==100000) {
  512:    names(tab)[pmatch("RawRate",names(tab))]<-paste('Rate x ',per,sep="")
  513:    names(tab)[pmatch("AdjRate",names(tab))]<-paste('Adj. Rate x ',per,sep="")
  514:   }
  515:   if (is.null(pop)==FALSE) {
  516:    names(tab)[pmatch("Denom",names(tab))]<-'Population'
  517:   }
  518:   # Graphs
  519:   fortab<-fortab[1:(dim(fortab)[1]-1),]
  520:   SEmax=max(fortab$SEAdj,na.rm=TRUE)
  521:
lmin<-min(min(fortab$LowAdj,na.rm=TRUE)-3*SEmax,(tot_Obs/tot_Denom)-3*SEmax,max(fortab$RawRate,na.rm=TRUE)-3*SEmax)
  522:
lmax<-max(max(fortab$UppAdj,na.rm=TRUE)+3*SEmax,(tot_Obs/tot_Denom)+3*SEmax,max(fortab$RawRate,na.rm=TRUE)+3*SEmax)
  523:   s<-c(strataname,as.character(fortab$Strata))
  524:   r<-c(names(tab)[5],as.character(fortab$AdjRate))
  525:   tabletext<-as.data.frame(cbind(s,r))
  526:
  527:   BIRO_forest(tabletext=tabletext,
  528:               values=c(NA,fortab$AdjRate),
  529:               lbound=c(NA,fortab$LowAdj),
```

```
530:                ubound=c(NA,fortab$UppAdj),
531:                limits=c(lmin,lmax),
532:                col=meta.colors(box="royalblue",line="darkblue", summary="royalblue",zero="red"),
533:                zero=(tot_Obs/tot_Denom)*per,
534:                xlab="",
535:                is.summary=c(FALSE,FALSE,FALSE,FALSE,FALSE),
536:                cex=2,
537:                graphwidth= unit(13,"inches"),
538:                csvindex=csvindex,
539:                dirgraph=dirgraph)
540:
541:    import_png(namegraph=paste("for",csvindex,".png",sep=""),
542:               caption=paste("Forestplot: ",title,sep=""),
543:               width=1,
544:               dirgraph=dirgraphlatex,
545:               texfile=texfile)
546:
547:    if (map==1) {
548:     if (exists("mapcod")) {
549:      if (maploaded!=shapefile) {
550:       maploaded=shapefile
551:       assign(maploaded,shapefile,envir=.GlobalEnv)
552:       mapfile<-readShapePoly(paste(dirmap,"/",shapefile,".shp",sep=""))
553:       mapfile<-mapfile[mapfile$ID %in% unique(mapcod$area_id),]
554:       assign("mapfile",mapfile,,envir=.GlobalEnv)
555:      }
556:      print(mapcod)
557:      print(maptab)
558:      myvalues<-merge(x=mapcod,y=maptab,by.x="centre_id",by.y="Strata",all.x=TRUE,all.y=FALSE)
559:      print(myvalues)
560:      BIRO_map(shape=mapfile,
561:               datavalues=maptab,
562:               valuesarea_id="area_id",
563:               shapearea_id="ID",
564:               frqvar="Adjusted Rate",
565:               labvar=labvarmap,
566:               breaks=mapth,
567:               legendpos="topright",
568:               dirgraph=dirgraph,
569:               namegraph=paste("map",csvindex,sep=""))
570:      }
571:     }
572:    }
573:    for (i in 1:dim(tab)[2]) {
574:     names(tab)[i]<-paste(unlist(strsplit(names(tab)[i],"_")),collapse="")
```

```
575:   }
576:   }
577:   write("BIRO_standardize finished",file="")
578:   if (nchar(logfile)>0) {
579:    write("BIRO_standardize finished",file=logfile,append=TRUE)
580:   }
581:
582:   if (is.null(chisq)==FALSE) {
583:    if (length(cov)<3) {
584:     latex(tab,file=texfile,append=TRUE,table.env=FALSE)
585:     latex(chitab,file=texfile,append=TRUE,table.env=FALSE)
586:    }
587:
588:    if (length(cov)==3) {
589:     mycov<-paste(unlist(strsplit(cov[3],"_")),collapse="")
590:
591:     for (k in 1:nlevels(factor(tab[,pmatch(mycov,names(tab))]))) {
592:      thislev<-levels(factor(tab[,pmatch(mycov,names(tab))]))[k]
593:      thisvar<-tab[,pmatch(mycov,names(tab))]
594:      printtab<-tab[thisvar==thislev,]
595:      latex(printtab,file=texfile,append=TRUE,table.env=FALSE)
596:
597:      for (i in 1:length(names(chitab))) {
598:       names(chitab)[i]<-paste(unlist(strsplit(names(chitab)[i],"_")),collapse="\\_")
599:      }
600:     latex(chilist[[k]],file=texfile,append=TRUE,table.env=FALSE)
601:     }
602:    }
603:   }
604:   return(tab)
605: }
606:
607:
608:
609:
610:
611:
```

```
    1: # ############################################################################
    2: #                                                                          #
    3: # Project: BIRO-Project (Funded by European Commission 2005-2008)          #
    4: # File: biro_trellis.r                                                   #
    5: #                                                                          #
#
    6: # ############################################################################
    7: # ------------------------
    8: # Authors:
    9: # Luca Rossi       <redsluke@gmail.com)
   10: # Fabrizio Carinci <research@fabcarinci.net)
   11: # Institutions: Serectrix snc
   12: # Created: 2008-09-07
   13: # Version: 2008-10-23
   14: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
   15: # Programming Language: R 2.8.0
   16: # ------------------------
   17: # COPYRIGHT INFORMATION
   18: #
   19: # This file is free software; you can redistribute it and/or modify
   20: # it under the terms of the GNU General Public License as published by
   21: # the Free Software Foundation; either version 2, or (at your option)
   22: # any later version.
   23: #
   24: # This file is distributed in the hope that it will be useful,
   25: # but WITHOUT ANY WARRANTY; without even the implied warranty of
   26: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   27: # GNU General Public License for more details.
   28: #
   29: # You should have received a copy of the GNU General Public License
   30: # along with this file; see the file COPYING.  If not, write to
   31: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
   32: #
   33: # In short: you may use this file any way you like, as long as you
   34: # don't charge money for it, remove this notice, or hold anyone liable
   35: # for its results.
   36: #
   37: # biro_trellis.r is part of WP Statistical Engine of the BIRO Project
   38: # GPL Copyright, The BIRO Project
   39: #
   40: # ------------------------
   41: #
   42: # CONTENT
   43: #
   44: #  BIRO_histtrellis
```

```
45: #  BIRO_densitytrellis
46: #  BIRO_boxtrellis
47: #
48: # ###########################################################################
49:
50:
51: # ###########################################################################
52: #  BIRO_histtrellis                                                         #
53: # ###########################################################################
54: # ------------------------
55: # Authors:
56: # Luca Rossi       <redsluke@gmail.com)
57: # Fabrizio Carinci <research@fabcarinci.net)
58: # Institutions: Serectrix snc
59: # Created: 2008-09-07
60: # Version: 2008-10-23
61: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
62: # Programming Language: R 2.8.0
63: # ------------------------
64: # DESCRIPTION
65: #
66: # BIRO_histtrellis draws Histograms possibly conditioned on other variables.
67: #
68: # DEPENDENCIES
69: #
70: #  R package: lattice
71: #
72: # ###########################################################################
73: # PARAMETERS(!=required)
74: #
75: # data                ! =  a data frame containing data
76: # var                 ! =  the name of the conditionated variable
77: # strata   [NULL]       =  a vector containing the names of the conditionating
78: #                          variables
79: # labvar              ! =  string value contining the label of the conditionated
80: #                          variable
81: # labclass          ! =  a vector containing string values (labels of the
82: #                          conditionating variable)
83: # lev_class           ! =  list containing the levels of strata
84: # dirout    [getwd()]   =  the path of the output directory
85: # ext        [csv]      =  the extention of the output file
86: # date    [Sys.Date()]  =  the date of analysis
87: # namegraph           ! =  the name of the output file (text, jpg,png, eps)
88: # dirgraph            ! =  the path of the output graph directory
89: # ###########################################################################
```

```
 90:
 91: BIRO_histtrellis<-function(data,
 92:                            var,
 93:                            strata=NULL,
 94:                            labvar,
 95:                            labclass,
 96:                            lev_class,
 97:                            dirout=getwd(),
 98:                            ext="csv",
 99:                            date=Sys.Date(),
100:                            namegraph,
101:                            dirgraph) {
102:
103:  #X11(width = 21, height =9 , pointsize=9)
104:
105:  if (is.null(strata)==FALSE) {
106:
107:   for (i in 1:length(strata)) {
108:    data[,pmatch(strata[i],names(data))]<-
109:    as.factor(data[,pmatch(strata[i],names(data))])
110:    levels(data[,pmatch(strata[i],names(data))])=lev_class[[i]]
111:   }
112:
113:
114:   formula<-paste(var,"~",paste(strata,collapse="+"),sep="")
115:   n<-NULL
116:   for (i in 1:length(lev_class)) {
117:    n<-c(n,length(lev_class[[i]]))
118:   }
119:   firstlay<-prod(n)/4
120:   formula<-paste("~ ",var," | ",paste(strata,collapse="+"),sep="")
121:
122:   write("preparing to do trellis function histogram",file="")
123:   trellis<-histogram(as.formula(as.character(formula)),
124:                      data=data,
125:                      layout=c(firstlay,4),
126:                      xlab=labvar,
127:                      type = "density",
128:                      panel = function(x, ...) {
129:                        panel.histogram(x, ...)
130:                        panel.mathdensity(dmath = dnorm, col = "black",
131:                        args = list(mean=mean(x),sd=sd(x)))
132:                      } )
133:
134:   write("function histogram done",file="")
```

```
135:
136:
137:  file = paste(dirgraph,"/i",namegraph,"g4_2a.svg",sep="")
138:  CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
139:  plot(trellis)
140:  dev.off()
141:
142:  file = paste(dirgraph,"/i",namegraph,"g4_2a.jpeg",sep="")
143:  jpeg(file, width=11.33, height=4.84, pointsize=9, units="in",
144:       bg = "transparent",res=9)
145:  plot(trellis)
146:  dev.off()
147:
148:  file = paste(dirgraph,"/i",namegraph,"g4_2a.png",sep="")
149:  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
150:  plot(trellis)
151:  dev.off()
152:
153:  file = paste(dirgraph,"/i",namegraph,"g4_2a.pdf",sep="")
154:  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
155:  plot(trellis)
156:  dev.off()
157:
158:  write("trellis histogram saved",file="")
159:
160:  }
161:  if (is.null(strata)==TRUE) {
162:
163:   write("preparing to do trellis function histogram",file="")
164:
165:   histogram(data[,pmatch(var,names(data))],
166:             xlab=labvar,
167:             type = "density",
168:             panel = function(x, ...) {
169:                panel.histogram(x, ...)
170:                panel.mathdensity(dmath = dnorm, col = "black",
171:                args = list(mean=mean(x),sd=sd(x)))
172:             } )
173:
174:   write("function histogram done",file="")
175:  }
176:  if (struc==TRUE) {
177:    cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
178:    cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
179:    cat("  <tr>",file=sthtml,fill=TRUE,append=TRUE)
```

```
  180:    cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/",namegraph,".png\">","</td>",sep=""
),file=sthtml,fill=TRUE,append=TRUE)
  181:    cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
  182:    cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
  183:    cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
  184:  }
  185:
  186: }
  187:
  188:
  189: # ##############################################################################
  190: #  BIRO_densitytrellis                                                         #
  191: # ##############################################################################
  192: # ------------------------
  193: # Authors:
  194: # Luca Rossi       <redsluke@gmail.com)
  195: # Fabrizio Carinci <research@fabcarinci.net)
  196: # Institutions: Serectrix snc
  197: # Created: 2008-09-07
  198: # Version: 2008-10-23
  199: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
  200: # Programming Language: R 2.8.0
  201: # ------------------------
  202: # DESCRIPTION
  203: #
  204: # BIRO_densitytrellis draws Kernel Density Plots possibly conditioned on other
  205: # variables.
  206: #
  207: # DEPENDENCIES
  208: #
  209: #  R package: lattice
  210: #
  211: # ##############################################################################
  212: # PARAMETERS(!=required)
  213: #
  214: # data                ! =  a data frame containing data
  215: # var                 ! =  the name of the conditionated variable
  216: # strata    [NULL]      =  a vector containing the names of the conditionating
  217: #                          variables
  218: # labvar              ! =  string value contining the label of the conditionated
  219: #                          variable
  220: # labclass            ! =  a vector containing string values (labels of the
  221: #                          conditionating variable)
  222: # lev_class           ! =  list containing the levels of strata
  223: # dirout    [getwd()]   =  the path of the output directory
```

```
224: # ext        [csv]      =  the extention of the output file
225: # date    [Sys.Date()]  =  the date of analysis
226: # namegraph           ! =  the name of the output file (text, jpg,png, eps)
227: # dirgraph            ! =  the path of the output graph directory
228: # ##############################################################################
229:
230: BIRO_densitytrellis<-function(data,
231:                               var,
232:                               strata=NULL,
233:                               labvar,
234:                               labclass,
235:                               lev_class,
236:                               dirout=getwd(),
237:                               ext="csv",
238:                               date=Sys.Date(),
239:                               namegraph,
240:                               dirgraph) {
241:
242:
243:
244:  #X11(width = 21, height =9 , pointsize=9)
245:
246:
247:  for (i in 1:length(strata)) {
248:   data[,pmatch(strata[i],names(data))]<-
249:   as.factor(data[,pmatch(strata[i],names(data))])
250:   levels(data[,pmatch(strata[i],names(data))])=lev_class[[i]]
251:
252:  }
253:
254:
255:  formula<-paste(var,"~",paste(strata,collapse="+"),sep="")
256:
257:
258:  if (is.null(strata)==FALSE) {
259:
260:  n<-NULL
261:  for (i in 1:length(lev_class)) {
262:   n<-c(n,length(lev_class[[i]]))
263:
264:  }
265:   firstlay<-prod(n)/4
266:
267:   formula<-paste("~ ",var," | ",paste(strata,collapse="+"),sep="")
268:
```

```
269:    write("Preparing to do trellis function densityplot",file="")
270:    if (nchar(logfile)>0) {
271:     write("Preparing to do trellis function densityplot",file=logfile,
272:           append=TRUE)
273:    }
274:
275:   trellis<-densityplot(as.formula(as.character(formula)),data=data,
276:                        layout=c(firstlay,4))
277:
278:
279:    write("function densityplot done",file="")
280:    if (nchar(logfile)>0) {
281:     write("function densityplot done",file=logfile,append=TRUE)
282:    }
283:
284:  write("Preparing to do SVG file",file="")
285:  if (nchar(logfile)>0) {
286:   write("Preparing to do SVG file",file=logfile,append=TRUE)
287:  }
288:
289:  file = paste(dirgraph,"/",namegraph,".svg",sep="")
290:  CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
291:  plot(trellis)
292:  dev.off()
293:
294:  write("SVG file: Done",file="")
295:  if (nchar(logfile)>0) {
296:   write("SVG file: Done",file=logfile,append=TRUE)
297:  }
298:
299:  write("Preparing to do JPEG file",file="")
300:  if (nchar(logfile)>0) {
301:   write("Preparing to do JPEG file",file=logfile,append=TRUE)
302:  }
303:
304:  file = paste(dirgraph,"/",namegraph,".jpeg",sep="")
305:  jpeg(file, width=11.33, height=4.84, pointsize=9, units="in",
306:       bg = "transparent",res=9)
307:  plot(trellis)
308:  dev.off()
309:
310:  write("JPEG file: Done",file="")
311:  if (nchar(logfile)>0) {
312:   write("JPEG file: Done",file=logfile,append=TRUE)
313:  }
```

```
314:
315:  write("Preparing to do PNG file",file="")
316:  if (nchar(logfile)>0) {
317:   write("Preparing to do PNG file",file=logfile,append=TRUE)
318:  }
319:
320:  file = paste(dirgraph,"/",namegraph,".png",sep="")
321:  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9 ,bg = "transparent")
322:  plot(trellis)
323:  dev.off()
324:
325:  write("PNG file: Done",file="")
326:  if (nchar(logfile)>0) {
327:   write("PNG file: Done",file=logfile,append=TRUE)
328:  }
329:
330:  write("Preparing to do PDF file",file="")
331:  if (nchar(logfile)>0) {
332:   write("Preparing to do PDF file",file=logfile,append=TRUE)
333:  }
334:
335:  file = paste(dirgraph,"/",namegraph,".pdf",sep="")
336:  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
337:  plot(trellis)
338:  dev.off()
339:
340:  write("PDF file: Done",file="")
341:  if (nchar(logfile)>0) {
342:   write("PDF file: Done",file=logfile,append=TRUE)
343:  }
344:
345:  }
346:  if (is.null(strata)==TRUE) {
347:
348:    write("Preparing to do trellis function densityplot",file="")
349:    if (nchar(logfile)>0) {
350:     write("Preparing to do trellis function densityplot",file=logfile,
351:           append=TRUE)
352:    }
353:
354:   trellis<-densityplot(data[,pmatch(var,names(data))],xlab=labvar)
355:
356:    write("function densityplot done",file="")
357:    if (nchar(logfile)>0) {
358:     write("function densityplot done",file=logfile,append=TRUE)
```

```
359:    }
360:
361:  write("Preparing to do SVG file",file="")
362:  if (nchar(logfile)>0) {
363:   write("Preparing to do SVG file",file=logfile,append=TRUE)
364:  }
365:
366:  file = paste(dirgraph,"/",namegraph,".svg",sep="")
367:  CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
368:  plot(trellis)
369:  dev.off()
370:
371:  write("SVG file: Done",file="")
372:  if (nchar(logfile)>0) {
373:   write("SVG file: Done",file=logfile,append=TRUE)
374:  }
375:
376:
377:  write("Preparing to do JPEG file",file="")
378:  if (nchar(logfile)>0) {
379:   write("Preparing to do JPEG file",file=logfile,append=TRUE)
380:  }
381:
382:  file = paste(dirgraph,"/",namegraph,".jpeg",sep="")
383:  jpeg(file, width=11.33, height=4.84, pointsize=9, units="in",
384:       bg = "transparent",res=9)
385:  plot(trellis)
386:  dev.off()
387:
388:  write("JPEG file: Done",file="")
389:  if (nchar(logfile)>0) {
390:   write("JPEG file: Done",file=logfile,append=TRUE)
391:  }
392:
393:  write("Preparing to do PNG file",file="")
394:  if (nchar(logfile)>0) {
395:   write("Preparing to do PNG file",file=logfile,append=TRUE)
396:  }
397:
398:
399:  file = paste(dirgraph,"/",namegraph,".png",sep="")
400:  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9,bg = "transparent")
401:  plot(trellis)
402:  dev.off()
403:
```

```
  write("PNG file: Done",file="")
  if (nchar(logfile)>0) {
   write("PNG file: Done",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")
  plot(trellis)
  dev.off()

  write("PDF file: Done",file="")
  if (nchar(logfile)>0) {
   write("PDF file: Done",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
```

```
448: # Version: 2008-10-23
449: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
450: # Programming Language: R 2.8.0
451: ## ------------------------
452: # DESCRIPTION
453: #
454: # BIRO_boxtrellis draws Box Plots possibly conditioned on other variables.
455: #
456: # DEPENDENCIES
457: #
458: #  R packages: grDevice, lattice, Cairo
459: #
460: # ##############################################################################
461: # PARAMETERS(!=required)
462: #
463: # data                ! =  a data frame containing data
464: # var                 ! =  the name of the conditionated variable
465: # strata    [NULL]      =  a vector containing the names of the conditionating
466: #                          variables
467: # labvar              ! =  string value contining the label of the conditionated
468: #                          variable
469: # labclass            ! =  a vector containing string values (labels of the
470: #                          conditionating variable)
471: # lev_class           ! =  list containing the levels of strata
472: # dirout    [getwd()]   =  the path of the output directory
473: # ext        [csv]      =  the extention of the output file
474: # date    [Sys.Date()]  =  the date of analysis
475: # namegraph           ! =  the name of the output file (text, jpg,png, eps)
476: # dirgraph            ! =  the path of the output graph directory
477: # ##############################################################################
478:
479: BIRO_boxtrellis<-function(data,
480:                           var,
481:                           strata=NULL,
482:                           labvar,
483:                           labclass,
484:                           lev_class,
485:                           dirout=getwd(),
486:                           ext="csv",
487:                           date=Sys.Date(),
488:                           namegraph,
489:                           dirgraph) {
490:
491:
492:
```

```
493:  #X11(width = 21, height =9 , pointsize=9)
494:
495:
496:  for (i in 1:length(strata)) {
497:   data[,pmatch(strata[i],names(data))]<-
498:   as.factor(data[,pmatch(strata[i],names(data))])
499:   levels(data[,pmatch(strata[i],names(data))])=lev_class[[i]]
500:
501:  }
502:
503:
504:  formula<-paste(var,"~",paste(strata,collapse="+"),sep="")
505:
506:  if (length(strata)>0) {
507:
508:  n<-NULL
509:  for (i in 1:length(lev_class)) {
510:   n<-c(n,length(lev_class[[i]]))
511:
512:  }
513:   firstlay<-prod(n)/8
514:
515:   nlev<-NULL
516:   for (i in 1:length(strata)) {
517:    nlev<-c(nlev,nlevels(factor(strata)))
518:   }
519:  }
520:
521:  if (is.null(strata)==FALSE) {
522:
523:   if (length(strata)==1) {
524:    secondpart<-strata[1]
525:   }
526:
527:   if (length(strata)>1) {
528:    secondpart<-paste(strata[1]," | ",paste(strata[2:length(strata)],
529:                      collapse="+"))
530:   }
531:
532:   formula<-paste(var," ~ ",secondpart)
533:   write("preparing to do trellis function bwplot",file="")
534:   trellis<-bwplot(as.formula(as.character(formula)),data=data,layout=c(firstlay,4))
535:   write("function bwplot done",file="")
536:
537:  file = paste(dirgraph,"/i",namegraph,"g4_3a.svg",sep="")
```

```
538:  CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
539:  plot(trellis)
540:  dev.off()
541:
542:  file = paste(dirgraph,"/i",namegraph,"g4_3a.jpeg",sep="")
543:  jpeg(file, width=11.33, height=4.84, pointsize=9, units="in",
544:       bg = "transparent",res=9)
545:  plot(trellis)
546:  dev.off()
547:
548:  file = paste(dirgraph,"/i",namegraph,"g4_3a.png",sep="")
549:  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9 , bg = "transparent")
550:  plot(trellis)
551:  dev.off()
552:
553:  file = paste(dirgraph,"/i",namegraph,"g4_3a.pdf",sep="")
554:  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
555:  plot(trellis)
556:  dev.off()
557:
558:  }
559:  if (is.null(strata)==TRUE) {
560:
561:   write("preparing to do trellis function bwplot",file="")
562:   trellis<-bwplot(data[,pmatch(var,names(data))],xlab=labvar)
563:   write("function bwplot done",file="")
564:
565:  file = paste(dirgraph,"/i",namegraph,"g4_3a.svg",sep="")
566:  CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
567:  plot(trellis)
568:  dev.off()
569:
570:  file = paste(dirgraph,"/i",namegraph,"g4_3a.jpeg",sep="")
571:  jpeg(file, width=11.33, height=4.84, pointsize=9, units="in",
572:       bg = "transparent",res=9)
573:  plot(trellis)
574:  dev.off()
575:
576:  file = paste(dirgraph,"/i",namegraph,"g4_3a.png",sep="")
577:  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9 , bg = "transparent")
578:  plot(trellis)
579:  dev.off()
580:
581:  file = paste(dirgraph,"/i",namegraph,"g4_3a.pdf",sep="")
582:  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
```

```
  583:  plot(trellis)
  584:  dev.off()
  585:
  586:  }
  587:  if (struc==TRUE) {
  588:    cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
  589:    cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
  590:    cat("  <tr>",file=sthtml,fill=TRUE,append=TRUE)
  591:    cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/",namegraph,".png\">","</td>",sep=""
),file=sthtml,fill=TRUE,append=TRUE)
  592:    cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
  593:    cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
  594:    cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
  595:  }
  596:
  597:
  598: }
  599:
  600:
  601:
```

```
# ##############################################################################
#                                                                              #
# 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: 2008-10-23
# OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
# 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
```

```
45: #  classlabel
46: #  classlabellist
47: #  BIRO_dframe
48: #
49: # ##############################################################################
50:
51: # ##############################################################################
52: #
53: # including also:
54: #
55: # sort.data.frame
56: # Author: Kevin Wright (kw.statr@gmail.com)
57: # with some ideas from Andy Liaw
58: # http://tolstoy.newcastle.edu.au/R/help/04/07/1076.html
59: #
60: # ##############################################################################
61:
62: # ##############################################################################
63: #  varclass
64: # ##############################################################################
65: # ------------------------
66: # Authors:
67: # Luca Rossi       <redsluke@gmail.com>
68: # Fabrizio Carinci <research@fabcarinci.net>
69: # Institutions: Serectrix snc
70: # Created: 2008-08-17
71: # Version: 2008-10-23
72: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
73: # Programming Language: R 2.8.0
74: # ------------------------
75: # DESCRIPTION
76: #
77: # varclass: the output is a vector with the same length of v,
78: #            with elements the values of v espressed in classes,
79: #            according the c thresholds vector
80: #
81: # DEPENDENCIES
82: #
83: #  R package: base
84: #
85: # ##############################################################################
86: # PARAMETERS(!=required)
87: #
88: # v        !   = a vector containing data
89: # c        !   = a vector containing the thresholds
```

```
 90: # ############################################################################
 91:
 92: varclass<-function(v,c) {
 93:
 94:  x<-rep(NA,length(v))
 95:  n<-length(c)
 96:
 97: for(i in 1:n) {
 98:  if (i==1) {
 99:   x<-ifelse(is.na(v)==FALSE & v < c[i],i,x)
100:  }
101:  if (i>1) {
102:   x<-ifelse(is.na(v)==FALSE & v >= c[i-1] & v < c[i],i,x)
103:  }
104:   x<-ifelse(is.na(v)==FALSE & v >= c[n],n+1,x)
105:  }
106:  return(x)
107: }
108:
109: # ############################################################################
110: #  classlabel
111: # ############################################################################
112: # ------------------------
113: # Authors:
114: # Luca Rossi       <redsluke@gmail.com>
115: # Fabrizio Carinci <research@fabcarinci.net>
116: # Institutions: Serectrix snc
117: # Created: 2008-08-17
118: # Version: 2008-10-23
119: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
120: # Programming Language: R 2.8.0
121: # ------------------------
122: # DESCRIPTION
123: #
124: # classlabel: create label from a thresholds vector. output is a vector
125: #
126: # DEPENDENCIES
127: #
128: #  R package: base
129: #
130: # ############################################################################
131: # PARAMETERS(!=required)
132: #
133: # v          !  =        a vector containing the thresholds
134: # ############################################################################
```

```
135:
136: classlabel<-function(v) {
137:  label<-rep(NA,length(v)+1)
138:  for (i in (1:length(v))) {
139:   if (i==1) {label[i]<-paste("0 -",v[1]-1)}
140:   if (i>1) {label[i]<-paste(v[i-1],"-",v[i]-1)}
141:  }
142:  label[length(v)+1]<-paste(v[length(v)],"+",sep=" ")
143:  return(label)
144: }
145:
146: # ##############################################################################
147: #  classlabellist
148: # ##############################################################################
149: # ------------------------
150: # Authors:
151: # Luca Rossi       <redsluke@gmail.com>
152: # Fabrizio Carinci <research@fabcarinci.net>
153: # Institutions: Serectrix snc
154: # Created: 2008-08-17
155: # Version: 2008-10-23
156: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
157: # Programming Language: R 2.8.0
158: # ------------------------
159: # DESCRIPTION
160: #
161: # classlabellist: create label from a thresholds vector. output is a list
162: #
163: # DEPENDENCIES
164: #
165: #  R package: base
166: #
167: # ##############################################################################
168: # PARAMETERS(!=required)
169: #
170: # v               ! =     a vector containing the thresholds
171: # ##############################################################################
172:
173: classlabellist<-function(v) {
174:  label<-rep(NA,length(v)+1)
175:  for (i in (1:length(v))) {
176:   if (i==1) {label[i]<-paste("0 -",v[1]-1)}
177:   if (i>1) {label[i]<-paste(v[i-1],"-",v[i]-1)}
178:  }
179:  label[length(v)+1]<-paste(v[length(v)],"+",sep=" ")
```

```
180:
181:  labellist<-list()
182:  for (i in 1:(length(v)+1)) {
183:  labellist[[i]]=i
184:  }
185:  names(labellist)<-label
186:  return(labellist)
187: }
188:
189:
190: # ############################################################################
191: # sort.data.frame
192: # ############################################################################
193: # Author: Kevin Wright (kw.statr@gmail.com)
194: # with some ideas from Andy Liaw
195: # http://tolstoy.newcastle.edu.au/R/help/04/07/1076.html
196: #
197: # ############################################################################
198: #
199: #  DESCRIPTION
200: #
201: #  A unified method for sorting data frames with mixed data types
202: #  is provided by the following function by Kevin Wright.
203: #
204: # ############################################################################
205:
206: sort.data.frame <- function(x, by){
207:     # Author: Kevin Wright (kw.statr@gmail.com)
208:     # with some ideas from Andy Liaw
209:     # http://tolstoy.newcastle.edu.au/R/help/04/07/1076.html
210:
211:     # x: A data.frame
212:     # by: A one-sided formula using + for ascending and - for descending
213:     #     Sorting is left to right in the formula
214:
215:     # Useage is:
216:     # library(nlme);
217:     # data(Oats)
218:     # sort(Oats, by= ~nitro-Variety)
219:
220:     if(by[[1]] != "~")
221:         stop("Argument 'by' must be a one-sided formula.")
222:
223:     # Make the formula into character and remove spaces
224:     formc <- as.character(by[2])
```

```
225:     formc <- gsub(" ", "", formc)
226:     # If the first character is not + or -, add +
227:     if(!is.element(substring(formc, 1, 1), c("+", "-")))
228:         formc <- paste("+", formc, sep = "")
229:
230:     # Extract the variables from the formula
231:     vars <- unlist(strsplit(formc, "[\\+\\-]"))
232:     vars <- vars[vars != ""] # Remove any extra "" terms
233:
234:     # Build a list of arguments to pass to "order" function
235:     calllist <- list()
236:     pos <- 1 # Position of + or -
237:     for(i in 1:length(vars)){
238:         varsign <- substring(formc, pos, pos)
239:         pos <- pos + 1 + nchar(vars[i])
240:         if(is.factor(x[, vars[i]])){
241:             if(varsign == "-") {
242:                 calllist[[i]] <- -rank(x[, vars[i]])
243:             } else {
244:                 calllist[[i]] <- rank(x[, vars[i]])
245:             }
246:         } else {
247:             if(varsign == "-") {
248:                 calllist[[i]] <- -x[, vars[i]]
249:             } else {
250:                 calllist[[i]] <- x[,vars[i]]
251:             }
252:         }
253:     }
254:     return(x[do.call("order", calllist), ])
255: }
256:
257:
258: # ############################################################################
259: #  BIRO_dframe
260: # ############################################################################
261: # ------------------------
262: # Authors:
263: # Luca Rossi       <redsluke@gmail.com>
264: # Fabrizio Carinci <research@fabcarinci.net>
265: # Institutions: Serectrix snc
266: # Created: 2008-08-17
267: # Version: 2008-10-23
268: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
269: # Programming Language: R 2.8.0
```

```
270: # ------------------------
271: # DESCRIPTION
272: #
273: # BIRO_dframe: print on screen or in a file the description of a data frame
274: #
275: # DEPENDENCIES
276: #
277: #  R package: base
278: #
279: # ##########################################################################
280: # PARAMETERS(!=required)
281: #
282: # dataname      !       =  the name (string) of the target data frame
283: # monitor       [TRUE]  =  logical. True if you want the description printed
284: #                          on the screen
285: # filelog       [""]    =  the name and the path of the log file
286: # ##########################################################################
287:
288: BIRO_dframe<-function(dataname,pos=-1,monitor=TRUE,filelog="") {
289:
290:  mydata<-get(dataname,pos = pos, envir = as.environment(pos))
291:
292:  if (monitor==TRUE) {
293:
294:    print("#############################################################################################",quote=FALSE)
295:    print(paste("#                    DATASET name:",dataname),quote=FALSE)
296:    print("#",quote=FALSE)
297:    print(paste("#                 Number of Rows:",dim(mydata)[1]),quote=FALSE)
298:    print(paste("#              Number of Columns:",dim(mydata)[2]),quote=FALSE)
299:    print("#+-----------------------------------------------------------------------------------",quote=FALSE)
300:    print("#                                  Not Determinated Values...",quote=FALSE)
301:    print("#",quote=FALSE)
302:    for (i in 1:dim(mydata)[2]) {
303:     NAnum<-length(mydata[is.na(mydata[,i]),i])
304:     Infnum<-length(mydata[is.infinite(mydata[,i]),i])
305:     print(paste("#| ",names(mydata)[i],NAnum+Infnum,sep=" "),quote=FALSE)
306:    }
307:    print("#############################################################################################",quote=FALSE)
308:
309:   }
310:
311:   if (nchar(filelog)>0) {
312:
313:     cat("#############################################################################################"
,file=filelog,append=TRUE,fill=TRUE)
```

```
  314:     cat(paste("# NOTE                DATASET name:",dataname),file=filelog,append=TRUE,fill=TRUE)
  315:     cat("#",file=filelog,append=TRUE,fill=TRUE)
  316:     cat(paste("#                   Number of Rows:",dim(mydata)[1]),file=filelog,append=TRUE,fill=TRUE)
  317:     cat(paste("#                Number of Columns:",dim(mydata)[2]),file=filelog,append=TRUE,fill=TRUE)
  318:     cat("#+-------------------------------------------------------------------------------------"
,file=filelog,append=TRUE,fill=TRUE)
  319:     cat("#                                  Not Determinated Values...",file=filelog,append=TRUE,fill=TRUE)
  320:     cat("#",file=filelog,append=TRUE,fill=TRUE)
  321:     for (i in 1:dim(mydata)[2]) {
  322:      NAnum<-length(mydata[is.na(mydata[,i]),i])
  323:      Infnum<-length(mydata[is.infinite(mydata[,i]),i])
  324:      cat(paste("#| ",names(mydata)[i],NAnum+Infnum,sep=" "),file=filelog,append=TRUE,fill=TRUE)
  325:     }
  326:   cat("#######################################################################################"
,file=filelog,append=TRUE,fill=TRUE)
  327:   }
  328:
  329:
  330:  }
  331:
  332:
  333:
  334:
  335:
```

```
  1: # ############################################################################
  2: #                                                                            #
  3: # Project: BIRO-Project (Funded by European Commission 2005-2008)            #
  4: # File: BIRO_webplots.r                                                      #
  5: #                                                                            #
#
  6: # ############################################################################
  7:
  8: # ------------------------
  9: # Authors:
 10: # Luca Rossi       <redsluke@gmail.com>
 11: # Fabrizio Carinci <research@fabcarinci.net>
 12: # Institutions: Serectrix snc
 13: # Created: 2007-11-24
 14: # Version: 2008-11-21
 15: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
 16: # Programming Language: R 2.8.0
 17: # ------------------------
 18: # COPYRIGHT INFORMATION
 19: #
 20: # This file is free software; you can redistribute it and/or modify
 21: # it under the terms of the GNU General Public License as published by
 22: # the Free Software Foundation; either version 2, or (at your option)
 23: # any later version.
 24: #
 25: # This file is distributed in the hope that it will be useful,
 26: # but WITHOUT ANY WARRANTY; without even the implied warranty of
 27: # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 28: # GNU General Public License for more details.
 29: #
 30: # You should have received a copy of the GNU General Public License
 31: # along with this file; see the file COPYING.  If not, write to
 32: # the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
 33: #
 34: # In short: you may use this file any way you like, as long as you
 35: # don't charge money for it, remove this notice, or hold anyone liable
 36: # for its results.
 37: #
 38: # BIRO_location.r is part of WP Statistical Engine of the BIRO Project
 39: # GPL Copyright, The BIRO Project
 40: #
 41: # ------------------------
 42: #
 43: # CONTENT
 44: #
```

```
45: #   BIRO_webplots
46: #
47: # ###########################################################################
48:
49: # ###########################################################################
50: #  BIRO_webplots                                                            #
51: # ###########################################################################
52: # ------------------------
53: # Authors:
54: # Luca Rossi       <redsluke@gmail.com>
55: # Fabrizio Carinci <research@fabcarinci.net>
56: # Institutions: Serectrix snc
57: # Created: 2007-11-24
58: # Version: 2008-11-21
59: # OS: Windows XP/Vista, Fedora Core 9 running Kernel:2.6.27.5-37.fc9.i686
60: # Programming Language: R 2.8.0
61: # ------------------------
62: # DESCRIPTION
63: #
64: # BIRO_webplots draws webplots for a conditionated distribution
65: #
66: # DEPENDENCIES
67: #
68: #  R packages: grDevice, graphics, Cairo
69: #
70: # ###########################################################################
71: # PARAMETERS(!=required)
72: #
73: # data        !             = dataset containing data
74: # var         !             = the name of the target variable
75: # labvar      !             = a vector containing the labels of var
76: # title       !             = a string containing the title of graph
77: # ext         [csv]         = the extension of the output files
78: # date        [System date] = date of analysis
79: # dirout      [current]     = the path of the output text files
80: # namegraph   !             = the coded name of the output files
81: # dirgraph    [current]     = the path of the output (jpg,eps,png files)
82: # dirdataout  !             = the path of the output (data files)
83: # ###########################################################################
84:
85:
86: BIRO_spider<-function(data,
87:                       var,
88:                       labvars,
89:                       title,
```

```
 90:                       dirout=getwd(),
 91:                       ext="csv",
 92:                       date=Sys.Date(),
 93:                       namegraph,
 94:                       dirgraph,
 95:                       dirdataout) {
 96:
 97:  write("Preparing to do SVG file",file="")
 98:  if (nchar(logfile)>0) {
 99:   write("Preparing to do SVG file",file=logfile,append=TRUE)
100:  }
101:
102:  file = paste(dirgraph,"/i",namegraph,"g4_7a.svg",sep="")
103:  CairoSVG(file, width = 11.33, height =4.84, pointsize=9, bg = "transparent")
104:  stars(data[,pmatch(var,names(data))], key.loc=c(2,35),
105:  main=title, frame=T)
106:
107:  dev.off()
108:
109:  write("SVG file: Done",file="")
110:  if (nchar(logfile)>0) {
111:   write("SVG file: done",file=logfile,append=TRUE)
112:  }
113:
114:
115:  write("Preparing to do JPEG file",file="")
116:  if (nchar(logfile)>0) {
117:   write("Preparing to do JPEG file",file=logfile,append=TRUE)
118:  }
119:
120:  file = paste(dirgraph,"/i",namegraph,"g4_7a.jpeg",sep="")
121:  jpeg(file, width= 11.33, height= 4.84, pointsize=9, bg = "transparent",
122:       units="in",res=9)
123:  stars(data[,pmatch(var,names(data))], key.loc=c(2,35),
124:  main=title, frame=T)
125:
126:  dev.off()
127:
128:  write("JPEG file: Done",file="")
129:  if (nchar(logfile)>0) {
130:   write("JPEG file: done",file=logfile,append=TRUE)
131:  }
132:
133:  write("Preparing to do PNG file",file="")
134:  if (nchar(logfile)>0) {
```

```
135:   write("Preparing to do PNG file",file=logfile,append=TRUE)
136:  }
137:
138:   file = paste(dirgraph,"/i",namegraph,"g4_7a.png",sep="")
139:  CairoPNG(file, width= 813.6, height= 348.48, pointsize=9, bg = "transparent")
140:  stars(data[,pmatch(var,names(data))], key.loc=c(2,35),
141:  main=title, frame=T)
142:
143:  dev.off()
144:
145:  write("PNG file: Done",file="")
146:  if (nchar(logfile)>0) {
147:   write("PNG file: done",file=logfile,append=TRUE)
148:  }
149:
150:  write("Preparing to do PDF file",file="")
151:  if (nchar(logfile)>0) {
152:   write("Preparing to do PDF file",file=logfile,append=TRUE)
153:  }
154:
155:  file = paste(dirgraph,"/i",namegraph,"g4_7a.pdf",sep="")
156:  pdf(file, width = 11.33, height =4.84 , pointsize = 9, bg = "transparent")
157:  stars(data[,pmatch(var,names(data))], key.loc=c(2,35),
158:  main=title, frame=T)
159:
160:  dev.off()
161:
162:  write("PDF file: Done",file="")
163:  if (nchar(logfile)>0) {
164:   write("PDF file: done",file=logfile,append=TRUE)
165:  }
166:
167:
168:
169:  write.csv(data[,pmatch(var,names(data))],
170:            paste(dirdataout,"/i",csvindex,"d4_7a.csv",sep=""),
171:            row.names = FALSE)
172:
173:  write("Webplot data stored",file="")
174:  if (nchar(logfile)>0) {
175:   write("Webplot data stored",file=logfile,append=TRUE)
176:  }
177:
178:  if (struc==TRUE) {
179:    cat("<table border='1'>",file=sthtml,fill=TRUE,append=TRUE)
```

```
  180:    cat(" <tbody>",file=sthtml,fill=TRUE,append=TRUE)
  181:    cat("  <tr>",file=sthtml,fill=TRUE,append=TRUE)
  182:    cat(paste("   <td><img src=\"",dirse,"/output/reports/#<datetime>/graphs/web",namegraph,".png\">","</td>",sep=""
),file=sthtml,fill=TRUE,append=TRUE)
  183:    cat("  </tr>",file=sthtml,fill=TRUE,append=TRUE)
  184:    cat(" </tbody>",file=sthtml,fill=TRUE,append=TRUE)
  185:    cat("</table>",file=sthtml,fill=TRUE,append=TRUE)
  186:  }
  187:
  188:
  189:
  190: }
  191:
  192:
  193:
```

Table of Contents