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

# R SCRIPT LINKED TO SCHOUTEN (2017), Statistical inference based on randomly generated auxiliary variables, JRSSB

#

# CODE FOR READING THE LISS-PANEL DATA SET AND REPRODUCING THE CALCULATIONS IN SECTION 5

#

# VERSION: MAY 16, 2017

# AUTHOR: BARRY SCHOUTEN (BSTN@CBS.NL)

#

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

library(foreign)

# READ LISS DATA SET

liss = read.spss("LISS-JRSSB.sav",use.value.labels=FALSE,use.missings=FALSE)

liss = as.data.frame(liss)

n = nrow(liss)

# RE-LABEL VARIABLES IN LISS-FILE IN ORDER TO FACILITATE CALCULATIONS OVER (SUBSETS OF) VARIABLES

# WITH EXCEPTION OF VARIABLES 16 - 21 (R09 TO R14 = 0-1 ATTRITION PER YEAR)

nvartot = length(names(liss))

nvar = nvartot - 6

for (i in 1:15){

names(liss)[i] = paste("V",i,sep="") }

for (i in 2:nvartot){

names(liss)[i] = paste("V",i-6,sep="") }

# MISSING VALUES ON VARIABLES IMPLY FILTER QUESTIONS, I.E QUESTIONS THAT ARE NOT APPLICABLE

# VALUES -2 IMPLIY REFUSALS BY THE RESPONDENTS TO PROVIDE ANSWERS TO THE CORRESPONDING QUESTIONS

# VALUES -3 IMPLY THAT A VARIABLE CONTAINS A MEASUREMENT ERROR/INCONSISTENCY

# RECODE MISSING VALUES TO -4, AND SET REFUSALS AND MEASUREMENT ERRORS TO MISSING

for (i in 1:nvar){

label = paste("V",i,sep="")

liss[[label]] = as.numeric(liss[[label]])

for (j in 1:n){

if (is.na(liss[[label]][j])) {liss[[label]][j] = -4}

if ((liss[[label]][j] == -2) | (liss[[label]][j] == -3)) {liss[[label]][j] = NA}

}

liss[[label]] = as.factor(liss[[label]])

}

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

# JRSSB ANALYSES

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

# PREPARATIONS

# SCRIPT FOR ESTIMATION OF R-INDICATORS AND CV'S (AVAILABLE AT WWW.RISQ-PROJECT.EU)

source("RISQ\_R-indicators\_v2.1.r")

# NUMBER OF CATEGORIES PER VARIABLE

Cmax = rep(0,nvar)

for (i in 1:nvar){

Cmax[i] = nlevels(liss[[ paste("V",i,sep="") ]] )

}

# AVERAGE NUMBER OF CATEGORIES

Chata = mean(Cmax)

# DIVERSITY (SET AT SIZE OF PANEL)

G = 2205

# SECTION 5.1, ESTIMATION OF LISS DIVERSITY

# CHISQUARE STATISTICS FOR ALL POSSIBLE PAIRS OF VARIABLES

lengte = nvar\*(nvar-1)\*0.5

chsqstat = rep(0,lengte)

Chat = rep(0,lengte)

teller = 0

telna = 0

Ghattel = 0

Ghatnoem = 0

for (i in 1:(nvar-1)){

lab1 = paste("V",i,sep="")

for (j in (i+1):nvar){

lab2 = paste("V",j,sep="")

#cat(lab1,lab2,"\n")

teller = teller + 1

chsqstat[teller] = chisq.test(table(liss[[lab1]],liss[[lab2]]),correct=FALSE)$statistic/n

Chat[teller] = (nlevels(liss[[lab1]]) -1) \* (nlevels(liss[[lab2]]) - 1)

if (!is.na(chsqstat[teller])) {

Ghattel = Ghattel + Chat[teller]

Ghatnoem = Ghatnoem + chsqstat[teller]

telna = telna + 1

}

}

}

# ESTIMATE FOR LISS-PANEL DIVERSITY G

Ghat = Ghattel/Ghatnoem

# SECTION 5.2, TABLE 1 ROW 1

#ATTRITION RATES PER YEAR

attrate = rep(0,5)

attrate[1] = mean(liss$R10)

attrate[2] = mean(liss$R11)

attrate[3] = mean(liss$R12)

attrate[4] = mean(liss$R13)

attrate[5] = mean(liss$R14)

# SECTION 5.2, TABLE 1 ROW 2

# ESTIMATE AVERAGE CV OF ATTRITION PROPENSITIES OVER ALL VARIABLES PER YEAR (2010 TO 2014)

n = nrow(liss)

CV1 = matrix(NA,nvar,5)

empty = 0 # COUNTER FOR NUMBER OF CONSTANT VARIABLES

for (b in 1:nvar){

cat("Variable:",b,"\n")

form10 = paste("R10 ~ V",b,sep="")

form11 = paste("R11 ~ V",b,sep="")

form12 = paste("R12 ~ V",b,sep="")

form13 = paste("R13 ~ V",b,sep="")

form14 = paste("R14 ~ V",b,sep="")

model10 = formula(form10)

model11 = formula(form11)

model12 = formula(form12)

model13 = formula(form13)

model14 = formula(form14)

index = rep(TRUE,n)

label = paste("V",b,sep="")

index = index & !is.na(liss[[ label ]])

liss2 = liss[index,]

n2 = nrow(liss2)

if (n2<n) { cat(n-n2,"\n") }

if (nlevels(liss2[[ label ]]) > 1) {

ind10 = getRIndicator(model10,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind11 = getRIndicator(model11,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind12 = getRIndicator(model12,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind13 = getRIndicator(model13,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind14 = getRIndicator(model14,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

CV1[b,1] = ind10$CV

CV1[b,2] = ind11$CV

CV1[b,3] = ind12$CV

CV1[b,4] = ind13$CV

CV1[b,5] = ind14$CV

} else { empty = empty + 1 }

}

CVavg = rep(0,5)

for (i in 1:5){

CVavg[i] = mean(CV1[,i],na.rm=TRUE)

}

# SECTION 5.2, TABLE 1 ROW 3 (ALSO TABLE 5 ROW 1)

# ESTIMATE TOTAL CV

CVmax = CVavg \* sqrt((G-1)/(Chata-1))

# SECTION 5.2, TABLE 1 ROW 4

# ESTIMATE CV OF ATTRITION PROPENSITIES FOR RANDOM DRAWS OF 10 VARIABLES PER YEAR (2010 TO 2014)

n = nrow(liss)

nvarsel = 10

nvarrest = nvar - 10

B = 200

CVb = matrix(0,B,5)

Rb = matrix(0,B,5)

for (b in 1:B){

cat("Iteration:",b,"\n")

X = sample(x=c(1:nvar),size=nvarsel)

Xrest = is.element(1:nvar,X)

form10 = "R10 ~ "

form11 = "R11 ~ "

form12 = "R12 ~ "

form13 = "R13 ~ "

form14 = "R14 ~ "

for (i in 1:(nvarsel-1)){

form10 = paste(form10,"V",X[i]," + ",sep="")

form11 = paste(form11,"V",X[i]," + ",sep="")

form12 = paste(form12,"V",X[i]," + ",sep="")

form13 = paste(form13,"V",X[i]," + ",sep="")

form14 = paste(form14,"V",X[i]," + ",sep="")

}

form10 = paste(form10,"V",X[nvarsel],sep="")

form11 = paste(form11,"V",X[nvarsel],sep="")

form12 = paste(form12,"V",X[nvarsel],sep="")

form13 = paste(form13,"V",X[nvarsel],sep="")

form14 = paste(form14,"V",X[nvarsel],sep="")

model10 = formula(form10)

model11 = formula(form11)

model12 = formula(form12)

model13 = formula(form13)

model14 = formula(form14)

index = rep(TRUE,n)

for (i in 1:nvarsel){

label = paste("V",X[i],sep="")

index = index & !is.na(liss[[ label ]])

}

liss2 = liss[index,]

n2 = nrow(liss2)

if (n2<n) { cat(n-n2,"\n") }

ind10b = getRIndicator(model10,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind11b = getRIndicator(model11,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind12b = getRIndicator(model12,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind13b = getRIndicator(model13,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind14b = getRIndicator(model14,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

CVb[b,1] = ind10b$CV

CVb[b,2] = ind11b$CV

CVb[b,3] = ind12b$CV

CVb[b,4] = ind13b$CV

CVb[b,5] = ind14b$CV

}

CVavg10 = rep(0,5)

for (i in 1:5){

CVavg10[i] = mean(CVb[,i],na.rm=TRUE)

}

# SECTION 5.2, TABLE 1 ROW 5

# PROPORTION OF TOTAL VARIATION

100 \* CVavg10/CVmax

# SECTION 5.2, TABLE 1 ROW 6

# ESTIMATE CV OF ATTRITION PROPENSITIES FOR DEMOGRAPHIC VARIABLES PER YEAR (2010 TO 2014)

# DEMOGRAPHIC X ARE V1 TO V9 and V12 TO V14

n = nrow(liss)

CVdemo = rep(0,5)

model10 = formula(R10 ~ V1 + V4 + V5 + V6 + V7 + V8 + V9 + V12 + V13 + V14)

model11 = formula(R11 ~ V1 + V4 + V5 + V6 + V7 + V8 + V9 + V12 + V13 + V14)

model12 = formula(R12 ~ V1 + V4 + V5 + V6 + V7 + V8 + V9 + V12 + V13 + V14)

model13 = formula(R13 ~ V1 + V4 + V5 + V6 + V7 + V8 + V9 + V12 + V13 + V14)

model14 = formula(R14 ~ V1 + V4 + V5 + V6 + V7 + V8 + V9 + V12 + V13 + V14)

ind10 = getRIndicator(model10,liss,sampleWeights=rep(16\*10^6/n,n),withPartials=FALSE,withPartialCV=FALSE)

ind11 = getRIndicator(model11,liss,sampleWeights=rep(16\*10^6/n,n),withPartials=FALSE,withPartialCV=FALSE)

ind12 = getRIndicator(model12,liss,sampleWeights=rep(16\*10^6/n,n),withPartials=FALSE,withPartialCV=FALSE)

ind13 = getRIndicator(model13,liss,sampleWeights=rep(16\*10^6/n,n),withPartials=FALSE,withPartialCV=FALSE)

ind14 = getRIndicator(model14,liss,sampleWeights=rep(16\*10^6/n,n),withPartials=FALSE,withPartialCV=FALSE)

CVdemo[1] = ind10$CV

CVdemo[2] = ind11$CV

CVdemo[3] = ind12$CV

CVdemo[4] = ind13$CV

CVdemo[5] = ind14$CV

# SECTION 5.2, TABLE 1 ROW 7

# PROPORTION OF TOTAL VARIATION

100 \* CVdemo/CVmax

# SECTION 5.2, TABLE 3

# PROPORTION OF TOTAL VARIATION TO MAXIMAL VARIATION

100 \* CVmax/sqrt((1-attrate)/attrate)

# SECTION 5.2, TABLE 3

# DERIVE PREFERENCES FOR YEAR PER VARIABLES BASED ON RANDOM DRAWS OF VARIABLES

B = 200

CVrank = matrix(0,B,5)

for (b in 1:B){

CVrank[b,] = rank(CVb[b,])

}

CVgem = apply(CVrank,2,mean)

CVbest = rep(0,5)

for (b in 1:B){

aantcvnul = sum(as.numeric(CVb[b,]==0))

if (aantcvnul==0) {

ranknul = 1

} else { ranknul = sum(1:aantcvnul)/aantcvnul }

for (i in 1:5){

if (CVrank[b,i] == ranknul) { CVbest[i] = CVbest[i] + 1 }

}

}

# SECTION 5.3, TABLE 4

# LABELS OF CORE STUDY TARGET VARIABELS

health = 168

dwelling = 343

sports = 591

survey = 672

# NUMBER OF VARIABLES DRAWN AND NUMBER OF DRAWS

nvarsel = 20

B = 200

# SELECT VARIABLES THAT ASSOCIATE MINIMALLY TO A TARGET VARIABLE

# FUNCTION TO SELECT VARIABLES THAT ASSOCIATE MINIMALLY TO A SPEFICIED TARGET VARIABLE

CVRkernvar = function(liss,targetvar,threshold,nvarsel=20,B=200){

n = nrow(liss)

nvarrest = nvar - nvarsel

CVb =matrix(0,B,5)

lab1 = paste("V",targetvar,sep="")

nempty = 0

for (b in 1:B){

cat("Iteration:",b,"\n")

Xtussen = sample(x=c(1:nvar),size=nvarsel)

X=rep(0,nvarsel)

teller=0

for (i in 1:nvarsel){

lab2 = paste("V",Xtussen[i],sep="")

noem = (nlevels(liss[[lab1]]) -1) \* (nlevels(liss[[lab2]]) - 1)

if (noem>0){

chsqvalue = (chisq.test(table(liss[[lab1]],liss[[lab2]]),correct=FALSE)$statistic)/(n\*noem)

} else { chisqvalue = 0 }

if (!is.na(chsqvalue)){

if (sqrt(chsqvalue)>threshold) {

teller = teller + 1

X[teller] = Xtussen[i] }

}

}

nvarselcv = teller

if (nvarselcv>0){

Xrest = is.element(1:nvar,X)

form10 = "R10 ~ "

form11 = "R11 ~ "

form12 = "R12 ~ "

form13 = "R13 ~ "

form14 = "R14 ~ "

if (nvarselcv>1){

for (i in 1:(nvarselcv-1)){

form10 = paste(form10,"V",X[i]," + ",sep="")

form11 = paste(form11,"V",X[i]," + ",sep="")

form12 = paste(form12,"V",X[i]," + ",sep="")

form13 = paste(form13,"V",X[i]," + ",sep="")

form14 = paste(form14,"V",X[i]," + ",sep="")

}

}

form10 = paste(form10,"V",X[nvarselcv],sep="")

form11 = paste(form11,"V",X[nvarselcv],sep="")

form12 = paste(form12,"V",X[nvarselcv],sep="")

form13 = paste(form13,"V",X[nvarselcv],sep="")

form14 = paste(form14,"V",X[nvarselcv],sep="")

model10 = formula(form10)

model11 = formula(form11)

model12 = formula(form12)

model13 = formula(form13)

model14 = formula(form14)

index = rep(TRUE,n)

for (i in 1:nvarselcv){

label = paste("V",X[i],sep="")

index = index & !is.na(liss[[ label ]])

}

liss2 = liss[index,]

n2 = nrow(liss2)

ind10b = getRIndicator(model10,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind11b = getRIndicator(model11,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind12b = getRIndicator(model12,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind13b = getRIndicator(model13,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind14b = getRIndicator(model14,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

CVb[b,1] = ind10b$CV

CVb[b,2] = ind11b$CV

CVb[b,3] = ind12b$CV

CVb[b,4] = ind13b$CV

CVb[b,5] = ind14b$CV

}

if (nvarselcv==0){ nempty = nempty + 1 }

}

list(CV=CVb, nempty = nempty)

}

# VARIABLE SELECTIONS PER TARGET VARIABLE. THRESHOLD IS SET TO AVERAGE CHISQUARE (INVERSE OF ESTIMATED POPULATION DIVERSITY)

threshold = sqrt(1/Ghat)

resHE = CVRkernvar(liss,health,threshold,B=200)

resSP = CVRkernvar(liss,sports,threshold,B=200)

resSU = CVRkernvar(liss,survey,threshold,B=200)

resDW = CVRkernvar(liss,dwelling,threshold,B=200)

# SELECTION OF DEMOGRAPHIC VARIABLES THAT ASSOCIATE MINIMALLY TO TARGET VARIABLES

# FUNCTION TO SELECT VARIABLES FROM A SPECIFIED SUBSET OF VARIABLES THAT ASSOCIATE MINIMALLY TO A SPECIFIED TARGET VARIABLE

CVRkernvarReg = function(liss,kernvar,threshold,x){

n = nrow(liss)

nvarsel = length(x)

nvarrest = nvar - nvarsel

CV = rep(0,5)

lab1 = paste("V",kernvar,sep="")

nempty = 0

Xtussen = x

X=rep(0,nvarsel)

teller=0

for (i in 1:nvarsel){

lab2 = paste("V",Xtussen[i],sep="")

noem = (nlevels(liss[[lab1]]) -1) \* (nlevels(liss[[lab2]]) - 1)

if (noem>0){

chisqvalue = (chisq.test(table(liss[[lab1]],liss[[lab2]]),correct=FALSE)$statistic)/(n\*noem)

} else { chisqvalue = 0 }

if (!is.na(chisqvalue)){

if (sqrt(chisqvalue)>threshold) {

teller = teller + 1

X[teller] = Xtussen[i] }

}

}

nvarselcv = teller

cat(X,"\n")

if (nvarselcv>0){

Xrest = is.element(1:nvar,X)

form10 = "R10 ~ "

form11 = "R11 ~ "

form12 = "R12 ~ "

form13 = "R13 ~ "

form14 = "R14 ~ "

if (nvarselcv>1){

for (i in 1:(nvarselcv-1)){

form10 = paste(form10,"V",X[i]," + ",sep="")

form11 = paste(form11,"V",X[i]," + ",sep="")

form12 = paste(form12,"V",X[i]," + ",sep="")

form13 = paste(form13,"V",X[i]," + ",sep="")

form14 = paste(form14,"V",X[i]," + ",sep="")

}

}

form10 = paste(form10,"V",X[nvarselcv],sep="")

form11 = paste(form11,"V",X[nvarselcv],sep="")

form12 = paste(form12,"V",X[nvarselcv],sep="")

form13 = paste(form13,"V",X[nvarselcv],sep="")

form14 = paste(form14,"V",X[nvarselcv],sep="")

model10 = formula(form10)

model11 = formula(form11)

model12 = formula(form12)

model13 = formula(form13)

model14 = formula(form14)

index = rep(TRUE,n)

for (i in 1:nvarselcv){

label = paste("V",X[i],sep="")

index = index & !is.na(liss[[ label ]])

}

liss2 = liss[index,]

n2 = nrow(liss2)

ind10 = getRIndicator(model10,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind11 = getRIndicator(model11,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind12 = getRIndicator(model12,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind13 = getRIndicator(model13,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

ind14 = getRIndicator(model14,liss2,sampleWeights=rep(16\*10^6/n2,n2),withPartials=FALSE,withPartialCV=FALSE)

CV[1] = ind10$CV

CV[2] = ind11$CV

CV[3] = ind12$CV

CV[4] = ind13$CV

CV[5] = ind14$CV

}

if (nvarselcv==0){ nempty = nempty + 1 }

list(CV=CV,nempty = nempty,nvar=nvarselcv,ind10=ind10,ind11=ind11,ind12=ind12,ind13=ind13,ind14=ind14)

}

X = c(1,4,5,6,7,8,9,12,13,14)

threshold = sqrt(1/Ghat)

resHEReg = CVRkernvarReg(liss,health,threshold,X)

resSPReg = CVRkernvarReg(liss,sports,threshold,X)

resSUReg = CVRkernvarReg(liss,survey,threshold,X)

resDWReg = CVRkernvarReg(liss,dwelling,threshold,X)

# SECTION 5.3, TABLE 5 ROWs 2 TO 5

# AVERAGE CV'S FOR SELECTED VARIABLES FROM RANDOM DRAWS PER TARGET VARIABLE

apply(resHE$CVb,2,mean)

apply(resSP$CVb,2,mean)

apply(resSU$CVb,2,mean)

apply(resDW$CVb,2,mean)

# SECTION 5.3, TABLE 5 ROWS 6 TO 9

# AVERAGE CV'S FOR SELECTED DEMOGRAPIC VARIABLES PER TARGET VARIABLE

apply(resHEreg$CVb,2,mean)

apply(resSPreg$CVb,2,mean)

apply(resSUreg$CVb,2,mean)

apply(resDWreg$CVb,2,mean)