**Data simulation script - R**

#Needed packages

library(survival)

library(MASS)

library(psych)

library(lme4)

library(plm)

library(geepack)

# Basic specs

mzcode0=0

dzcode1=1

Nrep=1

best=matrix(0,Nrep,12)

Nmz=50000 # N is twin pairs

Ndz=50000 # N is twin pairs

Nun=50000 # N can be changed accordingly

cmz=1

cdz=1

cun=1

# provide different scenario’s – change if needed

scens=c("A","B","C", "D")

for (scen in scens) {

if (scen == "A") {

#scenario A, no confounding only direct effect

# direct effect X->Y

bxy=.35

# confounding = 0

SA=matrix(c(

1,.0,

.0,1),2,2)

SC=matrix(c(

1,.0,

.0,1),2,2)

SE=matrix(c(

1,.0,

.0,1),2,2)

}# end A

if (scen=="B") {

# direct effect X->Y

bxy=.0

# Scenario B, confounding by A, no direct efect

SA=matrix(c(

1,.5,

.5,1),2,2)

SC=matrix(c(

1,.5,

.5,1),2,2)

SE=matrix(c(

1,.0,

.0,1),2,2)

} # end B

if (scen=="C") {

# direct effect X->Y

bxy= 0

# Scenario C confounding by C

SA=matrix(c(

1,o,

o,1),2,2)

SC=matrix(c(

1,.5,

.55,1),2,2)

SE=matrix(c(

1,.0,

.0,1),2,2)

} # end C

if (scen=="D") {

# direct effect X->Y

bxy=.35

# Scenario D direct effect and confounding by A and C

SA=matrix(c(

1,0.5,

0.5,1),2,2)

SC=matrix(c(

1,.5,

.5,1),2,2)

SE=matrix(c(

1,.0,

.0,1),2,2)

} # end D

# variance components. Can be changed according to preferred specs. Chosen variance components for life events (X) and the effect on depression (Y).

h2=c(0.45,0.40)

c2=c(0.22,0.10)

e2=(1-h2-c2)

h=diag(sqrt(h2))

e=diag(sqrt(e2))

c=diag(sqrt(c2))

SA=h%\*%SA%\*%h

SC=c%\*%SC%\*%c

SE=e%\*%SE%\*%e

#

Smz=matrix(0,4,4)

Sdz=matrix(0,4,4)

Sph=SA+SC+SE

Smz[1:2,1:2]=Smz[3:4,3:4]=Sdz[1:2,1:2]=Sdz[3:4,3:4]=Sph

Smz[1:2,3:4]=Smz[3:4,1:2]=SA+SC

Sdz[1:2,3:4]=Sdz[3:4,1:2]=.5\*SA+SC

# generate matrixes

irep=1

cdatmz=matrix(0,Nmz,13)

cdatdz=matrix(0,Ndz,13)

cdatmz[,12]=mzcode0 # zyg mz

cdatdz[,12]=dzcode1 # zyg dz

cdatun=matrix(0,Nun,5) # unrelated matrix

# exact data sim MZ, DZ, and unrelated sample

cdatmz[,1:4]=mvrnorm(Nmz,rep(0,4),Sigma=Smz/cmz,emp=T)

cdatdz[,1:4]=mvrnorm(Ndz,rep(0,4),Sigma=Sdz/cdz, emp=T)

cdatun[,1:2]=mvrnorm(Nun,rep(0,2),Sigma=Sdz[1:2,1:2]/cun, emp=T)

# y varx = 1 residual

sde=1 # sde = sqrt(1-bxy^2) # if sde=1, the probit and the linear reg parameters will both equal bxy

cdatmz[,2]=bxy\*cdatmz[,1]+sde\*cdatmz[,2]

cdatmz[,4]=bxy\*cdatmz[,3]+sde\*cdatmz[,4]

cdatdz[,2]=bxy\*cdatdz[,1]+sde\*cdatdz[,2]

cdatdz[,4]=bxy\*cdatdz[,3]+sde\*cdatdz[,4]

cdatun[,2]=bxy\*cdatun[,1]+sde\*cdatun[,2]

# difference scores

cdatmz[,5]=cdatmz[,1]-cdatmz[,3] # difference score x mz

cdatmz[,6]=cdatmz[,2]-cdatmz[,4] # difference score y mz

cdatdz[,5]=cdatdz[,1]-cdatdz[,3] # difference score x dz

cdatdz[,6]=cdatdz[,2]-cdatdz[,4] # difference score y dz

# Create Y as binary.

py=.5

thry=qnorm(py, lower=F)

cdatmz[cdatmz[,2]>thry,7]=1

cdatmz[cdatmz[,4]>thry,8]=1

cdatdz[cdatdz[,2]>thry,7]=1

cdatdz[cdatdz[,4]>thry,8]=1

cdatun[cdatun[,2]>thry,3]=1

# Create X as binary

px=.5

thrx=qnorm(px, lower=F)

cdatmz[cdatmz[,1]>thrx,10]=1

cdatmz[cdatmz[,3]>thrx,11]=1

cdatdz[cdatdz[,1]>thrx,10]=1

cdatdz[cdatdz[,3]>thrx,11]=1

cdatun[cdatun[,1]>thrx,4]=1

# mean x continuous within pairs

cdatmz[,9]=(cdatmz[,1]+cdatmz[,3])/2

cdatdz[,9]=(cdatdz[,1]+cdatdz[,3])/2

# create family number

cdatmz[,13]=c(1:Nmz)

cdatdz[,13]=c(1:Ndz)+Nmz

# assign column names

colnames(cdatmz)=colnames(cdatdz)=c('x1','y1','x2','y2','difx','dify','dy1','dy2','xm','dx1','dx2','zyg01','famnr')

cdatmz=as.data.frame(cdatmz)

cdatdz=as.data.frame(cdatdz)

# also for unrelated (separate dataset). Note family number is arbitrary here as the created dataset is unrelated

cdatun[,5]=c(1:Nun)

cdatun=as.data.frame(cdatun)

colnames(cdatun)=c('x','y','dy','dx','famnr')

# Check if data is correctly simulated by looking at twin correlations. Are these as expected according to given variance components?

# for continuous data

cor.test(cdatmz$x1, cdatmz$x2, method = "pearson", use = "complete.obs")

cor.test(cdatdz$x1, cdatdz$x2, method = "pearson", use = "complete.obs")

cor.test(cdatmz$y1, cdatmz$y2, method = "pearson", use = "complete.obs")

cor.test(cdatdz$y1, cdatdz$y2, method = "pearson", use = "complete.obs")

# Extra check. also - cross twin cross trait?

cor.test(cdatmz$x1, cdatmz$y2, method = "pearson", use = "complete.obs")

cor.test(cdatmz$x2, cdatmz$y1, method = "pearson", use = "complete.obs")

cor.test(cdatdz$x1, cdatdz$y2, method = "pearson", use = "complete.obs")

cor.test(cdatdz$x2, cdatdz$y1, method = "pearson", use = "complete.obs")

# And check for Binary data as well.

tetrachoric(as.matrix(cdatmz[,c("dx1","dx2")]))

tetrachoric(as.matrix(cdatdz[,c("dx1","dx2")]))

tetrachoric(as.matrix(cdatmz[,c("dy1","dy2")]))

tetrachoric(as.matrix(cdatdz[,c("dy1","dy2")]))

# cross twin cross trait

tetrachoric(as.matrix(cdatmz[,c("dx1","dy2")]))

tetrachoric(as.matrix(cdatmz[,c("dx2","dy1")]))

tetrachoric(as.matrix(cdatdz[,c("dx1","dy2")]))

tetrachoric(as.matrix(cdatdz[,c("dx2","dy1")]))

# Restructure MZ & DZ to long format to conduct the analyses.

cdatall=matrix(0,(Nmz+Ndz)\*2,8)

colnames(cdatall)=c('famnr','ppn','x','y','dx','dy','xm','zyg')

cdatall=as.data.frame(cdatall)

cdatall[,1]=c(cdatmz$famnr,cdatmz$famnr,cdatdz$famnr,cdatdz$famnr ) # note famnr 2x as we have 2MZ and 2DZ twins now in long format

cdatall[,4]=c(cdatmz$y1,cdatmz$y2,cdatdz$y1,cdatdz$y2)

cdatall[,6]=c(cdatmz$dy1,cdatmz$dy2,cdatdz$dy1,cdatdz$dy2)

cdatall[,3]=c(cdatmz$x1, cdatmz$x2, cdatdz$x1, cdatdz$x2)

cdatall[,5]=c(cdatmz$dx1,cdatmz$dx2,cdatdz$dx1,cdatdz$dx2)

cdatall[,7]=c(cdatmz$xm, cdatmz$xm, cdatdz$xm, cdatdz$xm)

cdatall[,8]=c(cdatmz$zyg01, cdatmz$zyg01, cdatdz$zyg01, cdatdz$zyg01)

ix=sort.int(cdatall$famnr,index.return=T)$ix # sort according to famnr

cdatall=cdatall[ix,]

cdatall[,2]=rep(c(1,2),(Nmz+Ndz)) #create num 1 or 2 to identify which twin.

# split data for DZ and MZ

cdatlongmz = cdatall[cdatall$zyg==0, c('famnr','ppn','x','y','dx','dy','xm','zyg')]

cdatlongdz = cdatall[cdatall$zyg==1, c('famnr','ppn','x','y','dx','dy','xm','zyg')]

#create data files for each scenario

assign(paste0("cdatun\_", scen), cdatun)

assign(paste0("cdatlongmz\_", scen),cdatlongmz)

assign(paste0("cdatlongdz\_", scen), cdatlongdz)}

**CTDT analyses – R code**

library(plm)

library(lmtest)

library(survival)

# Scenario A – unrelated sample

# Linear regression

lm(y ~ x, data = cdatun\_A)

lm(y ~ dx, data = cdatun\_A)

# Logistic regression

glm(dy ~ x, data = cdatun\_A, family = binomial)

glm(dy ~ dx, data = cdatun\_A, family = binomial)

# Scenario A – DZ twins

Cdatlongdz\_A <- Cdatlongdz.frame(Cdatlongdz\_A, index = c("famnr"))

plm(y ~ x, data = Cdatlongdz\_A, model = "within")

plm(y ~ dx, data = Cdatlongdz\_A, model = "within")

clogit(dy ~ x + strata(famnr), data = Cdatlongdz\_A)

clogit(dy ~ dx + strata(famnr), data = Cdatlongdz\_A)

# Scenario A – MZ twins

CdatlongMZ\_A <- CdatlongMZ.frame(Cdatlongmz\_A, index = c("famnr"))

#Fixed effects analyses

plm(y ~ x, data = CdatlongMZ\_A, model = "within")

plm(y ~ dx, data = CdatlongMZ\_A, model = "within")

# Conditional Logistic regression

clogit(dy ~ x + strata(famnr), data = Cdatlongmz\_A)

clogit(dy ~ dx + strata(famnr), data = Cdatlongmz\_A)

# Scenario B - Unrelated

# Linear regression

lm(y ~ x, data = cdatun\_B)

lm(y ~ dx, data = cdatun\_B)

# Logistic regression

glm(dy ~ x, data = cdatun\_B, family = binomial)

glm(dy ~ dx, data = cdatun\_B, family = binomial)

# Scenario B - DZ twins

Cdatlongdz\_B <- Cdatlongdz.frame(Cdatlongdz\_B, index = c("famnr"))

# Fixed effects

plm(y ~ x, data = Cdatlongdz\_B, model = "within")

plm(y ~ dx, data = Cdatlongdz\_B, model = "within")

# CLR

clogit(dy ~ x + strata(famnr), data = Cdatlongdz\_B)

clogit(dy ~ dx + strata(famnr), data = Cdatlongdz\_B)

# Scenario B - MZ twins

CdatlongMZ\_B <- CdatlongMZ.frame(Cdatlongmz\_B, index = c("famnr"))

#Fixed effects

plm(y ~ x, data = CdatlongMZ\_B, model = "within")

plm(y ~ dx, data = CdatlongMZ\_B, model = "within")

#CLR

clogit(dy ~ x + strata(famnr), data = Cdatlongmz\_B)

clogit(dy ~ dx + strata(famnr), data = Cdatlongmz\_B)

# Scenario C - Unrelated

# Linear regression

lm(y ~ x, data = cdatun\_C)

lm(y ~ dx, data = cdatun\_C)

# Logistic regression

glm(dy ~ x, data = cdatun\_C, family = binomial)

glm(dy ~ dx, data = cdatun\_C, family = binomial)

# Scenario C - DZ twins

Cdatlongdz\_C <- Cdatlongdz.frame(Cdatlongdz\_C, index = c("famnr"))

#Fixed effects

plm(y ~ x, data = Cdatlongdz\_C, model = "within")

plm(y ~ dx, data = Cdatlongdz\_C, model = "within")

# CLR

clogit(dy ~ x + strata(famnr), data = Cdatlongdz\_C)

clogit(dy ~ dx + strata(famnr), data = Cdatlongdz\_C)

# Scenario C - MZ twins

CdatlongMZ\_C <- CdatlongMZ.frame(Cdatlongmz\_C, index = c("famnr"))

#Fixed effects

plm(y ~ x, data = CdatlongMZ\_C, model = "within")

plm(y ~ dx, data = CdatlongMZ\_C, model = "within")

# CLR

clogit(dy ~ x + strata(famnr), data = Cdatlongmz\_C)

clogit(dy ~ dx + strata(famnr), data = Cdatlongmz\_C)

# Scenario D - Unrelated

# Linear regression

lm(y ~ x, data = cdatun\_D)

lm(y ~ dx, data = cdatun\_D)

# Logistic regression

glm(dy ~ x, data = cdatun\_D, family = binomial)

glm(dy ~ dx, data = cdatun\_D, family = binomial)

# Scenario D - DZ twins

Cdatlongdz\_D <- Cdatlongdz.frame(Cdatlongdz\_D, index = c("famnr"))

# Fixed Effects

plm(y ~ x, data = Cdatlongdz\_D, model = "within")

plm(y ~ dx, data = Cdatlongdz\_D, model = "within")

# CLR

clogit(dy ~ x + strata(famnr), data = Cdatlongdz\_D)

clogit(dy ~ dx + strata(famnr), data = Cdatlongdz\_D)

# Scenario D - MZ twins

CdatlongMZ\_D <- CdatlongMZ.frame(Cdatlongmz\_D, index = c("famnr"))

# Fixed effects

plm(y ~ x, data = CdatlongMZ\_D, model = "within")

plm(y ~ dx, data = CdatlongMZ\_D, model = "within")

# CLR

clogit(dy ~ x + strata(famnr), data = Cdatlongmz\_D)

clogit(dy ~ dx + strata(famnr), data = Cdatlongmz\_D)

**Data Analysis CTCD - STATA**

\\ Unrelated Sample Scenario A

import delimited \*choose own directory \* unrelated\_simulated\_data\_scenarioA.csv, clear

\\ linear regression

reg y x

reg y dx

\\ Logistic regression

logit dy x

logit dy dx

\* DZ Twin Sample Scenario A

clear

clear matrix

import delimited \*choose own directory \* simulated\_data\_DZ\_scenarioA.csv, clear

xtset famnr  
\\ fixed effects regression

xtreg y x, fe

xtreg y dx, fe

\\ CLR

clogit dy x, group(famnr)

clogit dy dx, group(famnr)

\* MZ Twin Sample Scenario A

clear

clear matrix

import delimited \*choose own directory \* simulated\_data\_MZ\_scenarioA.csv, clear

xtset famnr

\\ fixed effects regression

xtreg y x, fe

xtreg y dx, fe

\\ CLR

clogit dy x, group(famnr)

clogit dy dx, group(famnr)

\* Unrelated Sample Scenario B

clear

clear matrix

import delimited \*choose own directory \* unrelated\_simulated\_data\_scenarioB.csv, clear

\\ linear regression

reg y x

reg y dx

\\ logistic regression

logit dy x

logit dy dx

\* DZ Twin Sample Scenario B

clear

clear matrix

import delimited \*choose own directory \* simulated\_data\_DZ\_scenarioB.csv, clear

xtset famnr

\\ fixed effects regression

xtreg y x, fe

xtreg y dx, fe

\\ CLR

clogit dy x, group(famnr)

clogit dy dx, group(famnr)

\* MZ Twin Sample Scenario B

clear

clear matrix

import delimited \*choose own directory \* simulated\_data\_MZ\_scenarioB.csv, clear

xtset famnr

\\ Fixed effects regression

xtreg y x, fe

xtreg y dx, fe

\\ CLR

clogit dy x, group(famnr)

clogit dy dx, group(famnr)

\* Unrelated Sample Scenario C

clear

clear matrix

import delimited \*choose own directory \* unrelated\_simulated\_data\_scenarioC.csv, clear

reg y x

reg y dx

logit dy x

logit dy dx

\* DZ Twin Sample Scenario C

clear

clear matrix

import delimited \*choose own directory \* simulated\_data\_DZ\_scenarioC.csv, clear

xtset famnr

xtreg y x, fe

xtreg y dx, fe

clogit dy x, group(famnr)

clogit dy dx, group(famnr)

\* MZ Twin Sample Scenario C

clear

clear matrix

import delimited \*choose own directory \* simulated\_data\_MZ\_scenarioC.csv, clear

xtset famnr

xtreg y x, fe

xtreg y dx, fe

clogit dy x, group(famnr)

clogit dy dx, group(famnr)

\* Unrelated Sample Scenario D

clear

clear matrix

import delimited \*choose own directory \* unrelated\_simulated\_data\_scenarioD.csv, clear

reg y x

reg y dx

logit dy x

logit dy dx

\* DZ Twin Sample Scenario D

clear

clear matrix

import delimited \*choose own directory \* simulated\_data\_DZ\_scenarioD.csv, clear

xtset famnr

xtreg y x, fe

xtreg y dx, fe

clogit dy x, group(famnr)

clogit dy dx, group(famnr)

\* MZ Twin Sample Scenario D

clear

clear matrix

import delimited \*choose own directory \* simulated\_data\_MZ\_scenarioD.csv, clear

xtset famnr

xtreg y x, fe

xtreg y dx, fe

clogit dy x, group(famnr)

clogit dy dx, group(famnr)

**Data Analysis CTCD - SPSS**

\* Encoding: UTF-8.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Unrelated analyses \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*read in file: unrelated simulated data select scenario A, B, C or D.

PRESERVE.

SET DECIMAL DOT.

GET DATA /TYPE=TXT

/FILE=

"\*choose own directory \*"

/ENCODING='UTF8'

/DELIMITERS=","

/QUALIFIER='"'

/ARRANGEMENT=DELIMITED

/FIRSTCASE=2

/DATATYPEMIN PERCENTAGE=95.0

/VARIABLES=

x AUTO

y AUTO

dy AUTO

dx AUTO

famnr AUTO

/MAP.

RESTORE.

CACHE.

EXECUTE.

\* linear regression y~x.

REGRESSION

/MISSING LISTWISE

/STATISTICS COEFF OUTS CI(95) R ANOVA

/CRITERIA=PIN(.05) POUT(.10)

/NOORIGIN

/DEPENDENT y

/METHOD=ENTER x.

\* linear regression y~dx.

REGRESSION

/MISSING LISTWISE

/STATISTICS COEFF OUTS CI(95) R ANOVA

/CRITERIA=PIN(.05) POUT(.10)

/NOORIGIN

/DEPENDENT y

/METHOD=ENTER dx.

\* Logistic regression dy~x.

LOGISTIC REGRESSION VARIABLES dy

/METHOD=ENTER x

/PRINT=CI(95)

/CRITERIA=PIN(0.05) POUT(0.10) ITERATE(20) CUT(0.5).

\* Logistic regression dy~dx.

LOGISTIC REGRESSION VARIABLES dy

/METHOD=ENTER dx

/PRINT=CI(95)

/CRITERIA=PIN(0.05) POUT(0.10) ITERATE(20) CUT(0.5).

\*\*\*\* Repeat above for other scenario’s \*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DZ Twin analyses \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* read in file DZ twins scenario A, B, C or D.

PRESERVE.

SET DECIMAL DOT.

GET DATA /TYPE=TXT

/FILE= "\*Choose file from own directory\*"

/ENCODING='UTF8'

/DELIMITERS=","

/QUALIFIER='"'

/ARRANGEMENT=DELIMITED

/FIRSTCASE=2

/DATATYPEMIN PERCENTAGE=95.0

/VARIABLES=

famnr AUTO

ppn AUTO

x AUTO

y AUTO

dx AUTO

dy AUTO

xm AUTO

zyg AUTO

/MAP.

RESTORE.

CACHE.

EXECUTE.

\* Fixed effects regression, y ~x.

MIXED y WITH x

/CRITERIA=DFMETHOD(SATTERTHWAITE) CIN(95) MXITER(100) MXSTEP(10) SCORING(1)

SINGULAR(0.000000000001) HCONVERGE(0.00000001, RELATIVE) LCONVERGE(0, ABSOLUTE) PCONVERGE(0,

ABSOLUTE)

/FIXED=x | SSTYPE(3)

/METHOD=REML

/PRINT=SOLUTION.

\* Fixed effects y~dx.

\* note: SPSS DOES NOT select the discordant twins pairs prior to analysis. This needs to be done manually:

\* Identify Duplicate Cases.

SORT CASES BY famnr(A) dx(A).

MATCH FILES

/FILE=\*

/BY famnr dx

/FIRST=PrimaryFirst

/LAST=PrimaryLast.

DO IF (PrimaryFirst).

COMPUTE MatchSequence=1-PrimaryLast.

ELSE.

COMPUTE MatchSequence=MatchSequence+1.

END IF.

LEAVE MatchSequence.

FORMATS MatchSequence (f7).

MATCH FILES

/FILE=\*

/DROP=PrimaryFirst PrimaryLast.

VARIABLE LABELS MatchSequence 'Sequential count of matching cases'.

VARIABLE LEVEL MatchSequence (SCALE).

FREQUENCIES VARIABLES=MatchSequence.

EXECUTE.

\*select only dx discordant twins.

Select if MatchSequence =0.

\* fixed effects analyses y ~dx.

MIXED y WITH dx

/CRITERIA=DFMETHOD(SATTERTHWAITE) CIN(95) MXITER(100) MXSTEP(10) SCORING(1)

SINGULAR(0.000000000001) HCONVERGE(0.00000001, RELATIVE) LCONVERGE(0, ABSOLUTE) PCONVERGE(0,

ABSOLUTE)

/FIXED=dx | SSTYPE(3)

/METHOD=REML

/PRINT=SOLUTION.

\* Read original file in file in again.

PRESERVE.

SET DECIMAL DOT.

GET DATA /TYPE=TXT

/FILE= "\*choose from own directory\* "

/ENCODING='UTF8'

/DELIMITERS=","

/QUALIFIER='"'

/ARRANGEMENT=DELIMITED

/FIRSTCASE=2

/DATATYPEMIN PERCENTAGE=95.0

/VARIABLES=

famnr AUTO

ppn AUTO

x AUTO

y AUTO

dx AUTO

dy AUTO

xm AUTO

zyg AUTO

/MAP.

RESTORE.

CACHE.

EXECUTE.

\* CLR dy ~x. For full documentation provided by IBS see: https://www.ibm.com/support/pages/conditional-logistic-regression-using-coxreg

\* SPSS does not automatically select the dy discordant twins - so select twins first.

\* Identify Duplicate Cases.

SORT CASES BY famnr(A) dy(A).

MATCH FILES

/FILE=\*

/BY famnr dy

/FIRST=PrimaryFirst

/LAST=PrimaryLast.

DO IF (PrimaryFirst).

COMPUTE MatchSequence=1-PrimaryLast.

ELSE.

COMPUTE MatchSequence=MatchSequence+1.

END IF.

LEAVE MatchSequence.

FORMATS MatchSequence (f7).

MATCH FILES

/FILE=\*

/DROP=PrimaryFirst PrimaryLast.

VARIABLE LABELS MatchSequence 'Sequential count of matching cases'.

VARIABLE LEVEL MatchSequence (SCALE).

FREQUENCIES VARIABLES=MatchSequence.

EXECUTE.

\*select discordant dy twins.

Select if MatchSequence = 0.

EXECUTE.

\*recode the dependent variable so that it has a value of 1 for the cases and 2 for the controls.

RECODE dy (1=1) (0=2).

EXECUTE.

\* Create a copy of this variable with another name. We'll call this variable STATUS. (Technically, all that's needed here is for all cases to share some property not shared by the controls.).

RECODE dy (ELSE=Copy) INTO copy\_dy.

EXECUTE.

\* CLR dy ~x.

COXREG dy

/STATUS=copy\_dy(1)

/STRATA=famnr

/METHOD=ENTER x

/PRINT=CI(95) CORR

/CRITERIA=PIN(.05) POUT(.10) ITERATE(20).

\* CLR dy~dx

\* Note: SPSS does handle the discordant dx (exposure) variable correctly, so here we do not need to select those manually, only for the outcome variable dy.

COXREG dy

/STATUS=copy\_dy(1)

/STRATA=famnr

/CONTRAST (dx)=Indicator(1)

/METHOD=ENTER dx

/PRINT=CI(95) CORR

/CRITERIA=PIN(.05) POUT(.10) ITERATE(20).

\*\*\*\*\* Repeat above for the other scenario’s \*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* MZ Twin analyses \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* read in file MZ twins scenario A, B, C or D.

PRESERVE.

SET DECIMAL DOT.

GET DATA /TYPE=TXT

/FILE= "\*Choose file from own directory\*"

/ENCODING='UTF8'

/DELIMITERS=","

/QUALIFIER='"'

/ARRANGEMENT=DELIMITED

/FIRSTCASE=2

/DATATYPEMIN PERCENTAGE=95.0

/VARIABLES=

famnr AUTO

ppn AUTO

x AUTO

y AUTO

dx AUTO

dy AUTO

xm AUTO

zyg AUTO

/MAP.

RESTORE.

CACHE.

EXECUTE.

\* Fixed effects regression, y ~x.

MIXED y WITH x

/CRITERIA=DFMETHOD(SATTERTHWAITE) CIN(95) MXITER(100) MXSTEP(10) SCORING(1)

SINGULAR(0.000000000001) HCONVERGE(0.00000001, RELATIVE) LCONVERGE(0, ABSOLUTE) PCONVERGE(0,

ABSOLUTE)

/FIXED=x | SSTYPE(3)

/METHOD=REML

/PRINT=SOLUTION.

\* Fixed effects y~dx.

\* note: SPSS DOES NOT select the discordant twins pairs prior to analysis. This needs to be done manually:

\* Identify Duplicate Cases.

SORT CASES BY famnr(A) dx(A).

MATCH FILES

/FILE=\*

/BY famnr dx

/FIRST=PrimaryFirst

/LAST=PrimaryLast.

DO IF (PrimaryFirst).

COMPUTE MatchSequence=1-PrimaryLast.

ELSE.

COMPUTE MatchSequence=MatchSequence+1.

END IF.

LEAVE MatchSequence.

FORMATS MatchSequence (f7).

MATCH FILES

/FILE=\*

/DROP=PrimaryFirst PrimaryLast.

VARIABLE LABELS MatchSequence 'Sequential count of matching cases'.

VARIABLE LEVEL MatchSequence (SCALE).

FREQUENCIES VARIABLES=MatchSequence.

EXECUTE.

\*select only dx discordant twins.

Select if MatchSequence =0.

\* fixed effects analyses y ~dx.

MIXED y WITH dx

/CRITERIA=DFMETHOD(SATTERTHWAITE) CIN(95) MXITER(100) MXSTEP(10) SCORING(1)

SINGULAR(0.000000000001) HCONVERGE(0.00000001, RELATIVE) LCONVERGE(0, ABSOLUTE) PCONVERGE(0,

ABSOLUTE)

/FIXED=dx | SSTYPE(3)

/METHOD=REML

/PRINT=SOLUTION.

\* Read original file in file in again.

PRESERVE.

SET DECIMAL DOT.

GET DATA /TYPE=TXT

/FILE= "\*choose from own directory\* "

/ENCODING='UTF8'

/DELIMITERS=","

/QUALIFIER='"'

/ARRANGEMENT=DELIMITED

/FIRSTCASE=2

/DATATYPEMIN PERCENTAGE=95.0

/VARIABLES=

famnr AUTO

ppn AUTO

x AUTO

y AUTO

dx AUTO

dy AUTO

xm AUTO

zyg AUTO

/MAP.

RESTORE.

CACHE.

EXECUTE.

\* CLR dy ~x. For full documentation provided by IBS see: https://www.ibm.com/support/pages/conditional-logistic-regression-using-coxreg

\* SPSS does not automatically select the dy discordant twins - so select twins first.

\* Identify Duplicate Cases.

SORT CASES BY famnr(A) dy(A).

MATCH FILES

/FILE=\*

/BY famnr dy

/FIRST=PrimaryFirst

/LAST=PrimaryLast.

DO IF (PrimaryFirst).

COMPUTE MatchSequence=1-PrimaryLast.

ELSE.

COMPUTE MatchSequence=MatchSequence+1.

END IF.

LEAVE MatchSequence.

FORMATS MatchSequence (f7).

MATCH FILES

/FILE=\*

/DROP=PrimaryFirst PrimaryLast.

VARIABLE LABELS MatchSequence 'Sequential count of matching cases'.

VARIABLE LEVEL MatchSequence (SCALE).

FREQUENCIES VARIABLES=MatchSequence.

EXECUTE.

\*select discordant dy twins.

Select if MatchSequence = 0.

EXECUTE.

\*recode the dependent variable so that it has a value of 1 for the cases and 2 for the controls.

RECODE dy (1=1) (0=2).

EXECUTE.

\* Create a copy of this variable with another name. We'll call this variable STATUS. (Technically, all that's needed here is for all cases to share some property not shared by the controls.).

RECODE dy (ELSE=Copy) INTO copy\_dy.

EXECUTE.

\* CLR dy ~x.

COXREG dy

/STATUS=copy\_dy(1)

/STRATA=famnr

/METHOD=ENTER x

/PRINT=CI(95) CORR

/CRITERIA=PIN(.05) POUT(.10) ITERATE(20).

\* CLR dy~dx

\* Note: SPSS does handle the discordant dx (exposure) variable correctly, so here we do not need to select those manually, only for the outcome variable dy.

COXREG dy

/STATUS=copy\_dy(1)

/STRATA=famnr

/CONTRAST (dx)=Indicator(1)

/METHOD=ENTER dx

/PRINT=CI(95) CORR

/CRITERIA=PIN(.05) POUT(.10) ITERATE(20).

\*\*\*\*\* Repeat above for the other scenario’s \*\*\*\*\*\*