**Sample Size Calculation and Optimal Design for Multivariate Regression-Based Norming**

**Online Supplement B**

**Table of Contents**

[R Code to Find Robust Designs (online supplement A, section 2.2) 2](#_Toc107226651)

[R Code to Compute the Required Sample Size With Equations (17) and (18) (Main Text) 14](#_Toc107226652)

# R Code to Find Robust Designs (online supplement A, section 2.2)

### Robustness of the optimal design: Maximin Design   
N<-16\*6\*5\*13  
## Generate the design matrix for each design under each model

# Optimal designs in Table 2  
# true model = model 1 = E(Y)= b0+b1X1+b2X2  
p<-3 # number of parameters in the model  
# d1= (1,0), (1,1), (-1,0), (-1,1), with equal weight=1/4  
X\_d1\_M1<-matrix(cbind(rep(1,times=N),rep(c(1,-1),each=2),rep(c(0,1),times=2)),N,3)  
# d2= (1,0), (0,0), (-1,0), (1,1), (0,1), (-1,1) with equal weight=1/6  
X\_d2\_M1<-matrix(cbind(rep(1,times=N),rep(c(1,0,-1),each=2),rep(c(0,1),times=2)),N,3)  
# d3= (-1,0), (1,0), (-1,1), (1,1) with equal weight=3/16 and (0,1), (0,0) with equal weight=2/16  
X\_d3\_M1<-matrix(cbind(rep(1,times=N),c(rep(c(1,-1), times=36),rep(0,times=24)),c(rep(c(0,1),each=36),rep(c(0,1),each=12))),N,3)  
  
l=0;g=0;d=0;e=0  
for(i in 1:N){  
 ifelse(all(X\_d1\_M1[i,]==rbind(c(1,1,0))),l<-l+1,l<-l)  
 ifelse(all(X\_d2\_M1[i,]==rbind(c(1,1,0))),g<-g+1,g<-g)  
 ifelse(all(X\_d3\_M1[i,]==rbind(c(1,1,0))),d<-d+1,d<-d)  
 ifelse(all(X\_d3\_M1[i,]==rbind(c(1,0,0))),e<-e+1,e<-e)}  
l==N/4 # N times weigth

g==N/6

d==N\*(3/16)

e==N\*(2/16)

# true model = model 2 = E(Y)= b0+b1X1+b2X2+b3x1x1  
p<-4 # number of parameters in the model  
# d1= (1,0), (1,1), (-1,0), (-1,1), with equal weight=1/4  
X\_d1\_M2<-matrix(cbind(rep(1,times=N),rep(c(1,-1),each=2),rep(c(0,1),times=2),rep(c(1,-1),each=2)^2),N,p)  
# d2= (1,0), (0,0), (-1,0), (1,1), (0,1), (-1,1) with equal weight=1/6  
X\_d2\_M2<-matrix(cbind(rep(1,times=N),rep(c(1,0,-1),each=2),rep(c(0,1),times=2),X\_d2\_M1[,2]^2),N,p)  
# d3= (-1,0), (1,0), (-1,1), (1,1) with equal weight=3/16 and (0,1), (0,0) with equal weight=2/16  
X\_d3\_M2<-matrix(cbind(rep(1,times=N),c(rep(c(1,-1), times=36),rep(0,times=24)),c(rep(c(0,1),each=36),rep(c(0,1),each=12)),  
 c(rep(c(1,-1), times=36),rep(0,times=24))^2),N,p)  
  
l=0;g=0;d=0;e=0  
for(i in 1:N){  
 ifelse(all(X\_d2\_M2[i,]==rbind(c(1,1,0,1))),l<-l+1,l<-l)  
 ifelse(all(X\_d2\_M2[i,]==rbind(c(1,1,1,1))),g<-g+1,g<-g)  
 ifelse(all(X\_d3\_M2[i,]==rbind(c(1,1,0,1))),d<-d+1,d<-d)  
 ifelse(all(X\_d3\_M2[i,]==rbind(c(1,0,0,0))),e<-e+1,e<-e)}  
  
l==N/6 # N times weigth

g==N/6

d==N\*(3/16)

e==N\*(2/16)

# true model = model 3 = E(Y)= b0+b1X1+b2X2+b4x1x2  
p<-4 # number of parameters in the model  
# d1= (1,0), (1,1), (-1,0), (-1,1), with equal weight=1/4  
X\_d1\_M3<-matrix(cbind(rep(1,times=N),rep(c(1,-1),each=2),rep(c(0,1),times=2),rep(c(1,-1),each=2)\*rep(c(0,1),times=2)),N,p)  
# d2= (1,0), (0,0), (-1,0), (1,1), (0,1), (-1,1) with equal weight=1/6  
X\_d2\_M3<-matrix(cbind(rep(1,times=N),rep(c(1,0,-1),each=2),rep(c(0,1),times=2),X\_d2\_M1[,2]\*X\_d2\_M1[,3]),N,p)  
  
# d3= (-1,0), (1,0), (-1,1), (1,1) with equal weight=3/16 and (0,1), (0,0) with equal weight=2/16  
X\_d3\_M3<-matrix(cbind(rep(1,times=N),c(rep(c(1,-1), times=36),rep(0,times=24)),c(rep(c(0,1),each=36),rep(c(0,1),each=12)),  
 c(rep(c(1,-1), times=36),rep(0,times=24))\*c(rep(c(0,1),each=36),rep(c(0,1),each=12))),N,p)  
  
l=0;g=0;d=0;e=0  
for(i in 1:N){  
 ifelse(all(X\_d1\_M3[i,1:3]==rbind(c(1,1,0))),l<-l+1,l<-l)  
 ifelse(all(X\_d2\_M3[i,1:3]==rbind(c(1,1,1))),g<-g+1,g<-g)  
 ifelse(all(X\_d3\_M3[i,1:3]==rbind(c(1,1,0))),d<-d+1,d<-d)  
 ifelse(all(X\_d3\_M3[i,1:3]==rbind(c(1,0,0))),e<-e+1,e<-e)}  
  
l==N/4 # N times weigth

g==N/6

d==N\*(3/16)

e==N\*(2/16)

# true model = model 4 = E(Y)= b0+b1X1+b2X2+b3x1x1+b4x1x2  
p<-5 # number of parameters in the model  
# d1= (1,0), (1,1), (-1,0), (-1,1), with equal weight=1/4  
X\_d1\_M4<-matrix(cbind(rep(1,times=N),rep(c(1,-1),each=2),rep(c(0,1),times=2),rep(c(1,-1),each=2)^2,X\_d1\_M2[,2]\*X\_d1\_M2[,3]),N,p)  
# d2= (1,0), (0,0), (-1,0), (1,1), (0,1), (-1,1) with equal weight=1/6  
X\_d2\_M4<-matrix(cbind(rep(1,times=N),rep(c(1,0,-1),each=2),rep(c(0,1),times=2),X\_d2\_M1[,2]^2,X\_d2\_M2[,2]\*X\_d2\_M2[,3]),N,p)  
  
# d3= (-1,0), (1,0), (-1,1), (1,1) with equal weight=3/16 and (0,1), (0,0) with equal weight=2/16  
X\_d3\_M4<-matrix(cbind(rep(1,times=N),c(rep(c(1,-1), times=36),rep(0,times=24)),c(rep(c(0,1),each=36),rep(c(0,1),each=12)),  
 c(rep(c(1,-1), times=36),rep(0,times=24))^2,X\_d3\_M2[,2]\*X\_d3\_M2[,3]),N,p)  
  
l=0;g=0;d=0;e=0  
for(i in 1:N){  
 ifelse(all(X\_d2\_M4[i,1:3]==rbind(c(1,1,1))),g<-g+1,g<-g)  
 ifelse(all(X\_d3\_M4[i,1:3]==rbind(c(1,1,0))),d<-d+1,d<-d)  
 ifelse(all(X\_d3\_M4[i,1:3]==rbind(c(1,0,0))),e<-e+1,e<-e)}  
  
# N times weigth  
g==N/6

d==N\*(3/16)

e==N\*(2/16)

# true model = model 5 = E(Y)= b0+b1X1+b2X2+b3x1x1+b4x1x2+b5x1x1x2  
p<-6 # number of parameters in the model  
# d1= (1,0), (1,1), (-1,0), (-1,1), with equal weight=1/4  
X\_d1\_M5<-matrix(cbind(rep(1,times=N),rep(c(1,-1),each=2),rep(c(0,1),times=2),rep(c(1,-1),each=2)^2,X\_d1\_M2[,2]\*X\_d1\_M2[,3],X\_d1\_M2[,4]\*X\_d1\_M2[,3]),N,p)  
# d2= (1,0), (0,0), (-1,0), (1,1), (0,1), (-1,1) with equal weight=1/6  
X\_d2\_M5<-matrix(cbind(rep(1,times=N),rep(c(1,0,-1),each=2),rep(c(0,1),times=2),X\_d2\_M1[,2]^2,X\_d2\_M2[,2]\*X\_d2\_M2[,3],X\_d2\_M2[,4]\*X\_d2\_M2[,3]),N,p)  
  
# d3= (-1,0), (1,0), (-1,1), (1,1) with equal weight=3/16 and (0,1), (0,0) with equal weight=2/16  
X\_d3\_M5<-matrix(cbind(rep(1,times=N),c(rep(c(1,-1), times=36),rep(0,times=24)),c(rep(c(0,1),each=36),rep(c(0,1),each=12)),  
 c(rep(c(1,-1), times=36),rep(0,times=24))^2,X\_d3\_M2[,2]\*X\_d3\_M2[,3],X\_d3\_M2[,4]\*X\_d3\_M2[,3]),N,p)  
  
l=0;g=0;d=0;e=0  
for(i in 1:N){  
 ifelse(all(X\_d2\_M5[i,1:3]==rbind(c(1,1,1))),g<-g+1,g<-g)  
 ifelse(all(X\_d3\_M5[i,1:3]==rbind(c(1,1,0))),d<-d+1,d<-d)  
 ifelse(all(X\_d3\_M5[i,1:3]==rbind(c(1,0,0))),e<-e+1,e<-e)}  
  
#l==N/4 # N times weigth  
g==N/6

d==N\*(3/16)

e==N\*(2/16)

# Equidistant age levels designs (with more than 3 age levels)   
AGE<-seq(from=20,to=80,by=1)  
d\_tilde<-min(AGE)+(max(AGE)-min(AGE))/2  
Age\_scaled<-(AGE-d\_tilde)/(max(AGE)-d\_tilde)  
# 13 age levels (equal weight 1/26 per sex's level)  
seq(20,80,by=5)

ThirteenEquiDistpts<-Age\_scaled[c(which(AGE==seq(20,80,by=5)[1]),which(AGE==seq(20,80,by=5)[2]),which(AGE==seq(20,80,by=5)[3]),which(AGE==seq(20,80,by=5)[4]),which(AGE==seq(20,80,by=5)[5]),which(AGE==seq(20,80,by=5)[6]), which(AGE==seq(20,80,by=5)[7]),which(AGE==seq(20,80,by=5)[8]),which(AGE==seq(20,80,by=5)[9]),which(AGE==seq(20,80,by=5)[10]),which(AGE==seq(20,80,by=5)[11]),which(AGE==seq(20,80,by=5)[12]), which(AGE==seq(20,80,by=5)[13]))]  
X\_13equip<-matrix(cbind(rep(1,times=N),rep(ThirteenEquiDistpts,times=(N/13)),rep(c(0,1),each=13)),N,3)  
  
X\_d13\_M1<-X\_13equip # model 1 =true model  
X\_d13\_M2<-cbind(X\_13equip,X\_13equip[,2]^2) # model 2 =true model  
X\_d13\_M3<-cbind(X\_13equip,X\_13equip[,2]\*X\_13equip[,3]) # model 3 =true model  
X\_d13\_M4<-cbind(X\_13equip,X\_13equip[,2]^2,X\_13equip[,2]\*X\_13equip[,3]) # model 4 =true model  
X\_d13\_M5<-cbind(X\_13equip,X\_13equip[,2]^2,X\_13equip[,2]\*X\_13equip[,3],(X\_13equip[,2]^2)\*X\_13equip[,3]) # model 5 =true model  
  
l=0;g=0;d=0;e=0  
for(i in 1:N){  
 ifelse(all(X\_d13\_M5[i,1:3]==rbind(c(1,1,0))),l<-l+1,l<-l)  
 ifelse(all(X\_d13\_M1[i,1:3]==rbind(c(1,1,1))),g<-g+1,g<-g)  
 ifelse(all(X\_d13\_M2[i,1:3]==rbind(c(1,1,0))),d<-d+1,d<-d)  
 ifelse(all(X\_d13\_M3[i,1:3]==rbind(c(1,1,0))),e<-e+1,e<-e)}  
l==N/26 # 1/26 is the weight

g==N/26

d==N/26

e==N/26

## Compute (X'X)^-1 for each design under each model   
# under model (1)  
V\_d1\_M1<-solve(t(X\_d1\_M1)%\*%X\_d1\_M1) # design with 2 age levels  
V\_d2\_M1<-solve(t(X\_d2\_M1)%\*%X\_d2\_M1) # balanced design with 3 age levels  
V\_d3\_M1<-solve(t(X\_d3\_M1)%\*%X\_d3\_M1) # unbalanced design with 3 age levels  
V\_d7\_M1<-solve(t(X\_d13\_M1)%\*%X\_d13\_M1) # design with 13 age levels  
  
# under model (2)  
#V\_d1\_M2<-solve(t(X\_d1\_M2)%\*%X\_d1\_M2) # the effect of Age^2 is not identifiable for a design with only 2 age levels  
V\_d2\_M2<-solve(t(X\_d2\_M2)%\*%X\_d2\_M2)  
V\_d3\_M2<-solve(t(X\_d3\_M2)%\*%X\_d3\_M2)  
V\_d7\_M2<-solve(t(X\_d13\_M2)%\*%X\_d13\_M2)  
  
# under model (3)  
V\_d1\_M3<-solve(t(X\_d1\_M3)%\*%X\_d1\_M3)  
V\_d2\_M3<-solve(t(X\_d2\_M3)%\*%X\_d2\_M3)  
V\_d3\_M3<-solve(t(X\_d3\_M3)%\*%X\_d3\_M3)  
V\_d7\_M3<-solve(t(X\_d13\_M3)%\*%X\_d13\_M3)  
  
# under model (4)  
#V\_d1\_M4<-solve(t(X\_d1\_M4)%\*%X\_d1\_M4) # the effect of Age^2 is not identifiable for a design with only 2 age levels  
V\_d2\_M4<-solve(t(X\_d2\_M4)%\*%X\_d2\_M4)  
V\_d3\_M4<-solve(t(X\_d3\_M4)%\*%X\_d3\_M4)  
V\_d7\_M4<-solve(t(X\_d13\_M4)%\*%X\_d13\_M4)  
  
# under model (5)  
#V\_d1\_M5<-solve(t(X\_d1\_M5)%\*%X\_d1\_M5) # the effect of Age^2 is not identifiable for a design with only 2 age levels  
V\_d2\_M5<-solve(t(X\_d2\_M5)%\*%X\_d2\_M5)  
V\_d3\_M5<-solve(t(X\_d3\_M5)%\*%X\_d3\_M5)  
V\_d7\_M5<-solve(t(X\_d13\_M5)%\*%X\_d13\_M5)  
  
  
## Maximin Design based on the Efficiency criterion   
# Compute the maximum standardized prediction variance over x0  
# generate x0  
AGE<-seq(from=20,to=80,by=1)  
d\_tilde<-min(AGE)+(max(AGE)-min(AGE))/2  
Age\_scaled<-(AGE-d\_tilde)/(max(AGE)-d\_tilde)  
Sex<-c(0,1)  
  
X\_comb<-expand.grid(Age\_scaled,Sex)  
colnames(X\_comb)<-c("Age","Sex")  
  
x0<-data.frame(1,X\_comb$Age,X\_comb$Sex,X\_comb$Age^2,X\_comb$Age\*X\_comb$Sex,(X\_comb$Age^2)\*X\_comb$Sex)  
colnames(x0)<-c("Int", "Age","Sex", "Age2", "AgeSex", "Age2Sex")  
  
# note: d1 = 2 age levels design, d2 = 3 age levels balanced design, d3 = 3 age levels unbalanced design  
# d7 = 13 age levels design  
  
# under model (1)  
D\_d1\_M1<-max(N\*diag(data.matrix(x0[,1:3])%\*%V\_d1\_M1%\*%t(data.matrix(x0[,1:3]))))  
D\_d2\_M1<-max(N\*diag(data.matrix(x0[,1:3])%\*%V\_d2\_M1%\*%t(data.matrix(x0[,1:3]))))  
D\_d3\_M1<-max(N\*diag(data.matrix(x0[,1:3])%\*%V\_d3\_M1%\*%t(data.matrix(x0[,1:3]))))  
D\_d7\_M1<-max(N\*diag(data.matrix(x0[,1:3])%\*%V\_d7\_M1%\*%t(data.matrix(x0[,1:3]))))  
  
# under model (2): recall that the effect of Age^2 is not identifiable for d1 (i.e. design with 2 age levels)  
D\_d2\_M2<-max(N\*diag(data.matrix(x0[,1:4])%\*%V\_d2\_M2%\*%t(data.matrix(x0[,1:4]))))  
D\_d3\_M2<-max(N\*diag(data.matrix(x0[,1:4])%\*%V\_d3\_M2%\*%t(data.matrix(x0[,1:4]))))  
D\_d7\_M2<-max(N\*diag(data.matrix(x0[,1:4])%\*%V\_d7\_M2%\*%t(data.matrix(x0[,1:4]))))  
  
# under model (3)  
D\_d1\_M3<-max(N\*diag(data.matrix(x0[,c(1:3,5)])%\*%V\_d1\_M3%\*%t(data.matrix(x0[,c(1:3,5)]))))  
D\_d2\_M3<-max(N\*diag(data.matrix(x0[,c(1:3,5)])%\*%V\_d2\_M3%\*%t(data.matrix(x0[,c(1:3,5)]))))  
D\_d3\_M3<-max(N\*diag(data.matrix(x0[,c(1:3,5)])%\*%V\_d3\_M3%\*%t(data.matrix(x0[,c(1:3,5)]))))  
D\_d7\_M3<-max(N\*diag(data.matrix(x0[,c(1:3,5)])%\*%V\_d7\_M3%\*%t(data.matrix(x0[,c(1:3,5)]))))  
  
# under model (4): recall that the effect of Age^2 is not identifiable for d1 (i.e. design with 2 age levels)  
D\_d2\_M4<-max(N\*diag(data.matrix(x0[,-6])%\*%V\_d2\_M4%\*%t(data.matrix(x0[,-6]))))  
D\_d3\_M4<-max(N\*diag(data.matrix(x0[,-6])%\*%V\_d3\_M4%\*%t(data.matrix(x0[,-6]))))  
D\_d7\_M4<-max(N\*diag(data.matrix(x0[,-6])%\*%V\_d7\_M4%\*%t(data.matrix(x0[,-6]))))  
  
# under model (5): recall that the effect of Age^2 is not identifiable for d1 (i.e. design with 2 age levels)  
D\_d2\_M5<-max(N\*diag(data.matrix(x0)%\*%V\_d2\_M5%\*%t(data.matrix(x0))))  
D\_d3\_M5<-max(N\*diag(data.matrix(x0)%\*%V\_d3\_M5%\*%t(data.matrix(x0))))  
D\_d7\_M5<-max(N\*diag(data.matrix(x0)%\*%V\_d7\_M5%\*%t(data.matrix(x0))))  
  
library(htmlTable)  
# Table S.A.1, online supplement A  
write.table(  
round(rbind(cbind(D\_d1\_M1,0,D\_d1\_M3,0,0),  
cbind(D\_d2\_M1,D\_d2\_M2,D\_d2\_M3,D\_d2\_M4,D\_d2\_M5),  
cbind(D\_d3\_M1,D\_d3\_M2,D\_d3\_M3,D\_d3\_M4,D\_d3\_M5),  
cbind(D\_d7\_M1,D\_d7\_M2,D\_d7\_M3,D\_d7\_M4,D\_d7\_M5)),2)  
, file = "Maximum\_SPV.txt", sep = ",", quote = FALSE, row.names = F)  
### Maximin Design based on the RE criterion

# generate x0  
AGE<-seq(from=20,to=80,by=1)  
d\_tilde<-min(AGE)+(max(AGE)-min(AGE))/2  
Age\_scaled<-(AGE-d\_tilde)/(max(AGE)-d\_tilde)  
  
Sex<-c(0,1)  
  
X\_comb<-expand.grid(Age\_scaled,Sex)  
colnames(X\_comb)<-c("Age","Sex")  
  
x0<-data.frame(1,X\_comb$Age,X\_comb$Sex,X\_comb$Age^2,X\_comb$Age\*X\_comb$Sex,(X\_comb$Age^2)\*X\_comb$Sex)  
colnames(x0)<-c("Int", "Age","Sex", "Age2", "AgeSex", "Age2Sex")  
  
### Find the minimum RE across x0 values --> results in Table S.A.2, online supplement A  
  
# note: d1 = 2 age levels design, d2 = 3 age levels balanced design, d3 = 3 age levels unbalanced design  
# d7 = 13 age levels design  
  
#### RE based on the trace of V(z\_hat)  
Z1<-seq(from=-3,to=3,by=0.5)  
Z2<-cbind(Z1,c(Z1[1:7]+0.5,Z1[-c(1:7)]-0.5),c(Z1[1:7]+2,Z1[-c(1:7)]-2))  
  
RE\_trace<-function(x0,V\_ODE,V\_NON\_ODE,Z1,Z2,P){ # RE based on the trace   
   
NUM<-(P\*(N\*diag(data.matrix(x0)%\*%V\_ODE%\*%t(data.matrix(x0)))))+(((Z1^2)+(Z2^2))/2)  
DEN<-(P\*(N\*diag(data.matrix(x0)%\*%V\_NON\_ODE%\*%t(data.matrix(x0)))))+(((Z1^2)+(Z2^2))/2)  
  
RE<-sqrt(NUM/DEN)   
   
 return(RE)  
}  
# steps 1-3: to find the RE maximin design, step 4: to check whether it depends on Z1 and Z2  
# step 1): find the minimum RE over x0, for each design under each model, given Z1, Z2   
# step 2): find the smallest minimum RE over x0, for each design across all models, given Z1, Z2   
# step 3): find the design with the highest smallest minimum RE over x0, given Z1, Z2   
# step 4): check that the design in step 3 is the same for all values of given Z1, Z2  
RE\_d1\_mod1<-RE\_d2\_mod1<-RE\_d3\_mod1<-RE\_d7\_mod1<-matrix(0,length(Z1),(dim(Z2)[2]))  
RE\_d2\_mod2<-RE\_d3\_mod2<-RE\_d7\_mod2<-matrix(0,length(Z1),(dim(Z2)[2]))  
RE\_d1\_mod3<-RE\_d2\_mod3<-RE\_d3\_mod3<-RE\_d7\_mod3<-matrix(0,length(Z1),(dim(Z2)[2]))  
RE\_d2\_mod4<-RE\_d3\_mod4<-RE\_d7\_mod5<-matrix(0,length(Z1),(dim(Z2)[2]))  
RE\_d2\_mod5<-RE\_d3\_mod5<-RE\_d7\_mod4<-matrix(0,length(Z1),(dim(Z2)[2]))  
Min\_RE\_d2<-Min\_RE\_d3<-Min\_RE\_d7<-matrix(0,length(Z1),(dim(Z2)[2]))  
Maximin\_RE<-matrix(0,length(Z1),(dim(Z2)[2]))  
  
  
for(j in 1:(dim(Z2)[2])){  
for(i in 1:length(Z1)){  
   
## step 1): find the minimum RE over x0, for each design under each model, given Z1, Z2, and rho   
   
# Model (1)  
RE\_d1\_mod1[i,j]<-min(RE\_trace(x0=x0[,1:3],V\_ODE=V\_d1\_M1,V\_NON\_ODE=V\_d1\_M1,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d2\_mod1[i,j]<-min(RE\_trace(x0=x0[,1:3],V\_ODE=V\_d1\_M1,V\_NON\_ODE=V\_d2\_M1,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d3\_mod1[i,j]<-min(RE\_trace(x0=x0[,1:3],V\_ODE=V\_d1\_M1,V\_NON\_ODE=V\_d3\_M1,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d7\_mod1[i,j]<-min(RE\_trace(x0=x0[,1:3],V\_ODE=V\_d1\_M1,V\_NON\_ODE=V\_d7\_M1,Z1=Z1[i],Z2=Z2[i,j],P=2))  
  
# Model (2)  
RE\_d2\_mod2[i,j]<-min(RE\_trace(x0=x0[,1:4],V\_ODE=V\_d2\_M2,V\_NON\_ODE=V\_d2\_M2,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d3\_mod2[i,j]<-min(RE\_trace(x0=x0[,1:4],V\_ODE=V\_d2\_M2,V\_NON\_ODE=V\_d3\_M2,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d7\_mod2[i,j]<-min(RE\_trace(x0=x0[,1:4],V\_ODE=V\_d2\_M2,V\_NON\_ODE=V\_d7\_M2,Z1=Z1[i],Z2=Z2[i,j],P=2))  
  
# Model (3)  
RE\_d1\_mod3[i,j]<-min(RE\_trace(x0=x0[,c(1:3,5)],V\_ODE=V\_d1\_M3,V\_NON\_ODE=V\_d1\_M3,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d2\_mod3[i,j]<-min(RE\_trace(x0=x0[,c(1:3,5)],V\_ODE=V\_d1\_M3,V\_NON\_ODE=V\_d2\_M3,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d3\_mod3[i,j]<-min(RE\_trace(x0=x0[,c(1:3,5)],V\_ODE=V\_d1\_M3,V\_NON\_ODE=V\_d3\_M3,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d7\_mod3[i,j]<-min(RE\_trace(x0=x0[,c(1:3,5)],V\_ODE=V\_d1\_M3,V\_NON\_ODE=V\_d7\_M3,Z1=Z1[i],Z2=Z2[i,j],P=2))  
  
# Model (4)  
RE\_d2\_mod4[i,j]<-min(RE\_trace(x0=x0[,-6],V\_ODE=V\_d3\_M4,V\_NON\_ODE=V\_d2\_M4,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d3\_mod4[i,j]<-min(RE\_trace(x0=x0[,-6],V\_ODE=V\_d3\_M4,V\_NON\_ODE=V\_d3\_M4,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d7\_mod4[i,j]<-min(RE\_trace(x0=x0[,-6],V\_ODE=V\_d3\_M4,V\_NON\_ODE=V\_d7\_M4,Z1=Z1[i],Z2=Z2[i,j],P=2))  
  
# Model (5)  
RE\_d2\_mod5[i,j]<-min(RE\_trace(x0=x0,V\_ODE=V\_d2\_M5,V\_NON\_ODE=V\_d2\_M5,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d3\_mod5[i,j]<-min(RE\_trace(x0=x0,V\_ODE=V\_d2\_M5,V\_NON\_ODE=V\_d3\_M5,Z1=Z1[i],Z2=Z2[i,j],P=2))  
RE\_d7\_mod5[i,j]<-min(RE\_trace(x0=x0,V\_ODE=V\_d2\_M5,V\_NON\_ODE=V\_d7\_M5,Z1=Z1[i],Z2=Z2[i,j],P=2))  
  
## step 2): find the smallest minimum RE over x0, for each design across all models, given Z1, Z2, and rho  
# for each design, find the minimum RE across models  
Min\_RE\_d2[i,j]<-min(c(RE\_d2\_mod1[i,j],RE\_d2\_mod2[i,j],RE\_d2\_mod3[i,j],RE\_d2\_mod4[i,j],RE\_d2\_mod5[i,j]))  
Min\_RE\_d3[i,j]<-min(c(RE\_d3\_mod1[i,j],RE\_d3\_mod2[i,j],RE\_d3\_mod3[i,j],RE\_d3\_mod4[i,j],RE\_d3\_mod5[i,j]))  
Min\_RE\_d7[i,j]<-min(c(RE\_d7\_mod1[i,j],RE\_d7\_mod2[i,j],RE\_d7\_mod3[i,j],RE\_d7\_mod4[i,j],RE\_d7\_mod5[i,j]))  
  
### step 3): find the design with the highest smallest minimum RE over x0, given Z1, Z2, and rho   
# find the design with the highest minimum RE  
Maximin\_RE[i,j]<-max(c(Min\_RE\_d2[i,j],Min\_RE\_d3[i,j],Min\_RE\_d7[i,j]))  
  
}}  
  
### step 4): check that the design in step 3 is the same for all values of given Z1, Z2, and rho  
all(Maximin\_RE==Min\_RE\_d2) # the optimal design for models (2) and (5) is the RE maximin design, because it yields the highest minimum RE

# furthermore, the RE maximin design does not depend on Z1, Z2  
all(Maximin\_RE>Min\_RE\_d3)

all(Maximin\_RE>Min\_RE\_d7)

# row = Z1, col = Z2  
# Z1=Z2=-2: conjunctive rule  
  
Table\_trace<-matrix(0,4,5)  
  
  
Table\_trace[1,1]<-RE\_d1\_mod1[which(Z1==-2),1]  
Table\_trace[2,1]<-RE\_d2\_mod1[which(Z1==-2),1]  
Table\_trace[3,1]<-RE\_d3\_mod1[which(Z1==-2),1]  
Table\_trace[4,1]<-RE\_d7\_mod1[which(Z1==-2),1]  
  
Table\_trace[2,2]<-RE\_d2\_mod2[which(Z1==-2),1]  
Table\_trace[3,2]<-RE\_d3\_mod2[which(Z1==-2),1]  
Table\_trace[4,2]<-RE\_d7\_mod2[which(Z1==-2),1]  
  
Table\_trace[1,3]<-RE\_d1\_mod3[which(Z1==-2),1]  
Table\_trace[2,3]<-RE\_d2\_mod3[which(Z1==-2),1]  
Table\_trace[3,3]<-RE\_d3\_mod3[which(Z1==-2),1]  
Table\_trace[4,3]<-RE\_d7\_mod3[which(Z1==-2),1]  
  
Table\_trace[2,4]<-RE\_d2\_mod4[which(Z1==-2),1]  
Table\_trace[3,4]<-RE\_d3\_mod4[which(Z1==-2),1]  
Table\_trace[4,4]<-RE\_d7\_mod4[which(Z1==-2),1]  
  
Table\_trace[2,5]<-RE\_d2\_mod5[which(Z1==-2),1]  
Table\_trace[3,5]<-RE\_d3\_mod5[which(Z1==-2),1]  
Table\_trace[4,5]<-RE\_d7\_mod5[which(Z1==-2),1]  
  
# Table S.A.2 lower half, online supplement A  
library(htmlTable)  
write.table(round(Table\_trace,3), file = "RE\_trace\_conjunctive.txt", sep = ",", quote = FALSE, row.names = F)  
  
# Z1=-2 and Z2=0: disjunctive rule  
  
Table\_trace[1,1]<-RE\_d1\_mod1[which(Z1==-2),3]  
Table\_trace[2,1]<-RE\_d2\_mod1[which(Z1==-2),3]  
Table\_trace[3,1]<-RE\_d3\_mod1[which(Z1==-2),3]  
Table\_trace[4,1]<-RE\_d7\_mod1[which(Z1==-2),3]  
  
Table\_trace[2,2]<-RE\_d2\_mod2[which(Z1==-2),3]  
Table\_trace[3,2]<-RE\_d3\_mod2[which(Z1==-2),3]  
Table\_trace[4,2]<-RE\_d7\_mod2[which(Z1==-2),3]  
  
Table\_trace[1,3]<-RE\_d1\_mod3[which(Z1==-2),3]  
Table\_trace[2,3]<-RE\_d2\_mod3[which(Z1==-2),3]  
Table\_trace[3,3]<-RE\_d3\_mod3[which(Z1==-2),3]  
Table\_trace[4,3]<-RE\_d7\_mod3[which(Z1==-2),3]  
  
Table\_trace[2,4]<-RE\_d2\_mod4[which(Z1==-2),3]  
Table\_trace[3,4]<-RE\_d3\_mod4[which(Z1==-2),3]  
Table\_trace[4,4]<-RE\_d7\_mod4[which(Z1==-2),3]  
  
Table\_trace[2,5]<-RE\_d2\_mod5[which(Z1==-2),3]  
Table\_trace[3,5]<-RE\_d3\_mod5[which(Z1==-2),3]  
Table\_trace[4,5]<-RE\_d7\_mod5[which(Z1==-2),3]  
  
# Table S.A.2 lower half, online supplement A  
write.table(round(Table\_trace,3), file = "RE\_trace\_disjunctive.txt", sep = ",", quote = FALSE, row.names = F)  
  
#### RE based on the V(delta0\_hat)  
  
library(chi)  
Mahalanobis<-seq(0.15, 3.05, by=0.05)   
#round(pchi(q=c(0.15,3.05), df=2, ncp = 0, lower.tail = TRUE, log.p = FALSE),3)\*100  
  
RE\_delta0<-function(x0,V\_ODE,V\_NON\_ODE,Mahalanobis){ # RE based on delta\_0   
   
 NUM<-N\*diag(data.matrix(x0)%\*%V\_ODE%\*%t(data.matrix(x0)))+((Mahalanobis^2)/2)  
 DEN<-N\*diag(data.matrix(x0)%\*%V\_NON\_ODE%\*%t(data.matrix(x0)))+((Mahalanobis^2)/2)  
   
 RE<-sqrt(NUM/DEN)   
   
 return(RE)  
}  
# steps 1-3: to find the RE maximin design, step 4: to check whether it depends on delta\_0  
# step 1): find the minimum RE over x0, for each design under each model, given delta\_0   
# step 2): find the smallest minimum RE over x0, for each design across all models, given delta\_0   
# step 3): find the design with the highest smallest minimum RE over x0, given delta\_0   
# step 4): check that the design in step 3 is the same for all values of given delta\_0  
RE\_d1\_mod1<-RE\_d2\_mod1<-RE\_d3\_mod1<-RE\_d7\_mod1<-matrix(0,length(Mahalanobis))  
RE\_d2\_mod2<-RE\_d3\_mod2<-RE\_d7\_mod2<-matrix(0,length(Mahalanobis))  
RE\_d1\_mod3<-RE\_d2\_mod3<-RE\_d3\_mod3<-RE\_d7\_mod3<-matrix(0,length(Mahalanobis))  
RE\_d2\_mod4<-RE\_d3\_mod4<-RE\_d7\_mod5<-matrix(0,length(Mahalanobis))  
RE\_d2\_mod5<-RE\_d3\_mod5<-RE\_d7\_mod4<-matrix(0,length(Mahalanobis))  
Min\_RE\_d2<-Min\_RE\_d3<-Min\_RE\_d7<-matrix(0,length(Mahalanobis))  
Maximin\_RE<-matrix(0,length(Mahalanobis))  
  
  
for(i in 1:length(Mahalanobis)){  
   
### step 1): find the minimum RE over x0, for each design under each model, given delta\_0   
   
# Model (1)  
RE\_d1\_mod1[i]<-min(RE\_delta0(x0=x0[,1:3],V\_ODE=V\_d1\_M1,V\_NON\_ODE=V\_d1\_M1,Mahalanobis=Mahalanobis[i]))  
RE\_d2\_mod1[i]<-min(RE\_delta0(x0=x0[,1:3],V\_ODE=V\_d1\_M1,V\_NON\_ODE=V\_d2\_M1,Mahalanobis=Mahalanobis[i]))  
RE\_d3\_mod1[i]<-min(RE\_delta0(x0=x0[,1:3],V\_ODE=V\_d1\_M1,V\_NON\_ODE=V\_d3\_M1,Mahalanobis=Mahalanobis[i]))  
RE\_d7\_mod1[i]<-min(RE\_delta0(x0=x0[,1:3],V\_ODE=V\_d1\_M1,V\_NON\_ODE=V\_d7\_M1,Mahalanobis=Mahalanobis[i]))  
   
# Model (2)  
RE\_d2\_mod2[i]<-min(RE\_delta0(x0=x0[,1:4],V\_ODE=V\_d2\_M2,V\_NON\_ODE=V\_d2\_M2,Mahalanobis=Mahalanobis[i]))  
RE\_d3\_mod2[i]<-min(RE\_delta0(x0=x0[,1:4],V\_ODE=V\_d2\_M2,V\_NON\_ODE=V\_d3\_M2,Mahalanobis=Mahalanobis[i]))  
RE\_d7\_mod2[i]<-min(RE\_delta0(x0=x0[,1:4],V\_ODE=V\_d2\_M2,V\_NON\_ODE=V\_d7\_M2,Mahalanobis=Mahalanobis[i]))  
   
# Model (3)  
RE\_d1\_mod3[i]<-min(RE\_delta0(x0=x0[,c(1:3,5)],V\_ODE=V\_d1\_M3,V\_NON\_ODE=V\_d1\_M3,Mahalanobis=Mahalanobis[i]))  
RE\_d2\_mod3[i]<-min(RE\_delta0(x0=x0[,c(1:3,5)],V\_ODE=V\_d1\_M3,V\_NON\_ODE=V\_d2\_M3,Mahalanobis=Mahalanobis[i]))  
RE\_d3\_mod3[i]<-min(RE\_delta0(x0=x0[,c(1:3,5)],V\_ODE=V\_d1\_M3,V\_NON\_ODE=V\_d3\_M3,Mahalanobis=Mahalanobis[i]))  
RE\_d7\_mod3[i]<-min(RE\_delta0(x0=x0[,c(1:3,5)],V\_ODE=V\_d1\_M3,V\_NON\_ODE=V\_d7\_M3,Mahalanobis=Mahalanobis[i]))  
   
# Model (4)  
RE\_d2\_mod4[i]<-min(RE\_delta0(x0=x0[,-6],V\_ODE=V\_d3\_M4,V\_NON\_ODE=V\_d2\_M4,Mahalanobis=Mahalanobis[i]))  
RE\_d3\_mod4[i]<-min(RE\_delta0(x0=x0[,-6],V\_ODE=V\_d3\_M4,V\_NON\_ODE=V\_d3\_M4,Mahalanobis=Mahalanobis[i]))  
RE\_d7\_mod4[i]<-min(RE\_delta0(x0=x0[,-6],V\_ODE=V\_d3\_M4,V\_NON\_ODE=V\_d7\_M4,Mahalanobis=Mahalanobis[i]))  
   
# Model (5)  
RE\_d2\_mod5[i]<-min(RE\_delta0(x0=x0,V\_ODE=V\_d2\_M5,V\_NON\_ODE=V\_d2\_M5,Mahalanobis=Mahalanobis[i]))  
RE\_d3\_mod5[i]<-min(RE\_delta0(x0=x0,V\_ODE=V\_d2\_M5,V\_NON\_ODE=V\_d3\_M5,Mahalanobis=Mahalanobis[i]))  
RE\_d7\_mod5[i]<-min(RE\_delta0(x0=x0,V\_ODE=V\_d2\_M5,V\_NON\_ODE=V\_d7\_M5,Mahalanobis=Mahalanobis[i]))  
   
### step 2): find the smallest minimum RE over x0, for each design across all models, given delta\_0  
# for each design, find the minimum RE across models  
Min\_RE\_d2[i]<-min(c(RE\_d2\_mod1[i],RE\_d2\_mod2[i],RE\_d2\_mod3[i],RE\_d2\_mod4[i],RE\_d2\_mod5[i]))  
Min\_RE\_d3[i]<-min(c(RE\_d3\_mod1[i],RE\_d3\_mod2[i],RE\_d3\_mod3[i],RE\_d3\_mod4[i],RE\_d3\_mod5[i]))  
Min\_RE\_d7[i]<-min(c(RE\_d7\_mod1[i],RE\_d7\_mod2[i],RE\_d7\_mod3[i],RE\_d7\_mod4[i],RE\_d7\_mod5[i]))  
   
### step 3): find the design with the highest smallest minimum RE over x0, given delta\_0  
# find the design with the highest minimum RE  
Maximin\_RE[i]<-max(c(Min\_RE\_d2[i],Min\_RE\_d3[i],Min\_RE\_d7[i]))  
   
}  
  
### step 4): check that the design in step 3 is the same for all values of given delta\_0  
all(Maximin\_RE==Min\_RE\_d2) # the optimal design for models (2) and (5) is the RE maximin design, because it yields the highest minimum RE

## [1] TRUE

# furthermore, the RE maximin design does not depend on delta\_0  
all(Maximin\_RE>Min\_RE\_d3)

## [1] TRUE

all(Maximin\_RE>Min\_RE\_d7)

## [1] TRUE

Table\_delta0<-matrix(0,4,5)  
  
Table\_delta0[1,1]<-RE\_d1\_mod1[which(Mahalanobis==2.45)]  
Table\_delta0[2,1]<-RE\_d2\_mod1[which(Mahalanobis==2.45)]  
Table\_delta0[3,1]<-RE\_d3\_mod1[which(Mahalanobis==2.45)]  
Table\_delta0[4,1]<-RE\_d7\_mod1[which(Mahalanobis==2.45)]  
  
Table\_delta0[2,2]<-RE\_d2\_mod2[which(Mahalanobis==2.45)]  
Table\_delta0[3,2]<-RE\_d3\_mod2[which(Mahalanobis==2.45)]  
Table\_delta0[4,2]<-RE\_d7\_mod2[which(Mahalanobis==2.45)]  
  
Table\_delta0[1,3]<-RE\_d1\_mod3[which(Mahalanobis==2.45)]  
Table\_delta0[2,3]<-RE\_d2\_mod3[which(Mahalanobis==2.45)]  
Table\_delta0[3,3]<-RE\_d3\_mod3[which(Mahalanobis==2.45)]  
Table\_delta0[4,3]<-RE\_d7\_mod3[which(Mahalanobis==2.45)]  
  
Table\_delta0[2,4]<-RE\_d2\_mod4[which(Mahalanobis==2.45)]  
Table\_delta0[3,4]<-RE\_d3\_mod4[which(Mahalanobis==2.45)]  
Table\_delta0[4,4]<-RE\_d7\_mod4[which(Mahalanobis==2.45)]  
  
Table\_delta0[2,5]<-RE\_d2\_mod5[which(Mahalanobis==2.45)]  
Table\_delta0[3,5]<-RE\_d3\_mod5[which(Mahalanobis==2.45)]  
Table\_delta0[4,5]<-RE\_d7\_mod5[which(Mahalanobis==2.45)]  
  
# Table S.A.2 upper half, online supplement A  
library(htmlTable)  
write.table(round(Table\_delta0,3), file = "RE\_delta.txt", sep = ",", quote = FALSE, row.names = F)

# R Code to Compute the Required Sample Size With Equations (17) and (18) (Main Text)

***# R code to compute the Mahalanobis distance as a function of Z1, Z2, and rho,***   
***## and to compute the required sample size for the Mahalanobis distance***

***############# FUNCTIONS:***   
  
***### To compute percentiles of the chi distribution***   
*#install.packages("chi") # to install the package "chi", see https://cran.r-project.org/web/packages/chi/chi.pdf*  
library(chi)  
P<-2 *# number of outcomes = degrees of freedom of the chi distribution*  
PR<-c(0.90,0.95,0.99) *# percentile rank score*  
*#Example: compute the 90th, 95th, and 99th percentile of the chi distribution with 2 degrees of freedom*  
qchi(p=PR, df=P, ncp = 0, lower.tail = TRUE, log.p = FALSE)

***### Function to compute the Mahalanobis distance from the Z-scores and the correlation between them***  
*# Z1 = Z-score for test score 1*  
*# Z2 = Z-score for test score 2*  
*# rho = correlation between Z1 and Z2*  
  
Mahala\_dist\_rho<-**function**(Z1,Z2,rho){  
 delta\_0<-sqrt(((Z1^2)+(Z2^2)-2\*rho\*Z1\*Z2)/(1-rho^2))  
 return(delta\_0)  
}  
  
*# Examples*  
round(Mahala\_dist\_rho(Z1=-2,Z2=-2,rho=0),2) *# rho=0*

round(Mahala\_dist\_rho(Z1=-2,Z2=-2,rho=0.6),2) *# rho=0.6*

round(Mahala\_dist\_rho(Z1=-2,Z2=-2,rho=-0.6),2) *# rho=-0.6*

***### Function to compute the required sample size for the Mahalanobis distance-based approach (i.e. equation 17)***  
*# alpha = Type I error rate*  
*# gamma = Type II error rate*  
*# k = number of predictors in the multivariate regression model*  
*# Mahala\_c = cut-off point for decision making (e.g. a percentile of the chi distribution)*  
*# delta\_Mahala = effect size = tested person's Mahalanobis distance - Mahala\_c*  
  
N\_star\_Mahala<-**function**(alpha,gamma,k,Mahala\_c,delta\_Mahala){  
   
 N<-((qnorm(1-alpha)\*sqrt(k+1+((Mahala\_c^2)/2))+qnorm(1-gamma)\*sqrt(k+1+(((Mahala\_c+delta\_Mahala)^2)/2)))/delta\_Mahala)^2  
 return(N)  
}  
  
  
*# Examples:*  
N\_star\_Mahala(alpha=0.05,gamma=0.2,k=2,Mahala\_c=2.15,delta\_Mahala=0.25)

N\_star\_Mahala(alpha=0.01,gamma=0.2,k=3,Mahala\_c=2.45,delta\_Mahala=0.35)

N\_star\_Mahala(alpha=0.05,gamma=0.1,k=5,Mahala\_c=2.45,delta\_Mahala=0.35)

*# To obtain the size of the normative sample such that half the confidence interval width for Mahala\_c*   
*# equals the desired margin of estimation error, that is, equation (18)*  
  
*# Example*   
N\_star\_Mahala(alpha=0.05/2,gamma=0.5,k=5,Mahala\_c=2.45,delta\_Mahala=0.15)

*# note that alpha should be halved and power (i.e. 1-gamma) should be 0.5*