

```

# eAppendix 2

# Simulation comparing SI, SA, OWA, PCA, and CME methods for 2 informants
# Data generated from Congeneric Measurement Model with Rounding (negative values rounded to zero)
#  $X_{ij} = \delta_{.j} + \lambda_{.j} \cdot T_{.i} + \epsilon_{.ij}$ ,
# where  $\text{var}(\epsilon_{.ij}) = \text{sig}^2_{.j}$ 
# and where  $\text{cor}(T_{.i}, \epsilon_{.ij}) = \rho_{.j} = 0$ 
# and where  $\text{cor}(\epsilon_{.i1}, \epsilon_{.i2}) = \eta = 0$ 

# Sample size
N <- 250

# Fixed parameters:

# Mean and variance of T (true predictor) BEFORE rounding
T.mu <- 0
T.sigsq <- 1
T.sig <- sqrt(T.sigsq)

# Mean of T AFTER rounding (based on mixture of point mass at zero and truncated normal)
Tnew.mu <- pnorm((-1*T.mu/T.sig))*0 +
  (1 - pnorm((-1*T.mu/T.sig)))*(T.mu + T.sig*(dnorm(-1*T.mu/T.sig)/(1 - pnorm((-1*T.mu/T.sig)))))

# Regression parameters
a <- 1
b <- 1
epsilon.sd <- 1 # Error for regression equation

# Informants
K <- 2

# Congeneric model parameters
rho.1 <- 0
rho.2 <- 0

```

```
eta <- 0
```

```
# Parameters to be varied:
```

```
# Congeneric model parameters
```

```
delta.1 <- c(-0.5, -0.25, 0, 0.25, 0.5)
```

```
delta.2 <- c(-0.5, -0.25, 0, 0.25, 0.5)
```

```
lambda.1 <- c(0.4, 0.75, 1, 1.1, 4/3)
```

```
lambda.2 <- c(0.4, 0.75, 1, 1.1, 4/3)
```

```
sigsq.1 <- c(0, 0.1, 0.5)
```

```
sigsq.2 <- c(0, 0.1, 0.5)
```

```
# Create arrays to save results in
```

```
incl <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),  
                        length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),  
                        length(eta), length(epsilon.sd)))
```

```
dimnames(incl) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)
```

```
results.corrX1X2 <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),  
                                   length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),  
                                   length(eta), length(epsilon.sd)))
```

```
dimnames(results.corrX1X2) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)
```

```
results.corrThetaY <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),  
                                       length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),  
                                       length(eta), length(epsilon.sd)))
```

```
dimnames(results.corrThetaY) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)
```

```
# Classical Measurement Error approach
```

```

corr.results.ME.na.a <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
                                     length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
                                     length(eta), length(epsilon.sd)))
dimnames(corr.results.ME.na.a) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

corr.results.ME.est.a <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
                                     length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
                                     length(eta), length(epsilon.sd)))
dimnames(corr.results.ME.est.a) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

corr.results.ME.mse.a <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
                                     length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
                                     length(eta), length(epsilon.sd)))
dimnames(corr.results.ME.mse.a) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

corr.results.ME.fail.a <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
                                     length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
                                     length(eta), length(epsilon.sd)))
dimnames(corr.results.ME.fail.a) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

corr.results.ME.exceed.a <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
                                     length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
                                     length(eta), length(epsilon.sd)))
dimnames(corr.results.ME.exceed.a) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

cov.results.ME.est.a <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
                                     length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
                                     length(eta), length(epsilon.sd)))
dimnames(cov.results.ME.est.a) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

cov.results.ME.mse.a <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
                                     length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
                                     length(eta), length(epsilon.sd)))
dimnames(cov.results.ME.mse.a) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

```

```

cov.results.ME.fail.a <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(cov.results.ME.fail.a) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

# Single informant approach (just regress on X1)
corr.results.N.est <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(corr.results.N.est) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

corr.results.N.mse <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(corr.results.N.mse) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

cov.results.N.est <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(cov.results.N.est) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

cov.results.N.mse <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(cov.results.N.mse) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

# Simple Average approach
corr.results.AV.est <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(corr.results.AV.est) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

corr.results.AV.mse <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))

```

```

dimnames(corr.results.AV.mse) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

cov.results.AV.est <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(cov.results.AV.est) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

cov.results.AV.mse <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(cov.results.AV.mse) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

# Optimal Weighted Average Approach
corr.results.AVW.est <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(corr.results.AVW.est) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

corr.results.AVW.mse <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(corr.results.AVW.mse) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

cov.results.AVW.est <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(cov.results.AVW.est) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

cov.results.AVW.mse <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
length(eta), length(epsilon.sd)))
dimnames(cov.results.AVW.mse) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

```

```

# PCA Approach (with covariances)
corr.results.PCA1.est <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
    length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
    length(eta), length(epsilon.sd)))
dimnames(corr.results.PCA1.est) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

corr.results.PCA1.mse <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
    length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
    length(eta), length(epsilon.sd)))
dimnames(corr.results.PCA1.mse) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

cov.results.PCA1.est <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
    length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
    length(eta), length(epsilon.sd)))
dimnames(cov.results.PCA1.est) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

cov.results.PCA1.mse <- array(0, dim = c(length(delta.1),length(delta.2),length(lambda.1),length(lambda.2),
    length(sigsq.1),length(sigsq.2),length(rho.1),length(rho.2),
    length(eta), length(epsilon.sd)))
dimnames(cov.results.PCA1.mse) <- list(delta.1,delta.2,lambda.1,lambda.2,sigsq.1,sigsq.2,rho.1,rho.2,eta,epsilon.sd)

# Begin iterations

round.rho <- 0 # Should CME estimates of rho that exceed 1 be rounded down to 1? (0 = no)

R <- 1000 # Number of iterations
cnter <- 1 # Counter
draw <- 0

library(mvtnorm)
set.seed(333)
for(r in 1:length(delta.1))
{
  for(s in 1:length(delta.2))

```

```

{
  for(t in 1:length(lambda.1))
  {
    for(u in 1:length(lambda.2))
    {
      for(v in 1:length(sigsq.1))
      {
        for(x in 1:length(sigsq.2))
        {
          for(y in 1:length(rho.1))
          {
            for(z in 1:length(rho.2))
            {
              for(z2 in 1:length(eta))
              {
                for(z3 in 1:length(epsilon.sd))
                {
                  # Correlation storage for iterations
                  r.est.true <- c()
                  r.est.ME.a <- c()
                  r.est.naive <- c()
                  r.est.ave <- c()
                  r.est.wtave <- c()
                  r.est.pca1 <- c()

                  r.est.ME.a.fail <- c()
                  r.est.ME.a.exceed <- c()

                  # Covariance storage for iterations
                  b.est.true <- c()
                  b.est.ME.a <- c()
                  b.est.naive <- c()
                  b.est.ave <- c()
                  b.est.wtave <- c()
                  b.est.pca1 <- c()

                  b.est.ME.a.fail <- c()

                  # Parameters for this experiment

```

```

Delta.1 <- delta.1[r]
Delta.2 <- delta.2[s]
Lambda.1 <- lambda.1[t]
Lambda.2 <- lambda.2[u]
Sigsq.1 <- sigsq.1[v]
Sigsq.2 <- sigsq.2[x]
Rho.1 <- rho.1[y]
Rho.2 <- rho.2[z]
Eta <- eta[z2]
Epsilon.sd <- epsilon.sd[z3]

# Generate T and epsilons
# Calculate mean and covariance for (T before rounding, Epsilon1, Epsilon2) distribution
mn <- rep(0,1+K)
mn[1] <- T.mu

sig <- diag(3)
sig[1,1] <- T.sigsq
sig[2,2] <- Sigsq.1
sig[3,3] <- Sigsq.2
sig[1,2] <- sig[2,1] <- sqrt(T.sigsq)*sqrt(Sigsq.1)*Rho.1
sig[1,3] <- sig[3,1] <- sqrt(T.sigsq)*sqrt(Sigsq.2)*Rho.2
sig[2,3] <- sig[3,2] <- sqrt(Sigsq.1)*sqrt(Sigsq.2)*Eta

for(k in 1:R)
{
  rm(draws)
  draws <- rmvnorm(N, mean = mn, sigma = sig)

  # Standard version
  T.i <- draws[,1]
  T.i[T.i < 0] <- 0 # Round -> can't be less than zero
  # Center T by after-rounding mean
  X1.i <- Delta.1 + Tnew.mu + Lambda.1*(T.i - Tnew.mu) + draws[,2]
  X1.i[X1.i < 0] <- 0 # Can't be less than zero
  X2.i <- Delta.2 + Tnew.mu + Lambda.2*(T.i - Tnew.mu) + draws[,3]
  X2.i[X2.i < 0] <- 0 # Can't be less than zero
}

```

```

epsilon.i <- rnorm(N, mean = 0, sd = Epsilon.sd) # regression error - also epsilon
Y.i <- a + b*T.i + epsilon.i

# Calculate averages for informants
X <- cbind(X1.i,X2.i)
X.grandmean <- mean(X)
X.indmean <- apply(X,1,mean)
X.varmean <- apply(X,2,mean)
X.indmeandevgrand <- X.indmean - rep(X.grandmean,N)
X.inddevindmean <- X - cbind(X.indmean,X.indmean)

# True regression (using T, rather than Xs)
r.est.true[k] <- cor(x = T.i, y = Y.i, method = "pearson", use = "complete.obs")
b.est.true[k] <- lm(Y.i ~T.i)$coef[2]

# Classical measurement error method
sigma.sq <- sum(as.vector(X.inddevindmean^2))/(N*(K-1))
S.T.unst <- sqrt(sum(X.indmeandevgrand^2) - N*(sigma.sq/K)) # Can be NA
lambda.num <- sum(X.indmeandevgrand^2) - N*(sigma.sq/K)
S.y.unst <- sqrt(sum((Y.i - mean(Y.i))^2))
cov.unst <- sum(X.indmeandevgrand*(Y.i - mean(Y.i)))
if(lambda.num>0)
{
  b.est.ME.a[k] <- sum(X.indmeandevgrand*Y.i)/(sum(X.indmeandevgrand^2) - N*(sigma.sq/K))
  b.est.ME.a.fail[k] <- 0
  r.est.ME.a[k] <- cov.unst/(S.T.unst*S.y.unst)
  r.est.ME.a.fail[k] <- 0
}
if(lambda.num<=0)
{
  b.est.ME.a[k] <- NA # Wouldn't fail normally, but sign would be flipped
  b.est.ME.a.fail[k] <- 1
  r.est.ME.a[k] <- NA # Would fail normally b/c of square root of negative
  r.est.ME.a.fail[k] <- 1
  print("Both Est Fail")
}
if(is.na(r.est.ME.a[k]))

```

```

{
  r.est.ME.a.exceed[k] <- NA
}
if(!is.na(r.est.ME.a[k]) & abs(r.est.ME.a[k])<=1)
{
  r.est.ME.a.exceed[k] <- 0
}
if(!is.na(r.est.ME.a[k]) & abs(r.est.ME.a[k])>1)
{
  r.est.ME.a.exceed[k] <- 1
  if(round.rho==1)
  {
    r.est.ME.a[k] <- 1*sign(r.est.ME.a[k]) # Round to appropriate value!
  }
  print("Rho Est Exceed")
}

# Single informant approach
r.est.naive[k] <- cor(x = X1.i, y = Y.i, method = "pearson", use = "complete.obs")
lm.naive <- lm(Y.i ~ X1.i)
b.est.naive[k] <- lm.naive$coef[2]

# Simple average approach
r.est.ave[k] <- cor(x = X.indmean, y = Y.i, method = "pearson", use = "complete.obs")
lm.ave <- lm(Y.i ~ X.indmean)
b.est.ave[k] <- lm.ave$coef[2]

# Optimal weighted average approach
p <- seq(0,1,length=40)
p.corr <- c()
for(p.ind in 1:length(p))
{
  pp <- p[p.ind]
  X.indweightedmean <- pp*X1.i + (1-pp)*X2.i
  p.corr[p.ind] <- cor(x = X.indweightedmean, y = Y.i, method = "pearson", use = "complete.obs")
}

```

```

}
p.best <- p[p.corr==max(p.corr)]
#X.indbestweightedmean <- p.best*X1.i + (1-p.best)*X2.i
r.est.wtave[k] <- max(p.corr)

p <- seq(0,1,length=40)
p.res <- c()
for(p.ind in 1:length(p))
{
  pp <- p[p.ind]
  X.indweightedmean <- pp*X1.i + (1-pp)*X2.i
  p.res[p.ind] <- sum((lm(Y.i ~ X.indweightedmean)$resid)^2)
}
p.best <- p[p.res==min(p.res)]
X.indbestweightedmean <- p.best*X1.i + (1-p.best)*X2.i
b.est.wtave[k] <- lm(Y.i ~ X.indbestweightedmean)$coef[2]

```

```

# PCA approach (using covariances) - with standardized loadings
X.pca1 <- summary(princomp(~., data = data.frame(X), cor = F, scores = T))$scores[,1]
loading <- summary(princomp(~., data = data.frame(X), cor = F, scores = T))$loadings[,1]
loading <- loading/sum(loading)
X.pca1.b <- loading[1]*(X1.i - mean(X1.i)) + loading[2]*(X2.i - mean(X2.i))
r.est.pca1[k] <- cor(x = X.pca1.b, y = Y.i, method = "pearson", use = "complete.obs")
lm.pca1 <- lm(Y.i ~ X.pca1.b)
b.est.pca1[k] <- lm.pca1$coef[2]

```

```

} # End replications

```

```

incl[r,s,t,u,v,x,y,z,z2,z3] <- 1

```

```

results.corrX1X2[r,s,t,u,v,x,y,z,z2,z3] <- sig[2,3]/(sqrt(sig[2,2])*sqrt(sig[3,3]))

```

```

results.corrThetaY[r,s,t,u,v,x,y,z,z2,z3] <- mean(r.est.true)

# Correlation results:
r.true <- mean(r.est.true)

corr.results.N.est[r,s,t,u,v,x,y,z,z2,z3] <- mean(r.est.naive - r.true)
corr.results.AV.est[r,s,t,u,v,x,y,z,z2,z3] <- mean(r.est.ave - r.true)
corr.results.AVW.est[r,s,t,u,v,x,y,z,z2,z3] <- mean(r.est.wtave - r.true)
corr.results.PCA1.est[r,s,t,u,v,x,y,z,z2,z3] <- mean(r.est.pca1 - r.true)
corr.results.ME.fail.a[r,s,t,u,v,x,y,z,z2,z3] <- sum(r.est.ME.a.fail==1)
corr.results.ME.exceed.a[r,s,t,u,v,x,y,z,z2,z3] <- sum(r.est.ME.a.exceed==1, na.rm = T) # How many iterations rho is >1
corr.results.ME.na.a[r,s,t,u,v,x,y,z,z2,z3] <- mean(is.na(r.est.ME.a))
corr.results.ME.est.a[r,s,t,u,v,x,y,z,z2,z3] <- mean(r.est.ME.a - r.true, na.rm = T) # Method fails for some iterations

corr.results.N.mse[r,s,t,u,v,x,y,z,z2,z3] <- mean((r.est.naive - r.true)^2)
corr.results.AV.mse[r,s,t,u,v,x,y,z,z2,z3] <- mean((r.est.ave - r.true)^2)
corr.results.AVW.mse[r,s,t,u,v,x,y,z,z2,z3] <- mean((r.est.wtave - r.true)^2)
corr.results.PCA1.mse[r,s,t,u,v,x,y,z,z2,z3] <- mean((r.est.pca1 - r.true)^2)
corr.results.ME.mse.a[r,s,t,u,v,x,y,z,z2,z3] <- mean((r.est.ME.a - r.true)^2, na.rm = T)

# Covariance results:

```

```

b.true <- b

cov.results.N.est[r,s,t,u,v,x,y,z,z2,z3] <- mean(b.est.naive - b.true)
cov.results.AV.est[r,s,t,u,v,x,y,z,z2,z3] <- mean(b.est.ave - b.true)
cov.results.AVW.est[r,s,t,u,v,x,y,z,z2,z3] <- mean(b.est.wtave - b.true)
cov.results.PCA1.est[r,s,t,u,v,x,y,z,z2,z3] <- mean(b.est.pca1 - b.true)
cov.results.ME.fail.a[r,s,t,u,v,x,y,z,z2,z3] <- sum(b.est.ME.a.fail==1)
cov.results.ME.est.a[r,s,t,u,v,x,y,z,z2,z3] <- mean(b.est.ME.a - b.true, na.rm = T) #

cov.results.N.mse[r,s,t,u,v,x,y,z,z2,z3] <- mean((b.est.naive - b.true)^2)
cov.results.AV.mse[r,s,t,u,v,x,y,z,z2,z3] <- mean((b.est.ave - b.true)^2)
cov.results.AVW.mse[r,s,t,u,v,x,y,z,z2,z3] <- mean((b.est.wtave - b.true)^2)
cov.results.PCA1.mse[r,s,t,u,v,x,y,z,z2,z3] <- mean((b.est.pca1 - b.true)^2)
cov.results.ME.mse.a[r,s,t,u,v,x,y,z,z2,z3] <- mean((b.est.ME.a - b.true)^2, na.rm = T)

print(cnter)
cnter <- cnter + 1

} # End epsilon.sd
} # End eta
} # End rho.2
} # End rho.1
} # End sigsq.2
} # End sigsq.1
} # End lambda.2

```

```

    } # End lambda.1
  } # End delta.2
} # End delta.1

warnings()

results <- matrix(0, ncol = 10 + 1 + 1 + 14 + 14 + 1 + 3,
  nrow = length(delta.1)*length(delta.2)*length(lambda.1)*length(lambda.2)*length(sigsq.1)*
  length(sigsq.2)*length(rho.1)*length(rho.2)*length(eta)*length(epsilon.sd))

results[,1] <- rep(delta.1,
  times = length(delta.2)*length(lambda.1)*length(lambda.2)*length(sigsq.1)*
  length(sigsq.2)*length(rho.1)*length(rho.2)*length(eta)*length(epsilon.sd))
results[,2] <- rep(rep(delta.2, times = rep(length(delta.1), times = length(delta.2))),
  times = length(lambda.1)*length(lambda.2)*length(sigsq.1)*
  length(sigsq.2)*length(rho.1)*length(rho.2)*length(eta)*length(epsilon.sd))
results[,3] <- rep(rep(lambda.1, times = rep(length(delta.1)*length(delta.2), times = length(lambda.1))),
  times = length(lambda.2)*length(sigsq.1)*length(sigsq.2)*length(rho.1)*
  length(rho.2)*length(eta)*length(epsilon.sd))
results[,4] <- rep(rep(lambda.2, times = rep(length(delta.1)*length(delta.2)*length(lambda.1), times = length(lambda.2))),
  times = length(sigsq.1)*length(sigsq.2)*length(rho.1)*length(rho.2)*length(eta)*length(epsilon.sd))
results[,5] <- rep(rep(sigsq.1, times = rep(length(delta.1)*length(delta.2)*length(lambda.1)*length(lambda.2),
  times = length(sigsq.1))),
  times = length(sigsq.2)*length(rho.1)*length(rho.2)*length(eta)*length(epsilon.sd))
results[,6] <- rep(rep(sigsq.2, times = rep(length(delta.1)*length(delta.2)*length(lambda.1)*length(lambda.2)*length(sigsq.1),
  times = length(sigsq.2))),
  times = length(rho.1)*length(rho.2)*length(eta)*length(epsilon.sd))

results[,11] <- round(as.vector(results.corrX1X2),2)
results[,12] <- round(as.vector(results.corrThetaY),2)

results[,13] <- round(as.vector(corr.results.N.est),2)
results[,14] <- round(as.vector(corr.results.AV.est),2)
results[,15] <- round(as.vector(corr.results.AVW.est),2)
results[,16] <- round(as.vector(corr.results.ME.est.a),2)

```

```

results[,18] <- round(as.vector(corr.results.PCA1.est),2)
results[,20] <- round(as.vector(corr.results.N.mse),2)
results[,21] <- round(as.vector(corr.results.AV.mse),2)
results[,22] <- round(as.vector(corr.results.AVW.mse),2)
results[,23] <- round(as.vector(corr.results.ME.mse.a),2)
results[,25] <- round(as.vector(corr.results.PCA1.mse),2)

results[,13+14] <- round(as.vector(cov.results.N.est),2)
results[,14+14] <- round(as.vector(cov.results.AV.est),2)
results[,15+14] <- round(as.vector(cov.results.AVW.est),2)
results[,16+14] <- round(as.vector(cov.results.ME.est.a),2)
results[,18+14] <- round(as.vector(cov.results.PCA1.est),2)
results[,20+14] <- round(as.vector(cov.results.N.mse),2)
results[,21+14] <- round(as.vector(cov.results.AV.mse),2)
results[,22+14] <- round(as.vector(cov.results.AVW.mse),2)
results[,23+14] <- round(as.vector(cov.results.ME.mse.a),2)
results[,25+14] <- round(as.vector(cov.results.PCA1.mse),2)

results[,27+14] <- round(as.vector(corr.results.ME.na.a),2)

results[,28+14] <- round(as.vector(corr.results.ME.fail.a),2)
results[,29+14] <- round(as.vector(corr.results.ME.exceed.a),2)
results[,30+14] <- round(as.vector(cov.results.ME.fail.a),2)

# Output results of interest
results.online <- results[,c(1:6,c(13,14,15,18,16),c(13,14,15,18,16)+7,c(13,14,15,18,16)+14,c(13,14,15,18,16)+21,42:44)]
colnames(results.online) <- c("delta.1", "delta.2", "lambda.1", "lambda.2", "sigmasq.1", "sigmasq.2",
                             "SI.Rho.Bias", "SA.Rho.Bias", "OWA.Rho.Bias", "PCA.Rho.Bias", "CME.Rho.Bias",
                             "SI.Rho.MSE", "SA.Rho.MSE", "OWA.Rho.MSE", "PCA.Rho.MSE", "CME.Rho.MSE",
                             "SI.Beta.Bias", "SA.Beta.Bias", "OWA.Beta.Bias", "PCA.Beta.Bias", "CME.Beta.Bias",
                             "SI.Beta.MSE", "SA.Beta.MSE", "OWA.Beta.MSE", "PCA.Beta.MSE", "CME.Beta.MSE",
                             "NumberIterationsFail.CME.Rho", "NumberIterationsExceedOne.CME.Rho", "NumberIterationsFail.CME.Beta")
write.table(results.online, file = "JavarasEtAlSimulationResults.csv", row.names = F, col.names = T, sep = ",")

```

q()