```
> ##### Example 1: Handling Dependence Due to Multiple ESs within a Study #####
> ##### Step 1: Load Working Directory, Functions #####
>
> setwd("F:/Cochrane-C2 10")
> robust.se <- function(data,X.full,rho) {
+ p <- ncol(X.full)-2
+ N <- max(data$study)
+ sumXWX <- 0
+ sumXWy<- 0
+ sumXWJWX <- 0
+ sumXWVWX <- 0
+  sumXW.sig.m.v.WX <- 0
+
+ for (i in (1: N)) {
+ tab <- data[data$study == i,]
+ W <- diag(tab$weights,tab$k[1])
+ tab2 <- X.full[X.full$study==i,]
+ tab3 <- cbind(tab2[-c(1)])
+ X <- data.matrix(tab3)
+ dimnames(X) <- NULL
+ y <- cbind(tab$effectsize)
+ one <- cbind(rep(1,tab$k[1]))
+ J <- one%*%t(one)
+  sigma <- (tab$s%*%t(tab$s))
+ vee <- diag(tab$s^2,tab$k[1])
+ SigmV <- sigma - vee
+
+ sumXWX <- sumXWX + t(X)%*%W%*%X
+ sumXWy <- sumXWy + t(X)%*%W%*%y
+ sumXWJWX <- sumXWJWX + t(X)%*%W%*%J%*%W%*%X
+ sumXWVWX <- sumXWVWX + t(X)%*%W%*%vee%*%W%*%X
+ sumXW.sig.m.v.WX <- sumXW.sig.m.v.WX + t(X)%*%W%*%SigmV%*%W%*%X
+ }
+
+
+ b <- solve(sumXWX)%*%sumXWy
+ X <- data.matrix(X.full[-c(1)])
+ dimnames(X) <- NULL
+ data$pred <- X%*%b
+ data$e <- data$effectsize - data$pred
+ W <- diag(data$weights)
+ sumW <- sum(data$weights)
+ Qe <-t(data$e)%*%W%*%data$e
+
+ denom <- sumW - sum(diag(solve(sumXWX)%*%sumXWJWX))
+ term1 <- (Qe - N + sum(diag(solve(sumXWX)%*%sumXWVWX))) /denom
+ term2 <- (sum(diag(solve(sumXWX)%*%sumXW.sig.m.v.WX ))) / denom
+ tau.sq1 <- term1 + rho*term2
+ tau.sq <- ifelse(tau.sq1<0,0,tau.sq1)
+ data$r.weights <- 1/(data$k*(data$meanvar + tau.sq))
+
+ sumXWX.r <- 0
+ sumXWy.r<- 0
+ for (i in (1: N)) {
+ tab <- data[data$study == i,]
```

```
+ W <- diag(tab$r.weights,tab$k[1])
+ tab2 <- X.full[X.full$study==i,]
+ tab3 <- cbind(tab2[-c(1)])
+ X <- data.matrix(tab3)
+ dimnames(X) <- NULL
+
+ y <- cbind(tab$effectsize)
+
+ sumXWX.r <- sumXWX.r + t(X)%*%W%*%X
+ sumXWy.r <- sumXWy.r + t(X)%*%W%*%y
+ }
+
+ b.r <- solve(sumXWX.r)%*%sumXWy.r
+ X <- data.matrix(X.full[-c(1)])
+ dimnames(X) <- NULL
+ data$pred.r <- X%*%b.r
+ data$e.r <- cbind(data$effectsize) - data$pred.r
+
+ sumXWeeWX.r <- 0
+ for (i in (1:N)) {
+ tab <- data[data$study== i,]
+ sigma.hat.r <- tab$e.r%*%t(tab$e.r)
+ W <- diag(tab$r.weights,tab$k[1])
+ tab2 <- X.full[X.full$study==i,]
+ tab3 <- cbind(tab2[-c(1)])
+ X <- data.matrix(tab3)
+ dimnames(X) <- NULL
+
+ sumXWeeWX.r <- sumXWeeWX.r + t(X)%*%W%*%sigma.hat.r%*%W%*%X
+ }
+
+ VR.r <- solve(sumXWX.r)%*%sumXWeeWX.r%*%solve(sumXWX.r)
+
+ SE <- c(rep(0,p+1))
+ for (i in (1:(p+1))) {
+ SE[i] <-sqrt(VR.r[i,i])*sqrt(N/(N-(p+1)))
+ }
+
+ labels <- c(colnames(X.full[2:length(X.full)]))
+ output <- cbind(labels,b.r,SE)
+ colnames(output) <- c("beta", "estimate","SE")
+ list("Tau.sq estimate" = tau.sq, "Estimates & Robust SE's"= output)
+ }
>
> compare <- function(data,X.full) {
+ rho <- seq(0,1,.1)
+ tau.sq.v <- function(data,X.full,rho) {
+ p <- ncol(X.full)-2
+ N <- max(data$study)
+ sumXWX <- 0
+ sumXWy<- 0
+ sumXWJWX <- 0
+ sumXWVWX <- 0
+  sumXW.sig.m.v.WX <- 0
+
```

```
+ for (i in (1: N)) {
+ tab <- data[data$study == i,]
+ W <- diag(tab$weights,tab$k[1])
+ tab2 <- X.full[X.full$study==i,]
+ tab3 <- cbind(tab2[-c(1)])
+ X <- data.matrix(tab3)
+ dimnames(X) <- NULL
+ y <- cbind(tab$effectsize)
+ one <- cbind(rep(1,tab$k[1]))
+ J <- one%*%t(one)
+  sigma <- (tab$s%*%t(tab$s))
+ vee <- diag(tab$s^2,tab$k[1])
+ SigmV <- sigma - vee
+
+ sumXWX <- sumXWX + t(X)%*%W%*%X
+ sumXWy <- sumXWy + t(X)%*%W%*%y
+ sumXWJWX <- sumXWJWX + t(X)%*%W%*%J%*%W%*%X
+ sumXWVWX <- sumXWVWX + t(X)%*%W%*%vee%*%W%*%X
+ sumXW.sig.m.v.WX <- sumXW.sig.m.v.WX + t(X)%*%W%*%SigmV%*%W%*%X
+ }
+
+ b <- solve(sumXWX)%*%sumXWy
+ X <- data.matrix(X.full[-c(1)])
+ dimnames(X) <- NULL
+ data$pred <- X%*%b
+ data$e <- data$effectsize - data$pred
+ W <- diag(data$weights)
+ sumW <- sum(data$weights)
+ Qe <-t(data$e)%*%W%*%data$e
+
+ denom <- sumW - sum(diag(solve(sumXWX)%*%sumXWJWX))
+ term1 <- (Qe - N + sum(diag(solve(sumXWX)%*%sumXWVWX))) /denom
+ term2 <- (sum(diag(solve(sumXWX)%*%sumXW.sig.m.v.WX ))) / denom
+ tau.sq1 <- term1 + rho*term2
+ tau.sq <- ifelse(tau.sq1<0,0,tau.sq1)
+ tau.sq
+ }
+
+ tau.sq <- c(rep(0,length(rho)))
+ for (i in (1:length(rho))) {
+ tau.sq[i] <-tau.sq.v(data,X.full,rho[i])
+ }
+ output <- cbind(rho,tau.sq)
+ colnames(output) <- c("rho","tau.sq")
+ output
+ }
>
> ##### Step 2: Import Data #####
> ##### Option 1 is to use single data file and use subset statements when
reading data into R#
> ##### Note that you can import .out and .csv files using read.csv()
> ##### Or, you can import .dta, .sav, or .sas files using the foreign package
>
> fulldata <-read.csv('SingleDataset.out')
> library(foreign)
```

```
> fullstatadata <-read.dta('SingleDataset.dta')
> fullspssdata <-read.spss('SingleDataset.sav')
> fullsasdata <-read.ssd('SingleDataset.sas')
>
> ##### Option 2 is to use two data files (Data and Design)#####
> Data <- read.csv('Data.out')
> Design <- read.csv('Design.out')
>
> ##### Step 3: Run Regression Models #####
> ##### First Using Option 1 with Subset Statements with a Single Dataset #####
> ##### Model 1 Using Measurement Variables Only #####
> m1data <- subset(fulldata, select=c(2, 4:(length(fulldata)-2)))
> m1X.full <- subset(fulldata, select=c(2:3, 10:(length(fulldata)-2)))
> robust.se(m1data, m1X.full, .80)
$`Tau.sq estimate`
          [,1]
[1,] 0.1555256

$`Estimates & Robust SE's`
     beta        estimate               SE
[1,] "intercept" "0.18630479413004"      "0.132006733576191"
[2,] "m_alc"     "0.123802814013168"     "0.082135017041981"
[3,] "c_alc"     "-0.0348254904792215"   "0.0869716509129128"
[4,] "m_dvdays"  "-9.92045828493657e-05" "0.000479674081477361"
[5,] "c_dvdays"  "0.000620081307137491"  "0.000335592787736102"
[6,] "m_sreport" "-0.0939147923791257"   "0.138076652302361"
[7,] "c_sreport" "0.236548796098185"     "0.110438826242001"

> compare(m1data, m1X.full)
      rho    tau.sq
 [1,] 0.0 0.1550079
 [2,] 0.1 0.1550726
 [3,] 0.2 0.1551373
 [4,] 0.3 0.1552021
 [5,] 0.4 0.1552668
 [6,] 0.5 0.1553315
 [7,] 0.6 0.1553962
 [8,] 0.7 0.1554609
 [9,] 0.8 0.1555256
[10,] 0.9 0.1555903
[11,] 1.0 0.1556550
>
> ##### Model 2 Using All Variables #####
> m2data <- subset(fulldata, select=c(2, 4:length(fulldata)))
> m2X.full <- subset(fulldata, select=c(2:3, 10:length(fulldata)))
> robust.se(m2data, m2X.full, .80)
$`Tau.sq estimate`
          [,1]
[1,] 0.1575305

$`Estimates & Robust SE's`
      beta        estimate               SE
 [1,] "intercept" "0.315589813176703"    "0.613132003622982"
 [2,] "m_alc"     "0.178236711697457"    "0.145253471425459"
 [3,] "c_alc"     "-0.0348608123005546"  "0.087816477610962"
```

```
 [4,] "m_dvdays"  "-0.00011561031501047" "0.000494346693335574"
 [5,] "c_dvdays"  "0.00062002671737237" "0.000339205202066935"
 [6,] "m_sreport" "-0.0976866431873535"  "0.135634037315058"
 [7,] "c_sreport" "0.236656250036475"    "0.11156292115301"
 [8,] "permale"   "0.000867538398664154" "0.00306113075273208"
 [9,] "age"       "-0.0120225861931168"  "0.0343015830882078"

> compare(m2data, m2X.full)
      rho    tau.sq
 [1,] 0.0 0.1566894
 [2,] 0.1 0.1567945
 [3,] 0.2 0.1568997
 [4,] 0.3 0.1570048
 [5,] 0.4 0.1571100
 [6,] 0.5 0.1572151
 [7,] 0.6 0.1573202
 [8,] 0.7 0.1574254
 [9,] 0.8 0.1575305
[10,] 0.9 0.1576357
[11,] 1.0 0.1577408
>
> ##### Next Using Option 2 with Two Separate Datasets #####
> ##### Model 1 Using Measurement Variables Only #####
> m1sdata <- data.frame(Data)
> m1sX.full <- subset(Design, select=c(1:8))
> robust.se(m1sdata, m1sX.full, .80)
$`Tau.sq estimate`
          [,1]
[1,] 0.1555256

$`Estimates & Robust SE's`
     beta        estimate                SE
[1,] "intercept" "0.18630479413004"      "0.132006733576191"
[2,] "m_alc"     "0.123802814013168"     "0.082135017041981"
[3,] "c_alc"     "-0.0348254904792215"   "0.0869716509129128"
[4,] "m_dvdays"  "-9.92045828493657e-05" "0.000479674081477361"
[5,] "c_dvdays"  "0.000620081307137491"  "0.000335592787736102"
[6,] "m_sreport" "-0.0939147923791257"   "0.138076652302361"
[7,] "c_sreport" "0.236548796098185"     "0.110438826242001"

> compare(m1sdata, m1sX.full)
      rho    tau.sq
 [1,] 0.0 0.1550079
 [2,] 0.1 0.1550726
 [3,] 0.2 0.1551373
 [4,] 0.3 0.1552021
 [5,] 0.4 0.1552668
 [6,] 0.5 0.1553315
 [7,] 0.6 0.1553962
 [8,] 0.7 0.1554609
 [9,] 0.8 0.1555256
[10,] 0.9 0.1555903
[11,] 1.0 0.1556550
>
> ##### Model 2 Using All Variables #####
```

```
> m2data <- data.frame(Data)
> m2X.full <- data.frame(Design)
> robust.se(m2data, m2X.full, .80)
$`Tau.sq estimate`
          [,1]
[1,] 0.1575305

$`Estimates & Robust SE's`
      beta        estimate               SE
 [1,] "intercept" "0.315589813176703"    "0.613132003622982"
 [2,] "m_alc"     "0.178236711697457"    "0.145253471425459"
 [3,] "c_alc"     "-0.0348608123005546"  "0.087816477610962"
 [4,] "m_dvdays"  "-0.00011561031501047" "0.000494346693335574"
 [5,] "c_dvdays"  "0.000620026717377237" "0.000339205202066935"
 [6,] "m_sreport" "-0.0976866431873535"  "0.135634037315058"
 [7,] "c_sreport" "0.236656250036475"    "0.11156292115301"
 [8,] "permale"   "0.000867538398664154" "0.00306113075273208"
 [9,] "age"       "-0.0120225861931168"  "0.0343015830882078"

> compare(m2data, m2X.full)
      rho    tau.sq
 [1,] 0.0 0.1566894
 [2,] 0.1 0.1567945
 [3,] 0.2 0.1568997
 [4,] 0.3 0.1570048
 [5,] 0.4 0.1571100
 [6,] 0.5 0.1572151
 [7,] 0.6 0.1573202
 [8,] 0.7 0.1574254
 [9,] 0.8 0.1575305
[10,] 0.9 0.1576357
[11,] 1.0 0.1577408
>
> ##### Compare to naive standard errors calculated in metafor package #####
> library(metafor)
> rma.uni(effectsize, sei=s, mods=cbind(m_alc, c_alc, m_dvdays, c_dvdays,
m_sreport, c_sreport), data=fulldata, method="DL")

Mixed-Effects Model (k = 438; tau^2 estimator: DL)

tau^2 (estimate of residual amount of heterogeneity): 0.0960
tau (sqrt of the estimate of residual heterogeneity): 0.3099

Test for Residual Heterogeneity:
QE(df = 431) = 1574.1815, p-val < .0001

Test of Moderators (coefficient(s) 2,3,4,5,6,7):
QM(df = 6) = 19.6037, p-val = 0.0033

Model Results:

           estimate      se     zval    pval    ci.lb   ci.ub
intrcpt      0.0197  0.0765   0.2571  0.7971  -0.1302  0.1696
m_alc        0.1066  0.0503   2.1181  0.0342   0.0080  0.2053   *
c_alc       -0.0989  0.0767  -1.2891  0.1974  -0.2493  0.0515
```

```
m_dvdays    -0.0000  0.0002  -0.0135  0.9892  -0.0004  0.0004
c_dvdays     0.0006  0.0003   1.7706  0.0766  -0.0001  0.0012   .
m_sreport    0.0528  0.0753   0.7009  0.4834  -0.0948  0.2003
c_sreport    0.2303  0.0802   2.8696  0.0041   0.0730  0.3875  **

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> rma.uni(effectsize, sei=s, mods=cbind(m_alc, c_alc, m_dvdays, c_dvdays,
m_sreport, c_sreport, permale, age), data=fulldata, method="DL")

Mixed-Effects Model (k = 438; tau^2 estimator: DL)

tau^2 (estimate of residual amount of heterogeneity): 0.0967
tau (sqrt of the estimate of residual heterogeneity): 0.3109

Test for Residual Heterogeneity:
QE(df = 429) = 1574.1047, p-val < .0001

Test of Moderators (coefficient(s) 2,3,4,5,6,7,8,9):
QM(df = 8) = 20.3610, p-val = 0.0091

Model Results:

           estimate      se     zval    pval    ci.lb   ci.ub
intrcpt      0.3436  0.3661   0.9386  0.3480  -0.3739  1.0610
m_alc        0.0910  0.0775   1.1739  0.2404  -0.0609  0.2428
c_alc       -0.0988  0.0769  -1.2849  0.1988  -0.2495  0.0519
m_dvdays    -0.0000  0.0002  -0.0247  0.9803  -0.0004  0.0004
c_dvdays     0.0006  0.0003   1.7635  0.0778  -0.0001  0.0012   .
m_sreport    0.0530  0.0755   0.7020  0.4827  -0.0950  0.2009
c_sreport    0.2303  0.0804   2.8637  0.0042   0.0727  0.3880  **
permale     -0.0014  0.0019  -0.7122  0.4764  -0.0051  0.0024
age         -0.0135  0.0168  -0.8042  0.4213  -0.0463  0.0194

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

>
>
> ##### Compare to naive results using study level mean effect sizes only #####
> badchoice <- read.csv('NeverUseThisModel.out')
> rma.uni(meaneffectsize, sei=means, mods=cbind(meanalc, meandvdays,
meansreport), data=badchoice, method="DL")

Mixed-Effects Model (k = 109; tau^2 estimator: DL)

tau^2 (estimate of residual amount of heterogeneity): 0.1101
tau (sqrt of the estimate of residual heterogeneity): 0.3318

Test for Residual Heterogeneity:
QE(df = 105) = 381.1302, p-val < .0001

Test of Moderators (coefficient(s) 2,3,4):
QM(df = 3) = 1.7928, p-val = 0.6165
```

```
Model Results:

             estimate      se     zval    pval    ci.lb   ci.ub
intrcpt        0.1909  0.1455   1.3122  0.1894  -0.0942  0.4760
meanalc        0.1247  0.0996   1.2527  0.2103  -0.0704  0.3199
meandvdays    -0.0001  0.0004  -0.2700  0.7872  -0.0010  0.0008
meansreport   -0.1024  0.1405  -0.7288  0.4661  -0.3778  0.1730

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> rma.uni(meaneffectsize, sei=means, mods=cbind(meanalc, meandvdays,
meansreport, permale, age), data=badchoice, method="DL")

Mixed-Effects Model (k = 109; tau^2 estimator: DL)

tau^2 (estimate of residual amount of heterogeneity): 0.1129
tau (sqrt of the estimate of residual heterogeneity): 0.3360

Test for Residual Heterogeneity:
QE(df = 103) = 379.8489, p-val < .0001

Test of Moderators (coefficient(s) 2,3,4,5,6):
QM(df = 5) = 1.9164, p-val = 0.8606

Model Results:

             estimate      se     zval    pval    ci.lb   ci.ub
intrcpt        0.2752  0.6481   0.4247  0.6711  -0.9950  1.5454
meanalc        0.1679  0.1481   1.1335  0.2570  -0.1224  0.4581
meandvdays    -0.0001  0.0005  -0.2908  0.7712  -0.0010  0.0008
meansreport   -0.1052  0.1427  -0.7376  0.4608  -0.3849  0.1744
permale        0.0007  0.0030   0.2455  0.8061  -0.0052  0.0067
age           -0.0087  0.0342  -0.2535  0.7999  -0.0757  0.0584

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1



> ##### Example 2: Handling Dependence Due to Multiple Studies within
Labs/Research Groups #####
> ##### Step 1: Load Working Directory, Functions #####
>
> setwd("F:/Cochrane-C2 10")
> robust.hier.se <- function(data,X.full) {
+ #this program assumes W = V^(-1) where V = data$vareffsize
+ p <- ncol(X.full)-2 # number of X variables other than intercept
+ N <- max(data$study)
+ sumXWX <- 0
+ sumXWy<- 0
+ sumXWJWX <- 0
+ sumXWVWX <- 0
```

```
+  sumXW.sig.m.v.WX <- 0
+  sumXWWX <- 0
+ sumXJX <- 0
+ sumXWJJX <- 0
+ sumXJJWX <- 0
+  tr.sumJJ <- 0
+  sumXJWX<- 0
+  sumXWJX <- 0
+
+ #for calcing Q1 and Qe:
+ for (i in (1: N)) {
+ tab <- data[data$study == i,]
+ W <- diag((1/tab$vareffsize),tab$k[1])
+ tab2 <- X.full[X.full$study==i,]
+ tab3 <- cbind(tab2[-c(1)])
+ X <- data.matrix(tab3)
+ dimnames(X) <- NULL
+
+ y <- cbind(tab$effectsize)
+ one <- cbind(rep(1,tab$k[1]))
+ J <- one%*%t(one)
+  sigma <- (sqrt(tab$vareffsize))%*%t(sqrt(tab$vareffsize))
+ vee <- diag(tab$vareffsize,tab$k[1])
+ SigmV <- sigma - vee
+
+ tr.sumJJ <- tr.sumJJ + sum(diag(J%*%J))
+ sumXJX <- sumXJX + t(X)%*%J%*%X
+ sumXWJJX <- sumXWJJX + t(X)%*%W%*%J%*%J%*%X
+ sumXJJWX <- sumXJJWX + t(X)%*%J%*%J%*%W%*%X
+ sumXWWX <- sumXWWX + t(X)%*%W%*%W%*%X
+ sumXWX <- sumXWX + t(X)%*%W%*%X
+ sumXWy <- sumXWy + t(X)%*%W%*%y
+ sumXWJWX <- sumXWJWX + t(X)%*%W%*%J%*%W%*%X
+ sumXWVWX <- sumXWVWX + t(X)%*%W%*%vee%*%W%*%X
+ sumXW.sig.m.v.WX <- sumXW.sig.m.v.WX + t(X)%*%W%*%SigmV%*%W%*%X
+
+ sumXJWX<- sumXJWX + t(X)%*%J%*%W%*%X
+  sumXWJX <- sumXWJX + t(X)%*%W%*%J%*%X
+ }
+
+ sumV <- sum(data$vareffsize)
+ W <- diag(1/data$vareffsize)
+ sumW <- sum(W)
+
+ #calc Qe = (y-xb)'W(y-xb)
+ b <- solve(sumXWX)%*%sumXWy
+ X <- data.matrix(X.full[-c(1)])
+ dimnames(X) <- NULL
+ data$pred <- X%*%b
+ data$e <- data$effectsize - data$pred
+ Qe <-t(data$e)%*%W%*%data$e
+
+ #calc Qa = (y-xb)'A(y-xb), A=diag(J)
+ sumEJE <- 0
+ for (i in (1: N)) {
```

```
+ tab <- data[data$study == i,]
+ e <- tab$e
+ one <- cbind(rep(1,tab$k[1]))
+ J <- one%*%t(one)
+ sumEJE <- sumEJE + t(e)%*%J%*%e
+ }
+
+ Qa <- sumEJE
+ V.i <- solve(sumXWX)
+
+ # Qe terms
+ A <- sumW - sum(diag(V.i%*%sumXWJWX))
+ B <- sumW - sum(diag(V.i%*%sumXWWX))
+ C <- length(data$k) - (p+1)
+
+ # Qa terms, Qa = y'(I-H)'J.s(I-H)y, A = J.s - H'J.s-J.sH + H'J.sH
+ a1 <- tr.sumJJ - sum(diag(V.i%*%sumXJJWX)) - sum(diag(V.i%*%sumXWJJX)) +
sum(diag(V.i%*%sumXJX%*%V.i%*%sumXWJWX))
+
+ b1 <- length(data$k) - sum(diag(V.i%*%sumXWJX)) - sum(diag(V.i%*%sumXJWX)) +
sum(diag(V.i%*%sumXJX%*%V.i%*%sumXWWX))
+
+ c1 <- sumV - sum(diag(V.i%*%sumXJX))
+
+ #est omega2 and tau2
+ omega.sq1 <- ((Qa - c1)*A - (Qe - C)*a1)/(b1*A - B*a1)
+  omega.sq <- ifelse(omega.sq1<0,0,omega.sq1)
+
+ tau.sq1 <- ((Qe-C)/A) - omega.sq1*(B/A)
+ tau.sq <- ifelse(tau.sq1<0,0,tau.sq1)
+
+ #now random effects model:
+ data$r.weights <- (1/(data$vareffsize + tau.sq + omega.sq))
+ sumXWX.r <- 0
+ sumXWy.r<- 0
+ for (i in (1: N)) {
+ tab <- data[data$study == i,]
+ W <- diag(tab$r.weights,tab$k[1])
+ tab2 <- X.full[X.full$study==i,]
+ tab3 <- cbind(tab2[-c(1)])
+ X <- data.matrix(tab3)
+ dimnames(X) <- NULL
+
+ y <- cbind(tab$effectsize)
+
+ sumXWX.r <- sumXWX.r + t(X)%*%W%*%X
+ sumXWy.r <- sumXWy.r + t(X)%*%W%*%y
+ }
+
+ b.r <- solve(sumXWX.r)%*%sumXWy.r
+ X <- data.matrix(X.full[-c(1)])
+ dimnames(X) <- NULL
+ data$pred.r <- X%*%b.r
+ data$e.r <- cbind(data$effectsize) - data$pred.r
+
```

```
+ sumXWeeWX.r <- 0
+ for (i in (1:N)) {
+ tab <- data[data$study== i,]
+ sigma.hat.r <- tab$e.r%*%t(tab$e.r)
+ W <- diag(tab$r.weights,tab$k[1])
+ tab2 <- X.full[X.full$study==i,]
+ tab3 <- cbind(tab2[-c(1)])
+ X <- data.matrix(tab3)
+ dimnames(X) <- NULL
+
+ sumXWeeWX.r <- sumXWeeWX.r + t(X)%*%W%*%sigma.hat.r%*%W%*%X
+ }
+
+ VR.r <- solve(sumXWX.r)%*%sumXWeeWX.r%*%solve(sumXWX.r)
+
+ SE <- c(rep(0,p+1))
+ t <- c(rep(0,p+1))
+ for (i in (1:(p+1))) {
+ SE[i] <-sqrt(VR.r[i,i])*sqrt(N/(N-(p+1)))
+ t[i] <- b.r[i]/SE[i]
+ }
+
+ labels <- c(colnames(X.full[2:length(X.full)]))
+ output <- cbind(labels,b.r,SE,t)
+ colnames(output) <- c("beta", "estimate","SE", "t-test")
+
+ list("Omega squared estimate" = omega.sq,
+ "Tau squared estimate" = tau.sq, "Estimates & Robust SE's"= output)
+ }
>
> ##### Step 2: Import Data #####
> ##### Option 1 is to use single data file and use subset statements when
reading data into R#
> ##### Note that you can import .out and .csv files using read.csv()
> ##### Or, you can import .dta, .sav, or .sas files using the foreign package
>
> fulldata <-read.csv('HierSingleDataset.out')
> library(foreign)
> fullstatadata <-read.dta('HierSingleDataset.dta')
> fullspssdata <-read.spss('HierSingleDataset.sav')
> fullsasdata <-read.ssd('HierSingleDataset.sas')
>
> ##### Option 2 is to use two data files (Data and Design)#####
> HierData <- read.csv('HierData.out')
> HierDesign <- read.csv('HierDesign.out')
>
> ##### Step 3: Run Regression Models #####
> ##### First Using Option 1 with Subset Statements with a Single Dataset #####
> ##### Model 1 Using Measurement Variables Only #####
> m1sdata <- subset(fulldata, select=c(2, 4:(length(fulldata)-4)))
> m1sX.full <- subset(fulldata, select=c(2:3, 9:(length(fulldata)-4)))
> robust.hier.se(m1sdata, m1sX.full)
$`Omega squared estimate`
          [,1]
[1,] 0.1173877
```

```
$`Tau squared estimate`
           [,1]
[1,] 0.01927008

$`Estimates & Robust SE's`
     beta        estimate              SE                   t-test
     [1,] "intercept" "-0.234330168081584"   "0.211527439702871"
     "-1.10780033271685"
[2,] "m_alc"     "0.565247766835447"    "0.334807573985549"
"1.6882765228598"
[3,] "c_alc"     "0.488968812973844"    "0.110742068304773"
"4.41538450977954"
[4,] "m_dvdays"  "0.00198029924178322"  "0.00151137893110917"
"1.31025992292344"
[5,] "c_dvdays"  "0.000744240665116459" "0.000245568524529953"
"3.03068427251019"
[6,] "m_sreport" "0.165845493069739"    "0.284413408323727"
"0.583114185956273"
[7,] "c_sreport" "0.543935991613606"    "0.166073003135117"
"3.27528244413729"

>
> ##### Model 2 Using All Variables #####
> m2sdata <- subset(fulldata, select=c(2, 4:length(fulldata)))
> m2sX.full <- subset(fulldata, select=c(2:3, 9:length(fulldata)))
> robust.hier.se(m2sdata, m2sX.full)
$`Omega squared estimate`
           [,1]
[1,] 0.04519917

$`Tau squared estimate`
          [,1]
[1,] 0.1488219

$`Estimates & Robust SE's`
      beta        estimate              SE                   t-test
       [1,] "intercept" "2.09630338024676"     "2.38089272910835"
      "0.880469478787409"
 [2,] "m_alc"     "0.646841036353826"    "0.329437959341352"
 "1.96346844075607"
 [3,] "c_alc"     "0.546337757870382"    "0.145793258913871"
 "3.74734580967928"
 [4,] "m_dvdays"  "0.00147104181915973"  "0.00207243642776259"
 "0.709812759249686"
 [5,] "c_dvdays"  "0.000718752688577079" "0.000341060363194616"
 "2.1074060962251"
 [6,] "m_sreport" "0.267109255155533"    "0.354788936408506"
 "0.75286805124042"
 [7,] "c_sreport" "0.560838496782526"    "0.251078397899464"
 "2.23371863718477"
 [8,] "m_permale" "-0.00392669090278498" "0.0123267233703778"
 "-0.318551068666079"
 [9,] "c_permale" "0.00347610338997933"  "0.00599649664918209"
 "0.579689040675686"
```

```
[10,] "m_age"     "-0.126421311187227"   "0.107957295317999"
"-1.1710307378009"
[11,] "c_age"     "0.000220363014701173" "0.111858371154197"
"0.00197001809008467"

>
> ##### Next Using Option 2 with Two Separate Datasets #####
> ##### Model 1 Using Measurement Variables Only #####
> m1data <- data.frame(HierData)
> m1X.full <- subset(HierDesign, select=c(1:8))
> robust.hier.se(m1data, m1X.full)
$`Omega squared estimate`
          [,1]
[1,] 0.1173877

$`Tau squared estimate`
           [,1]
[1,] 0.01927008

$`Estimates & Robust SE's`
     beta        estimate               SE                     t-test
     [1,] "intercept" "-0.234330168081584"   "0.211527439702871"
     "-1.10780033271685"
[2,] "m_alc"     "0.565247766835447"    "0.334807573985549"
"1.6882765228598"
[3,] "c_alc"     "0.488968812973844"    "0.110742068304773"
"4.41538450977954"
[4,] "m_dvdays"  "0.00198029924178322"  "0.00151137893110917"
"1.31025992292344"
[5,] "c_dvdays"  "0.000744240665116459" "0.000245568524529953"
"3.03068427251019"
[6,] "m_sreport" "0.165845493069739"    "0.284413408323727"
"0.583114185956273"
[7,] "c_sreport" "0.543935991613606"    "0.166073003135117"
"3.27528244413729"

>
> ##### Model 2 Using All Variables #####
> m2data <- data.frame(HierData)
> m2X.full <- data.frame(HierDesign)
> robust.hier.se(m2data, m2X.full)
$`Omega squared estimate`
           [,1]
[1,] 0.04519917

$`Tau squared estimate`
          [,1]
[1,] 0.1488219

$`Estimates & Robust SE's`
      beta        estimate               SE                     t-test
       [1,] "intercept" "2.09630338024676"     "2.38089272910835"
      "0.880469478787409"
 [2,] "m_alc"     "0.646841036353826"    "0.329437959341352"
 "1.96346844075607"
```

```
 [3,] "c_alc"     "0.546337757870382"    "0.145793258913871"
 "3.74734580967928"
 [4,] "m_dvdays"  "0.00147104181915973"  "0.00207243642776259"
 "0.709812759249686"
 [5,] "c_dvdays"  "0.000718752688577079" "0.000341060363194616"
 "2.1074060962251"
 [6,] "m_sreport" "0.267109255155533"    "0.354788936408506"
 "0.75286805124042"
 [7,] "c_sreport" "0.560838496782526"    "0.251078397899464"
 "2.23371863718477"
 [8,] "m_permale" "-0.00392669090278498" "0.0123267233703778"
 "-0.318551068666079"
 [9,] "c_permale" "0.00347610338997933"  "0.00599649664918209"
 "0.579689040675686"
[10,] "m_age"     "-0.126421311187227"   "0.107957295317999"
"-1.1710307378009"
[11,] "c_age"     "0.000220363014701173" "0.111858371154197"
"0.00197001809008467"

>
> ##### Compare to naive standard errors calculated in metafor package #####
> library(metafor)
> rma.uni(effectsize, sei=s, mods=cbind(m_alc, c_alc, m_dvdays, c_dvdays,
m_sreport, c_sreport), data=fulldata, method="DL")

Mixed-Effects Model (k = 68; tau^2 estimator: DL)

tau^2 (estimate of residual amount of heterogeneity): 0.1302
tau (sqrt of the estimate of residual heterogeneity): 0.3608

Test for Residual Heterogeneity:
QE(df = 61) = 224.5222, p-val < .0001

Test of Moderators (coefficient(s) 2,3,4,5,6,7):
QM(df = 6) = 46.1167, p-val < .0001

Model Results:

           estimate      se     zval    pval    ci.lb   ci.ub
intrcpt     -0.2328  0.2572  -0.9054  0.3653  -0.7369  0.2712
m_alc        0.5639  0.2843   1.9833  0.0473   0.0066  1.1212    *
c_alc        0.4862  0.1585   3.0666  0.0022   0.1755  0.7969   **
m_dvdays     0.0020  0.0014   1.4436  0.1489  -0.0007  0.0047
c_dvdays     0.0007  0.0006   1.3382  0.1808  -0.0003  0.0018
m_sreport    0.1637  0.3949   0.4146  0.6785  -0.6103  0.9378
c_sreport    0.5445  0.1437   3.7878  0.0002   0.2627  0.8262  ***

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> rma.uni(effectsize, sei=s, mods=cbind(m_alc, c_alc, m_dvdays, c_dvdays,
m_sreport, c_sreport, m_permale, c_permale, m_age, c_age), data=fulldata,
method="DL")

Mixed-Effects Model (k = 68; tau^2 estimator: DL)
```

```
tau^2 (estimate of residual amount of heterogeneity): 0.1162
tau (sqrt of the estimate of residual heterogeneity): 0.3409

Test for Residual Heterogeneity:
QE(df = 57) = 190.7503, p-val < .0001

Test of Moderators (coefficient(s) 2,3,4,5,6,7,8,9,10,11):
QM(df = 10) = 52.9615, p-val < .0001

Model Results:

           estimate      se     zval    pval    ci.lb   ci.ub
intrcpt      2.0863  2.1019   0.9926  0.3209  -2.0333  6.2060
m_alc        0.6503  0.2792   2.3289  0.0199   0.1030  1.1976    *
c_alc        0.5300  0.1600   3.3131  0.0009   0.2164  0.8435  ***
m_dvdays     0.0014  0.0014   1.0155  0.3099  -0.0013  0.0042
c_dvdays     0.0008  0.0005   1.3898  0.1646  -0.0003  0.0018
m_sreport    0.2557  0.4074   0.6276  0.5303  -0.5429  1.0542
c_sreport    0.5642  0.1424   3.9609  <.0001   0.2850  0.8434  ***
m_permale   -0.0031  0.0100  -0.3144  0.7532  -0.0227  0.0164
c_permale    0.0040  0.0034   1.1996  0.2303  -0.0026  0.0107
m_age       -0.1288  0.0926  -1.3901  0.1645  -0.3104  0.0528
c_age        0.0073  0.0506   0.1437  0.8857  -0.0919  0.1065

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

>
>
> ##### Compare to naive results using study level mean effect sizes only #####
> badchoice <- read.csv('HierNeverUseThisModel.out')
> rma.uni(meaneffectsize, sei=means, mods=cbind(meanalc, meandvdays,
meansreport), data=badchoice, method="DL")

Mixed-Effects Model (k = 15; tau^2 estimator: DL)

tau^2 (estimate of residual amount of heterogeneity): 0
tau (sqrt of the estimate of residual heterogeneity): 0

Test for Residual Heterogeneity:
QE(df = 11) = 10.9963, p-val = 0.4436

Test of Moderators (coefficient(s) 2,3,4):
QM(df = 3) = 16.3747, p-val = 0.0010

Model Results:

             estimate      se     zval    pval    ci.lb   ci.ub
intrcpt       -0.1420  0.2269  -0.6261  0.5312  -0.5867  0.3026
meanalc        0.5972  0.2586   2.3089  0.0210   0.0902  1.1041  *
meandvdays     0.0013  0.0012   1.0270  0.3044  -0.0011  0.0037
meansreport    0.1619  0.3365   0.4811  0.6304  -0.4976  0.8214

---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> rma.uni(meaneffectsize, sei=means, mods=cbind(meanalc, meandvdays,
meansreport, meanpermale, meanage), data=badchoice, method="DL")

Mixed-Effects Model (k = 15; tau^2 estimator: DL)

tau^2 (estimate of residual amount of heterogeneity): 0.0009
tau (sqrt of the estimate of residual heterogeneity): 0.0293

Test for Residual Heterogeneity:
QE(df = 9) = 9.1212, p-val = 0.4262

Test of Moderators (coefficient(s) 2,3,4,5,6):
QM(df = 5) = 18.0651, p-val = 0.0029

Model Results:

             estimate      se     zval    pval    ci.lb   ci.ub
intrcpt        0.0950  2.4470   0.0388  0.9690  -4.7010  4.8911
meanalc        0.6061  0.2666   2.2733  0.0230   0.0835  1.1286  *
meandvdays     0.0009  0.0013   0.7139  0.4753  -0.0017  0.0036
meansreport    0.3006  0.3620   0.8304  0.4063  -0.4089  1.0100
meanpermale    0.0073  0.0102   0.7126  0.4761  -0.0127  0.0272
meanage       -0.0502  0.1130  -0.4440  0.6570  -0.2716  0.1712

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```