Supplemental material 7 - Statistical code for RStudio version 1.4.1717

```
####LOAD REQUISITE PACKAGES####
if (!require("pacman")) install.packages("pacman")
pacman::p_load(readxl, plyr, robumeta,metafor, clubSandwich, ggplot2, psych, devtools)


####LOAD AND PREPARE DATA####

meta_data<-
read_excel("DataExtractionForm_Exercise,CogFunc&Dementia_SR&MA_NoFW_ForMeta-
Analysis_20210513.xlsx")

spaceless <- function(x) {colnames(x) <- gsub(" ", "_", colnames(x));x}

meta_data_spaceless <- spaceless(meta_data)

colnames(meta_data_spaceless)

meta_data2 <- rename(meta_data_spaceless, c("Author_Year"="Study",
"Study_Name_(Acronym)"="Acronym", "Parent_Study=1;_Secondary=2"="Study_P1_S2",
                                    "Randomization_level_(Individual,_Cluster)"="Randomization",

"Type_of_dementia_(AD,_VCI,_Mixed_types_of_dementia,_unspecified_dementia)"="Dementi
a_Type",

"Dementia_severity_(Mild,_Mixed_severity,_Unspecified)"="Dementia_Severity",
                                    "Outcome_Type_(Cognitive_domain)"="Cognitive_Domain",
                                    "Outcome_(Neuropsychological_test)"="Neuropsych_Test",
                                    "Participants_(Community-dwelling=1;_Living_in_long-
term_care_facilities=2;_Both=3)"="Participants",
                                    "Total_Sample_Size"="Total_N","%Female"="Pct_Female",
                                    "Type_of_analyses_(Intent-to-treat,_Per-protocol,_Complete-
case,_Not_reported)"="ITT_PPT_CC",
                                    "Intervention_Mode__(AT;_RT;_MT;_Other)"="INT_Type",
                                    "Intervention_Length__(weeks)"="INT_Length",

"Intervention_Session_Duration__(min)"="INT_SessionDuration",
                                    "Intervention_Exercise_Frequency__(days/week)"="INT_Freq",

"Intervention_Intensity_(Low,_Moderate,_High,_Mixed_intensity,_Unspecified)"="INT_Intensi
ty",
                                    "Intervention_%_Adherence"="INT_Adherence",
```

```
"Control_Group_(Usual_care;_Cognitive_training;_Active_training;_Other_activities)"="CON_
Type",
                                "Control_Group_Length_(weeks)"="CON_Length",

"Control_Group_Session_Duration__(min)"="CON_SessionDuration",
                                "Control_Group_Frequency__(days/week)"="CON_Freq",
                                "Control_%Adherence"="CON_Adherence",
                                "RoB2_-_Domain_1_or_1a_(Randomization)"="Bias_Rand",
                                "RoB2_-_Domain_1b_(Cluster_-
_Timing_recruitment_and_randomization)"="Bias_Timing_Cluster",
                                "RoB2_-
_Domain_2_(Deviation_from_intended_intervention)"="Bias_Int_Dev",
                                "RoB2_-_Domain_3_(Missing)"="Bias_Missing",
                                "RoB2_-_Domain_4_(Outcome_measurement)"="Bias_Measure",
                                "RoB2_-_Domain_5_(Selected_result)"="Bias_SelResult",
                                "Overall_Risk_of_Bias"="Bias_Overall",
                                "Intervention_Group_Size"="INT_n",
                                "Control_Group_Size"="CON_n",
                                "Intervention_Mean_Baseline"="INT_Mean_Pre",
                                "Intervention_SD_Baseline"="INT_SD_Pre",
                                "Intervention_Mean_Follow-Up"="INT_Mean_Post",
                                "Intervention_SD_Follow-up"="INT_SD_Post",
                                "Control_Mean_Baseline"="CON_Mean_Pre",
                                "Control_SD_Baseline"="CON_SD_Pre",
                                "Control_Mean_Follow-up"="CON_Mean_Post",
                                "Control_SD_Follow-up"="CON_SD_Post",

"Effect_Sizes_(Standardized_towards_expected_direction)"="SMD",

"Effect_Size_Calculated_(1=_All_info_present_(M,SD,n);_2=_median_data;_3=_ES_Calculate
d_by_paper,_4=Change_score,_5=Contacted_the_authors,_6=More_than_one_method)"="How
_d_Calculated"))

colnames(meta_data2)

meta_data_final<-subset(meta_data2, select = -c(Acronym, Country))


####RENAMING STUDIES WITH MORE THAN ONE INTERVENTION OR CONTROL
GROUP#####

colnames(meta_data_final)

meta_data_final$Study <- ifelse(meta_data_final$Study == "Enette et al. (2020)" &
meta_data_final$AT_or_MT_Type=="Continuous",
```

```
                    "Enette et al. (2020) (Continuous AT)", meta_data_final$Study)

meta_data_final$Study <- ifelse(meta_data_final$Study == "Enette et al. (2020)" &
meta_data_final$AT_or_MT_Type=="Intermittent",
                        "Enette et al. (2020) (Intermittent AT)", meta_data_final$Study)

meta_data_final$Study <- ifelse(meta_data_final$Study == "Sanders et al. (2020)" &
meta_data_final$AT_or_MT_Type=="Low intensity",
                       "Sanders et al. (2020) (Low-intensity)", meta_data_final$Study)

meta_data_final$Study <- ifelse(meta_data_final$Study == "Sanders et al. (2020)" &
meta_data_final$AT_or_MT_Type=="High intensity",
                       "Sanders et al. (2020) (Low- and high-intensity)", meta_data_final$Study)

table(meta_data_final$Study)

meta_data_final$Study_INT_CON<-paste(meta_data_final$Study, meta_data_final$INT_Type,
                        meta_data_final$CON_Type, sep=' ')

meta_data_final$Study_INT_CON

####HEDGE'S G CALCULATION####

meta_data_final$var.d <- (((meta_data_final$INT_n + meta_data_final$CON_n) /
(meta_data_final$INT_n * meta_data_final$CON_n)) +
                ((meta_data_final$SMD^2) / (2*(meta_data_final$INT_n +
meta_data_final$CON_n))))

meta_data_final$J <- 1 - (3/(4 * (meta_data_final$INT_n + meta_data_final$CON_n -2) - 1))

meta_data_final$hedge.g <- meta_data_final$SMD * meta_data_final$J
meta_data_final$hedge.g

meta_data_final$var.g <- meta_data_final$J^2 * meta_data_final$var.d

meta_data_final$se.g<-
sqrt(meta_data_final$var.g/(meta_data_final$INT_n+meta_data_final$CON_n))

table(meta_data_final$Study, meta_data_final$INT_Type)
count(meta_data_final$INT_Type=="AT")

describeBy(meta_data_final$hedge.g)

describeBy(meta_data_final$hedge.g, meta_data_final$Cognitive_Domain)

####SUBSETTING BY INTERVENTION AND CONTROL GROUP######
```

```
subset.meta_data_final.wide.1<-ddply(meta_data_final, "INT_Type", transform, time =
seq_along(Study))

subset.meta_data_final.wide<-
reshape(subset.meta_data_final.wide.1,idvar="Study_INT_CON",timevar="time",
                    direction="wide",v.names=c("hedge.g","var.g"))

View(subset.meta_data_final.wide) # Looks fine (Bossers, Ennette, and Ho [2 Ex int]
                        # with 2 rows each, and Cheng a, Cheng b, Fonte, Henskens
                        # with 2 rows each [2 Con])

colnames(subset.meta_data_final.wide)

# Averaging Hedges.g for each outcome #

library(openxlsx)
write.xlsx(subset.meta_data_final.wide, "subset.meta_data_final.wide.xlsx")

subset.meta_data_final.wide$cog.var<-
rowMeans(subset.meta_data_final.wide[c(54,56,58,60,62,64,66,68,
                                       70,72,74,76,78,80,82,84,
                                       86,88,90,92,94,96,98,100,
                                       102,104,106,108,110,112,
                                       114,116,118,120,122,124,
                                       126,128,130,132,134,136,
                                       138,140,142,144,146,148,
                                       150,152,154,156,158,160,
                                       162,164,166,168,170,172,
                                       174,176,178,180,182,184,
                                       186,188,190,192,194,196,
                                       198,200,202,204,206,208,
                                       210,212,214,216)],na.rm=TRUE)

subset.meta_data_final.wide$cog.g<-
rowMeans(subset.meta_data_final.wide[c(55,57,59,61,63,65,67,
                                       69,71,73,75,77,79,81,
                                       83,85,87,89,91,93,95,
                                       97,99,101,103,105,107,
                                       109,111,113,115,117,
                                       119,121,123,125,127,
                                       129,131,133,135,137,
                                       139,141,143,145,147,
                                       149,151,153,155,157,
                                       159,161,163,165,167,169,
                                       171,173,175,177,179,181,
```

```
                                        183,185,187,189,191,193,
                                        195,197,199,201,203,205,
                                        207,209,211,213,215)],na.rm=TRUE)


###### OVERALL META-ANALYSIS ######

## Overall forest plot ##

View(meta_data_final)
overall.g<-robu(hedge.g~1,var=var.g,studynum=Study_INT_CON,meta_data_final) #Clustered
at level of overall study
print(overall.g)
sensitivity(overall.g)
forest.plot<-forest.robu(overall.g,es.lab="Neuropsych_Test",study.lab="Study_INT_CON",
                         "hedge.g" = effect.size)


## Type of dementia specific effect sizes ##

# Subsetting data #

subset.AD<-subset(meta_data_final, Dementia_Type=="1")
subset.VCI<-subset(meta_data_final, Dementia_Type=="2")
subset.Mixed<-subset(meta_data_final, Dementia_Type=="3")
subset.Unspecified<-subset(meta_data_final, Dementia_Type=="4")

## Dementia type-specific forest plots ##

# AD

describeBy(subset.AD$hedge.g, subset.AD$Cognitive_Domain)
describe(subset.AD$hedge.g)

ADOverall.g<-robu(hedge.g~1,var=var.g,studynum=Study_INT_CON,subset.AD)
print(ADOverall.g)
forest.AD<-forest.robu(ADOverall.g,es.lab="Neuropsych_Test",study.lab="Study_INT_CON",
                           "hedge.g" = effect.size)

# VCI

describeBy(subset.VCI$hedge.g, subset.VCI$Cognitive_Domain)
describe(subset.VCI$hedge.g)

VCIOverall.g<-robu(hedge.g~1,var=var.g,studynum=Study_INT_CON,subset.VCI)
print(VCIOverall.g)
```

```
forest.VCI<-forest.robu(VCIOverall.g,es.lab="Neuropsych_Test",study.lab="Study_INT_CON",
                        "hedge.g" = effect.size)

# Mixed types of dementia

describeBy(subset.Mixed$hedge.g, subset.Mixed$Cognitive_Domain)
describe(subset.Mixed$hedge.g)

MixedOverall.g<-robu(hedge.g~1,var=var.g,studynum=Study_INT_CON,subset.Mixed)
print(MixedOverall.g)
forest.mixed<-
forest.robu(MixedOverall.g,es.lab="Neuropsych_Test",study.lab="Study_INT_CON",
                        "hedge.g" = effect.size)

# Unspecified dementia

describeBy(subset.Unspecified$hedge.g, subset.Unspecified$Cognitive_Domain)
describe(subset.Unspecified$hedge.g)

UnspecifiedOverall.g<-
robu(hedge.g~1,var=var.g,studynum=Study_INT_CON,subset.Unspecified)
print(UnspecifiedOverall.g)
forest.unspecified<-
forest.robu(UnspecifiedOverall.g,es.lab="Neuropsych_Test",study.lab="Study_INT_CON",
                        "hedge.g" = effect.size)

#### LEAVE ONE OUT ANALYSIS + Z SCORES FOR ALL-CAUSE DEMENTIA AND
OVERALL COGNTIVE FUNCTION ####


leave_out_cog<-robu(hedge.g~1,var=var.g,studynum=Study_INT_CON,meta_data_final)
print(leave_out_cog)

leave.one.out <- function(rownum){

  # subset dataframe
  vec <- rep(TRUE, nrow(meta_data_final))
  vec[rownum] <- FALSE
  inf <- subset(meta_data_final, vec)

  # fit reference model
  rve.model.ref <- robu(hedge.g~1,var=var.g,studynum=Study_INT_CON,meta_data_final)

  # fit adjusted model
  rve.model <- robu(hedge.g~1,var=var.g,studynum=Study_INT_CON, data = inf)
```

```
 # return results
 results.vec <- c(g.study = round(meta_data_final$hedge.g[rownum], 3),
                  g.adj = round(rve.model$reg_table$b.r, 3),
                  se.g.adj = round(rve.model$reg_table$SE, 3),
                  p.adj = round(rve.model$reg_table$prob, 5),
                  diff.g = round(rve.model$reg_table$b.r - rve.model.ref$reg_table$b.r, 3),
                  diff.se.g = round(rve.model$reg_table$SE - rve.model.ref$reg_table$SE, 3),
                  diff.p = round(rve.model$reg_table$prob - rve.model.ref$reg_table$prob, 5),
                  diff.isq = round(rve.model$mod_info$I.2 - rve.model.ref$mod_info$I.2, 3))
 return(results.vec)
}

inf.cog <- sapply(1:nrow(meta_data_final), leave.one.out)
inf.cog <- data.frame(t(inf.cog))
inf.cog <- data.frame(label = paste(meta_data_final$Study_INT_CON,
meta_data_final$Cognitive_Domain, sep = " / "),
                      inf.cog)
inf.cog$z.g.study <- round(scale(inf.cog$g.study), 2)
inf.cog$g.min.sumeff.div.sdg <- round((meta_data_final$hedge.g -
leave_out_cog$reg_table$b.r) / sd(meta_data_final$hedge.g), 2)

View(inf.cog)

####ROBU MODERATION ANALYSES####

# note: All models contain all effect sizes, no outliers have been removed here.

# Convert Intervention frequency, intervention duration, and control length from character to
numeric
colnames(meta_data_final)

meta_data_final[, c(22,23,27,29)] <- sapply(meta_data_final[, c(22,23,27,29)], as.numeric)

meta_data_final_noLamb<-subset(meta_data_final, StudyID!="18") #Excluding Lamb b/c was
impossible to categorize duration and frequency


# Categorize intervention length to Northey MA

meta_data_final_noLamb$INT_Length_Cat<-NA
meta_data_final_noLamb$INT_Length_Cat[meta_data_final_noLamb$INT_Length>=8 &
meta_data_final_noLamb$INT_Length<=12]<-"Short"
meta_data_final_noLamb$INT_Length_Cat[meta_data_final_noLamb$INT_Length>12 &
meta_data_final_noLamb$INT_Length<=26]<-"Medium"
meta_data_final_noLamb$INT_Length_Cat[meta_data_final_noLamb$INT_Length>26]<-
"Long"
```

```
table(meta_data_final_noLamb$Study, meta_data_final_noLamb$INT_Length_Cat)

# Create volume variable (frequency*session duration) and categorize

meta_data_final_noLamb$INT_Volume <-
meta_data_final_noLamb$INT_Freq*meta_data_final_noLamb$INT_SessionDuration
View(meta_data_final_noLamb$INT_Volume)

# Categorize type of dementia

meta_data_final$Dementia_Type_Cat <- NA
meta_data_final$Dementia_Type_Cat[meta_data_final$Dementia_Type==1] <- "AD"
meta_data_final$Dementia_Type_Cat[meta_data_final$Dementia_Type==2] <- "VCI"
meta_data_final$Dementia_Type_Cat[meta_data_final$Dementia_Type==3] <- "Mixed"
meta_data_final$Dementia_Type_Cat[meta_data_final$Dementia_Type==4] <- "Unspecified"


# Categorize cog domain

meta_data_final$Cognitive_Domain_Cat <- NA
meta_data_final$Cognitive_Domain_Cat[meta_data_final$Cognitive_Domain==1] <- "Global
cognition"
meta_data_final$Cognitive_Domain_Cat[meta_data_final$Cognitive_Domain==2] <-
"Memory"
meta_data_final$Cognitive_Domain_Cat[meta_data_final$Cognitive_Domain==3] <-
"Executive function"
meta_data_final$Cognitive_Domain_Cat[meta_data_final$Cognitive_Domain==4] <-
"Processing speed"

# Categorize overall bias

meta_data_final$Bias_Overall_Cat <- NA
meta_data_final$Bias_Overall_Cat[meta_data_final$Bias_Overall==1] <- "Low"
meta_data_final$Bias_Overall_Cat[meta_data_final$Bias_Overall==2] <- "Some concerns"
meta_data_final$Bias_Overall_Cat[meta_data_final$Bias_Overall==3] <- "High"

# Categorize sex into >65% and <65% female (median split)

meta_data_final$Pct_Female_Cat<-NA
meta_data_final$Pct_Female_Cat[meta_data_final$Pct_Female<=65]<-"Low_Female"
meta_data_final$Pct_Female_Cat[meta_data_final$Pct_Female>65]<-"High_Female"

table(meta_data_final$Study, meta_data_final$Pct_Female_Cat)

## MODERATION ANALYSES ##
```

```
moderation.models <- list(

  # this list contains all moderation models with intercepts

  # Treatment-Level Moderator
  typetreat = robu(hedge.g~INT_Type, var.eff.size = var.g, studynum = Study_INT_CON, data =
meta_data_final),  ## Type of Training
  lengthtreat_cat = robu(hedge.g~INT_Length_Cat, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final_noLamb),  ## Length of Training Cat
  intensitytreat = robu(hedge.g~INT_Intensity, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),  ## Intensity of Training
  adheretreat = robu(hedge.g~INT_Adherence, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),  ## Training adherence
  volumetreat = robu(hedge.g~INT_Volume, var.eff.size = var.g, studynum = Study_INT_CON,
data = meta_data_final_noLamb),  ## Frequency of Training

  # Study-level Moderator
  typedementia = robu(hedge.g~Dementia_Type_Cat, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),
  overallbias = robu(hedge.g~Bias_Overall_Cat, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),

  # Participant-Level Moderators
  age = robu(hedge.g~scale(Mean_Age, scale= FALSE), var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),  ## Mean age
  mmse = robu(hedge.g~Mean_MMSE, var.eff.size = var.g, studynum = Study_INT_CON, data
= meta_data_final),  ## Mean MMSE
  pct.female_cat = robu(hedge.g~Pct_Female_Cat, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),  ## Pct female cat at 65%

  # Outcome-level Moderator
  cogdomain = robu(hedge.g~Cognitive_Domain_Cat,var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final)
)

moderation.models

# fit no-intercept models

moderation.models.noint <- list(   # this list contains all moderation models without intercepts

  # Treatment-Level Moderator
  typetreat = robu(hedge.g~INT_Type-1, var.eff.size = var.g, studynum = Study_INT_CON, data
= meta_data_final),  ## Type of Training
  lengthtreat_cat = robu(hedge.g~INT_Length_Cat-1, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final_noLamb),  ## Length of Training Cat
```

```
  intensitytreat = robu(hedge.g~INT_Intensity-1, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),  ## Intensity of Training
  adheretreat = robu(hedge.g~INT_Adherence, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),  ## Training adherence
  volumetreat = robu(hedge.g~INT_Volume, var.eff.size = var.g, studynum = Study_INT_CON,
data = meta_data_final_noLamb),  ## Volume of Training

  # Study-level Moderator
  typedementia = robu(hedge.g~Dementia_Type_Cat-1, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),
  overallbias = robu(hedge.g~Bias_Overall_Cat-1, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),

  # Participant-Level Moderators
  age = robu(hedge.g~scale(Mean_Age, scale= FALSE), var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),  ## Mean age
  mmse = robu(hedge.g~Mean_MMSE, var.eff.size = var.g, studynum = Study_INT_CON, data
= meta_data_final),  ## Mean MMSE
  pct.female_cat = robu(hedge.g~Pct_Female_Cat-1, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final),

  # Outcome-level Moderator
  cogdomain = robu(hedge.g~Cognitive_Domain_Cat-1,var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final)
)

moderation.models.noint


#### Run HTZ tests for categorical moderators ####

constraints = sapply(unname(unlist(lapply(moderation.models, function(x)return(x$p))))+1, seq,
from = 2)  # retrieve positions of coefficients to be tested in the HTZ tests
constraints


  # Treatment-Level Moderator
typetreat_htz = robu(hedge.g~INT_Type, var.eff.size = var.g, studynum = Study_INT_CON,
data = meta_data_final)  ## Type of Training
Wald_test(typetreat_htz,constraints = constrain_zero(2:4), vcov ="CR2")

lengthtreat_cat_htz = robu(hedge.g~INT_Length_Cat, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final_noLamb)  ## Lenght of Training
Wald_test(lengthtreat_cat_htz,constraints = constrain_zero(2:3), vcov ="CR2")
```

```
intensitytreat_htz = robu(hedge.g~INT_Intensity, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final)  ## Intensity of Training
Wald_test(intensitytreat_htz,constraints = constrain_zero(2:3), vcov ="CR2")

dementiatype_htz = robu(hedge.g~Dementia_Type_Cat, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final)  ## Type of dementia
Wald_test(dementiatype_htz,constraints = constrain_zero(2:4), vcov ="CR2")

cogdomain_htz = robu(hedge.g~Cognitive_Domain_Cat, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final)  ## Cog domaing
Wald_test(cogdomain_htz,constraints = constrain_zero(2:4), vcov ="CR2")

overallbias_htz = robu(hedge.g~Bias_Overall_Cat, var.eff.size = var.g, studynum =
Study_INT_CON, data = meta_data_final)  ## Cog domaing
Wald_test(overallbias_htz,constraints = constrain_zero(2:3), vcov ="CR2")



##############################################
# Small-study Effects and Publication Bias #
##############################################

# Egger's Regression Test
## for dependent effect sizes
pet.rve <- robu(hedge.g~se.g, var.eff.size = var.g, studynum = Study_INT_CON, data =
meta_data_final)
print(pet.rve)


# Funnel Plot
meta.res<-
rma.mv(yi=hedge.g,V=var.g,random=~factor(Study_INT_CON)|Study,data=meta_data_final,sla
b=Study)

funnel(meta.res, xlab = "Hedges' g for Cognitive Outcomes")

# Precision-effect estimate with standard errors - PEESE

meta_data_final$sesqrd.g<-meta_data_final$se.g^2

peese.rve <- robu(hedge.g~sesqrd.g, var.eff.size = var.g, studynum = Study_INT_CON, data =
meta_data_final)
print(peese.rve)
```