**R function for Calculating ICCs and averaged measurement ICCs with 95% CIs**

library(haven)

library(dplyr)

library(VCA)

library(stringr)

library(readxl)

##### R function for calculating ICCs and averaged measurement ICCs and their ##### 95% CIs

# nrep: numbers of repeated measurements for which you want to calculate the # averaged measurement ICCs

# subject: variable names for each subject(id), e.g. subject = "id\_freq"

# measurement: variable names for the outcome

# RandomEffect (NOTE: must be "factor"): variable names for random effects: # e.g. ear side, measurement day # RandomEffect=c("ear\_side","meas\_day")

# FixedEffect: variable names for fixed effects: e.g. ambient noise level # FixedEffect = "meanAmbientNoise"

# digit: number of digits you want to display

# data(NOTE: must be "data.frame" and dataset must be in long format): names # of the dataset

# Output in addition to the ICCs and their CIs:

# VC: variance components (e.g. sigma\_X^2, sigma^2)

# CV[%]: coefficient of variance[%]

ICC<-function(nrep=1,

subject = NULL,

measurement = NULL,

RandomEffect = NULL,

FixedEffect = NULL, # e.g. ambient noise level

digit = 2,

data){

fixed <- paste(FixedEffect, collapse = "+")

random <- ifelse(is.null(RandomEffect),"",paste("(", RandomEffect, ":", subject, ")", sep = "", collapse = "+"))

subject <- paste("(",subject,")",sep="")

form <- paste(threshold, "~", subject,"+",fixed, "+", random, sep = "")

form <- gsub("\\++$","",form)

form <- gsub("\\+\\+","+",form)

form <- as.formula(form)

fit <- remlMM(form, data)

# B: Estimates of squared sigmas

B <- fit$aov.tab[-1,2]

# var\_cov: Variance and covariance matrix

var\_cov <- fit$VarCov

var\_cov <- as.matrix(var\_cov)

B[is.na(B)] <- 0

var\_cov[is.na(var\_cov)] <- 0

print("The formula of the fitted model:")

print(form)

aov <- fit$aov.tab[,c(2,5,6)]

print(aov)

print("The variance and covariance matrix for the sigmas^2:")

print(fit$VarCov)

i=1

for(i in 1:nrep){

icc <- if(length(RandomEffect)==0)

{icc2(i,B,var\_cov,digit)}

else if(length(RandomEffect)==1)

{icc4(i,B,var\_cov,digit)}

else if(length(RandomEffect)==2)

{icc6(i,B,var\_cov,digit)}

else {print("error")}

print(paste0("ICC for ", i, " repeated measure and its 95% CI: ", icc))

i=i+1

}

}

#icc.format: function for formatting 95% CI of ICCs

icc.format<-function(icc,var,digit){

my.round<-function(x,n=digit){format(round(x, n), nsmall = n)}

icc\_p <- my.round(icc)

icc\_ll <- my.round(1/(1/icc + 1.96\* sqrt(var)))

icc\_ul <- my.round(1/(1/icc - 1.96\* sqrt(var)))

return(paste0(icc\_p,"(",icc\_ll,",",icc\_ul,")"))

}

#icc2: function for icc2 (see the manuscript)

icc2<-function(n,B,var\_cov,digit){

sigma\_sqx = B[1]

sigma\_sq = B[2]

der1n = -(sigma\_sq/n)/(sigma\_sqx^2)

der2n = (1/n)/sigma\_sqx

B\_dern <- as.matrix(c(der1n,der2n),nrow = 2)

icc <- sigma\_sqx/(sigma\_sqx+sigma\_sq/n)

var\_n = t(B\_dern)%\*%var\_cov%\*%B\_dern

return(icc.format(icc,var\_n,digit))

}

#icc4: function for icc4 (see the manuscript)

icc4<-function(n,B,var\_cov,digit){

sigma\_sqx = B[1]

sigma\_sqy = B[2]

sigma\_sq = B[3]

B\_dern <- as.matrix(c(-(sigma\_sq/n)/(sigma\_sqx+sigma\_sqy)^2,

-(sigma\_sq/n)/(sigma\_sqx+sigma\_sqy)^2,

(1/n)/(sigma\_sqx+sigma\_sqy)),nrow=3)

icc = (sigma\_sqx+sigma\_sqy)/(sigma\_sqx+sigma\_sqy+sigma\_sq/n)

var\_n = t(B\_dern)%\*%var\_cov%\*%B\_dern

return(icc.format(icc,var\_n,digit))

}

#icc6: function for icc6 (see the manuscript)

icc6<-function(n,B,var\_cov,digit){

sigma\_sqx = B[1]

sigma\_sqy = B[2]

sigma\_sqt = B[3]

sigma\_sq = B[4]

B\_dern <- as.matrix(c(-(sigma\_sq/n)/(sigma\_sqx+sigma\_sqy+sigma\_sqt)^2,

-(sigma\_sq/n)/(sigma\_sqx+sigma\_sqy+sigma\_sqt)^2,

-(sigma\_sq/n)/(sigma\_sqx+sigma\_sqy+sigma\_sqt)^2,

(1/n)/(sigma\_sqx+sigma\_sqy+sigma\_sqt)),nrow=4)

icc = (sigma\_sqx+sigma\_sqy+sigma\_sqt)/(sigma\_sqx+sigma\_sqy+sigma\_sqt+sigma\_sq/n)

var\_n = t(B\_dern)%\*%var\_cov%\*%B\_dern

return(icc.format(icc,var\_n,digit))

}

#### Examples of calling this function

htdata <- read\_sas("Path for your data")

htdata <- as.data.frame(htdata) %>%

mutate(id\_freq=as.factor(id\_freq),

meas\_day=as.factor(meas\_day))

# Within-participant ICC

htdata\_filtered <- htdata %>%

filter(frequency==1000&reliable==1&ear\_side=="Left")

# without ambient noise and the interaction term as covariates

ICC(nrep = 4,subject = "id\_freq", measurement = "threshold", RandomEffect = NULL, FixedEffect = NULL, digit = 2, data = htdata\_filtered)

# with ambient noise and the interaction term as covariates

ICC(nrep = 4,subject = "id\_freq", measurement = "threshold", RandomEffect = NULL, FixedEffect = "meanAmbientNoise", digit = 2, data = htdata\_filtered)

# Within-participant, within-ear side ICC

htdata\_filtered <- htdata %>%

filter(frequency==1000&reliable\_bothears==1)

ICC(nrep = 4,subject = "id\_freq", measurement = "threshold", RandomEffect = "ear\_side", FixedEffect = NULL, digit = 2, data = htdata\_filtered)

# Within-participant, within-ear side, within-measurement days ICC

htdata\_filtered <- htdata %>%

filter(frequency==1000&reliable\_bothears==1)

ICC(nrep = 4,subject = "id\_freq", measurement = "threshold", RandomEffect = c("ear\_side","meas\_day"),

FixedEffect = NULL, digit = 2, data = htdata\_filtered)