# Supplementary Material

## Supplementary Material A: R Code for Model Depicted in Table 3

install.packages("simr", dependencies = TRUE, repos='http://cran.rstudio.com/')

library(simr)

set.seed(123)

#### Specification of Input Parameters ####

### Specification of standardized input parameters ###

L1\_DE\_standardized <- .30 ## standardized L1 direct effect

L2\_DE\_standardized <- .30 ## standardized L2 direct effect

CLI\_E\_standardized = .50 ## standardized CLI effect

rand.sl = .09 ## standardized random slope variance component

ICC <- .30 ## standardized intraclass correlation coefficient

cor.i.sl = .00 ## Correlation between random slope and random intercept

alpha.S <- .05 ## significance level

Size.clus <- 20 ## L1 sample size (cluster size)

N.clus <- 40 ## L2 sample size (number of clusters)

### Derivation of a population model for the power analysis ###

## Specification of predictor variables ##

x <- scale(rep(1:Size.clus))

g <- as.factor(1:N.clus)

X <- cbind(expand.grid("x"=x, "g"=g))

X <- data.frame(X, Z=as.numeric(X$g))

X$Z <- scale(X$Z)

## Specification of the outcome variable ##

varL1 <- 1 ## uncond. L1 variance (fixed at 1)

s <- sqrt((varL1)\*(1-(L1\_DE\_standardized^2))) ## cond. L1 variance (Equation 11)

varL2 <- ICC/(1-ICC) ## uncond. L2 variance (Equation 10)

V1 <- varL2\*(1-(L2\_DE\_standardized^2)) ## cond. L2 variance (Equation 11)

## Adjustment of the random slope variance ##

varRS <- rand.sl\*varL1 ## uncond. random slope variance (Equation 13)

condRS <- varRS\*(1-(CLI\_E\_standardized^2)) ## cond. Random slope variance (Equation 11)

## Covariance of random intercept and slope (Equation 14) ##

cov.i.sl <- cor.i.sl\*sqrt(varL2)\*sqrt(varRS)

## Adjustment of fixed effects (Equation 15) ##

L1\_DE <- L1\_DE\_standardized\*sqrt(varL1)

L2\_DE <- L2\_DE\_standardized\*sqrt(varL2)

CLI\_E <- CLI\_E\_standardized\*sqrt(varRS)

#### Implementation of a Power Analysis in a Two-Level Model in SIMR ####

### Vector of fixed effects ###

b <- c(0, L1\_DE, L2\_DE, CLI\_E)

### Random intercept/slope variance-covariance matrix ###

V2 <- matrix(c(V1, cov.i.sl, cov.i.sl, rand\_sl.con), 2)

### Setting up the population model ###

model <- makeLmer(y ~ x + Z + x:Z + (x|g),

fixef=b, VarCorr=V2,

sigma=s, data=X)

print(model)

### Simulating power for the L1 direct effect ###

sim.ef <- powerSim(model,

fixed("x","kr"),

alpha=alpha.S,

nsim=1000)

print(sim.ef)

simdat\_Table2 <- cbind(effect="x",

Size.clus,

N.clus,

L1\_DE\_standardized,

L2\_DE\_standardized,

CLI\_E\_standardized,

summary(sim.ef))

### Simulating power for the L2 direct effect ###

sim.ef <- powerSim(model,

fixed("Z","kr"),

alpha=alpha.S,

nsim=1000)

print(sim.ef)

simdat\_Table2 <- rbind(simdat\_Table2,

cbind(effect="Z",

Size.clus,

N.clus,

L1\_DE\_standardized,

L2\_DE\_standardized,

CLI\_E\_standardized,

summary(sim.ef)))

### Simulating power for the CLI effect ###

sim.ef <- powerSim(model,

fixed("x:Z","kr"),

alpha=alpha.S,

nsim=1000)

print(sim.ef)

### Power of all effects is stored in the data frame simdat\_Table2 ###

simdat\_Table2 <- rbind(simdat\_Table2,

cbind(effect="x:Z",

Size.clus,

N.clus,

L1\_DE\_standardized,

L2\_DE\_standardized,

CLI\_E\_standardized,

summary(sim.ef)))

simdat\_Table2

## Supplementary Material B: R Code for Example 1

##### Example 1: R code for individuals-within-clusters example #####

set.seed(123)

#### Specification of Input Parameters ####

### Specification of standardized input parameters ###

L1\_DE\_standardized <- .10 ## standardized L1 direct effect

L2\_DE\_standardized <- .40 ## standardized L2 direct effect

ICC <- .10 ## standardized intraclass correlation coefficient

alpha.S <- .05 ## significance level

Size.clus <- 5 ## L1 sample size (cluster size)

N.clus <- 150 ## L2 sample size (number of clusters)

### Derivation of a population model for the power analysis ###

## Specification of predictor variables ##

x <- scale(rep(1:Size.clus))

g <- as.factor(1:N.clus)

X <- cbind(expand.grid("x"=x, "g"=g))

X <- data.frame(X, Z=as.numeric(X$g))

X$Z <- scale(X$Z)

## Specification of the outcome variable ##

varL1 <- 1 ## uncond. L1 variance (fixed at 1)

s <- sqrt((varL1)\*(1-(L1\_DE\_standardized^2))) ## cond. L1 variance (Equation 11)

varL2 <- ICC/(1-ICC) ## uncond. L2 variance (Equation 10)

V1 <- varL2\*(1-(L2\_DE\_standardized^2)) ## cond. L2 variance (Equation 11)

## Adjustment of fixed effects (Equation 15) ##

L1\_DE <- L1\_DE\_standardized\*sqrt(varL1)

L2\_DE <- L2\_DE\_standardized\*sqrt(varL2)

#### Implementation of a Power Analysis in a Two-Level Model in SIMR ####

### Vector of fixed effects ###

b <- c(0, L1\_DE, L2\_DE)

### Setting up the population model ###

model <- makeLmer(y ~ x + Z + (1|g), fixef=b, VarCorr=V1, sigma=s, data=X)

print(model)

### Simulating power for the L1 direct effect ###

sim.ef <- powerSim(model,

fixed("x","kr"),

nsim=1000)

print(sim.ef)

simdat\_E1 <- cbind(effect="x",

Size.clus,

N.clus,

L1\_DE\_standardized,

L2\_DE\_standardized,

summary(sim.ef))

### Simulating power for the L2 direct effect ###

sim.ef <- powerSim(model,

fixed("Z","kr"),

nsim=1000)

print(sim.ef)

### Power of all effects is stored in the data frame simdat\_E1 ###

simdat\_E1 <- rbind(simdat\_E1,cbind(effect="Z",

Size.clus,

N.clus,

L1\_DE\_standardized,

L2\_DE\_standardized,

summary(sim.ef)))

simdat\_E1

## Supplementary Material C: R Code for Example 2

##### Example 2: R code for longitudinal data #####

set.seed(123)

#### Specification of Input Parameters ####

### Specification of standardized input parameters ###

L1\_DE\_standardized <- .10 ## standardized L1 direct effect

CLI\_E\_standardized = .50 ## standardized CLI effect

rand.sl = .09 ## standardized random slope variance component

ICC <- .30 ## standardized intraclass correlation coefficient

cor.i.sl = .00 ## Correlation between random slope and random intercept

alpha.S <- .05 ## significance level

Size.clus <- 14 ## L1 sample size (cluster size)

N.clus <- 100 ## L2 sample size (number of clusters)

### Derivation of a population model for the power analysis ###

## Specification of predictor variables ##

x <- scale(rep(1:Size.clus))

g <- as.factor(1:N.clus)

X <- cbind(expand.grid("x"=x, "g"=g))

X <- data.frame(X, Z=as.numeric(X$g))

X$Z <- scale(X$Z)

## Specification of the outcome variable ##

varL1 <- 1 ## uncond. L1 variance (fixed at 1)

s <- sqrt((varL1)\*(1-(L1\_DE\_standardized^2))) ## cond. L1 variance (Equation 11)

varL2 <- ICC/(1-ICC) ## uncond. L2 variance (Equation 10)

## Adjustment of the random slope variance ##

varRS <- rand.sl\*varL1 ## uncond. random slope variance (Equation 13)

condRS <- varRS\*(1-(CLI\_E\_standardized^2)) ## cond. Random slope variance (Equation 11)

## Covariance of random intercept and slope (Equation 14) ##

cov.i.sl <- cor.i.sl\*sqrt(varL2)\*sqrt(varRS)

## Adjustment of fixed effects (Equation 15) ##

L1\_DE <- L1\_DE\_standardized\*sqrt(varL1)

CLI\_E <- CLI\_E\_standardized\*sqrt(varRS)

#### Implementation of a Power Analysis in a Two-Level Model in SIMR ####

### Vector of fixed effects ###

b <- c(0, L1\_DE, CLI\_E)

### Random intercept/slope variance-covariance matrix ###

V2 <- matrix(c(V1, cov.i.sl, cov.i.sl, rand\_sl.con), 2)

### Setting up the population model ###

model <- makeLmer(y ~ x + x:Z + (x|g), fixef=b, VarCorr=V2, sigma=s, data=X)

print(model)

### Simulating power for the L1 direct effect ###

sim.ef <- powerSim(model,

fixed("x","kr"),

nsim=1000)

print(sim.ef)

simdat\_E2 <- cbind(effect="x",

Size.clus,

N.clus,

L1\_DE\_standardized,

L2\_DE\_standardized,

summary(sim.ef))

### Simulating power for the CLI effect ###

sim.ef <- powerSim(model,

fixed("x:Z","kr"),

nsim=1000)

print(sim.ef)

### Power of all effects is stored in the data frame simdat\_E2 ###

simdat\_E2 <- rbind(simdat\_E2,cbind(effect="Z",

Size.clus,

N.clus,

L1\_DE\_standardized,

L2\_DE\_standardized,

summary(sim.ef)))

simdat\_E2