Supplementary Materials

Multivariate Behavioral Research

Assessing Omitted Confounder Bias in Multilevel Mediation Models

Davood Tofighi

School of Psychology

Georgia Institute of Technology

Ken Kelley

Department of Management

Mendoza College of Business

University of Notre Dame

Author Note

Correspondence concerning this article should be addressed to Davood Tofighi, School of Psychology J.S. Coon Bldg., Georgia Institute of Technology, 654 Cherry Street, Atlanta, Georgia 30332-0170. e-mail: dtofighi@psych.gatech.edu

Assessing Omitted Confounder Bias in Multilevel Mediation Models

Centering in $1 \rightarrow 1 \rightarrow 1$ Mediation Models

One important decision in specifying a $1 \rightarrow 1 \rightarrow 1$ model is the scaling (centering) of the Level-1 predictors. Researchers recommend centering the Level-1 predictors (Enders & Tofighi, 2007). The choice of centering depends on the particular research question (Enders, 2013; Enders & Tofighi, 2007). When a researcher is interested in *both* Between and Within relationships for a specific predictor, there are two choices of centering strategies: centering within cluster 2 (CWC2) and centering at the grand mean 2 (CGM2) (Kreft, de Leeuw, & Aiken, 1995). CWC2 centers variables within cluster (school) and adds the cluster (school) means as covariates into the model at Level 2; CGM2 centers a Level-1 predictor at the grand mean (e.g., across all schools) while also including the cluster (school) means as covariates into the model at Level 2. If one is interested in estimating differential Between and Within effects, one may use either CWC2 or CGM2. Both CWC2 and CGM2 are “equivalent” in that the coefficient estimates in CGM2 are linear transformation of the coefficients in CWC2 and vice versa (Kreft et al., 1995). In CGM2, the coefficient associated with the cluster mean predictor is the contextual effect, an estimate of the differential effect of Between and Within effects (Blalock, 1984); However, in CWC2 the coefficient is an estimate of the Between effect. In both CWC2 and CGM2, the coefficient associated with the centered Level-1 predictor is an estimate of the Within effect.

Another issue in applying CWC2 or CGM2 is the choice between the observed instead of latent cluster means to measure a Level-2 construct and to estimate Between (e.g., school-level) effects (Muthén & Muthén, 1998-2012). The reliability of using the observed instead of a latent cluster mean to represent a Level-2 construct plays an important role in estimating a Between effect. Each Level-1 observation might be considered an “item” that represents a Level-2 construct (Snijders & Bosker, 2012). Using the observed mean to represent a Level-2 construct when the number of observations (e.g., items) at Level-1 is “small” might result in less than perfect reliability (e.g., coefficient $\alpha < .8$, Cronbach,

1951). In this case, using the observed cluster means could also underestimate Between effects. On the other hand, Lüdtke et al. showed that using observed means could result in smaller standard errors of the Between coefficients. Finally, both methods yielded similar results for significance tests of Between coefficients. In sum, using latent cluster mean would yield less biased and efficient Between estimates than would the observed cluster mean. The choice between observed and latent cluster mean depends on the research question and design specifications of a particular multilevel study (Lüdtke et al., 2008).

Estimating A $1 \rightarrow 1 \rightarrow 1$ Model in Mplus

Mplus is one of the few programs that can estimate multilevel structural equation modeling (MSEM) with random slopes and latent cluster means. We highlight nuances in Mplus syntax to estimate an MSEM. Preacher, Zyphur, and Zhang (2010) provide a detailed explanation of estimating different types of multilevel model using observed and latent cluster means. We discuss two models and highlight subtle differences in Mplus syntax that might be confusing: (a) a $1 \rightarrow 1 \rightarrow 1$ model with random intercepts only and (b) a $1 \rightarrow 1 \rightarrow 1$ model with random slopes and intercepts (our focus)

A $1 \rightarrow 1 \rightarrow 1$ model with random intercepts only. Suppose that we want to estimate this model with latent cluster mean. The default and only option in Mplus is CWC2. Mplus does latent mean clustering, i.e., decomposing a Level-1 variable into Between and Within when the observed variable is (a) measured at Level-1 and (b) is not declared as Within in the VARIABLE command section. Also, for the random intercept model, one needs to specify the following command ANALYSIS: TYPE = TWOLEVEL. Example 9.1 from Mplus Manual Version 7 (Muthén & Muthén, 1998-2012, pp. 262–263) explains this example:

The difference between this part of the example and the first part is that the covariate x is decomposed into two latent variable parts instead of being treated as an observed variable as in conventional multilevel regression modeling. The

decomposition occurs when the covariate x is not mentioned on the WITHIN statement and is therefore modeled on both the within and between levels.

When a covariate is not mentioned on the WITHIN statement, it is decomposed into two uncorrelated latent variables,

$$x_{ij} = x_{wij} + x_{bj}$$

where i represents individual, j represents cluster, x_{wij} is the latent variable covariate used on the within level, and x_{bj} is the latent variable covariate used on the between level. This model is described in Muthén (1989, 1990, 1994).

The latent variable covariate x_b is not used in conventional multilevel analysis.

Using a latent covariate may, however, be advantageous when the observed cluster-mean covariate x_m does not have sufficient reliability resulting in biased estimation of the between-level slope (Asparouhov & Muthén, 2006b; Lüdtke et al., 2008).

For analytical results, we used CWC2 with latent mean centering to derive analytical results because of research questions and convenience. For estimating the $1 \rightarrow 1 \rightarrow 1$ model in sensitivity analysis section, however, we used CGM2 because it was the only choice in Mplus; CGM2 coefficients were linearly transformed to obtain CWC2 coefficients

We can use either CWC2 or CGM2 to estimate a $1 \rightarrow 1 \rightarrow 1$ model in the sensitivity analysis. We used CGM2 in Mplus. While for analytical derivations we used CWC2, for estimating the $1 \rightarrow 1 \rightarrow 1$ model in sensitivity analysis, one could use either CWC2 and CGM2 because these two methods are equivalent for the $1 \rightarrow 1 \rightarrow 1$ model. That is, the coefficient estimates in CGM2 are linear transformation of the coefficients in CWC2 and vice versa. For a $1 \rightarrow 1 \rightarrow 1$ model with random slopes and intercepts and latent mean centering, Mplus only allows CGM2 strategy. We used this option along with additional command in Mplus to compute the Between effect (See MODEL CONSTRAINT command). Because of the equivalency, the results of the proposed methodology remains the same for the reasons explained in the manuscript.

A $1 \rightarrow 1 \rightarrow 1$ with random slopes and intercepts. Suppose that we want to estimate this model with latent cluster mean. The default and only option in Mplus is CGM2 with latent mean clustering. Mplus uses the latent mean as a Level-2 predictor in the model when the observed variable is (a) measured at Level-1 and (b) is not declared as Within in the VARIABLE command section. It is important to note that for this model, the following command, which is different from the random intercepts model, should be specified: `TYPE = TWOLEVEL RANDOM;` An extension of Example 9.2 from Mplus Manual Version 7 (Muthén & Muthén, 1998-2012, p. 267) explains this example:

This [using latent mean] is achieved when the individual-level observed covariate is modeled in both the within and between parts of the model. This is requested by not mentioning the observed covariate x on the WITHIN statement in the VARIABLE command. When a random slope is estimated, the observed covariate x is used on the within level and the latent variable covariate x_{bj} is used on the between level.

Using observed X_{ij} at the within level and latent X_{bj} at the between level means CGM2 with latent mean clustering. Although not explicitly emphasized, Mplus switches to CGM2 when the latent cluster mean is used. That is, in specifying a random slopes model, if one uses observed cluster means, Mplus estimates CWC2. However, if one uses latent cluster mean, Mplus estimates CGM2. This point, not widely known, is mentioned in supplemental materials by Preacher et al. (2010) available at <http://dx.doi.org/10.1037/a0020141.supp>

Calculate Regression Effects

In this section, we describe steps to calculate the coefficients, ds , ls , ss , and qs necessary to compute adjusted results.

1. Obtain Between and Within estimate of sample variance-covariance matrices, $\mathbf{S}_B$

and $\mathbf{S}_W$, respectively. For example, Mplus (Muthén & Muthén, 1998-2012) can produce these estimates.

2. Use the Between and Within sample estimates to calculate the regression coefficients.

For Step 1, we provided sample Mplus syntax for the empirical example to specify $1 \rightarrow 1 \rightarrow 1$ model (See the section “Mplus Input” in this document). Note that we assume this model to be potentially “misspecified”. The estimates for this model are potentially “biased” as a result of the omitted confounders. The Mplus output (See the section “Mplus Output” in this document) contains the estimates of the Between and Within sample variance–covariance matrices for the variables X , M , and Y . From these matrices one can also calculate the Between and Within correlation matrices for X , M , and Y .

In Step 2, the correlation matrices are augmented with correlation coefficients between Z and X , M , and Y . These correlation coefficients are plausible values set by the researcher. As a result, the “augmented” Between and Within sample correlation matrices can be calculated for each set of plausible correlation values. The correlation coefficients are then used to calculate standardized regression coefficients for the Between and Within regression coefficients using computational formulas in Cohen, Cohen, West, and Aiken (2003, Chapter 3) and Mauro (1990). Unstandardized coefficients can also be calculated using standard deviations obtained from the diagonal elements of the Between and Within sample covariances. Note that for identification purposes, Between and Within parts of the latent proxy variables are scaled to have a mean of zero and SD of one. To illustrate, we show the calculation for l_{ZX} in Equation B4 and l_{YZ} in Equation B5. The calculations for the coefficients ds , ss , and qs follow a similar computational approach.

$$l_{ZX} = \frac{\rho_{ZX}^{(W)}}{\sigma_{\eta_{Xij}}}$$

In the formula above, $\rho_{ZX}^{(W)}$ and $\sigma_{\eta_{Xij}}$ denote the population within–cluster correlation between Z_{ij} (η_{Zij}) and X_{ij} (η_{Xij}) and within–cluster population standard deviation of

$X_{ij}(\eta_{Xij})$; superscript “W” denotes Within. To compute an estimate, $\rho_{ZX}^{(W)}$ is replaced by plausible values set by the researcher and $\sigma_{\eta_{Xij}}$ is replaced with a within-cluster estimate obtained from software output such as Mplus. For example, l_{YZ} in Equation B5 is calculated as follows:

$$l_{YZ} = \sigma_{\eta_{Yij}} \frac{\rho_{YZ}^{(W)}(1 - \rho_{XM}^{(W)2}) + \rho_{YX}^{(W)}(\rho_{ZM}^{(W)}\rho_{XM}^{(W)} - \rho_{ZX}^{(W)}) + \rho_{YM}^{(W)}(\rho_{ZX}^{(W)}\rho_{XM}^{(W)} - \rho_{ZM}^{(W)})}{1 + 2\rho_{XM}^{(W)}\rho_{ZM}^{(W)}\rho_{ZX}^{(W)} - \rho_{XM}^{(W)2} - \rho_{ZM}^{(W)2} - \rho_{ZX}^{(W)2}}$$

All the population ρ values are replaced by the sample estimates from the “augmented” Within correlation matrix that includes plausible values for the correlations between Z and X , M , and Y as well as the sample estimates of correlations between X , M , and Y ; $\sigma_{\eta_{Yij}}$ is the population within-cluster standard deviation of $Y_{ij}(\eta_{Yij})$, which can be estimated from Mplus output. It should be noted that plausible values for the correlations between Z and X , M , and Y are chosen so that the resulting augmented matrices are positive definite (Rousseeuw & Molenberghs, 1994).

R Code

In this section, we explain R functions used to conduct a sensitivity analysis for a two-level $1 \rightarrow 1 \rightarrow 1$ model with the random intercepts and slopes. We also explain specific R commands used to generate the results for the empirical example in the current manuscript. To follow the materials presented below readers need to have basic familiarity with the statistical software environment R (R Development Core Team, 2014).

Two main functions to conduct sensitivity analysis for the $1 \rightarrow 1 \rightarrow 1$ model are **SensitivityB** and **SensitivityW**. These functions are located in a file called **sensitivity.R**. To load these functions into your R session, run the following command:

```
source("sensitivity.R")
```

Input. Below is the list of the input arguments for both functions `SensitivityB` and `SensitivityW`:

- **R:** A 3×3 Between or Within correlation matrix. **R** needs to be a matrix with row and column dimension names "x", "m", and "y". For example, dimension names for **R** can be set as follows:

```
dimnames(R) <- list(c("m", "x", "y") , c("m", "x", "y") )
```

- **est:** A vector of estimates for coefficients a and b . The vector elements must have names for a and b specified as "a" and "b".
- **sd:** A vector of Between or Within estimates of SDs for X , M , and Y . These values are the square root of diagonal of the Between or Within variance–covariance matrices. The vector elements must have the names in the following format: "x", "m", and "y".
- **rZXRange:** A vector of plausible values for correlation between X and Z .
- **rZMRange:** A vector of plausible values for correlation between M and Z .
- **rZYRange:** A vector of plausible values for correlation between Y and Z .
- **covajbj:** Covariance between random coefficients a_j and b_j . This argument is only used in `SensitivityW` function.
- **ncpus:** Number of processors to run computations in parallel.

Output. Each function generates a data frame with the variables described below. For example, function `SensitivityB` generates a data frame with the following variables:

- **rxz:** Correlation between X and Z

- `rmz`: Correlation between M and Z
- `ryz`: Correlation between Y and Z
- `abBetBiased`: Unadjusted (“biased”) indirect effect estimate
- `abBetTrue`: Adjusted (“unbiased”) indirect effect estimate

Then, one can use the resulting data frames to produce sensitivity contour plots. Below, we show R commands used to generate the results in the empirical example:

```
estW <- c(0.086, 0.433)
estB <- c(0.024, 0.784)
names(estW) <- names(estB) <- c("a", "b") # names the vector elements
## Read the estimates of between and within covariance matrices
## produced by Mplus. See the Mplus output section.
sw_vec <- scan("spw.txt") # Within estimate
sb_vec <- scan("estsigb.txt") #Between estimate
## vechr.reverse function from lavaan package converts a vector to
## a matrix
library(lavaan)
SW <- vechr.reverse(sw_vec) ## within covariance matrix
SB <- vechr.reverse(sb_vec) ## between covariance matrix
nam <- c("m", "y", "x") ##dim names
dimnames(SW) <- dimnames(SB) <- list(nam, nam) # assigning dim names
RW <- cov2cor(SW) ##correlation matrix
RB <- cov2cor(SB) ##correlation matrix
sdW <- sqrt(diag(SW)) ##compute the SDs from diagonal values
sdB <- sqrt(diag(SB)) ##compute the SDs from diagonal values
## Covariance between aj and bj
```

```

covajbj <- 0
## Generating plausible values

stp <- 0.01 # the amount of step

rZXRange <- c(0, 0.1, 0.3, 0.5)
rZMRange <- seq(-0.5, 0.5, stp)
rZYRange <- seq(-0.5, 0.5, stp)

library(parallel) ## load parallel library for parallel computing

resB <- SensitivityB(RB, est = estB, sd = sdB, rZXRange = rZXRange,
  rZMRange = rZMRange, rZYRange = rZYRange, ncpus = ncpus)

resW <- SensitivityW(RW, est = estW, sd = sdW, covajbj = covajbj,
  rZXRange = rZXRange, rZMRange = rZMRange, rZYRange = rZYRange,
  ncpus = ncpus)

## To generate sensitivity contour plots using lattice library

library(lattice)
library(grid)
library(gridExtra)

## Between plot

trellis.par.set(theme = col.whitebg()) # Set the coloring scheme

przx0 <- contourplot(abBetTrue ~ rzm * ryz, resB, main = expression(r[ZX] ==
  0), xlab = expression(r[ZM]), ylab = expression(r[ZY]), region = TRUE,
  subset = rzx == "rZX=0", par.settings = list(fontsize = list(text = 8,
    points = 10)), cuts = 6)

przx.1 <- contourplot(abBetTrue ~ rzm * ryz, resB, main = expression(r[ZX] ==
  0.1), xlab = expression(r[ZM]), ylab = expression(r[ZY]), region = TRUE,
  subset = rzx == "rZX=0.1", par.settings = list(fontsize = list(text = 8,
    points = 10)))

przx.3 <- contourplot(abBetTrue ~ rzm * ryz, resB, main = expression(r[ZX] ==

```

```

0.3), xlab = expression(r[ZM]), ylab = expression(r[ZY]), region = TRUE,
subset = rZX == "rZX=0.3", par.settings = list(fontsize = list(text = 8,
points = 10)))
grid.arrange(przx0, przx.1, przx.3, ncol = 2)

## Within plot
trellis.par.set(theme = col.whitebg())

przxw0 <- contourplot(abWithTrue ~ rzm * ryz, resW, main = expression(r[ZX] ==
0), xlab = expression(r[ZM]), ylab = expression(r[ZY]), region = TRUE,
subset = rZX == "rZX=0", par.settings = list(fontsize = list(text = 8,
points = 10))), cuts = 6)

przxw.1 <- contourplot(abWithTrue ~ rzm * ryz, resW, main = expression(r[ZX] ==
0.1), xlab = expression(r[ZM]), ylab = expression(r[ZY]), region = TRUE,
subset = rZX == "rZX=0.1", par.settings = list(fontsize = list(text = 8,
points = 10)))

przxw.3 <- contourplot(abWithTrue ~ rzm * ryz, resW, main = expression(r[ZX] ==
0.3), xlab = expression(r[ZM]), ylab = expression(r[ZY]), region = TRUE,
subset = rZX == "rZX=0.3", par.settings = list(fontsize = list(text = 8,
points = 10)))

grid.arrange(przxw0, przxw.1, przxw.3, ncol = 2)

```

Sensitivity.R Function

```

## There are two main functions to conduct sensitivity analysis for a
## 1-1-1 model called SensitivityB and SensitivityW located in a file
## 'sensitivity.R'. To load these functions into your R session, run
## the following command:
## source('sensitivity.R')

```

```

## Input and output values for each function is listed below.

## R: A 3 by 3 Between or Within correlation matrix. R needs to be a
## matrix with row and column dimension names 'x', 'm', and 'y'. For
## example, dimension names for R can be set as follows: dimnames(R)
## <- list(c('m', 'x', 'y') , c('m', 'x', 'y'))

## est: A vector of estimates for coefficients a and b. The vector
## elements must have names for a and b as 'a' and 'b'.

## sd: A vector of between and estimates SDs of X, M, and Y . These
## values are the square root of diagonal of the Between or Within
## variance-covariance matrices. The vector elements must have the
## names of the following format: 'x', 'm', and 'y'.

## rZXRange: A vector of plausible values for correlation between X
## and Z.

## rZMRange: A vector of plausible values for correlation between M
## and Z.

## rZYRange: A vector of plausible values for correlation between Y
## and Z.

## covajbj: Covariance between random coefficients aj and bj. This
## argument is only used in SensitivityW function.

## ncpus: Number of processors to run computations in parallel.

#### Extra Utility Functions

b_1p <- function(ryx, sx, sy) {
  b <- ryx * sy/sx
  return(as.vector(b))
}

b1_2p <- function(ry1, ry2, r12, sy, s1) {
  # x1 is the primary variable

```

```

    beta1 <- (ry1 - ry2 * r12)/(1 - r12^2)

    b1 <- beta1 * sy/s1

    return(as.vector(b1))
}

corCompute <- function(x) {
  ## accepts list of parameters and return a cor matrix

  x <- x[1:6]

  corM <- lavaan::vech.reverse(x, FALSE)

  corM <- corM + diag(1, 4)

  corM <- corM + diag(1, 4)

  return(corM)
}

bz_3p <- function(x) {
  rzx <- x[1]
  rzm <- x[2]
  ryz <- x[3]
  rxm <- x[4]
  ryx <- x[5]
  rym <- x[6]
  sx <- x[7]
  sm <- x[8]
  sy <- x[9]
  sz <- x[10]

  num <- ryz * (1 - rxm^2) + ryx * (rzm * rxm - rzx) + rym * (rzx *
    rxm - rzm)

  den <- 1 + 2 * rxm * rzm * rzx - rxm^2 - rzm^2 - rzx^2

  b <- as.vector(num/den * sy)

```

```

    return(b)
}

### Main Functions

SensitivityB <- function(R, est, sd, rZXRange = c(-0.3, -0.1, 0, 0.1,
0.3), rZMRange = seq(-0.99, 0.99, 0.02), rZYRange = seq(-0.99, 0.99,
0.02), ncpus = 3L) {
  RYX_B = R["x", "y"]  ##Observed correlation between the independent and dependent v
  RMX_B = R["x", "m"]  ##Observed correlation between the independent variable and th
  RYM_B = R["m", "y"]  ##Observed correlation between the mediator and the dependent
  #
  aBiased_B = est["a"]
  bBiased_B = est["b"]
  sx_B = sd["x"]
  sm_B = sd["m"]
  sy_B = sd["y"]
  rVecM = rZMRange  #seq(-.99,.99,.02)
  rVecY <- rZYRange  #seq(-.5,.5,.01)
  rVecX <- rZXRange  #c(-.3,-.1,0,.1,.3)
  input <- expand.grid(rzx = rVecX, rzm = rVecM, ryz = rVecY, rxm = RMX_B,
    ryx = RYX_B, rym = RYM_B, sx = sx_B, sm = sm_B, sy = sy_B, sz = 1,
    aBiased = aBiased_B, bBiased = bBiased_B)
  input <- data.frame(input)
  inputList <- split(input, 1:nrow(input))
  ## We would like to acknowledge that following code is adopted from
  ## from the lavaan package
  have_mc <- have_snow <- FALSE
  if (.Platform$OS.type != "windows" && ncpus > 1L)

```

```

    have_mc <- TRUE else if (.Platform$OS.type == "windows" && ncpus > 1L)
      have_snow <- TRUE
  res1 <- if (ncpus > 1L && (have_mc || have_snow)) {
    if (have_mc) {
      parallel::mclapply(inputList, trueCoefB, mc.cores = ncpus)
    } else if (have_snow) {
      cl <- parallel::makePSOCKcluster(rep("localhost", ncpus)) #
      if (RNGkind()[1L] == "L'Ecuyer-CMRG")
        parallel::clusterSetRNGStream(cl) #
      parallel::clusterExport(cl, list("corCompute", "trueCoefB",
        "b_1p", "b1_2p", "bz_3p"))
      res <- parallel::parLapply(cl, inputList, trueCoefB) #
      parallel::stopCluster(cl) #
      res
    }
  } else lapply(inputList, trueCoefB)
  res1 <- do.call("rbind", res1)
  res1 <- data.frame(res1)
  ord1 <- with(res1, order(rzm, ryz))
  resN <- res1[ord1, ]
  resN$rx <- factor(resN$rx, levels = sort(unique(rVecX)), labels = paste("rZX=",
    sort(unique(rVecX)), sep = ""))
  return(resN)
}

SensitivityW <- function(R, est, sd, covajbj = 0, rZXRange = c(-0.3,
  -0.1, 0, 0.1, 0.3), rZMRange = seq(-0.99, 0.99, 0.02), rZYRange = seq(-0.99,
  0.99, 0.02), ncpus = 3L) {

```

```

ryx = R["x", "y"] ##Observed correlation between the independent and dependent variables
rxm = R["x", "m"] ##Observed correlation between the independent variable and the mediator
rym = R["m", "y"] ##Observed correlation between the mediator and the dependent variable
aBiased = est["a"]
bBiased = est["b"]
sx = sd["x"]
sm = sd["m"]
sy = sd["y"]
rVecM <- rZMRange
rVecY <- rZYRange
rVecX <- rZXRange
input <- expand.grid(rzx = rVecX, rzm = rVecM, ryz = rVecY, rxm = rxm,
  ryx = ryx, rym = rym, sx = sx, sm = sm, sy = sy, sz = 1, aBiased = aBiased,
  bBiased = bBiased, covajbj)
input <- data.frame(input)
inputList <- split(input, 1:nrow(input))
## We would like to acknowledge that following code is adopted from
## from the lavaan package
have_mc <- have_snow <- FALSE
if (.Platform$OS.type != "windows" && ncpus > 1L)
  have_mc <- TRUE else if (.Platform$OS.type == "windows" && ncpus > 1L)
  have_snow <- TRUE
have_mc
have_snow
res1 <- if (ncpus > 1L && (have_mc || have_snow)) {
  if (have_mc) {
    parallel::mclapply(inputList, trueCoefW, mc.cores = ncpus)
  }
}

```



```

    } else if (have_snow) {
      cl <- parallel::makePSOCKcluster(rep("localhost", ncpus)) #
      if (RNGkind()[1L] == "L'Ecuyer-CMRG")
        parallel::clusterSetRNGStream(cl) #
      parallel::clusterExport(cl, list("corCompute", "trueCoefW",
        "b_1p", "b1_2p", "bz_3p"))
      res <- parallel::parLapply(cl, inputList, trueCoefW) #
      parallel::stopCluster(cl) #
      res
    }
  } else lapply(inputList, trueCoefW)
res1 <- do.call("rbind", res1)
res1 <- data.frame(res1)
ord1 <- with(res1, order(rzm, ryz))
resN <- res1[ord1, ]
resN$rx <- factor(resN$rx, levels = sort(unique(rVecX)), labels = paste("rZX=",
  sort(unique(rVecX)), sep = ""))
return(resN)
}

### Additional Utility Functions within coefficients adjusted values
trueCoefW <- function(x) {
  x <- unlist(x)
  xMat <- corCompute(x)
  if (matrixcalc::is.positive.definite(xMat)) {
    rx <- x[1]
    rzm <- x[2]
    ryz <- x[3]

```

```

rxm <- x[4]
ryx <- x[5]
rym <- x[6]
sx <- x[7]
sm <- x[8]
sy <- x[9]
sz <- x[10]
aBiased <- x[11]
bBiased <- x[12]
covajbj <- x[13]
lzx <- b_1p(rzx, sx, sz)
lyz <- bz_3p(x)
lmz <- b1_2p(ry1 = rzm, ry2 = rxm, r12 = rzx, sy = sm, s1 = sz) #M is y and z
qzx <- b1_2p(ry1 = rzx, ry2 = rzm, r12 = rxm, sy = sz, s1 = sx)
qzm <- b1_2p(ry1 = rzm, ry2 = rzx, r12 = rxm, sy = sz, s1 = sm)
aBias <- lmz * lzx
bBias <- lyz * qzm
cpBias <- lyz * qzm
abBiased <- as.vector(aBiased * bBiased) + covajbj
a <- aBiased - aBias #unbiased (adjusted) estimate
b <- bBiased - bBias #unbiased (adjusted)
ab <- as.vector(a * b) # unbiased (adjusted) estimate of indirect effect
grp = ifelse(ryz < 0, 1, 0) # This is used to distinguish two curves (if exists
res <- c(rzx, rzm, ryz, abWithBiased = abBiased, abWithTrue = ab,
        grp = as.vector(grp))
} else res <- c(rzx, rzm, ryz, abWithBiased = NA, abWithTrue = NA,
               grp = NA)

```

```

    return(res)
}

### between coefficients adjusted values
trueCoefB <- function(x) {
  x <- unlist(x)
  xMat <- corCompute(x)
  if (matrixcalc::is.positive.definite(xMat)) {
    rzx <- x[1]
    rzm <- x[2]
    ryz <- x[3]
    rxm <- x[4]
    ryx <- x[5]
    rym <- x[6]
    sx <- x[7]
    sm <- x[8]
    sy <- x[9]
    sz <- x[10]
    aBiased <- x[11]
    bBiased <- x[12]
    dzx <- b_1p(rzx, sx, sz)
    dyz <- bz_3p(x)
    dmz <- b1_2p(ry1 = rzm, ry2 = rxm, r12 = rzx, sy = sm, s1 = sz) #M is y and z
    szx <- b1_2p(ry1 = rzx, ry2 = rzm, r12 = rxm, sy = sz, s1 = sx)
    szm <- b1_2p(ry1 = rzm, ry2 = rzx, r12 = rxm, sy = sz, s1 = sm)
    aBias <- dmz * dzx
    bBias <- dyz * szm
    cpBias <- dyz * szm
  }
}

```

```

    abBiased <- as.vector(aBiased * bBiased)

    a <- aBiased - aBias  #unbiased (adjusted) estimate
    b <- bBiased - bBias  #unbiased (adjusted) estimate

    ab <- as.vector(a * b)  # unbiased (adjusted) estimate of indirect effect

    grp = ifelse(ryz < 0, 1, 0)  # This is used to distinguish two curves (if exists)

    res <- c(rzx, rzm, ryz, abBetBiased = abBiased, abBetTrue = ab,
            grp = as.vector(grp))
  } else res <- c(rzx, rzm, ryz, abBetBiased = NA, abBetTrue = NA, grp = NA)

  return(res)
}

```

Mplus Input

```

TITLE: MSEM for mediation analysis

random intercepts and slopes;

DATA: FILE = mplusdata.txt;

VARIABLE:

NAMES = id x m y;

USEVARIABLES= id x m y;

MISSING=.;

CLUSTER=id;

DEFINE:

standardize x m y;

ANALYSIS: TYPE IS TWOLEVEL RANDOM;

MODEL: ! model specification follows

%WITHIN% ! Model for Within effects follows

sa | m ON x; ! regress m on x, call the random slope sa

sb | y ON m; ! regress y on m, call the random slope sb

```

```

sc | y ON x; ! regress y on x, call the random slope sc
%BETWEEN% ! Model for Between effects follows
sa sb sc x m y; ! estimate Level-2 (residual) variances for sa, sb,
    sc, x, m, and y
sa WITH sc; !      ! estimate Level-2 covariances of sa with sc, x, m,
    and y
sa WITH sb (covab); ! estimate Level-2 covariance of sa and sb, call
    it cab
sb WITH sc ;    ! estimate Level-2 covariances of sb with sc, x, m,
    and y
m ON x (acon); ! regress m on x, call the slope acon, acon =
    contextual effect
!, not the Between slope
y ON m (bcon); ! regress y on m, call the slope bcon, bcon =
    contextual effect
!, not the Between slope
y ON x (cpcon); ! regress y on x
[sa] (aw); ! estimate the mean of sa, call it aw
[sb] (bw); ! estimate the mean of sb, call it bw
[sc] (cpw); ! estimate the mean of sc, call it cpw
MODEL CONSTRAINT: ! section for computing indirect effects
NEW(ab bb cpb indb indw indwWRONG); ! name the indirect effects
ab=aw+acon;    ! compute Between a path
bb=bw+bcon;    ! compute Between b path
cpb=cpw+cpcon; ! compute Between b path
indw=aw*bw;    ! compute the Within indirect effect
indwWRONG=aw*bw+covab; ! compute the Within indirect effect--not
    correct
indb=ab*bb;    ! compute the Between indirect effect

```

```

OUTPUT: samp TECH1 TECH8; ! request parameter specifications,
        starting values,
! optimization history, and confidence intervals for all effects
Savedata:
        sample = spw.txt; !Saves Within variance-covariance estimate
        sigb = estsigb.txt; !Saves Between variance-covariance estimate

```

Mplus Output

Mplus VERSION 7.3 (Mac)

MUTHEN & MUTHEN

11/15/2014 12:21 PM

INPUT INSTRUCTIONS

```

TITLE: MSEM for mediation analysis
random intercepts and slopes;
DATA: FILE = mplusdata.txt;
VARIABLE:
NAMES = id x m y;
USEVARIABLES= id x m y;
MISSING=.;
CLUSTER=id;
DEFINE:
standardize x m y;
ANALYSIS: TYPE IS TWOLEVEL RANDOM;
MODEL: ! model specification follows
%WITHIN% ! Model for Within effects follows
sa | m ON x; ! regress m on x, call the random slope sa

```

```

sb | y ON m; ! regress y on m, call the random slope sb
sc | y ON x; ! regress y on x, call the random slope sc
%BETWEEN% ! Model for Between effects follows
sa sb sc x m y; ! estimate Level-2 (residual) variances for sa, sb
, sc, x, m, and y
sa WITH sc; ! ! estimate Level-2 covariances of sa with sc, x, m
, and y
sa WITH sb (covab); ! estimate Level-2 covariance of sa and sb,
call it cab
sb WITH sc ; ! estimate Level-2 covariances of sb with sc, x, m,
and y
m ON x (acon); ! regress m on x, call the slope acon, acon =
contextual effect
!, not the Between slope
y ON m (bcon); ! regress y on m, call the slope bcon, bcon =
contextual effect
!, not the Between slope
y ON x (cpcon); ! regress y on x
[sa] (aw); ! estimate the mean of sa, call it aw
[sb] (bw); ! estimate the mean of sb, call it bw
[sc] (cpw); ! estimate the mean of sc, call it cpw
MODEL CONSTRAINT: ! section for computing indirect effects
NEW(ab bb cpb indb indw indwWRONG); ! name the indirect effects
ab=aw+acon; ! compute Between a path
bb=bw+bcon; ! compute Between b path
cpb=cpw+cpcon; ! compute Between b path
indw=aw*bw; ! compute the Within indirect effect
indwWRONG=aw*bw+covab; ! compute the Within indirect effect--not
correct

```

```

indb=ab*bb; ! compute the Between indirect effect
OUTPUT: samp TECH1 TECH8; ! request parameter specifications,
      starting values,
! optimisation history, and confidence intervals for all effects
Savedata:
      sample = spw.txt; !Saves Within variance-covariance estimate
      sigb = estsigb.txt; !Saves Between variance-covariance
      estimate

```

INPUT READING TERMINATED NORMALLY

MSEM for mediation analysis
 random intercepts and slopes;

SUMMARY OF ANALYSIS

Number of groups	1
Number of observations	12000
Number of dependent variables	2
Number of independent variables	1
Number of continuous latent variables	3
Observed dependent variables	

Continuous

M

Y

Observed independent variables

X

Continuous latent variables

SA

SB

SC

Variables with special functions

Cluster `variable`

ID

Estimator	MLR
Information matrix	OBSERVED
Maximum number of iterations	100
Convergence criterion	0.100D-05
Maximum number of EM iterations	500
Convergence criteria for the EM algorithm	
Loglikelihood change	0.100D-02
Relative loglikelihood change	0.100D-05
Derivative	0.100D-03
Minimum variance	0.100D-03
Maximum number of steepest descent iterations	20
Maximum number of iterations for H1	2000
Convergence criterion for H1	0.100D-03
Optimization algorithm	EMA

Input `data` file(s)

mplusdata.txt

Input data format FREE

SUMMARY OF DATA

Number of missing data patterns 1

Number of clusters 300

Average cluster size 40.000

Estimated Intraclass Correlations for the Y Variables

Intraclass		Intraclass	
Intraclass			
Variable	Correlation	Variable	Correlation
Correlation			
M	0.649	Y	0.526
0.207		X	

COVARIANCE COVERAGE OF DATA

Minimum covariance coverage value 0.100

PROPORTION OF DATA PRESENT

Covariance Coverage			
	M	Y	X
	-----	-----	-----
M	1.000		
Y	1.000	1.000	
X	1.000	1.000	1.000

SAMPLE STATISTICS

NOTE: The sample statistics for **within** and **between** refer to the maximum-likelihood estimated **within** and **between** covariance matrices, respectively.

ESTIMATED SAMPLE STATISTICS FOR **WITHIN**

Means			
	M	Y	X
	-----	-----	-----
1	0.000	0.000	0.000

Covariances

	M	Y	X
	-----	-----	-----

M	0.351		
Y	0.167	0.474	
X	0.062	0.096	0.793

Correlations

	M	Y	X
	-----	-----	-----
M	1.000		
Y	0.408	1.000	
X	0.117	0.156	1.000

ESTIMATED SAMPLE STATISTICS FOR BETWEEN

Means

	M	Y	X
	-----	-----	-----
1	0.000	0.000	0.000

Covariances

	M	Y	X
	-----	-----	-----
M	0.649		
Y	0.475	0.526	
X	-0.001	-0.024	0.207

Correlations			
	M	Y	X
	-----	-----	-----
M	1.000		
Y	0.813	1.000	
X	-0.002	-0.072	1.000

MAXIMUM LOG-LIKELIHOOD VALUE FOR THE UNRESTRICTED (H1) **MODEL** IS
-39127.241

UNIVARIATE HIGHER-ORDER MOMENT DESCRIPTIVE STATISTICS

UNIVARIATE HIGHER-ORDER MOMENT DESCRIPTIVE STATISTICS

	Variable/		Mean/	Skewness/	Minimum/	% with
	Percentiles					
	Sample Size	Variance	Kurtosis	Maximum	Min/Max	
	20%/60%	40%/80%	Median			
M		0.000	-0.083	-4.254	0.01%	
	-0.826	-0.208	0.029			
	12000.000	1.000	0.230	4.809	0.01%	
	0.271	0.820				
Y		0.000	0.002	-4.677	0.01%	
	-0.794	-0.229	0.018			

	12000.000	1.000	0.718	5.018	0.01%
	0.247	0.783			
X		0.000	0.023	-4.113	0.01%
	-0.843	-0.259	-0.008		
	12000.000	1.000	-0.048	3.750	0.01%
	0.245	0.847			

THE MODEL ESTIMATION TERMINATED NORMALLY

MODEL FIT INFORMATION

Number of Free Parameters 21

Loglikelihood

H0 Value	-29160.280
H0 Scaling Correction Factor	1.1811
for MLR	

Information Criteria

Akaike (AIC)	58362.559
Bayesian (BIC)	58517.805
Sample-Size Adjusted BIC	58451.069
$(n^* = (n + 2) / 24)$	

MODEL RESULTS

		Two-Tailed			
		Estimate	S.E.	Est./S.E.	P-Value
Within Level					
Variances					
X		0.793	0.010	77.038	0.000
Residual Variances					
M		0.177	0.002	76.502	0.000
Y		0.102	0.001	70.322	0.000
Between Level					
M	ON				
X		-0.062	0.112	-0.556	0.579
Y	ON				
M		0.350	0.082	4.293	0.000
X		-0.063	0.092	-0.685	0.493
SA	WITH				
SC		0.011	0.010	1.062	0.288
SB		0.004	0.021	0.206	0.837

SB	WITH				
SC	-0.005	0.016	-0.320	0.749	
Means					
X	0.000	0.028	0.000	1.000	
SA	0.086	0.027	3.216	0.001	
SB	0.433	0.043	10.162	0.000	
SC	0.078	0.021	3.713	0.000	
Intercepts					
M	0.006	0.048	0.131	0.895	
Y	-0.067	0.042	-1.593	0.111	
Variances					
X	0.207	0.017	12.345	0.000	
SA	0.208	0.018	11.874	0.000	
SB	0.510	0.045	11.284	0.000	
SC	0.125	0.011	11.316	0.000	
Residual Variances					
M	0.684	0.061	11.301	0.000	
Y	0.501	0.073	6.874	0.000	
New/Additional Parameters					
AB	0.024	0.109	0.217	0.828	
BB	0.784	0.071	10.994	0.000	
CPB	0.015	0.090	0.165	0.869	
INDB	0.019	0.086	0.216	0.829	
INDW	0.037	0.012	3.056	0.002	

INDWWRON	0.042	0.024	1.746	0.081
----------	-------	-------	-------	-------

QUALITY OF NUMERICAL RESULTS

Condition Number for the Information Matrix	0.165E
-05	
(ratio of smallest to largest eigenvalue)	

TECHNICAL 1 OUTPUT

PARAMETER SPECIFICATION FOR WITHIN

	NU			
	M	Y	X	
	-----	-----	-----	
1	0	0	0	
	LAMBDA			
	SA	SB	SC	M
Y	-----	-----	-----	-----

M	0	0	0	0
	0			

Y	0	0	0	0
	0			
X	0	0	0	0
	0			

LAMBDA

	X

M	0
Y	0
X	0

THETA

	M	Y	X
	-----	-----	-----
M	0		
Y	0	0	
X	0	0	0

ALPHA

	SA	SB	SC	M
Y	-----	-----	-----	-----

1	0	0	0	0
	0

SA	0
SB	0
SC	0
M	0
Y	0
X	0

	PSI			
	SA	SB	SC	M
Y	-----	-----	-----	-----

SA	0			
SB	0	0		
SC	0	0	0	
M	0	0	0	1
Y	0	0	0	0
2				
X	0	0	0	0
0				

	PSI
	X

X	3

PARAMETER SPECIFICATION FOR BETWEEN

NU				
	M	Y	X	
	-----	-----	-----	
1	0	0	0	
LAMBDA				
	SA	SB	SC	M
Y	-----	-----	-----	-----

M	0	0	0	0
0				
Y	0	0	0	0
0				
X	0	0	0	0
0				
LAMBDA				
	X			

M	0			
Y	0			
X	0			

THETA			
	M	Y	X
	-----	-----	-----
M	0		
Y	0	0	
X	0	0	0

ALPHA				
	SA	SB	SC	M
	-----	-----	-----	-----
Y				

1	4	5	6	7
	8			

ALPHA	
	X

1	9

BETA				
	SA	SB	SC	M
Y	-----	-----	-----	-----

SA		0	0	0	0
	0				
SB		0	0	0	0
	0				
SC		0	0	0	0
	0				
M		0	0	0	0
	0				
Y		0	0	0	11
	0				
X		0	0	0	0
	0				

BETA	
X	

SA	0
SB	0
SC	0
M	10
Y	12
X	0

PSI				
	SA	SB	SC	M
Y				

	-----	-----	-----	-----

SA	13			
SB	14	15		
SC	16	17	18	
M	0	0	0	19
Y	0	0	0	0
	20			
X	0	0	0	0
	0			

	PSI
	X

X	21

PARAMETER SPECIFICATION FOR THE ADDITIONAL PARAMETERS

	NEW/ADDITIONAL PARAMETERS			
	AB	BB	CPB	INDB
INDW				
	-----	-----	-----	-----

1	22	23	24	25
	26			

NEW/ADDITIONAL PARAMETERS	
INDWWRON	

1	27

STARTING VALUES FOR WITHIN

NU			
M		Y	X
-----		-----	-----
1	0.000	0.000	0.000

LAMBDA				
SA		SB	SC	M
-----		-----	-----	-----
Y	-----			

M	0.000	0.000	0.000	1.000
0.000				
Y	0.000	0.000	0.000	0.000
1.000				
X	0.000	0.000	0.000	0.000
0.000				

LAMBDA	
	X

M	0.000
Y	0.000
X	1.000

THETA			
	M	Y	X

M	0.000		
Y	0.000	0.000	
X	0.000	0.000	0.000

ALPHA				
	SA	SB	SC	M

Y				

1	0.000	0.000	0.000	0.000
	0.000			

ALPHA	
	X

1	0.000

BETA				
	SA	SB	SC	M
Y	-----	-----	-----	-----

SA	0.000	0.000	0.000	0.000
0.000				
SB	0.000	0.000	0.000	0.000
0.000				
SC	0.000	0.000	0.000	0.000
0.000				
M	0.000	0.000	0.000	0.000
0.000				
Y	0.000	0.000	0.000	0.000
0.000				
X	0.000	0.000	0.000	0.000
0.000				

BETA	
	X

SA	0.000
SB	0.000
SC	0.000
M	0.000
Y	0.000

X 0.000

PSI				
	SA	SB	SC	M
Y				
	-----	-----	-----	-----

SA	0.000			
SB	0.000	0.000		
SC	0.000	0.000	0.000	
M	0.000	0.000	0.000	0.500
Y	0.000	0.000	0.000	0.000
0.500				
X	0.000	0.000	0.000	0.000
0.000				

PSI	
	X

X	0.500

STARTING VALUES FOR BETWEEN

NU
M Y X

	-----	-----	-----
1	0.000	0.000	0.000

	LAMBDA			
	SA	SB	SC	M
Y	-----	-----	-----	-----

M	0.000	0.000	0.000	1.000
	0.000			
Y	0.000	0.000	0.000	0.000
	1.000			
X	0.000	0.000	0.000	0.000
	0.000			

	LAMBDA	
	X	

M	0.000	
Y	0.000	
X	1.000	

	THETA		
	M	Y	X
	-----	-----	-----
M	0.000		

Y	0.000	0.000	
X	0.000	0.000	0.000

ALPHA				
	SA	SB	SC	M
Y				
	-----	-----	-----	-----

1	0.000	0.000	0.000	0.000
0.000				

ALPHA	
	X

1	0.000

BETA				
	SA	SB	SC	M
Y				
	-----	-----	-----	-----

SA	0.000	0.000	0.000	0.000
0.000				
SB	0.000	0.000	0.000	0.000
0.000				
SC	0.000	0.000	0.000	0.000

	0.000				
M		0.000	0.000	0.000	0.000
	0.000				
Y		0.000	0.000	0.000	0.000
	0.000				
X		0.000	0.000	0.000	0.000
	0.000				

	BETA
	X

SA	0.000
SB	0.000
SC	0.000
M	0.000
Y	0.000
X	0.000

	PSI			
	SA	SB	SC	M
Y				
	-----	-----	-----	-----

SA	1.000			
SB	0.000	1.000		
SC	0.000	0.000	1.000	
M	0.000	0.000	0.000	0.500

Y	0.000	0.000	0.000	0.000
	0.500			
X	0.000	0.000	0.000	0.000
	0.000			

	PSI
	X

X	0.500

STARTING VALUES FOR THE ADDITIONAL PARAMETERS

NEW/ADDITIONAL PARAMETERS				
	AB	BB	CPB	INDB
INDW				
	-----	-----	-----	-----

1	0.500	0.500	0.500	0.500
	0.500			

NEW/ADDITIONAL PARAMETERS	
	INDWWRON

1	0.500

TECHNICAL 8 OUTPUT

E STEP	ITER	LOGLIKELIHOOD	ABS CHANGE	REL CHANGE	ALGORITHM
	1	-0.36926775D+05	0.0000000	0.0000000	EM
	2	-0.29361604D+05	7565.1706780	0.2048695	EM
	3	-0.29162226D+05	199.3780926	0.0067904	EM
	4	-0.29160301D+05	1.9252962	0.0000660	EM
	5	-0.29160280D+05	0.0209272	0.0000007	EM
	6	-0.29160280D+05	0.0002962	0.0000000	EM

TECHNICAL 8 OUTPUT FOR THE H1 MODEL

E STEP	ITER	LOGLIKELIHOOD	ABS CHANGE	REL CHANGE	ALGORITHM
	1	-0.39133672D+05	0.0000000	0.0000000	EM
	2	-0.39127270D+05	6.4021543	0.0001636	EM
	3	-0.39127242D+05	0.0278645	0.0000007	EM
	4	-0.39127241D+05	0.0007304	0.0000000	EM
	5	-0.39127241D+05	0.0000229	0.0000000	EM

SAVEDATA INFORMATION

Sample/H1/Pooled-Within Matrix

Save file

spw.txt

Save type COVARIANCE

Save format Free

Estimated Sigma **Between** Matrix

Save file

estsigb.txt

Save type COVARIANCE

Save format Free

Beginning Time: 12:21:10

Ending Time: 12:21:10

Elapsed Time: 00:00:00

MUTHEN & MUTHEN

3463 Stoner Ave.

Los Angeles, CA 90066

Tel: (310) 391-9971

Fax: (310) 391-8971

Web: www.StatModel.com

Support: Support@StatModel.com

References

- Blalock, H. M. (1984). Contextual-effects models: Theoretical and methodological issues. *Annual Review of Sociology*, 10, 353–372. doi: 10.1146/annurev.so.10.080184.002033
- Cohen, J., Cohen, P., West, S. G., & Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences* (3rd ed.). Mahwah, NJ: Erlbaum.
- Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, 16, 297–334. doi: 10.1007/BF02310555
- Enders, C. K. (2013). Centring predictors and contextual effects. In M. A. Scott, J. S. Simonoff, & B. P. Marx (Eds.), *The Sage handbook of multilevel modeling*. Los Angeles, CA: SAGE.
- Enders, C. K., & Tofighi, D. (2007). Centering predictor variables in cross-sectional multilevel models: A new look at an old issue. *Psychological Methods*, 12, 121–138.
- Kreft, I. G. G., de Leeuw, J., & Aiken, L. S. (1995). The effect of different forms of centering in hierarchical linear models. *Multivariate Behavioral Research*, 30, 1–21.
- Lüdtke, O., Marsh, H., Robitzsch, A., Trautwein, U., Asparouhov, T., & Muthén, B. (2008). The multilevel latent covariate model: A new, more reliable approach to group-level effects in contextual studies. *Psychological Methods*, 13, 203–229.
- Mauro, R. (1990). Understanding L.O.V.E. (left out variables error): A method for estimating the effects of omitted variables. *Psychological Bulletin*, 108, 314–329. doi: 10.1037/0033-2909.108.2.314
- Muthén, L. K., & Muthén, B. O. (1998–2012). *Mplus user's guide* (7th ed.) [Computer software manual]. Los Angeles, CA: Muthén & Muthén.
- Preacher, K. J., Zyphur, M. J., & Zhang, Z. (2010). A general multilevel SEM framework for assessing multilevel mediation. *Psychological Methods*, 15, 209–233. doi: 10.1037/a0020141
- R Development Core Team. (2014). *R: A language and environment for statistical*

computing [Computer software manual]. Vienna, Austria. Retrieved from
<http://www.R-project.org>

Rousseeuw, P. J., & Molenberghs, G. (1994). The shape of correlation matrices. *The American Statistician*, 48, 276–279. doi: 10.2307/2684832

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis : An introduction to basic and advanced multilevel modeling* (2nd ed.). Thousand Oaks, CA: SAGE.