**Supplemental Materials**

**Estimating Interaction Effects With Incomplete Predictor Variables**

**by C. K. Enders et al., 2013, *Psychological Methods***

**http://dx.doi.org/10.1037/a0035314**

**Appendix A**

**Mplus Syntax for Latent Variable Analysis 1: Default Model**

data:

file = reading.txt;

variable:

names = x z xz y;

usevariables = x z xz y;

missing = all(-99);

model:

! regression model;

y on x (g1)

z (g2)

xz (g3);

! specify a normal distribution for the predictors;

x; z (zvar); xz;

x z xz with x z xz;

! mean structure parameters;

[y];

[x]; [z] (zbar); [xz];

model constraint:

! compute simple slopes;

new(b1zmean b1zhigh b1zlow);

b1zmean = g1 + g3\*zbar;

b1zhigh = g1 + g3\*(zbar + sqrt(zvar));

b1zlow = g1 + g3\*(zbar - sqrt(zvar));

output:

sampstat;

**Appendix B**

**Mplus Syntax for Latent Variable Analysis 2: Free Mean Model**

data:

file = reading.txt;

variable:

names = x z xz y;

usevariables = x z xz y;

missing = all(-99);

model:

! factor loadings;

latentx by x@1 xz (tauz);

latentz by z@1 xz (taux);

latentxz by xz@1;

! measurement intercepts;

[x] (taux); [z] (tauz); [xz] (tauxz);

! constrain residual variances to zero;

x@0; z@0; xz@0;

! latent means;

[latentx] (xmean); [latentz] (zmean); [latentxz] (xzmean);

! exogenous variable variances and covariances;

latentx (xvar); latentz (zvar); latentxz;

latentx with latentz (xzcov);

latentx with latentxz; latentz with latentxz;

! structural regression model;

[y] (g0);

y on latentx (g1)

latentz (g2)

latentxz (g3);

model constraint:

! implement mean structure constraints;

tauxz = taux\*tauz;

xmean = 0;

zmean = 0;

! compute simple slopes;

new(b1low b1high);

b1high = g1 + g3\*(zmean + sqrt(zvar));

b1low = g1 + g3\*(zmean - sqrt(zvar));

output:

sampstat;

**Appendix C**

**Mplus Syntax for Latent Variable Analysis 3: Constrained Mean Model**

data:

file = reading.txt;

variable:

names = x z xz y;

usevariables = x z xz y;

missing = all(-99);

model:

! factor loadings;

latentx by x@1 xz (tauz);

latentz by z@1 xz (taux);

latentxz by xz@1;

! measurement intercepts;

[x] (taux); [z] (tauz); [xz] (tauxz);

! constrain residual variances to zero;

x@0; z@0; xz@0;

! latent means;

[latentx] (xmean); [latentz] (zmean); [latentxz] (xzmean);

! exogenous variable variances and covariances;

latentx (xvar); latentz (zvar); latentxz;

latentx with latentz (xzcov);

latentx with latentxz; latentz with latentxz;

! structural regression model;

[y] (g0);

y on latentx (g1)

latentz (g2)

latentxz (g3);

model constraint:

! implement mean structure constraints;

xmean = 0;

zmean = 0;

xzmean = xzcov;

! compute simple slopes;

new(b1low b1high);

b1high = g1 + g3\*(zmean + sqrt(zvar));

b1low = g1 + g3\*(zmean - sqrt(zvar));

output:

sampstat;

**Appendix D**

**Parameter Constraints for Probing Interaction Effects**

The centered latent variable models define and as deviation scores by constraining and equal to zero. The lower order coefficients are consistent with a centered solution (e.g., is the conditional effect of *x* on *y* at the mean of *z*). Altering the mean structure constraint redefines and provides a mechanism for estimating and testing different conditional effects.

First, consider the simple slope of *x* at one standard deviation above the mean of *z*. In a complete-data interaction analysis, centering provides a mechanism for implementing the so-called “pick-a-point” approach to estimating simple slopes (Aiken & West, 1991). For example, if a researcher wants to compute the simple slope of *y* on *x* at one standard deviation above the mean of *z*, the steps are as follows: (a) center *x* at its mean, (b) center *z* at a value one standard deviation above its mean, (c) compute the product term, and (d) estimate the regression model. When implementing this approach, the *z* deviation scores are

|  |  |  |
| --- | --- | --- |
|  |  | (D1) |

where *zc* is the centering constant (i.e., the score value equal to one standard deviation above the mean). Note that the expected value (i.e., mean) of D1 equals negative one times the standard deviation of *z* (i.e., ).

In the context of the latent variable interaction model, constraining to negative one times the square root of (i.e., negative one times the standard deviation of ) effectively defines as *z*dev. The resulting coefficient is the conditional effect of *x* on *y* at one standard deviation above the mean of *z*. In this parameterization, no longer estimates the mean of *z* but instead gives the value of *z* at one standard deviation above the mean (i.e., the centering constant, *zc*). Estimating the simple slope of *y* on *x* at one standard deviation below the mean of *z* follows the same process, except that *z* is centered at one standard deviation below its mean. This change reverses the sign of the constraint, such that . As before, corresponds to the centering constant (i.e., the value of *z* at one standard deviation below the mean).

More generally, the mean structure constraints can produce conditional effects for any centering constant. To illustrate, suppose that a researcher wants to estimate the simple slope of *y* on *x* at a clinically meaningful value of *z* (e.g., the mean of a particular normative group). Constraining is difficult when *z* is incomplete because the value of the constraint depends on the unknown *z* mean (i.e., ). Although a preliminary missing at random (MAR)-based analysis could provide the necessary estimate, constraining to the desired centering constant (i.e., = ) and estimating achieves the desired result.

**Appendix E**

**SAS Syntax for Multiple Imputation Analysis 1**

data reading;

infile 'c:\reading.txt';

input x z xz y;

if x = -99 then x = . ;

if z = -99 then z = . ;

if xz = -99 then xz = . ;

if y = -99 then y = . ;

run;

/\* generate imputations \*/

proc mi data = reading seed = 25384 nimpute = 50 out = midata;

var x z xz y;

mcmc nbiter = 2000 niter = 2000;

run;

/\* estimate regression model \*/

proc reg data = midata outest = regparams covout noprint;

model y = x z xz;

by \_imputation\_;

run;

/\* pool raw score estimates \*/

proc mianalyze data = regparams edf = 70;

modeleffects intercept x z xz;

run;

/\* reshape regparams data set for pooling simple slopes \*/

data simpleslopes;

set regparams;

if \_type\_ = 'PARMS' then g0 = intercept;

if \_type\_ = 'PARMS' then g1 = x;

if \_type\_ = 'PARMS' then g2 = z;

if \_type\_ = 'PARMS' then g3 = xz;

if \_type\_ = 'COV' & \_name\_ = 'Intercept' then varg0

= intercept;

if \_type\_ = 'COV' & \_name\_ = 'x' then varg1 = x;

if \_type\_ = 'COV' & \_name\_ = 'z' then varg2 = z;

if \_type\_ = 'COV' & \_name\_ = 'xz' then varg3 = xz;

if \_type\_ = 'COV' & \_name\_ = 'x' then covg1g3 = xz;

if \_type\_ = 'COV' & \_name\_ = 'z' then covg2g3 = xz;

keep \_imputation\_ g0 g1 g2 g3 varg0 varg1 varg2 varg3

covg1g3 covg2g3;

run;

proc means data = simpleslopes noprint;

var g0 g1 g2 g3 varg0 varg1 varg2 varg3 covg1g3 covg2g3;

by \_imputation\_;

output out = simpleslopes (drop = \_type\_ \_freq\_)

mean = g0 g1 g2 g3 varg0 varg1 varg2 varg3 covg1g3 covg2g3;

run;

/\* estimate variable means for simple slopes \*/

proc means data = midata noprint;

var x z;

by \_imputation\_;

output out = means (drop = \_type\_ \_freq\_)

mean = xmean zmean std = xstd zstd;

run;

/\* merge means into simple slope file \*/

data simpleslopes;

merge simpleslopes means;

by \_imputation\_;

run;

/\* compute simple slopes and standard errors \*/

data simpleslopes;

set simpleslopes;

zhigh = zmean + zstd;

zmean = zmean;

zlow = zmean - zstd;

b1high = g1 + g3\*zhigh;

b1mean = g1 + g3\*zmean;

b1low = g1 + g3\*zlow;

b1highse = sqrt(varg1 + 2\*zhigh\*covg1g3 + (zhigh\*\*2)\*varg3);

b1meanse = sqrt(varg1 + 2\*zmean\*covg1g3 + (zmean\*\*2)\*varg3);

b1lowse = sqrt(varg1 + 2\*zlow\*covg1g3 + (zlow\*\*2)\*varg3);

run;

/\* pool simple slopes \*/

proc mianalyze data = simpleslopes edf = 70;

modeleffects b1high b1mean b1low;

stderr b1highse b1meanse b1lowse;

run;

**Appendix F**

**SAS and SPSS Syntax for Multiple Imputation Analysis 2: Free Mean Centering**

**SAS Syntax for Multiple Imputation Analysis 2: Free Mean Centering**

data reading;

infile 'c:\reading.txt';

input x z xz y;

if x = -99 then x = . ;

if z = -99 then z = . ;

if xz = -99 then xz = . ;

if y = -99 then y = . ;

run;

/\* generate imputations \*/

proc mi data = reading seed = 25384 nimpute = 50 out = midata;

var x z xz y;

mcmc nbiter = 2000 niter = 2000;

run;

/\* estimate variable means for centering \*/

proc means data = midata noprint;

var x z;

by \_imputation\_;

output out = means (drop = \_type\_ \_freq\_)

mean = xmean zmean std = xstd zstd;

run;

/\* add means to the imputed data \*/

data midata;

merge midata means;

by \_imputation\_;

run;

/\* rescale imputed predictors \*/

/\* simple slope of x at mean of z \*/

data midata;

set midata;

cx = xmean;

cz = zmean;

xdev = x - cx;

zdev = z - cz;

xzdev = xz - x\*cz - z\*cx + cx\*cz;

run;

/\* estimate regression model \*/

proc reg data = midata outest = regout covout noprint;

model y = xdev zdev xzdev;

by \_imputation\_;

run;

/\* pool simple slopes \*/

proc mianalyze data = regout edf = 70;

modeleffects intercept xdev zdev xzdev;

run;

/\* rescale imputed predictors \*/

/\* simple slope of x at one std. dev. above z mean \*/

data midata;

set midata;

cx = xmean;

cz = zmean + zstd;

xdev = x - cx;

zdev = z - cz;

xzdev = xz - x\*cz - z\*cx + cx\*cz;

run;

/\* estimate regression model \*/

proc reg data = midata outest = regout covout noprint;

model y = xdev zdev xzdev;

by \_imputation\_;

run;

/\* pool simple slopes \*/

proc mianalyze data = regout edf = 70;

modeleffects intercept xdev zdev xzdev;

run;

/\* rescale imputed predictors \*/

/\* simple slope of x at one std. dev. below z mean \*/

data midata;

set midata;

cx = xmean;

cz = zmean - zstd;

xdev = x - cx;

zdev = z - cz;

xzdev = xz - x\*cz - z\*cx + cx\*cz;

run;

/\* estimate regression model \*/

proc reg data = midata outest = regout covout noprint;

model y = xdev zdev xzdev;

by \_imputation\_;

run;

/\* pool simple slopes \*/

proc mianalyze data = regout edf = 70;

modeleffects intercept xdev zdev xzdev;

run;

**SPSS Syntax for Multiple Imputation Analysis 2: Free Mean Centering**

data list free file = 'c:\reading.txt'

/x z xz y.

recode x z xz y (-99 = sysmis).

exe.

\* generate imputations.

dataset declare midata.

multiple imputation x z xz y

/impute method = fcs maxiter = 2000 nimputations = 50

/outfile imputations = midata.

\* retain only imputed data sets.

dataset activate midata.

select if imputation\_ ge 1.

exe.

\* add means to the data.

dataset activate midata.

aggregate

/outfile = \* mode = addvariables

/break = imputation\_

/xmean = mean(x)

/zmean = mean(z)

/xstd = sd(x)

/zstd = sd(z).

\* rescale imputed predictors.

\* simple slope of x at mean of z.

dataset activate midata.

compute cx = xmean.

compute cz = zmean.

compute xdev = x - cx.

compute zdev = z - cz.

compute xzdev = xz - x\*cz - z\*cx + cx\*cz.

execute.

\* activate pooling facility.

split file layered by imputation\_.

\* estimate regression model.

regression

/dependent y

/method = enter xdev zdev xzdev.

\* rescale imputed predictors.

\* simple slope of x at one std. dev. above z mean.

dataset activate midata.

compute cx = xmean.

compute cz = zmean + zstd.

compute xdev = x - cx.

compute zdev = z - cz.

compute xzdev = xz - x\*cz - z\*cx + cx\*cz.

execute.

\* estimate regression model.

regression

/dependent y

/method = enter xdev zdev xzdev.

\* rescale imputed predictors.

\* simple slope of x at one std. dev. below z mean.

dataset activate midata.

compute cx = xmean.

compute cz = zmean - zstd.

compute xdev = x - cx.

compute zdev = z - cz.

compute xzdev = xz - x\*cz - z\*cx + cx\*cz.

execute.

\* estimate regression model.

regression

/dependent y

/method = enter xdev zdev xzdev.

**Appendix G**

**SAS and SPSS Syntax for Multiple Imputation Analysis 3: Constrained Mean Centering**

**SAS Syntax for Multiple Imputation Analysis 3: Constrained Mean Centering**

data reading;

infile 'c:\reading.txt';

input x z xz y;

if x = -99 then x = . ;

if z = -99 then z = . ;

if xz = -99 then xz = . ;

if y = -99 then y = . ;

run;

/\* generate imputations \*/

proc mi data = reading seed = 25384 nimpute = 50 out = midata;

var x z xz y;

mcmc nbiter = 2000 niter = 2000;

run;

/\* estimate variable means for centering \*/

proc means data = midata noprint;

var x z xz;

by \_imputation\_;

output out = means (drop = \_type\_ \_freq\_) mean = xmean zmean xzmean std = xstd zstd xzstd;

run;

/\* estimate covariance between x and z for centering \*/

proc corr data = midata cov outp = covxz noprint;

var x z;

by \_imputation\_;

run;

/\* restructure covariance matrix data file \*/

data covxz;

set covxz;

where \_type\_ = 'COV' and \_name\_ = 'x';

rename z = covxz;

drop \_type\_ \_name\_ x;

run;

/\* add centering parameters to data \*/

data midata;

merge midata means covxz;

by \_imputation\_;

run;

/\* rescale imputed predictors \*/

/\* simple slope of x at mean of z \*/

data midata;

set midata;

cx = xmean;

cz = zmean;

xdev = x - cx;

zdev = z - cz;

xzdev = xz - xzmean + covxz + xmean\*zmean - x\*cz - z\*cx

+ cx\*cz;

run;

/\* estimate regression model \*/

proc reg data = midata outest = regout covout noprint;

model y = xdev zdev xzdev;

by \_imputation\_;

run;

/\* pool simple slopes \*/

proc mianalyze data = regout edf = 70;

modeleffects intercept xdev zdev xzdev;

run;

/\* rescale imputed predictors \*/

/\* simple slope of x at one std. dev. above z mean \*/

data midata;

set midata;

cx = xmean;

cz = zmean + zstd;

xdev = x - cx;

zdev = z - cz;

xzdev = xz - xzmean + covxz + xmean\*zmean - x\*cz - z\*cx

+ cx\*cz;

run;

/\* estimate regression model \*/

proc reg data = midata outest = regout covout noprint;

model y = xdev zdev xzdev;

by \_imputation\_;

run;

/\* pool simple slopes \*/

proc mianalyze data = regout edf = 70;

modeleffects intercept xdev zdev xzdev;

run;

/\* rescale imputed predictors \*/

/\* simple slope of x at one std. dev. below z mean \*/

data midata;

set midata;

cx = xmean;

cz = zmean - zstd;

xdev = x - cx;

zdev = z - cz;

xzdev = xz - xzmean + covxz + xmean\*zmean - x\*cz - z\*cx

+ cx\*cz;

run;

/\* estimate regression model \*/

proc reg data = midata outest = regout covout noprint;

model y = xdev zdev xzdev;

by \_imputation\_;

run;

/\* pool simple slopes \*/

proc mianalyze data = regout edf = 70;

modeleffects intercept xdev zdev xzdev;

run;

**SPSS Syntax for Multiple Imputation Analysis 3: Constrained Mean Centering**

data list free file = 'c:\reading.txt'

/x z xz y.

recode x z xz y (-99 = sysmis).

exe.

\* generate imputations.

dataset declare midata.

multiple imputation x z xz y

/impute method = fcs maxiter = 2000 nimputations = 50

/outfile imputations = midata.

\* retain only imputed data sets.

dataset activate midata.

select if imputation\_ ge 1.

exe.

\* add means to the data.

dataset activate midata.

aggregate

/outfile = \* mode = addvariables

/break = imputation\_

/xmean = mean(x)

/zmean = mean(z)

/xzmean = mean(xz)

/xstd = sd(x)

/zstd = sd(z).

\* estimate covariance between x and z for centering.

dataset activate midata.

correlations x z

/matrix = out(\*).

mconvert.

\* restructure covariance matrix data file.

select if rowtype\_= 'COV'.

dataset name covxz.

filter off.

use all.

select if (varname\_ = "x").

execute.

rename variables (z = covxz).

delete variables rowtype\_ varname\_ x.

\* add centering parameters to data.

dataset activate midata.

match files

/file = \*

/table = 'covxz'

/by imputation\_.

execute.

\* rescale imputed predictors.

\* simple slope of x at mean of z.

dataset activate midata.

compute cx = xmean.

compute cz = zmean.

compute xdev = x - cx.

compute zdev = z - cz.

compute xzdev = xz - xzmean + covxz + xmean\*zmean - x\*cz –

z\*cx + cx\*cz;

execute.

\* activate pooling facility.

split file layered by imputation\_.

\* estimate regression model.

regression

/dependent y

/method = enter xdev zdev xzdev.

\* rescale imputed predictors.

\* simple slope of x at one std. dev. above z mean.

dataset activate midata.

compute cx = xmean.

compute cz = zmean + zstd.

compute xdev = x - cx.

compute zdev = z - cz.

compute xzdev = xz - xzmean + covxz + xmean\*zmean - x\*cz –

z\*cx + cx\*cz;

execute.

\* estimate regression model.

regression

/dependent y

/method = enter xdev zdev xzdev.

\* rescale imputed predictors.

\* simple slope of x at one std. dev. below z mean.

dataset activate midata.

compute cx = xmean.

compute cz = zmean - zstd.

compute xdev = x - cx.

compute zdev = z - cz.

compute xzdev = xz - xzmean + covxz + xmean\*zmean - x\*cz –

z\*cx + cx\*cz;

execute.

\* estimate regression model.

regression

/dependent y

/method = enter xdev zdev xzdev.

**Appendix H**

**Latent Variable Specification and Post-Imputation Centering for Quadratic Effects**

This appendix considers the latent variable specification and post-imputation centering for the following polynomial regression model.

|  |  |  |
| --- | --- | --- |
|  |  | (H1) |

First, consider the latent variable model. The measurement model for *x* is the same as that in Equation 2 of the main article. We establish a measurement model for the *x*2 quadratic term by squaring the *x* equation, as follows.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Replacing the product with a single latent variable and adding a residual term gives the measurement model for *x*2.

|  |  |  |
| --- | --- | --- |
|  |  | (H2) |

Consistent with the lower order predictor, the squared term has a loading of one on its corresponding latent variable , and its residual is fixed at zero. The default structural equation model (SEM) constrains the measurement intercepts to zero, in which case the measurement model reduces to an identity between the latent and manifest variable, as follows.

|  |  |  |
| --- | --- | --- |
|  |  | (H3) |

The structural regression is the same as Equation H1 but replaces manifest variables with latent variables, as follows.

|  |  |  |
| --- | --- | --- |
|  |  | (H4) |

As before, the SEM assumes that follows a normal distribution with a zero mean and a residual variance . The model further assumes that the latent exogenous variables (and thus the manifest predictors) are normally distributed with a mean vector and a covariance matrix . Again, the multivariate normality assumption is important for missing data handling.

The default latent variable is equivalent to a quadratic regression model with raw score (uncentered) variables. Centered versions of the quadratic latent variable model mimic the free and constrained mean models for interaction effects. Centering the lower order latent variable is identical to the interaction model (i.e., estimate the measurement intercept and constrain the latent mean to zero). To understand the constraints for , note that the expected value of a squared term is as follows.

|  |  |  |
| --- | --- | --- |
|  |  | (H5) |

In the centered models, the *x*2 measurement intercept captures the square of the *x* mean, leaving the quadratic latent mean to estimate the variance of *x*. This parameter can be freely estimated or constrained during estimation. The free mean model requires the following specifications: (a) the intercept is held equal to the square of the *x* measurement intercept, (b) the cross-loading of *x*2 on is set equal to , and (c) the latent variable mean is freely estimated. Alternatively, the constrained mean model estimates the *x*2 measurement intercept and constrains the latent mean. The model requires the following specifications: (a) the *x*2 measurement intercept is freely estimated, (b) the cross-loading of *x*2 on is set equal to , and (c) the quadratic latent variable mean is constrained at , the variance of the lower order latent variable (i.e., the variance of *x*).

Turning to multiple imputation, post-imputation centering for a squared term consists of the following steps: (a) form a power term by squaring the raw *x* scores, (b) impute all variables on their raw score metrics (including the squared term), and (c) rescale *x* and *x*2 following imputation. The centering equation for *x* is the same as Equation 15 in the main article. The free mean centering expression for *x*2 is

|  |  |  |
| --- | --- | --- |
|  |  | (H6) |

where is the imputed quadratic term, and is the conditioning value of *x* (e.g., the mean). The corresponding fixed mean centering expression is

|  |  |  |
| --- | --- | --- |
|  |  | (H7) |

where is the mean of the imputed squared term in data set *i*, and is the squared mean of *x*.