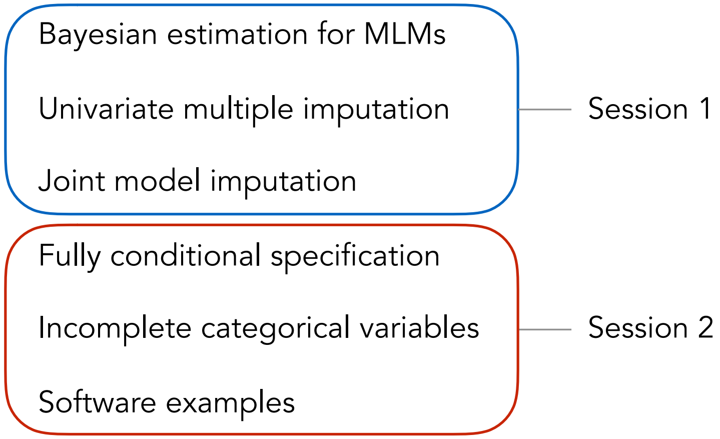
Multiple Imputation for Multilevel Data

Craig K. Enders
Brian T. Keller
University of California - Los Angeles
Department of Psychology

Work supported by IES award R305D150056

Overview



Bayesian estimation for MLMs

Univariate multiple imputation

Joint model imputation

Session 1

Fully conditional specification

Incomplete categorical variables

Software examples

Session 2

Why Imputation?

Dedicated multilevel programs restricts maximum likelihood estimation to incomplete outcomes

Multilevel SEM software is more flexible but typically imposes normality on incomplete predictors and may perform poorly in some cases

Imputation is flexible (e.g., mixtures of categorical and continuous variables are no problem)

Model Notation

Two-level model with observation *i* nested in cluster *j* (e.g., student *i* in school *j*)

$$y_{ij} = \gamma_0 + \gamma_1 x_{ij} + \gamma_2 w_j + u_{0j} + u_{1j} x_{ij} + \varepsilon_{ij}$$

$$\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim \mathrm{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_u) \qquad \varepsilon_{ij} \sim \mathrm{N}\left(0, \sigma_\varepsilon^2\right)$$

Bayesian Estimation For Multilevel Models

Bayesian Estimation And Imputation

Bayesian estimation (e.g., Gibbs sampler) is the mathematical machinery for imputation

Each algorithmic cycle is a complete-data Bayes analysis followed by an imputation step

A multilevel model generates imputations

Analysis Example

Random intercept model with a level-1 predictor

$$y_{ij} = \gamma_0 + \gamma_1 \, x_{ij} + u_{0j} + \varepsilon_{ij}$$

$$u_{0j} \sim \mathrm{N}(0, \boldsymbol{\Sigma}_u) \qquad \varepsilon_{ij} \sim \mathrm{N}\left(0, \sigma^2_\varepsilon\right)$$

Assume complete data, estimation steps do not change with missing values

Bayesian Paradigm

The Bayesian framework views parameters and level-2 residuals as random variables that follow a probability distribution (a posterior)

$$\theta = \left\{\boldsymbol{\gamma}, \boldsymbol{u}_j, \sigma^2_\varepsilon, \boldsymbol{\Sigma}_u\right\}$$

$$\boldsymbol{P}(\theta|\text{data}) \propto \boldsymbol{P}(\text{data}|\theta)\boldsymbol{P}(\theta)$$

Posterior

Likelihood

Prior

Gibbs Sampler

An iterative Gibbs sampler algorithm estimates quantities in θ one at a time, treating all other variables as known

Monte Carlo simulation “samples” parameter values from their conditional distributions

Repeating the sampling steps many times yields a distribution of each estimate

Gibbs Sampler Steps For One Iteration

Estimate regression coefficients

Estimate level-2 random effects

Estimate within-cluster residual variance

Estimate level-2 covariance matrix

Estimating Regression Coefficients

Regression coefficients are drawn from a multivariate normal distribution that conditions on random effects, variances, and the data

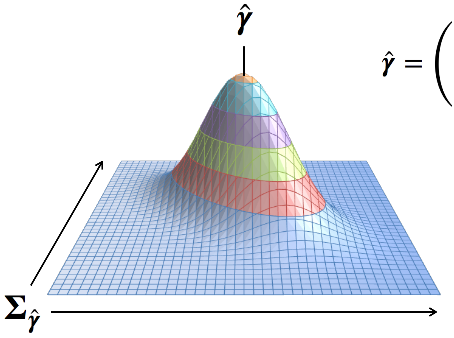
Current iteration

Previous iteration

$$\dot{\boldsymbol{\gamma}}^{(t)} \sim P\left(\boldsymbol{\gamma} \mid \overbrace{\dot{\boldsymbol{u}}_j^{(t-1)}, \dot{\sigma}_\varepsilon^{2(t-1)}, \dot{\boldsymbol{\Sigma}}_u^{(t-1)}}, \text{data}\right)$$

Conditional Distribution

$$\dot{\boldsymbol{\gamma}} \sim \text{MVN}\left(\hat{\boldsymbol{\gamma}}, \boldsymbol{\Sigma}_{\hat{\boldsymbol{\gamma}}}\right)$$


$\hat{\boldsymbol{\gamma}}$
$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\gamma}}}$

$$\hat{\boldsymbol{\gamma}} = \left(\sum_{j=1}^{J} \boldsymbol{x}_j^T \boldsymbol{x}_j\right)\left(\sum_{j=1}^{J} \boldsymbol{x}_j^T \left(\boldsymbol{y}_j - \boldsymbol{z}_j \dot{\boldsymbol{u}}_j\right)\right)$$

$$\boldsymbol{\Sigma}_{\hat{\boldsymbol{\gamma}}} = \left(\sum_{j=1}^{J} \boldsymbol{x}_j^T \boldsymbol{x}_j \dot{\sigma}_\varepsilon^{-2}\right)^{-1}$$

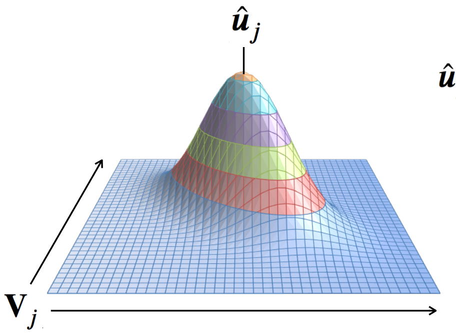
Estimating Level-2 Random Effects

Level-2 random effects are drawn from a multivariate normal distribution that conditions on the coefficients, variances, and the data

$$\dot{\boldsymbol{u}}_j^{(t)} \sim P\left(\boldsymbol{u}_j \mid \overset{\text{Updated estimates}}{\dot{\boldsymbol{\gamma}}^{(t)}}, \overbrace{\dot{\sigma}_\varepsilon^{2(t-1)}, \dot{\boldsymbol{\Sigma}}_u^{(t-1)}}^{\text{Previous iteration}}, \text{data}\right)$$

Conditional Distribution

$$\dot{\boldsymbol{u}}_j \sim \text{MVN}\left(\hat{\boldsymbol{u}}_j, \mathbf{V}_j\right)$$



$$\hat{\boldsymbol{u}}_j = \mathbf{V}_j\left(\dot{\sigma}_\varepsilon^{-2}\boldsymbol{z}_j^T\right)\left(\boldsymbol{y}_j - \boldsymbol{x}_j\dot{\boldsymbol{\gamma}}\right)$$

$$\mathbf{V}_j = \left(\dot{\boldsymbol{\Sigma}}_u^{-1} + \boldsymbol{z}_j^T\boldsymbol{z}_j\dot{\sigma}_\varepsilon^{-2}\right)^{-1}$$

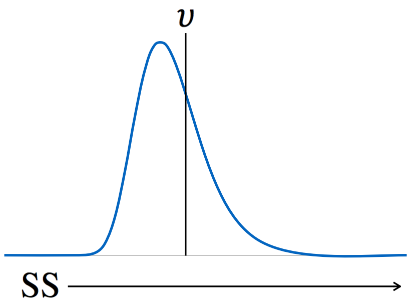
Estimating The Residual Variance

The within-cluster residual variance is drawn from an inverse Wishart distribution that conditions on the previous coefficients, random effects, level-2 covariance matrix, and the data

$$\dot{\sigma}_\varepsilon^{2(t)} \sim P\left(\boldsymbol{u}_j \mid \dot{\boldsymbol{\gamma}}^{(t)}, \dot{\boldsymbol{u}}_j^{(t)}, \dot{\boldsymbol{\Sigma}}_u^{(t-1)}, \text{data}\right)$$

Conditional Distribution

$$\dot{\sigma}_\varepsilon^2 \sim \text{IW}(\text{SS}, \upsilon)$$



$$\boldsymbol{\varepsilon}_j = \boldsymbol{y}_j - \left(\boldsymbol{x}_j\dot{\boldsymbol{\gamma}} + \boldsymbol{z}_j\dot{\boldsymbol{u}}_j\right)$$

$$\text{SS} = \sum_{j=1}^{J}\boldsymbol{\varepsilon}_j^T\boldsymbol{\varepsilon}_j + \text{SS}_{prior}$$

$$\upsilon = J + \upsilon_{prior}$$

Estimating Level-2 Covariance Matrix

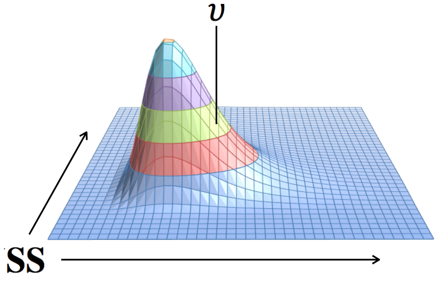
The level-2 covariance matrix is sampled from an inverse Wishart distribution that conditions on the previous coefficients, random effects, residual variance, and the data

$$\dot{\boldsymbol{\Sigma}}_u^{(t)} \sim P\left(\boldsymbol{\Sigma}_u \mid \dot{\boldsymbol{\gamma}}^{(t)}, \dot{\boldsymbol{u}}_j^{(t)}, \dot{\sigma}_\varepsilon^{2(t)}, \text{data}\right)$$

Iteration t is complete, start anew at iteration $t+1$

Conditional Distribution

$$\dot{\boldsymbol{\Sigma}}_u \sim \text{IW}(\mathbf{SS}, \upsilon)$$



$$\mathbf{SS} = \sum_{j=1}^{J} \dot{\boldsymbol{u}}_j^T \dot{\boldsymbol{u}}_j + \mathbf{SS}_{prior}$$

$$\upsilon = J + \upsilon_{prior}$$

Univariate Multiple Imputation

Multilevel Imputation

Imputation uses a model with an incomplete variable regressed on complete variables

Bayesian estimation steps are applied to the filled-in data from the previous iteration

Model parameters and level-2 residuals define a distribution from which imputations are sampled

Analysis And Imputation Models

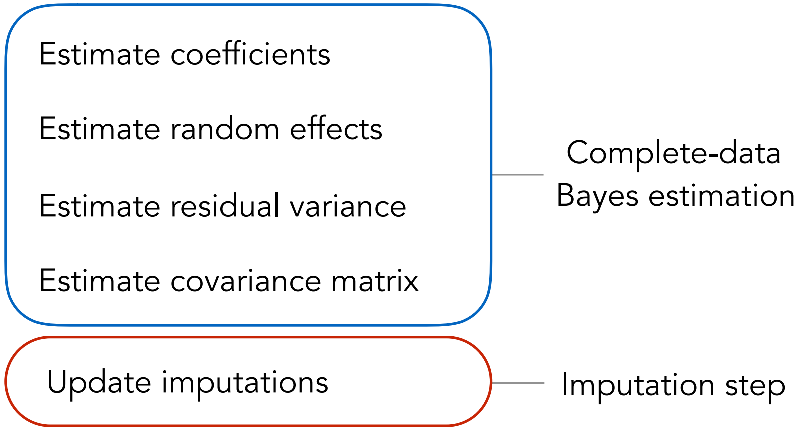
Random intercept analysis model with an incomplete predictor

$$y_{ij} = \gamma_0 + \gamma_1 x_{ij} + u_{0j} + \varepsilon_{ij}$$

Random intercept imputation model with the incomplete predictor as the outcome

$$x_{ij} = \gamma_0 + \gamma_1 \, y_{ij} + u_{0j} + \varepsilon_{ij}$$

Gibbs Sampler Steps



Estimate coefficients

Estimate random effects

Estimate residual variance

Estimate covariance matrix

Complete-data Bayes estimation

Update imputations

Imputation step

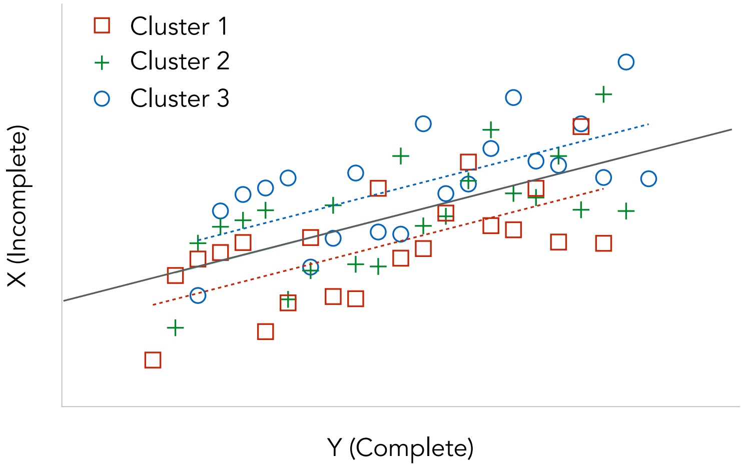
Distribution Of Missing Values

A normal distribution generates imputations, with center equal to the predicted value for observation *i* in cluster *j* and spread equal to the within-cluster residual variance

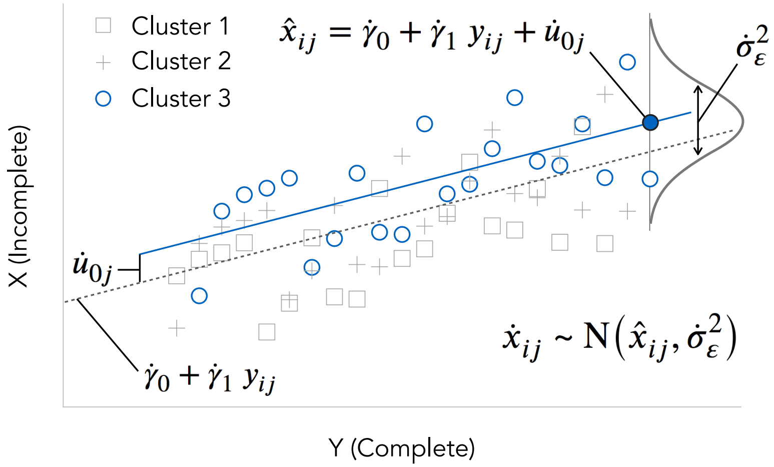
$$\dot{x}_{ij} \sim \mathrm{N}\left(\hat{x}_{ij}, \dot{\sigma}_{\varepsilon}^{2}\right)$$

$$\hat{x}_{ij} = \dot{\gamma}_0 + \dot{\gamma}_1 \, y_{ij} + \dot{u}_{0j}$$

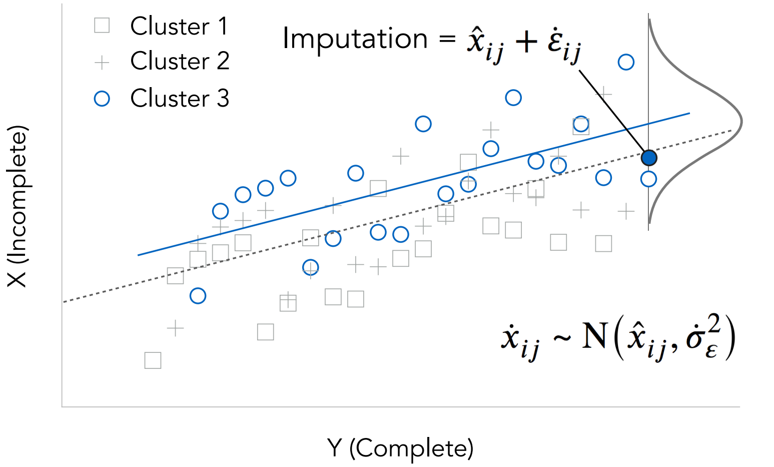
Random Intercept Imputation Model



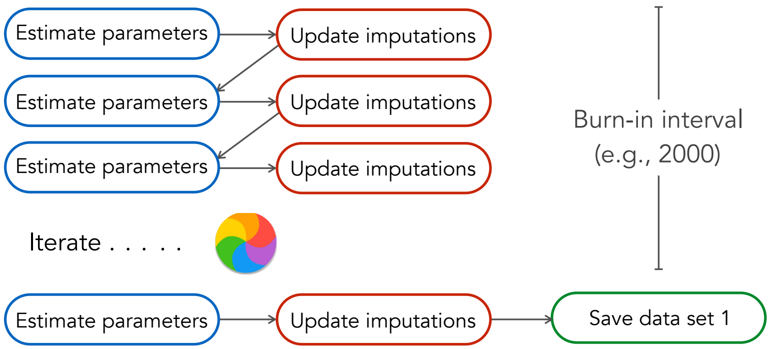
Random Intercept Imputation Model


□ Cluster 1
+ Cluster 2
○ Cluster 3
$\hat{x}_{ij} = \dot{\gamma}_0 + \dot{\gamma}_1 y_{ij} + \dot{u}_{0j}$
$\dot{\sigma}^2_\varepsilon$
$\dot{u}_{0j}$
$\dot{\gamma}_0 + \dot{\gamma}_1 y_{ij}$
$\dot{x}_{ij} \sim \mathrm{N}\left(\hat{x}_{ij}, \dot{\sigma}^2_\varepsilon\right)$
X (Incomplete)
Y (Complete)

Random Intercept Imputation Model


□ Cluster 1
+ Cluster 2
○ Cluster 3
Imputation = $\hat{x}_{ij} + \dot{\varepsilon}_{ij}$
$\dot{x}_{ij} \sim \mathrm{N}\left(\hat{x}_{ij}, \dot{\sigma}^2_\varepsilon\right)$
X (Incomplete)
Y (Complete)

Burn-In Period



Estimate parameters → Update imputations

Estimate parameters → Update imputations

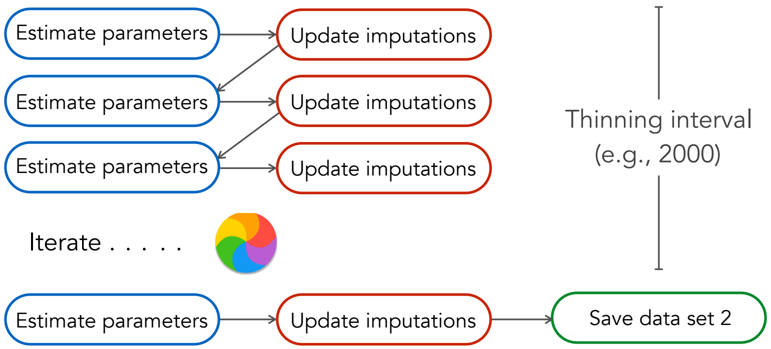
Estimate parameters → Update imputations

Iterate . . . . .

Estimate parameters → Update imputations → Save data set 1

Burn-in interval (e.g., 2000)

Thinning Interval



Estimate parameters → Update imputations

Estimate parameters → Update imputations

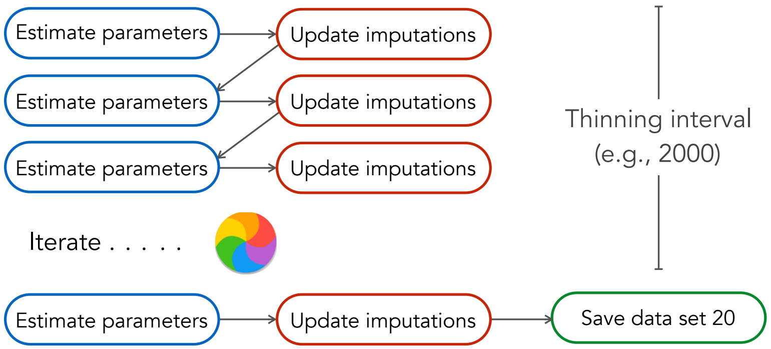
Estimate parameters → Update imputations

Iterate . . . . .

Estimate parameters → Update imputations → Save data set 2

Thinning interval (e.g., 2000)

Repeat Until Finished …



Estimate parameters → Update imputations

Estimate parameters → Update imputations

Estimate parameters → Update imputations

Iterate . . . . .

Estimate parameters → Update imputations → Save data set 20

Thinning interval (e.g., 2000)

Analysis And Pooling

The analysis model is fit to each data set, and the arithmetic average of the *M* estimates is the multiple imputation point estimate

$$\bar{\theta} = \frac{1}{M}\sum_{m=1}^{M}\hat{\theta}^{(m)}$$

Pooling assumes a normal sampling distribution

Pooling Standard Errors

$$\text{var}_w = \frac{1}{M}\sum_{m=1}^{M}\text{var}(\hat{\theta})^{(m)}$$

Average sampling variance

$$\text{var}_b = \frac{1}{M-1}\sum_{m=1}^{M}(\hat{\theta}^{(m)} - \bar{\theta})^2$$

Variance across imputations

$$SE = \sqrt{\text{var}_w + \text{var}_b + \frac{\text{var}_b}{M}}$$

Standard error

Multivariate Missing Data

Joint model imputation uses multivariate regression to impute the set of missing variables

Fully conditional specification imputes variables one at a time in a sequence

Both are multilevel extensions of major single-level imputation frameworks

Multivariate Imputation With The Joint Modeling Framework

Joint Model Imputation

Two forms:

1) Multivariate regression model with incomplete variables regressed on complete variables

2) Empty model treating all variables as outcomes

Available in Mplus, MLwiN, and R packages (e.g., jomo, pan, mlmmm)

Random Intercept Analysis Model

Two-level random intercept analysis with continuous level-1 and level-2 predictors

$$y_{ij} = \gamma_0 + \gamma_1 x_{ij} + \gamma_2 w_j + u_{0j} + \varepsilon_{ij}$$

All variables have missing data

Imputation Model

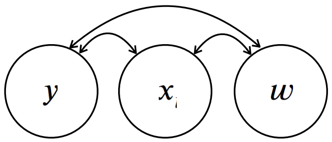
$$y_{ij} = \gamma_{0(y)} + u_{0j(y)} + \varepsilon_{ij(y)}$$

$$x_{ij} = \gamma_{0(x)} + u_{0j(x)} + \varepsilon_{ij(x)}$$

$$w_j = \gamma_{0(w)} + u_{0j(w)}$$

$$\begin{pmatrix} u_{0j(y)} \\ u_{0j(x)} \\ u_{0j(w)} \end{pmatrix} \sim \mathrm{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_u) \qquad \begin{pmatrix} \varepsilon_{ij(y)} \\ \varepsilon_{ij(x)} \\ \varepsilon_{ij(w)} \end{pmatrix} \sim \mathrm{MVN}(\mathbf{0}, \boldsymbol{\Sigma}_\varepsilon)$$

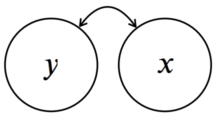
Covariance Structure



$$\boldsymbol{\Sigma}_u = \begin{pmatrix} \sigma^2_{u_{(y)}} & \sigma_{u_{(y,x)}} & \sigma_{u_{(y,w)}} \\ \sigma_{u_{(x,y)}} & \sigma^2_{u_{(x)}} & \sigma_{u_{(x,w)}} \\ \sigma_{u_{(w,y)}} & \sigma_{u_{(w,x)}} & \sigma^2_{u_{(w)}} \end{pmatrix}$$

Level-2

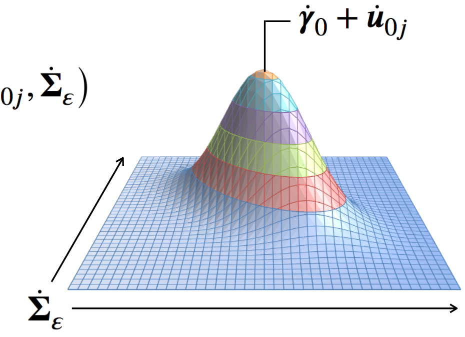
Level-1



$$\boldsymbol{\Sigma}_\varepsilon = \begin{pmatrix} \sigma^2_{\varepsilon_{(y)}} & \sigma_{\varepsilon_{(y,x)}} & 0 \\ \sigma_{\varepsilon_{(x,y)}} & \sigma^2_{\varepsilon_{(x)}} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Imputation Step

$$\begin{pmatrix} \dot{y}_{ij}^{(t)} \\ \dot{x}_{ij}^{(t)} \\ \dot{w}_j^{(t)} \end{pmatrix} \sim \text{MVN}\left(\dot{\boldsymbol{\gamma}}_0 + \dot{\boldsymbol{u}}_{0j}, \dot{\boldsymbol{\Sigma}}_\varepsilon\right)$$


$\dot{\boldsymbol{\gamma}}_0 + \dot{\boldsymbol{u}}_{0j}$
$\dot{\boldsymbol{\Sigma}}_\varepsilon$

Compatibility Of Imputation And Analysis

The imputation model is more flexible than the analysis model because it allows level-1 and level-2 covariance matrices to freely vary

The analysis model assumes a common slope

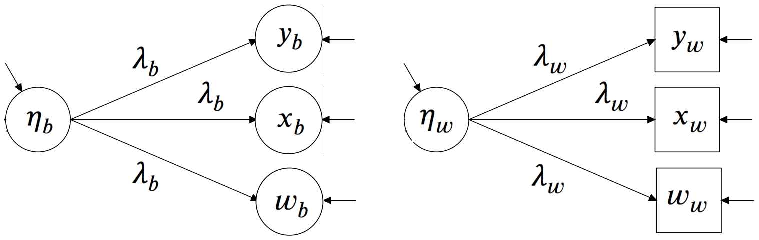
Imputations are appropriate for random intercept analyses that partition relations into within- and between-cluster parts

Compatible Analysis Models

Contextual effects analyses

$$y_{ij} = \gamma_0 + \gamma_1 x_{ij} + \gamma_2 \bar{x}_j + \gamma_3 w_j + u_{0j} + \varepsilon_{ij}$$

Multilevel SEM



R Package jomo

```
# load packages
library (jomo)

# read raw data
dat <- read.table("~/desktop/examples/ridata.csv", sep = ",")
names(dat) = c("cluster", "av1", "av2", "y", "x","w")
dat[dat == 999] <- NA

# jomo imputation
set.seed(90291)
dat$icept <- 1
l1miss <- c("y", "x")
l2miss <- c("w")
l1complete <- c("icept")
l2complete <- c("icept")
impdata <- jomo(dat[l1miss], Y2 = dat[l2miss], X = dat[l1complete],
  X2 = dat[l2complete], clus = dat$cluster,
  nburn = 2000, nbetween = 2000, nimp = 20, meth = "common")
```

Mplus

```
data:
file = ridata.csv;
variable:
names = cluster av1 av2 y x w;
usevariables = av1 av2 y x w;
missing = all(999);
analysis:
type = basic;
bseed = 90291;
data imputation:
impute = y x w;
ndatasets = 20;
save = imp*.dat;
thin = 1000;
output:
tech8;
```

Simulation Study

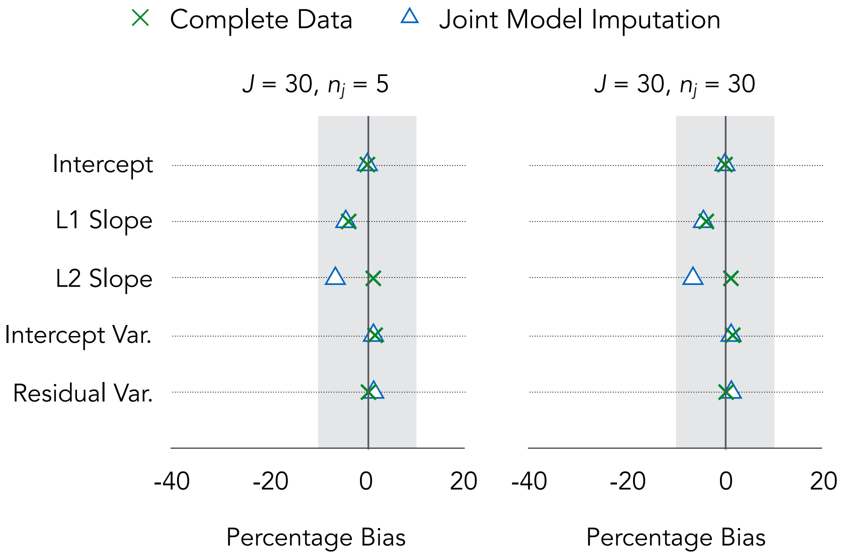
Random intercept model with 1000 replications

ICC = .25, medium effect sizes

30 clusters with 5 or 30 observations per cluster (i.e., N = 150 and 900)

15% MAR missing data on all analysis variables

20 imputations with R package jomo



Random Slope Analysis Model

Two-level random slope analysis with continuous level-1 and level-2 predictors

$$y_{ij} = \gamma_0 + \gamma_1\, x_{ij} + \gamma_2 w_j + u_{0j} + u_{1j} x_{ij} + \varepsilon_{ij}$$

All variables have missing data

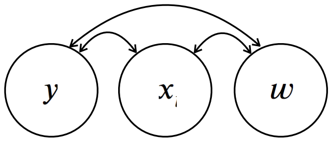
Joint Model Limitations

Within-cluster covariances must preserve level-1 relations, including the random coefficients

The classic formulation of the joint model assumes a common covariance matrix at level-1

Imputation ignores random slope variation

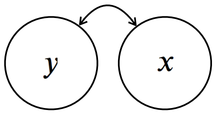
Covariance Structure Revisited



$$\boldsymbol{\Sigma}_u = \begin{pmatrix} \sigma^2_{u_{(y)}} & \sigma_{u_{(y,x)}} & \sigma_{u_{(y,w)}} \\ \sigma_{u_{(x,y)}} & \sigma^2_{u_{(x)}} & \sigma_{u_{(x,w)}} \\ \sigma_{u_{(w,y)}} & \sigma_{u_{(w,x)}} & \sigma^2_{u_{(w)}} \end{pmatrix}$$

Level-2

Level-1



$$\boldsymbol{\Sigma}_\varepsilon = \begin{pmatrix} \sigma^2_{\varepsilon_{(y)}} & \sigma_{\varepsilon_{(y,x)}} & 0 \\ \sigma_{\varepsilon_{(x,y)}} & \sigma^2_{\varepsilon_{(x)}} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Simulation Study

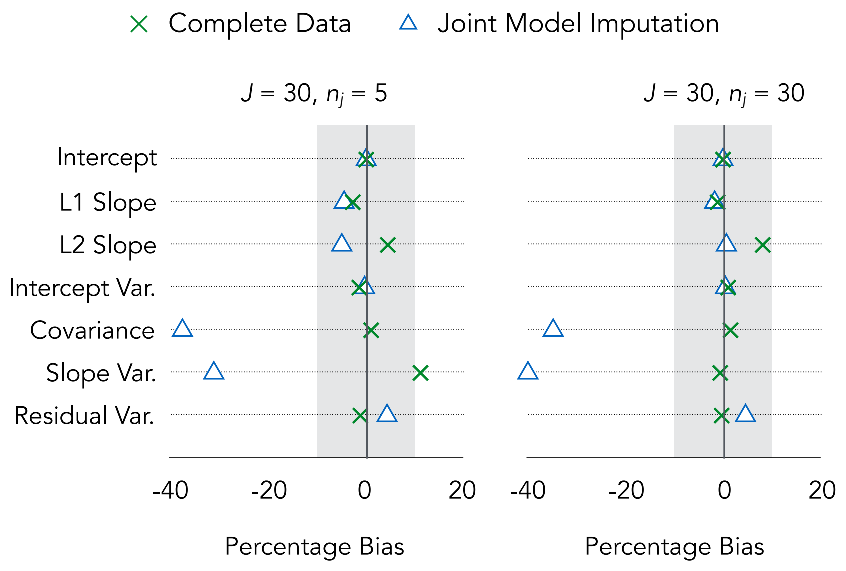
Random slope model with 1000 replications

ICC = .25, medium effect sizes

30 clusters with 5 or 30 observations per cluster (i.e., *N* = 150 and 900)

15% MAR missing data on all analysis variables

20 imputations with R package jomo

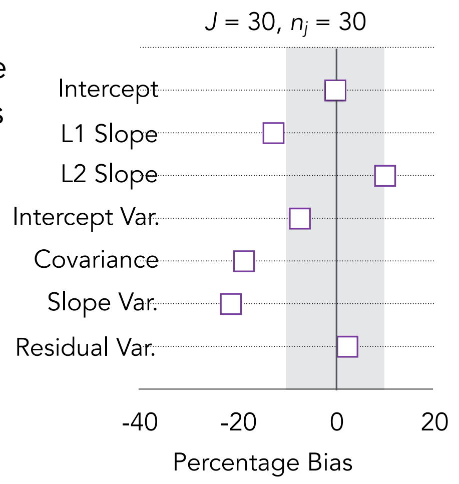


Brief Maximum Likelihood Detour

Mplus allows incomplete random slope predictors

Requires numerical integration and many latent variable products

Often yields severe bias



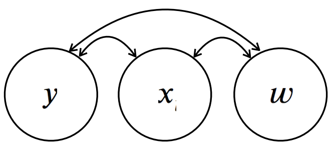
Joint Modeling With Random Level-1 Covariance Matrices

Yucel (2011) extended the joint model to incorporate random level-1 covariance matrices

Available in the R package jomo

Currently limited to 2-level models

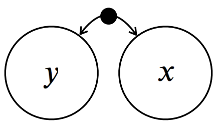
Covariance Structure



$$\boldsymbol{\Sigma}_u = \begin{pmatrix} \sigma^2_{u_{(y)}} & \sigma_{u_{(y,x)}} & \sigma_{u_{(y,w)}} \\ \sigma_{u_{(x,y)}} & \sigma^2_{u_{(x)}} & \sigma_{u_{(x,w)}} \\ \sigma_{u_{(w,y)}} & \sigma_{u_{(w,x)}} & \sigma^2_{u_{(w)}} \end{pmatrix}$$

Level-2

Level-1



$$\boldsymbol{\Sigma}_{\varepsilon_j} = \begin{pmatrix} \sigma^2_{\varepsilon_{j(y)}} & \sigma_{\varepsilon_{j(y,x)}} & 0 \\ \sigma_{\varepsilon_{j(x,y)}} & \sigma^2_{\varepsilon_{j(x)}} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Limitation Of Random Covariance Matrices

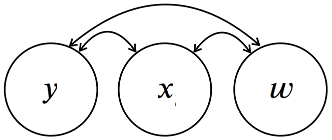
The between-cluster covariance matrix preserves random intercept variation, while the within-cluster matrices preserve random slopes

Elements of $\boldsymbol{\Sigma}_u$ in the analysis model depend on orthogonal sources of variation

Imputation assumes no correlation between the random intercepts and slopes

Covariance Structure

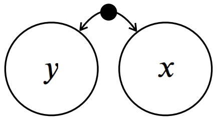
Intercept variation



$$\boldsymbol{\Sigma}_u = \begin{pmatrix} \boxed{\sigma^2_{u_{(y)}}} & \sigma_{u_{(y,x)}} & \sigma_{u_{(y,w)}} \\ \sigma_{u_{(x,y)}} & \sigma^2_{u_{(x)}} & \sigma_{u_{(x,w)}} \\ \sigma_{u_{(w,y)}} & \sigma_{u_{(w,x)}} & \sigma^2_{u_{(w)}} \end{pmatrix}$$

Level-2

Level-1



$$\boldsymbol{\Sigma}_{\varepsilon_j} = \begin{pmatrix} \sigma^2_{\varepsilon_{j(y)}} & \sigma_{\varepsilon_{j(y,x)}} & 0 \\ \boxed{\sigma_{\varepsilon_{j(x,y)}}} & \sigma^2_{\varepsilon_{j(x)}} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Slope variation

R Package jomo

```
# load packages
library (jomo)

# read raw data
dat <- read.table("~/desktop/examples/ridata.csv", sep = ",")
names(dat) = c("cluster", "av1", "av2", "y", "x","w")
dat[dat == 999] <- NA

# jomo imputation
set.seed(90291)
dat$icept <- 1
l1miss <- c("y", "x")
l2miss <- c("w")
l1complete <- c("icept")
l2complete <- c("icept")
impdata <- jomo(dat[l1miss], Y2 = dat[l2miss], X = dat[l1complete],
  X2 = dat[l2complete], clus = dat$cluster,
  nburn = 2000, nbetween = 2000, nimp = 20, meth = "random")
```

Simulation Study

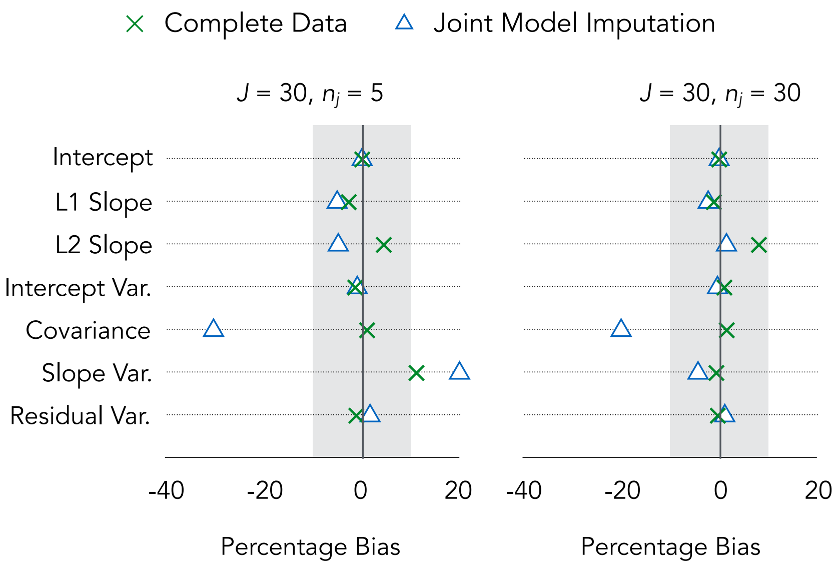
Random slope model with 1000 replications

ICC = .25, medium effect sizes

30 clusters with 5 or 30 observations per cluster (i.e., N = 150 and 900)

15% MAR missing data on all analysis variables

20 imputations with R package jomo



Multivariate Imputation With Fully Conditional Specification

Fully Conditional Specification

Variable-by-variable imputation

Uses a series of univariate regression models with an incomplete variable regressed on complete and previously imputed variables

Available in R package mice (2-level models with continuous variables) and the Blimp application for MacOS, Windows, and Linux

Random Intercept Analysis Model

Two-level random intercept analysis with continuous level-1 and level-2 predictors

$$y_{ij} = \gamma_0 + \gamma_1\, x_{ij} + \gamma_2 w_j + u_{0j} + \varepsilon_{ij}$$

All variables have missing data

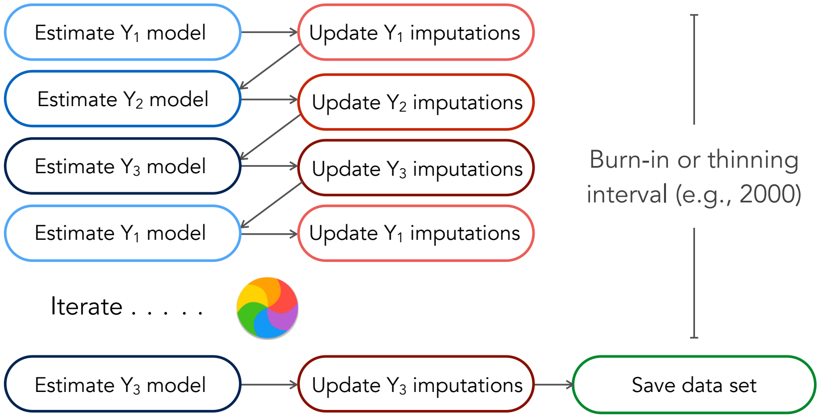
Overview Of Algorithmic Steps

Each incomplete variable has an imputation models tailored to match features of the analysis

A single iteration consists of estimation and imputation sequences for each missing variable

The imputed variable from one sequence serves as a predictor variable in all other sequences

Algorithmic Steps



Estimate Y₁ model

Update Y₁ imputations

Estimate Y₂ model

Update Y₂ imputations

Estimate Y₃ model

Update Y₃ imputations

Estimate Y₁ model

Update Y₁ imputations

Iterate . . . . .

Estimate Y₃ model

Update Y₃ imputations

Save data set

Burn-in or thinning interval (e.g., 2000)

Estimation And Imputation For *y*

Imputation model:

$$y_{ij} = \gamma_{0(y)} + \gamma_{1(y)}\, x_{ij} + \gamma_{2(y)}\, w_j + u_{0j(y)} + \varepsilon_{ij(y)}$$

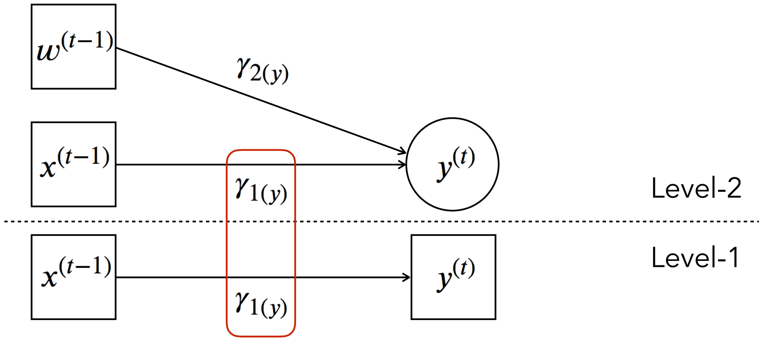
Bayesian estimation and imputation sequence:

$$\dot{\theta}^{(t)}_{(y)} \sim P\left(\theta_{(y)}\,|\dot{y}^{(t-1)}, \dot{x}^{(t-1)}, \dot{w}^{(t-1)}\right)$$

$$\dot{y}^{(t)} \sim P\left(y\,|\dot{x}^{(t-1)}, \dot{w}^{(t-1)}, \dot{\theta}^{(t)}_{(y)}\right)$$

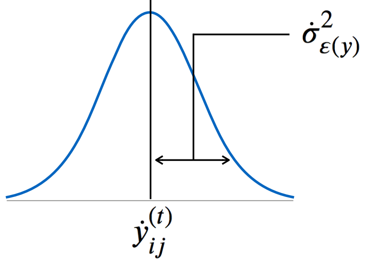
Imputation Model For *y*

$$y_{ij} = \gamma_{0(y)} + \gamma_{1(y)}\, x_{ij} + \gamma_{2(y)}\, w_j + u_{0j(y)} + \varepsilon_{ij(y)}$$



Imputation Step For y

$$\hat{y}_{ij}^{(t)} = \dot{\gamma}_{0(y)} + \dot{\gamma}_{1(y)}\, \dot{x}_{ij}^{(t-1)} + \dot{\gamma}_{2(y)}\, \dot{w}_{j}^{(t-1)} + \dot{u}_{0j(y)}$$



$$\dot{y}_{ij}^{(t)} \sim \mathrm{N}\left(\hat{y}_{ij}^{(t)}, \dot{\sigma}^2_{\varepsilon(y)}\right)$$

Estimation And Imputation For x

Imputation model:

$$x_{ij} = \gamma_{0(x)} + \gamma_{1(x)}\, y_{ij} + \gamma_{2(x)}\, w_j + u_{0j(x)} + \varepsilon_{ij(x)}$$

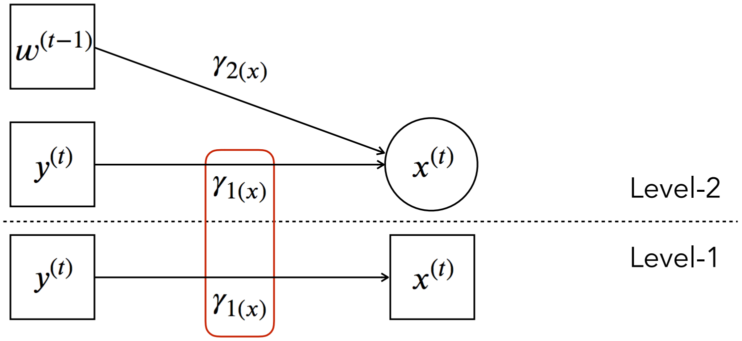
Bayesian estimation and imputation sequence:

$$\dot{\theta}_{(x)}^{(t)} \sim P\left(\theta_{(x)} \mid \dot{y}^{(t)}, \dot{x}^{(t-1)}, \dot{w}^{(t-1)}\right)$$

$$\dot{x}^{(t)} \sim P\left(x \mid \dot{y}^{(t)}, \dot{w}^{(t-1)}, \dot{\theta}_{(x)}^{(t)}\right)$$

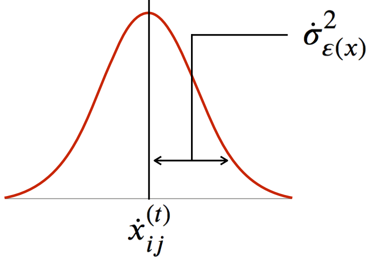
Imputation Model For x

$$x_{ij} = \gamma_{0(x)} + \gamma_{1(x)}\, y_{ij} + \gamma_{2(x)}\, w_j + u_{0j(x)} + \varepsilon_{ij(x)}$$



Imputation Step For x

$$\hat{x}_{ij}^{(t)} = \dot{\gamma}_{0(x)} + \dot{\gamma}_{1(x)}\, \dot{y}_{ij}^{(t)} + \dot{\gamma}_{2(x)}\, \dot{w}_{j}^{(t-1)} + \dot{u}_{0j(x)}$$



$$\dot{x}_{ij}^{(t)} \sim \mathrm{N}\left(\hat{x}_{ij}^{(t)}, \dot{\sigma}^2_{\varepsilon(x)}\right)$$

Estimation And Imputation For *w*

Imputation model:

$$w_j = \gamma_{0(w)} + \gamma_{1(w)}\,\bar{y}_j + \gamma_{2(w)}\,\bar{x}_j + u_{0j(w)}$$

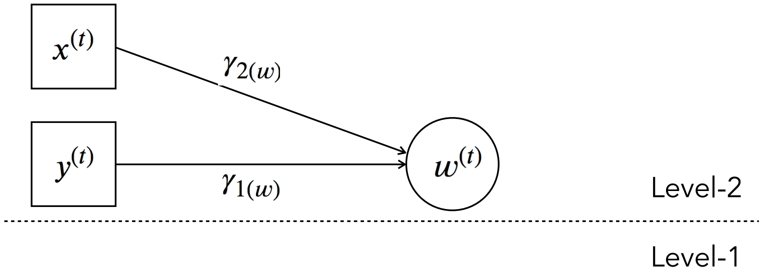
Bayesian estimation and imputation sequence:

$$\dot{\theta}^{(t)}_{(w)} \sim P\left(\theta_{(w)} \,|\, \dot{y}^{(t)}, \dot{x}^{(t)}, \dot{w}^{(t-1)}\right)$$

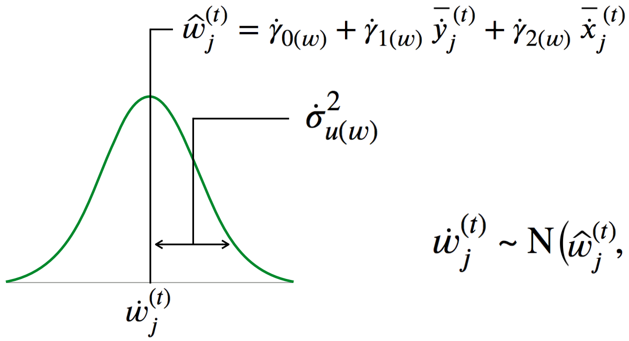
$$\dot{w}^{(t)} \sim P\left(w \,|\, \dot{y}^{(t)}, \dot{x}^{(t)}, \dot{\theta}^{(t)}_{(w)}\right)$$

Imputation Model For *w*

$$w_j = \gamma_{0(w)} + \gamma_{1(w)}\,\bar{y}_j + \gamma_{2(w)}\,\bar{x}_j + u_{0j(w)}$$



Imputation Step For *w*


$\hat{w}^{(t)}_j = \dot{\gamma}_{0(w)} + \dot{\gamma}_{1(w)}\,\bar{\dot{y}}^{(t)}_j + \dot{\gamma}_{2(w)}\,\bar{\dot{x}}^{(t)}_j$
$\dot{\sigma}^2_{u(w)}$
$\dot{w}^{(t)}_j$

$$\dot{w}^{(t)}_j \sim \mathrm{N}\left(\hat{w}^{(t)}_j, \dot{\sigma}^2_{u(w)}\right)$$

Blimp Syntax

```
DATA: ~/desktop/examples/ridata.csv;
VARIABLES: cluster av1 av2 y x w;
MISSING: 999;
MODEL: cluster ~ y x w;
NIMPS: 20;
THIN: 2000;
BURN: 2000;
SEED: 90291;
OUTFILE: ~/desktop/examples/imps.csv;
OPTIONS: stacked noclmeans prior1;
```

Simulation Study

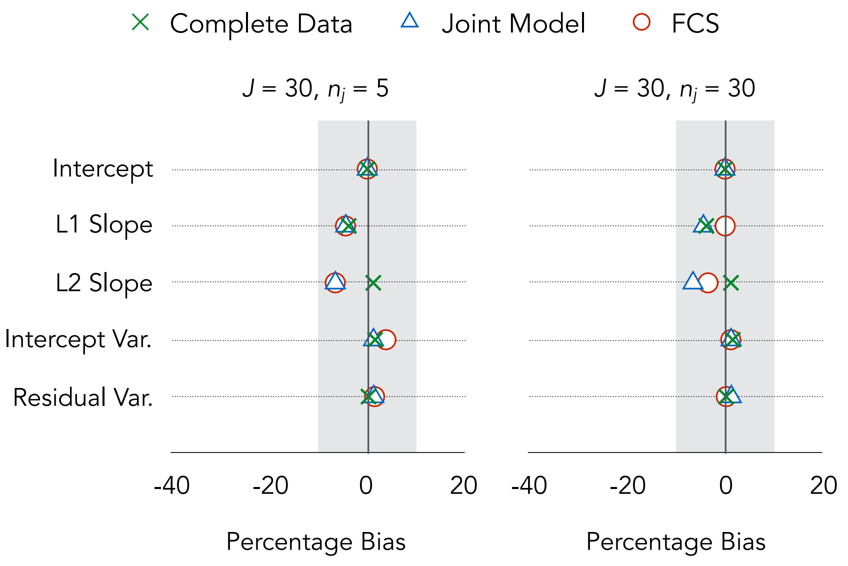
Random intercept model with 1000 replications

ICC = .25, medium effect sizes

30 clusters with 5 or 30 observations per cluster (i.e., N = 150 and 900)

15% MAR missing data on all analysis variables

20 imputations with the Blimp application



Limitations

The classic formulation of fully conditional specification assumes equal within- and between-cluster regression slopes

i.e., Equality constraints on the level-1 and level-2 model-implied covariance matrices

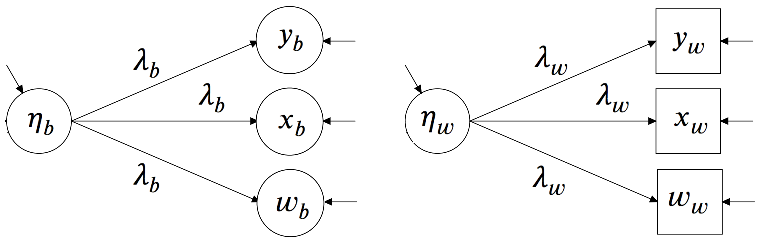
Not ideal for models that partition relations

Revisiting Models That Partition Variability

Contextual effects analyses

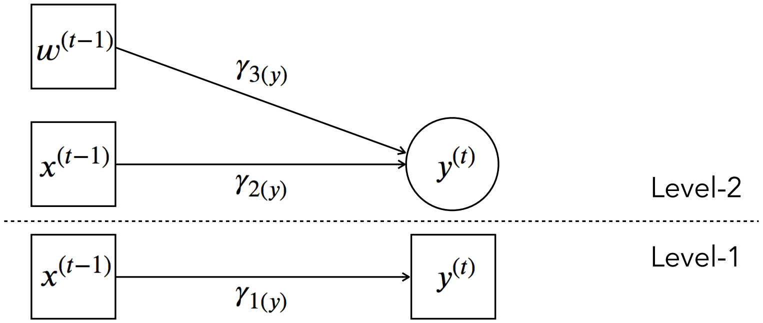
$$y_{ij} = \gamma_0 + \gamma_1 x_{ij} + \gamma_2 \bar{x}_j + \gamma_3 w_j + u_{0j} + \varepsilon_{ij}$$

Multilevel SEM



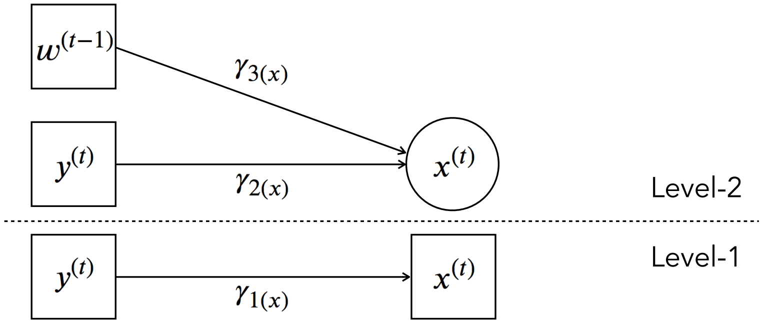
Partitioned Imputation Model For y

$$y_{ij} = \gamma_{0(y)} + \gamma_{1(y)}\, x_{ij} + \gamma_{2(y)}\, \bar{x}_j + \gamma_{3(y)}\, w_j + u_{0j(y)} + \varepsilon_{ij(y)}$$



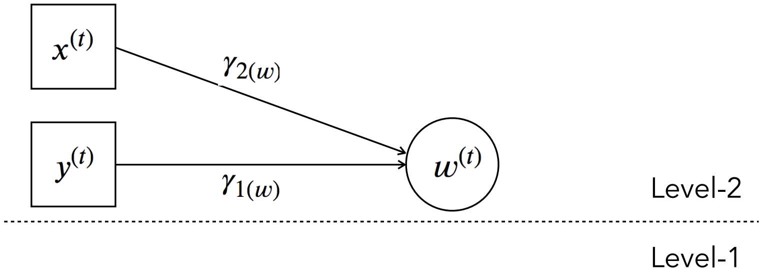
Partitioned Imputation Model For x

$$x_{ij} = \gamma_{0(x)} + \gamma_{1(x)}\, y_{ij} + \gamma_{2(x)}\, \bar{y}_j + \gamma_{3(x)}\, w_j + u_{0j(x)} + \varepsilon_{ij(x)}$$



Imputation Model For w

$$w_j = \gamma_{0(w)} + \gamma_{1(w)}\, \bar{y}_j + \gamma_{2(w)}\, \bar{x}_j + u_{0j(w)}$$



Blimp Syntax

```
DATA: ~/desktop/examples/ridata.csv;
VARIABLES: cluster av1 av2 y x w;
MISSING: 999;
MODEL: cluster ~ y x w;
NIMPS: 20;
THIN: 2000;
BURN: 2000;
SEED: 90291;
OUTFILE: ~/desktop/example/imps.csv;
OPTIONS: stacked clmeans prior1;
```

Random Slope Analysis Model

Two-level random slope analysis with continuous level-1 and level-2 predictors

$$y_{ij} = \gamma_0 + \gamma_1\, x_{ij} + \gamma_2 w_j + u_{0j} + u_{1j} x_{ij} + \varepsilon_{ij}$$

All variables have missing data

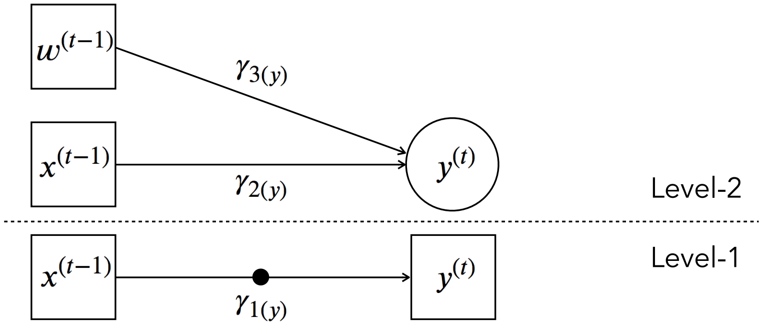
Reversed Random Coefficients

Fully conditional specification uses "reversed random coefficients" to preserve random slope variation

Imputation treats x as a random predictor of y, and y as a random predictor of x

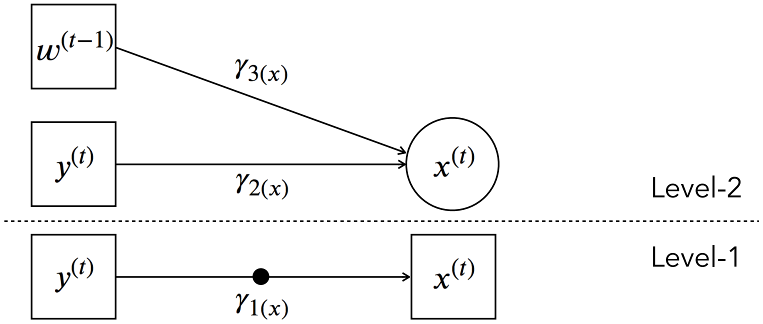
Reversed Coefficient Model For y

$$y_{ij} = \gamma_{0(y)} + \gamma_{1(y)}\, x_{ij} + \gamma_{2(y)}\, \bar{x}_j + \gamma_{3(y)}\, w_j + u_{0j(y)} + u_{1j(y)} x_{ij} + \varepsilon_{ij(y)}$$



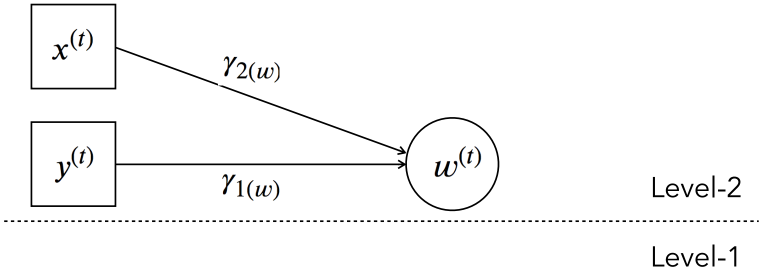
Reversed Coefficient Model For x

$$x_{ij} = \gamma_{0(x)} + \gamma_{1(x)}\, y_{ij} + \gamma_{2(x)}\, \bar{y}_j + \gamma_{3(x)}\, w_j + u_{0j(x)} + u_{1j(x)} y_{ij} + \varepsilon_{ij(x)}$$



Imputation Model For w

$$w_j = \gamma_{0(w)} + \gamma_{1(w)}\,\bar{y}_j + \gamma_{2(w)}\,\bar{x}_j + u_{0j(w)}$$



Blimp Syntax

```
DATA: ~/desktop/examples/rsdata.csv;
VARIABLES: cluster av1 av2 y x w;
MISSING: 999;
MODEL: cluster ~ y:x w;
NIMPS: 20;
THIN: 2000;
BURN: 2000;
SEED: 90291;
OUTFILE: ~/desktop/examples/imps.csv;
OPTIONS: stacked clmeans prior1;
```

Simulation Study

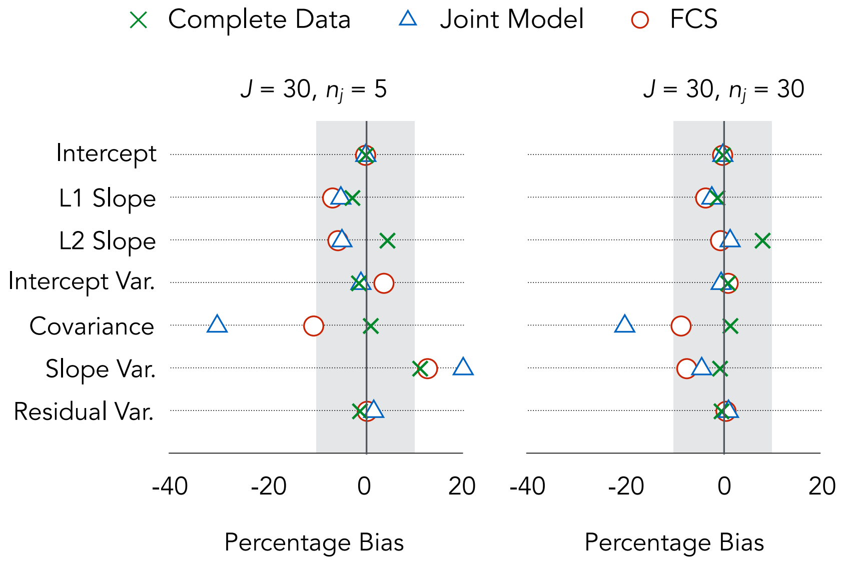
Random slope model with 1000 replications

ICC = .25, medium effect sizes

30 clusters with 5 or 30 observations per cluster (i.e., N = 150 and 900)

15% MAR missing data on all analysis variables

20 imputations with the Blimp application



Incomplete Categorical Variables

Complete Categorical Variables

Complete categorical variables function as predictors in fully conditional specification

Convert nominal (and maybe ordinal) variables to dummy or effect codes, à la regression

Blimp's NOMINAL command automatically creates the necessary code variables

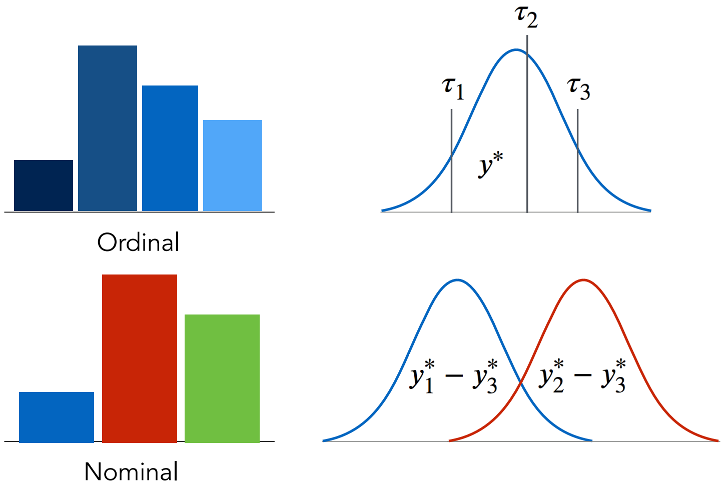
Latent Variable Imputation Framework

Blimp uses a latent variable (i.e., probit regression) formulation to impute categorical variables

Discrete responses arise from one or more underlying normal latent variables, denoted y^*

Cumulative and multinomial probit models impute ordinal and nominal variables, respectively

Latent Variable Transformations

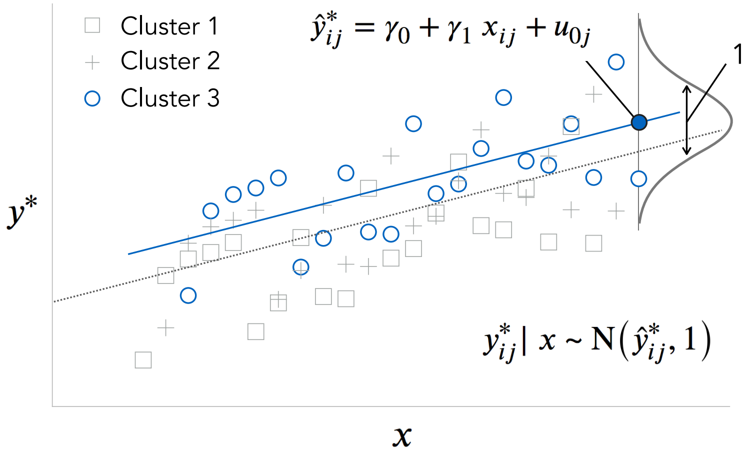


Latent Variable Scaling

Latent variable distributions are centered at a predicted value and have residual variance fixed at one for identification

$$y^*_{ij} \sim \mathrm{N}(\gamma_0 + \gamma_1\, x_{ij} + u_{0j}, 1)$$

Random Intercept Model



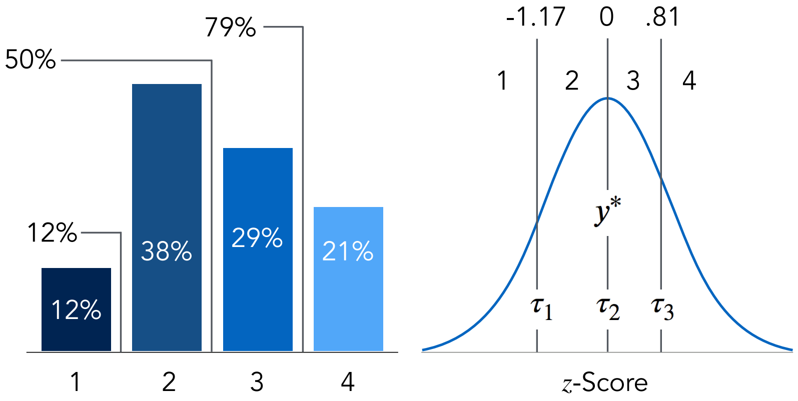
Threshold Parameters

Ordinal (or binary) variables with *K* response options require *K* - 1 threshold parameters

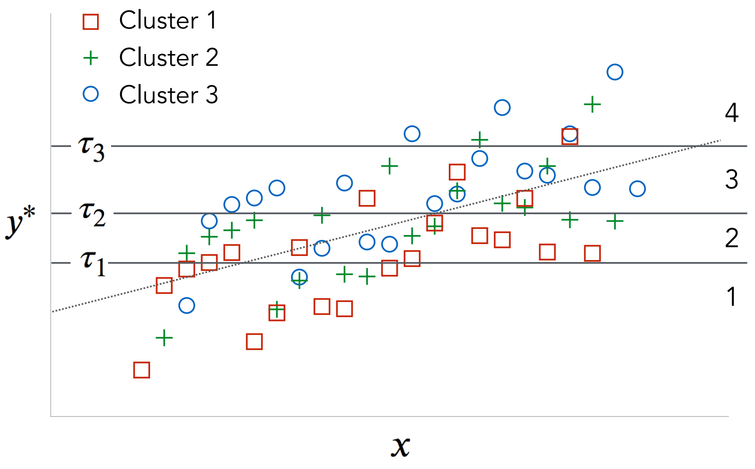
Thresholds are *z*-scores corresponding to the cumulative percentage of each response

Thresholds slice the continuous latent distribution into discrete response segments

Marginal Distribution Example



Multilevel Model Example



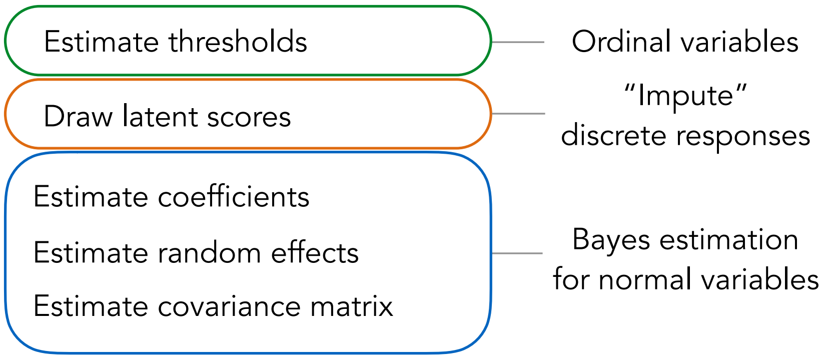
Complete-Data Bayesian Estimation

The Gibbs sampler first replaces discrete responses with latent variable scores

Threshold parameters (ordinal variables) are sampled using a Metropolis step

Bayesian estimation steps for normal data update parameters and level-2 residual terms for the underlying latent variable model

Gibbs Sampler Steps



Estimate thresholds — Ordinal variables

Draw latent scores — "Impute" discrete responses

Estimate coefficients

Estimate random effects

Estimate covariance matrix

— Bayes estimation for normal variables

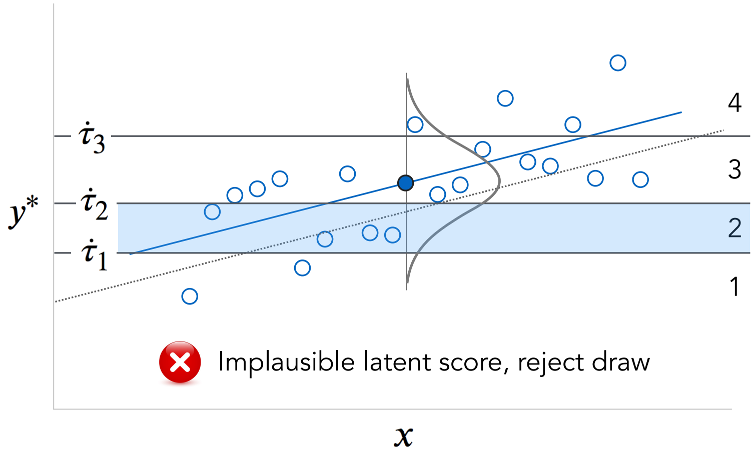
Latent Scores For Ordinal Variables

A discrete response restricts the plausible range of the latent scores

e.g., a score of y = 2 must have a latent score located between the appropriate thresholds

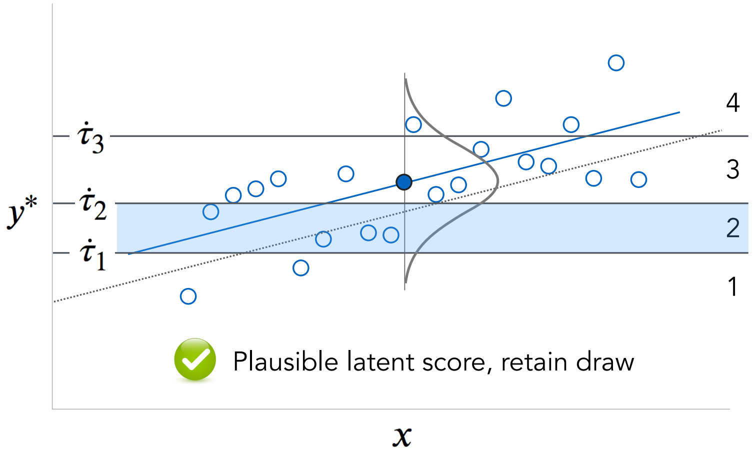
The latent variable scores are drawn from a normal distribution truncated at the thresholds

Truncated Normal Draw | $y = 2$



✗ Implausible latent score, reject draw

Truncated Normal Draw | $y = 2$



✓ Plausible latent score, retain draw

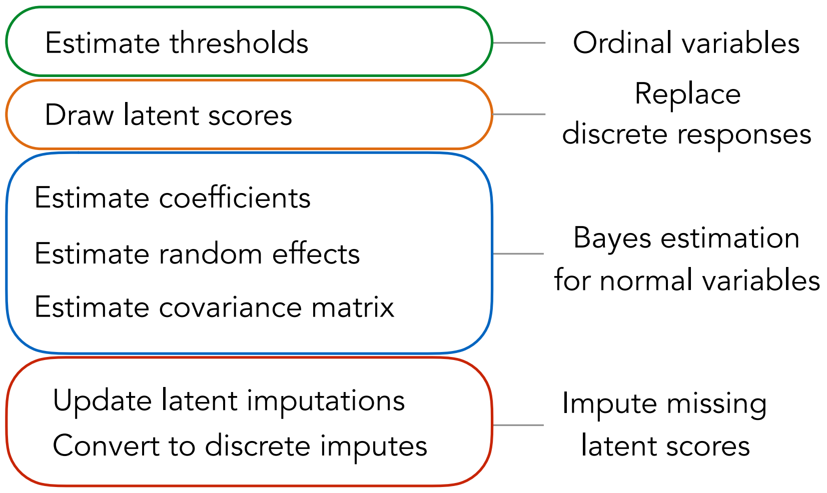
Incomplete Ordinal Variables

Identical procedure as complete data, with imputations generated at the end of each Bayesian estimation sequence

Latent scores for missing cases are unbounded because the truncation points are unknown

Latent imputes are subsequently discretized using threshold parameters

Gibbs Sampler Steps



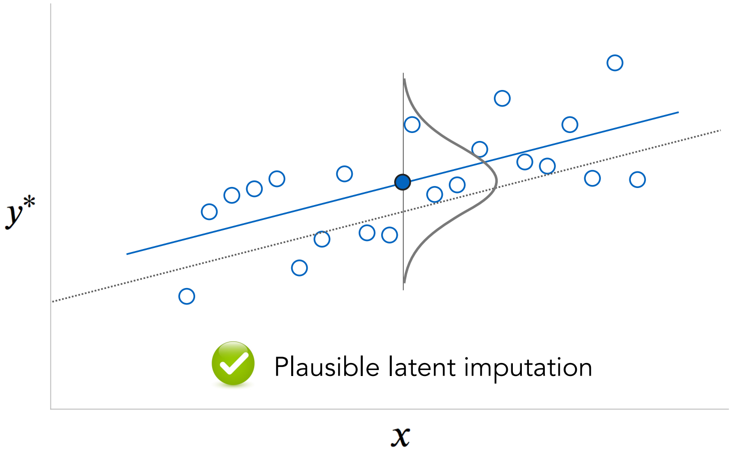
Estimate thresholds — Ordinal variables

Draw latent scores — Replace discrete responses

Estimate coefficients
Estimate random effects
Estimate covariance matrix — Bayes estimation for normal variables

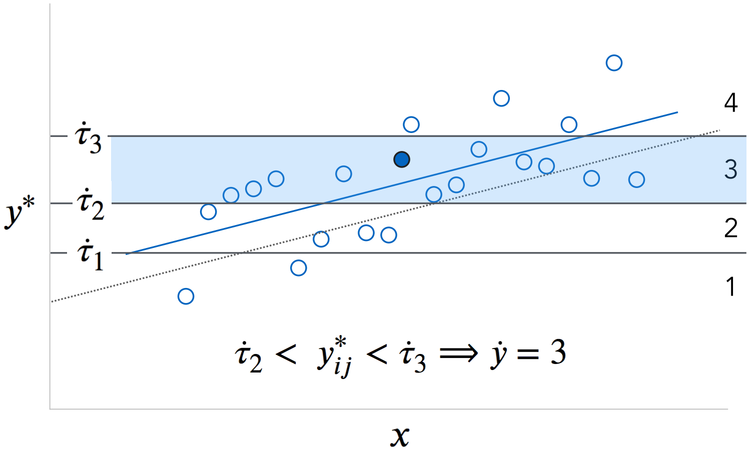
Update latent imputations
Convert to discrete imputes — Impute missing latent scores

Truncated Normal Draw | y = ?



✓ Plausible latent imputation

Generating Discrete Imputes



$$\dot\tau_2 < y^*_{ij} < \dot\tau_3 \Longrightarrow \dot y = 3$$

Multinomial Probit Model

The multinomial model defines K latent variables representing the response strength of each category

$$d_1^* = y_1^* - y_K^*$$

$$d_2^* = y_2^* - y_K^*$$

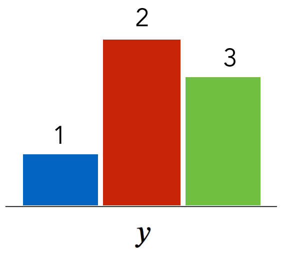
$$\vdots$$

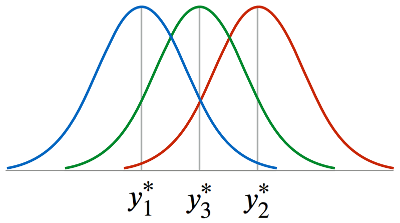
$$d_{K-1}^* = y_{K-1}^* - y_K^*$$

K categories require K-1 latent variable difference scores

Category K is the reference

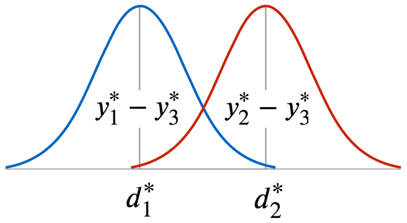
Example: 3-Category Nominal Variable

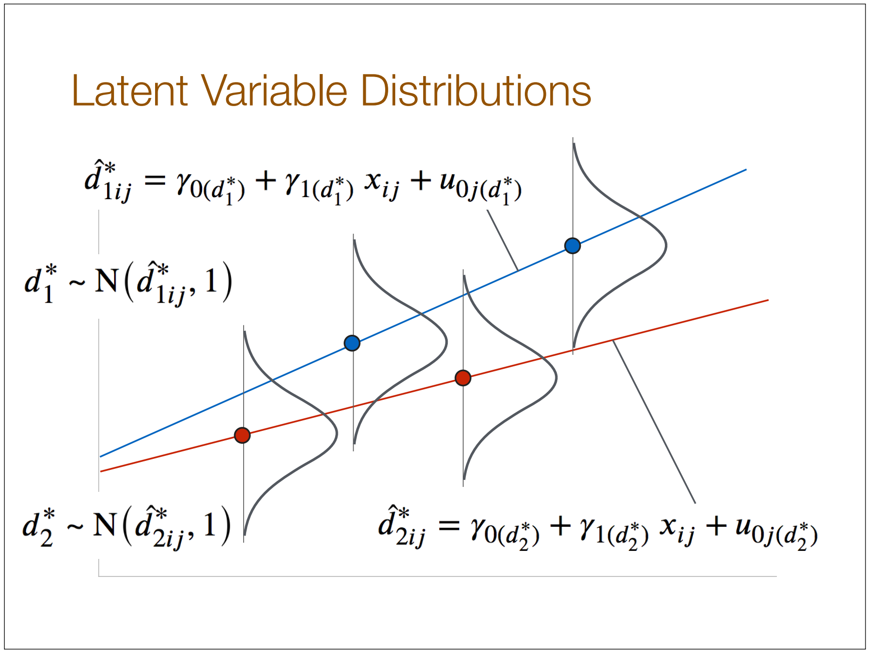




$$d_1^* = y_1^* - y_3^*$$

$$d_2^* = y_2^* - y_3^*$$





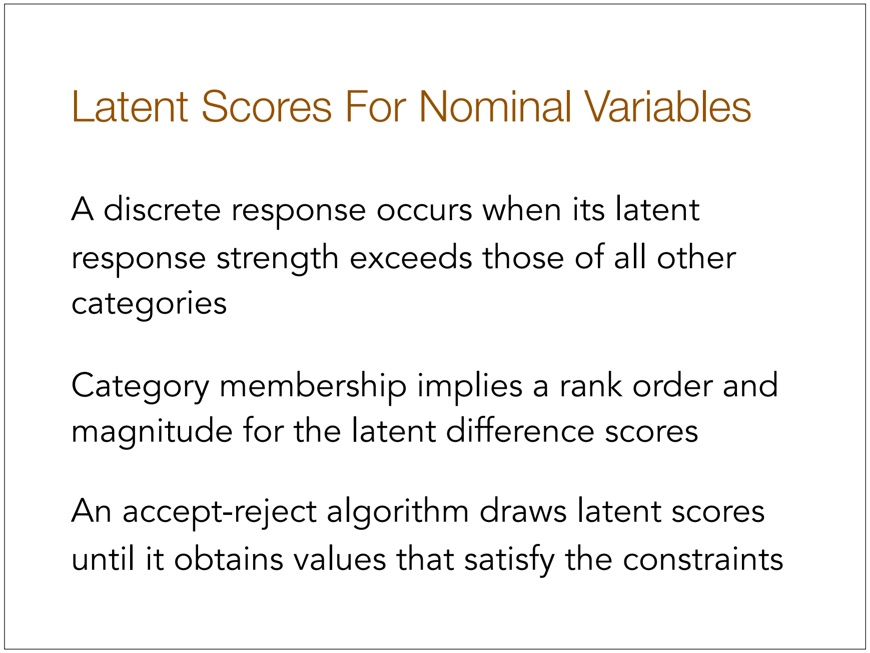
Latent Variable Distributions

$\hat{d}^*_{1ij} = \gamma_{0(d^*_1)} + \gamma_{1(d^*_1)}\, x_{ij} + u_{0j(d^*_1)}$

$d^*_1 \sim \mathrm{N}\left(\hat{d}^*_{1ij}, 1\right)$

$d^*_2 \sim \mathrm{N}\left(\hat{d}^*_{2ij}, 1\right)$

$\hat{d}^*_{2ij} = \gamma_{0(d^*_2)} + \gamma_{1(d^*_2)}\, x_{ij} + u_{0j(d^*_2)}$

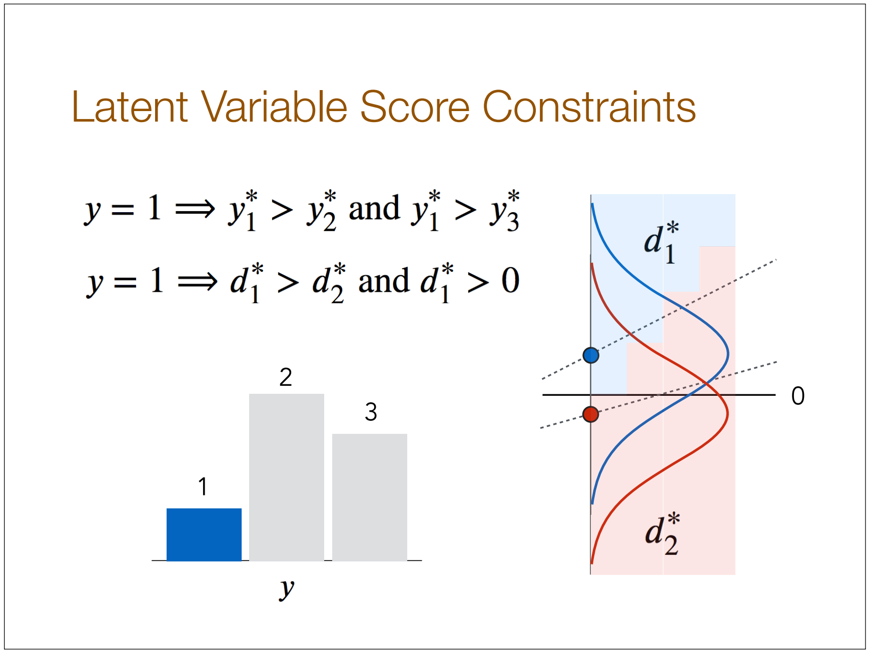


Latent Scores For Nominal Variables

A discrete response occurs when its latent response strength exceeds those of all other categories

Category membership implies a rank order and magnitude for the latent difference scores

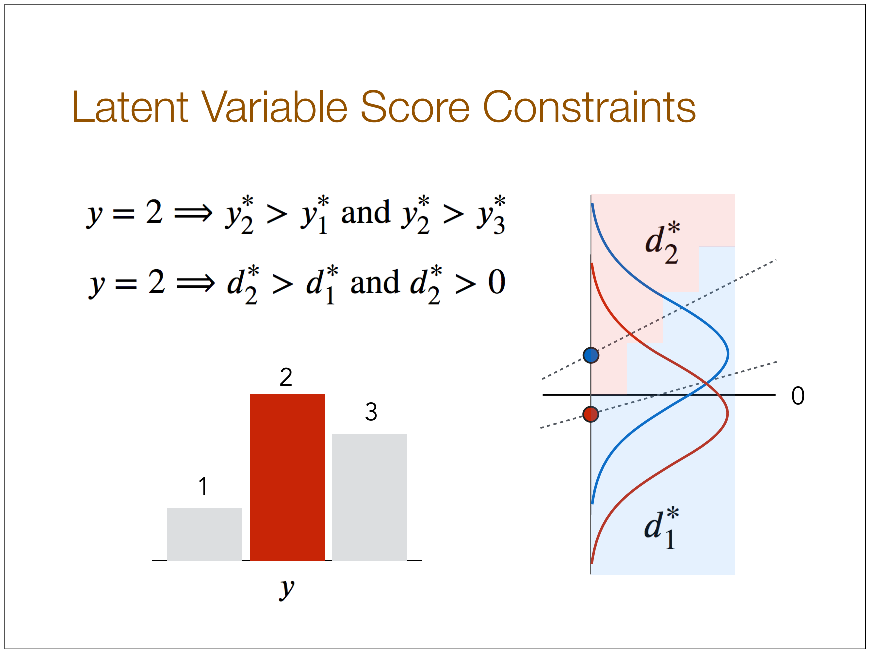
An accept-reject algorithm draws latent scores until it obtains values that satisfy the constraints



Latent Variable Score Constraints

$y = 1 \Longrightarrow y^*_1 > y^*_2$ and $y^*_1 > y^*_3$

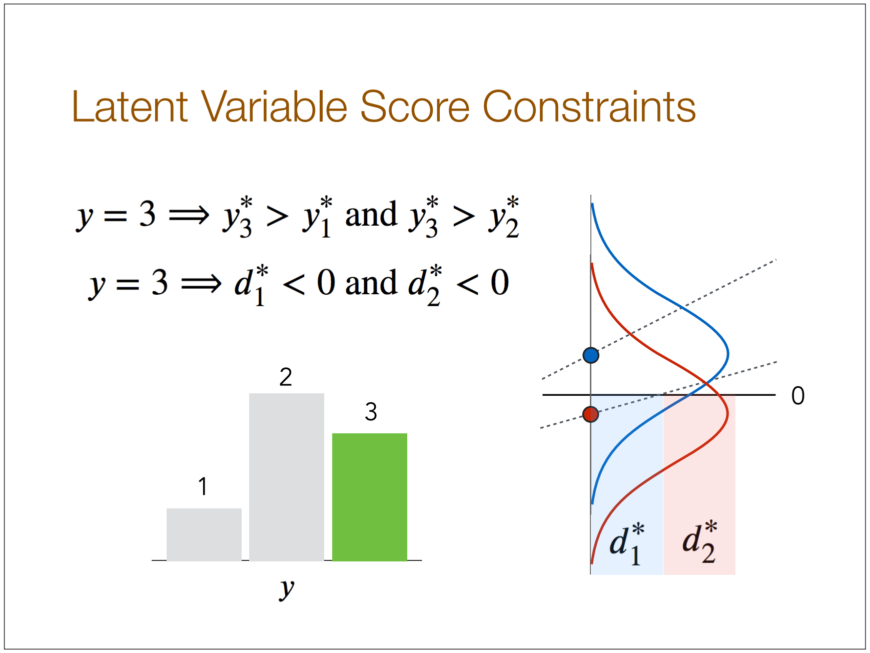
$y = 1 \Longrightarrow d^*_1 > d^*_2$ and $d^*_1 > 0$



Latent Variable Score Constraints

$y = 2 \Longrightarrow y^*_2 > y^*_1$ and $y^*_2 > y^*_3$

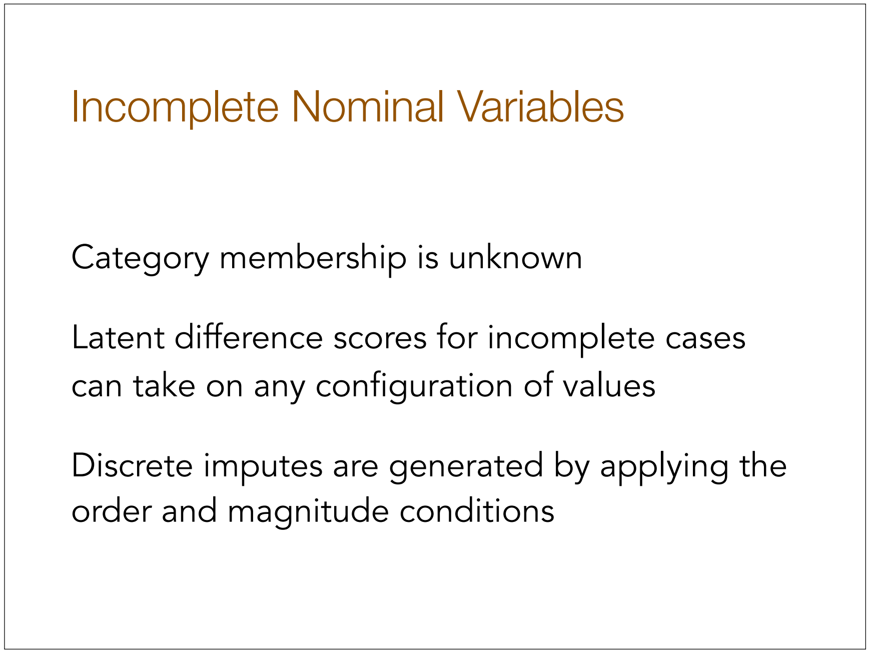
$y = 2 \Longrightarrow d^*_2 > d^*_1$ and $d^*_2 > 0$



Latent Variable Score Constraints

$$y = 3 \Longrightarrow y_3^* > y_1^* \text{ and } y_3^* > y_2^*$$

$$y = 3 \Longrightarrow d_1^* < 0 \text{ and } d_2^* < 0$$

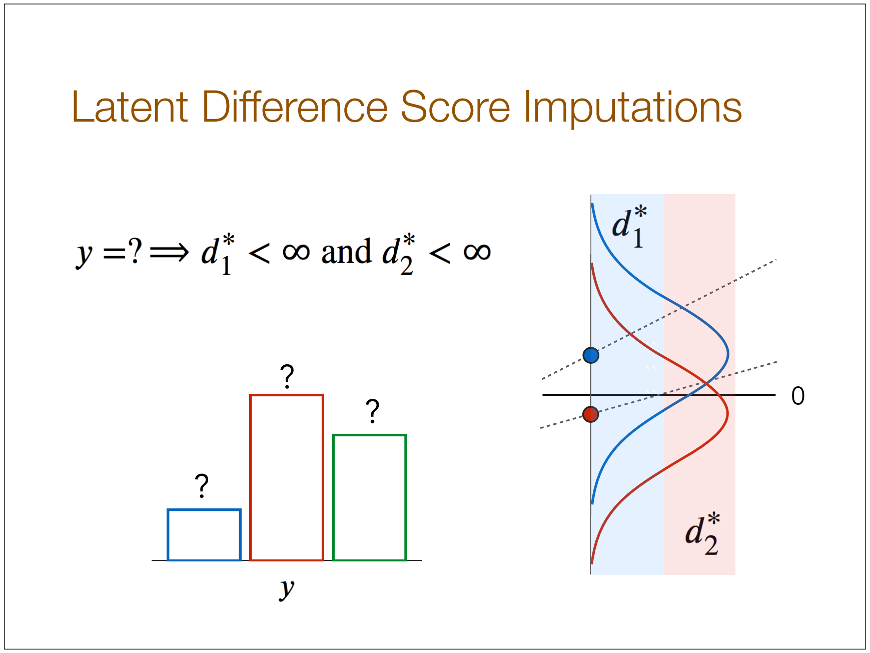


Incomplete Nominal Variables

Category membership is unknown

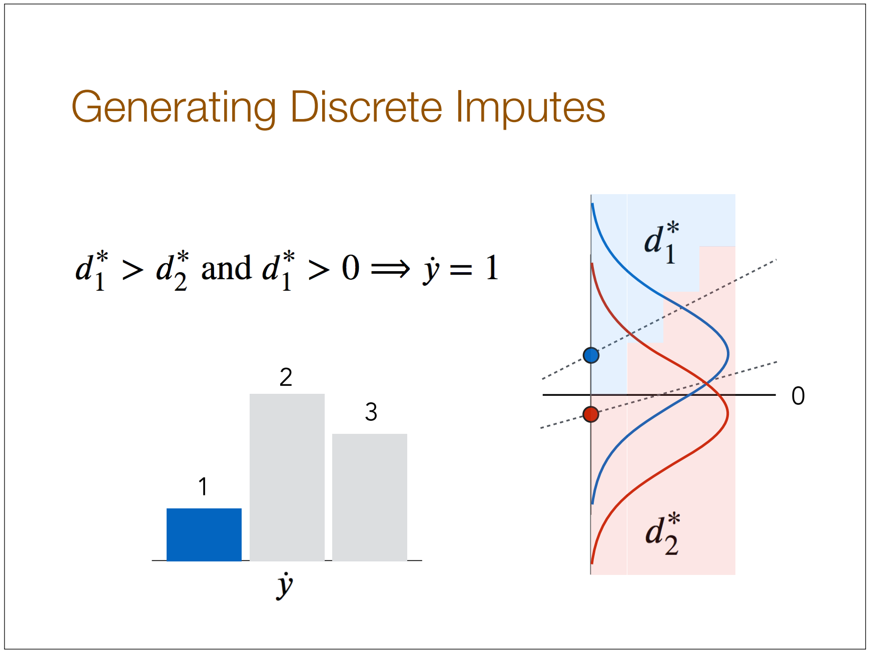
Latent difference scores for incomplete cases can take on any configuration of values

Discrete imputes are generated by applying the order and magnitude conditions



Latent Difference Score Imputations

$$y = ? \Longrightarrow d_1^* < \infty \text{ and } d_2^* < \infty$$

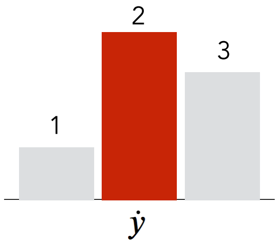


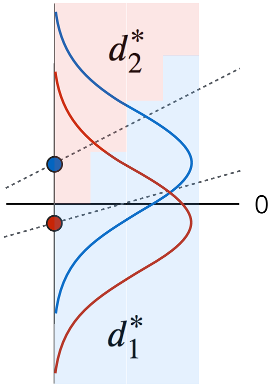
Generating Discrete Imputes

$$d_1^* > d_2^* \text{ and } d_1^* > 0 \Longrightarrow \dot{y} = 1$$

Generating Discrete Imputes

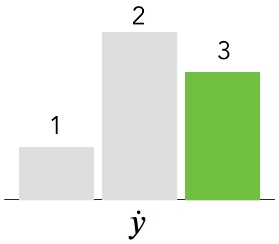
$d_2^* > d_1^*$ and $d_2^* > 0 \Longrightarrow \dot{y} = 2$

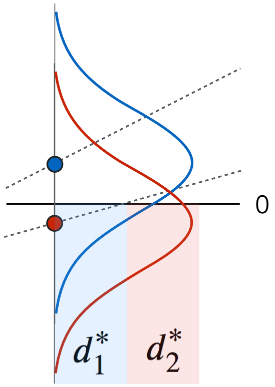




Generating Discrete Imputes

$d_1^* < 0$ and $d_2^* < 0 \Longrightarrow \dot{y} = 3$





Blimp Syntax

```
DATA: ~/desktop/examples/rsdata.csv;
VARIABLES: cluster av1 av2 y x w;
MISSING: 999;
MODEL: cluster ~ y x w;
ORDINAL: y;
NOMINAL: x w;
NIMPS: 20;
THIN: 2000;
BURN: 2000;
SEED: 90291;
OUTFILE: ~/desktop/examples/imps.csv;
OPTIONS: stacked clmeans prior1;
```

Two-Level Analysis Example

Download Information

The Blimp application for MacOS and Windows is freely available online (Linux by request)

www.appliedmissingdata.com/multilevel-imputation.html

The data and analysis scripts are also available

Motivating Example

Data from a cluster-randomized study investigating a novel math problem-solving curriculum

29 schools (level-2 units) were randomly assigned to an intervention or control condition

The average number of students (level-1 units) per school was 33.86, with a range of 13 to 61

Input Data

	Variable	Description	Missing	Metric
Level-2	school	School identifier variable		
	condition	Treatment code (0 = control, 1 = intervention)		Nominal
	esolpercent	Percentage of English as second language	*	Numeric
Level-1	student	Student identifier		
	abilitylev	Ability grouping (3-group classification)	*	Nominal
	female	Female dummy code		Nominal
	stanmath	Standardized math test scores	*	Numeric
	frlunch	Lunch assistance dummy code	*	Nominal
	efficacy	Math self-efficacy rating scale	*	Ordinal
	probsolve1	Math problem-solving score at baseline	*	Numeric
	probsolve7	Math problem-solving score at final wave	*	Ordinal

Analysis Model

The substantive analysis model predicts end-of-year problem-solving scores from intervention condition and pretest covariates

$$probsolve7_{ij} = \gamma_0 + \gamma_1\left(probsolve1_{ij}\right) + \gamma_2\left(efficacy_{ij}\right) +$$

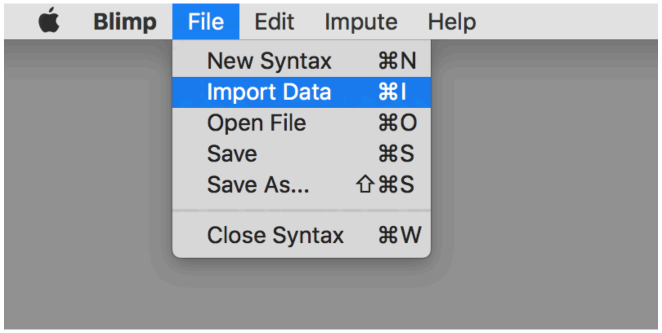
$$\gamma_3\left(abilitylev2_{ij}\right) + \gamma_4\left(abilitylev3_{ij}\right) + \gamma_5\left(female_{ij}\right) +$$

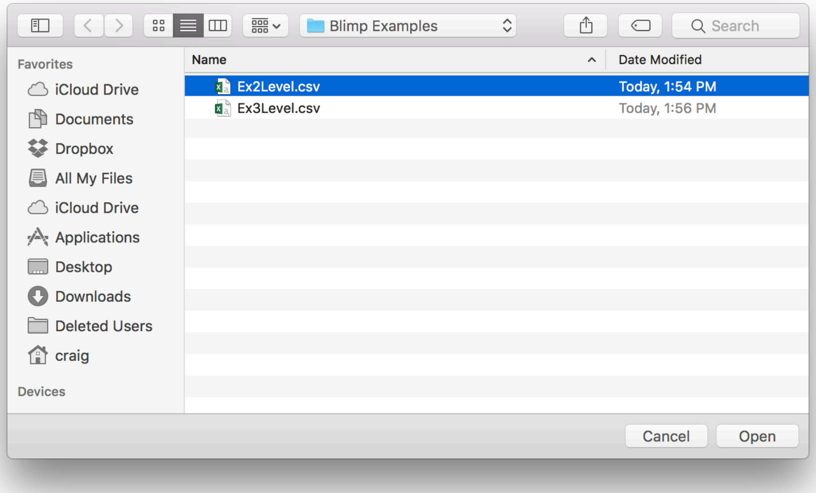
$$\gamma_6\left(esolpercent_j\right) + \gamma_7\left(condition_j\right) + u_{0j} + \varepsilon_{ij}$$

Blimp Syntax

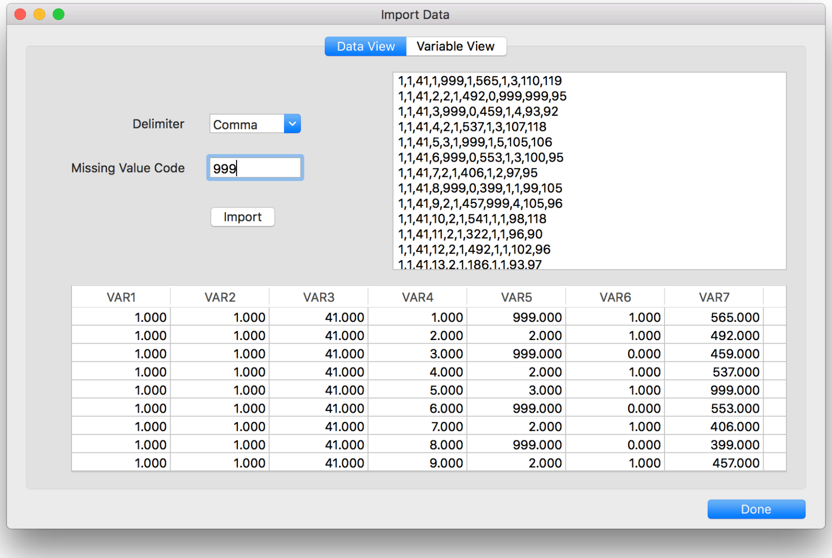
```
DATA: ~/Desktop/Blimp Examples/Ex2Level.csv;
VARIABLES: school condition esolpercent student
abilitylev
  female stanmath frlunch efficacy probsolve1 probsolve7;
ORDINAL: efficacy;
NOMINAL: condition abilitylev female frlunch;
MISSING: 999;
MODEL: school ~ condition esolpercent abilitylev female
  stanmath frlunch efficacy probsolve1 probsolve7;
NIMPS: 20;
THIN: 2000;
BURN: 2000;
SEED: 90291;
OUTFILE: ~/Desktop/Blimp Examples/Imps2Level.csv;
OPTIONS: stacked nopsr csv clmean prior1 hov;
```

Import Data





Name	Date Modified
Ex2Level.csv	Today, 1:54 PM
Ex3Level.csv	Today, 1:56 PM

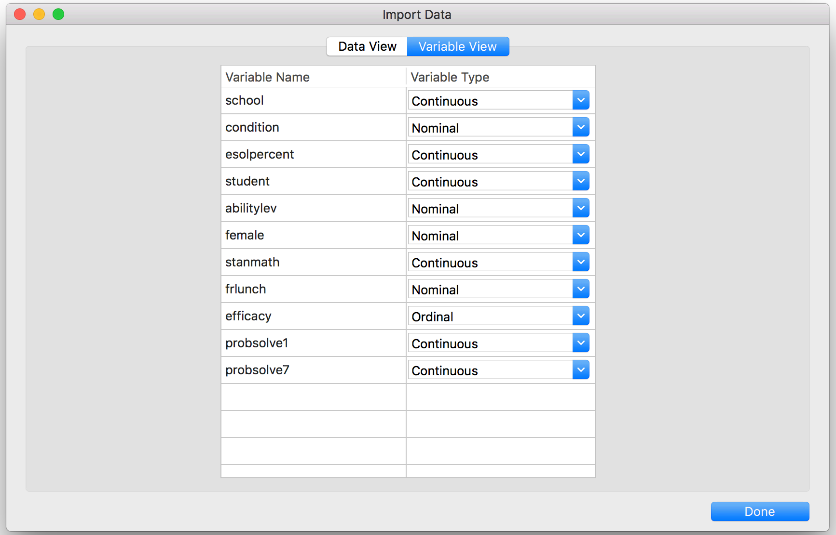


Delimiter: Comma

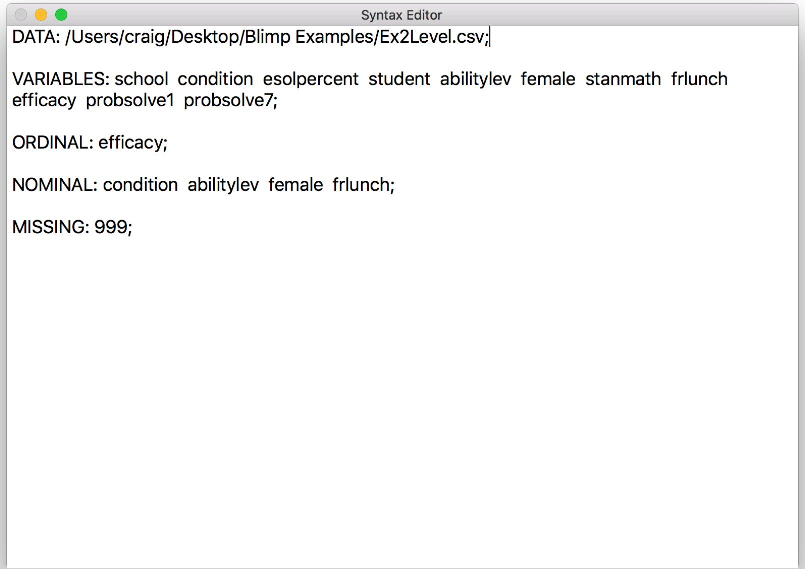
Missing Value Code: 999

```
1,1,41,1,999,1,565,1,3,110,119
1,1,41,2,2,1,492,0,999,999,95
1,1,41,3,999,0,459,1,4,93,92
1,1,41,4,2,1,537,1,3,107,118
1,1,41,5,3,1,999,1,5,105,106
1,1,41,6,999,0,553,1,3,100,95
1,1,41,7,2,1,406,1,2,97,95
1,1,41,8,999,0,399,1,1,99,105
1,1,41,9,2,1,457,999,4,105,96
1,1,41,10,2,1,541,1,1,98,118
1,1,41,11,2,1,322,1,1,96,90
1,1,41,12,2,1,492,1,1,102,96
1,1,41,13,2,1,186,1,1,93,97
```

VAR1	VAR2	VAR3	VAR4	VAR5	VAR6	VAR7
1.000	1.000	41.000	1.000	999.000	1.000	565.000
1.000	1.000	41.000	2.000	2.000	1.000	492.000
1.000	1.000	41.000	3.000	999.000	0.000	459.000
1.000	1.000	41.000	4.000	2.000	1.000	537.000
1.000	1.000	41.000	5.000	3.000	1.000	999.000
1.000	1.000	41.000	6.000	999.000	0.000	553.000
1.000	1.000	41.000	7.000	2.000	1.000	406.000
1.000	1.000	41.000	8.000	999.000	0.000	399.000
1.000	1.000	41.000	9.000	2.000	1.000	457.000

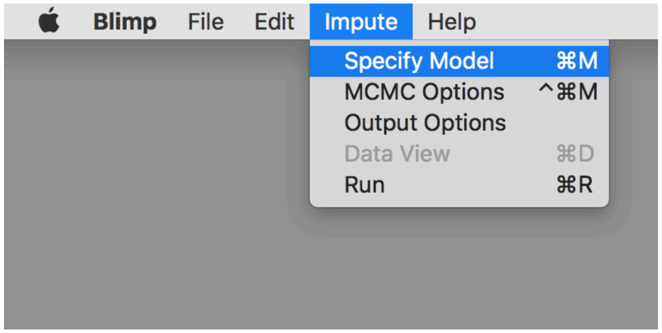


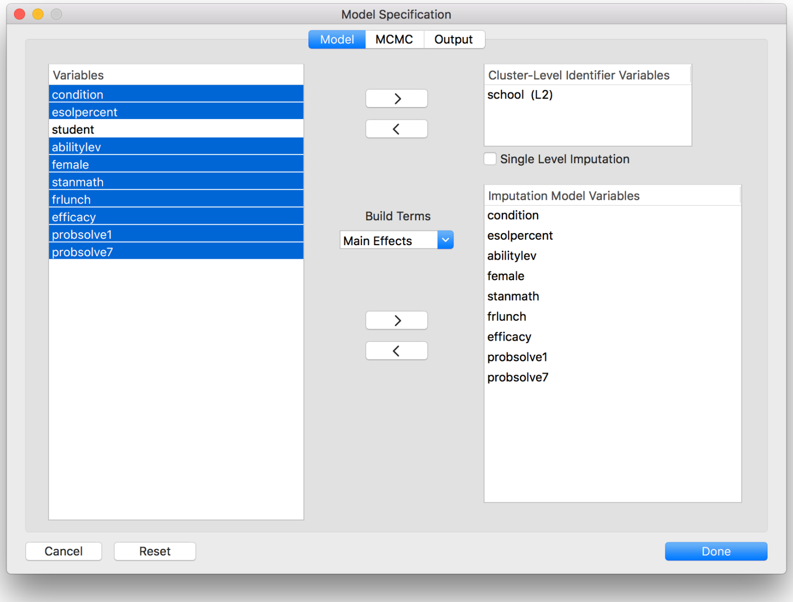
Variable Name	Variable Type
school	Continuous
condition	Nominal
esolpercent	Continuous
student	Continuous
abilitylev	Nominal
female	Nominal
stanmath	Continuous
frlunch	Nominal
efficacy	Ordinal
probsolve1	Continuous
probsolve7	Continuous



```
DATA: /Users/craig/Desktop/Blimp Examples/Ex2Level.csv;

VARIABLES: school  condition  esolpercent  student  abilitylev  female  stanmath  frlunch
efficacy  probsolve1  probsolve7;

ORDINAL: efficacy;

NOMINAL: condition  abilitylev  female  frlunch;

MISSING: 999;
```

Specify Imputation Model





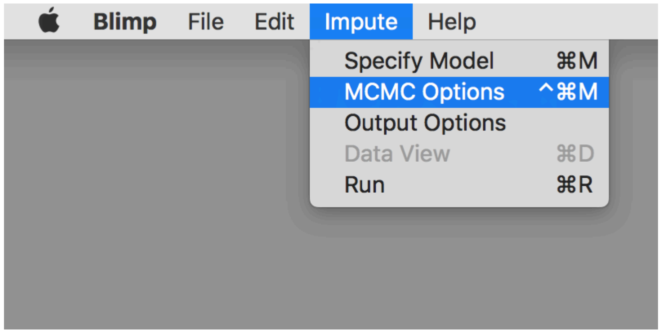
Variables
condition
esolpercent
student
abilitylev
female
stanmath
frlunch
efficacy
probsolve1
probsolve7

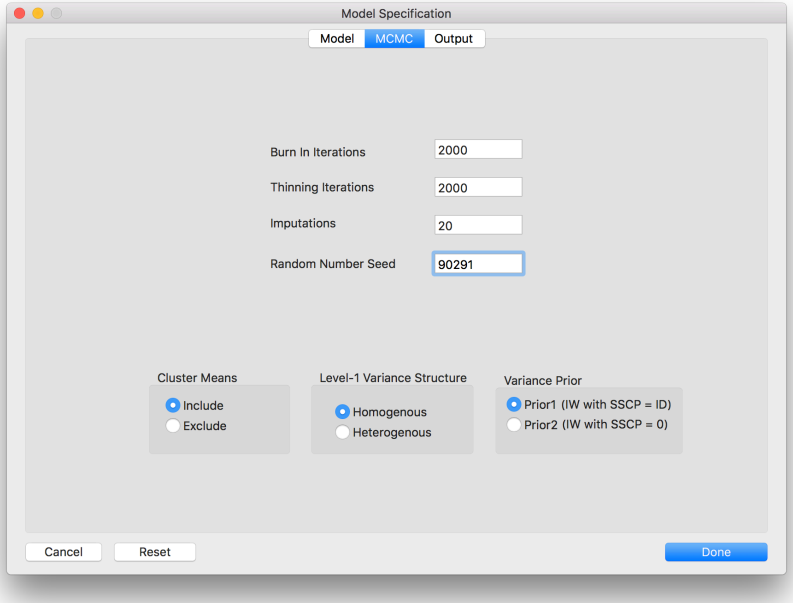
Cluster-Level Identifier Variables
school (L2)

□ Single Level Imputation

Imputation Model Variables
condition
esolpercent
abilitylev
female
stanmath
frlunch
efficacy
probsolve1
probsolve7

Specify Algorithmic Options





Burn In Iterations 2000

Thinning Iterations 2000

Imputations 20

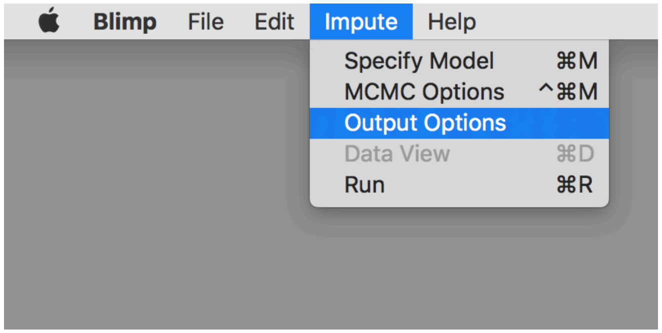
Random Number Seed 90291

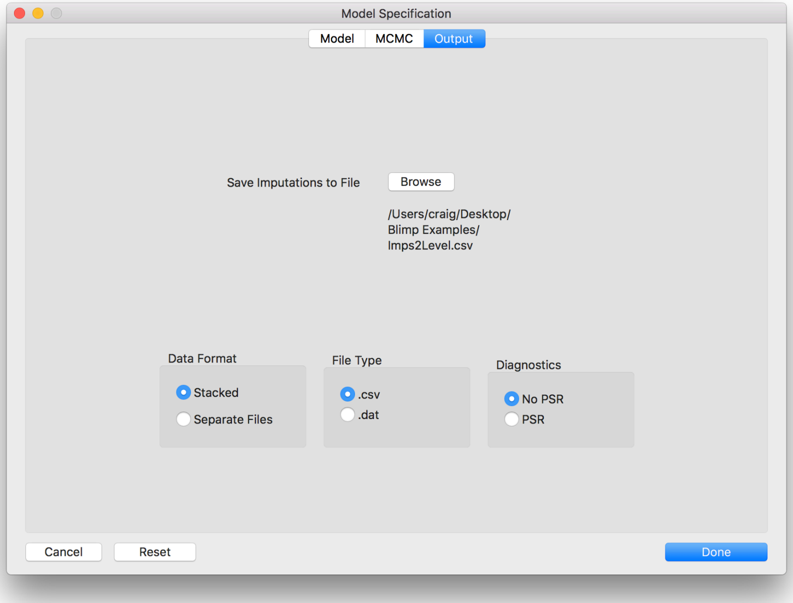
Cluster Means
- ● Include
- ○ Exclude

Level-1 Variance Structure
- ● Homogenous
- ○ Heterogenous

Variance Prior
- ● Prior1 (IW with SSCP = ID)
- ○ Prior2 (IW with SSCP = 0)

Specify Output Options





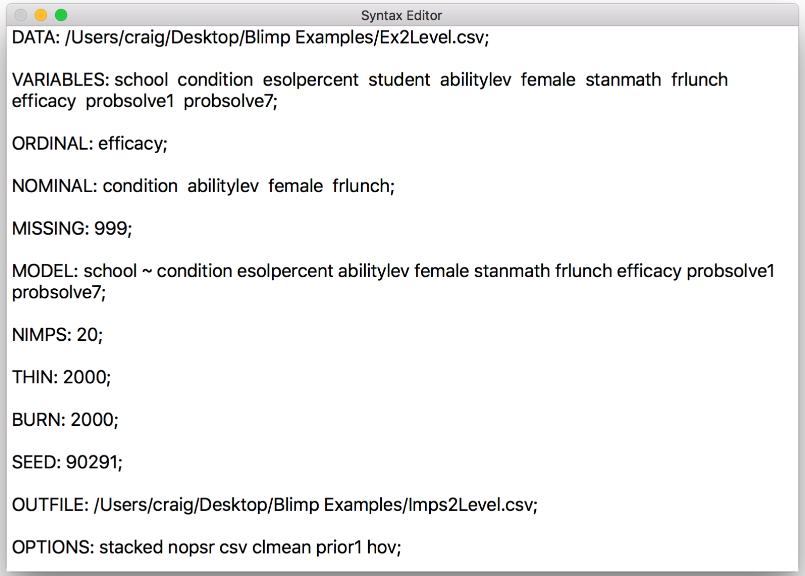
Save Imputations to File

/Users/craig/Desktop/
Blimp Examples/
Imps2Level.csv

Data Format
- ● Stacked
- ○ Separate Files

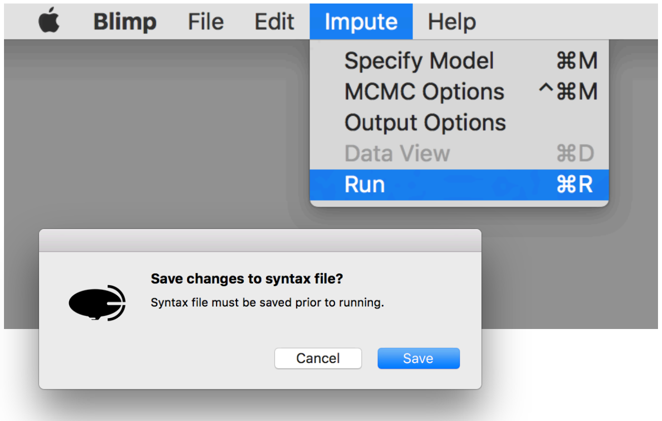
File Type
- ● .csv
- ○ .dat

Diagnostics
- ● No PSR
- ○ PSR



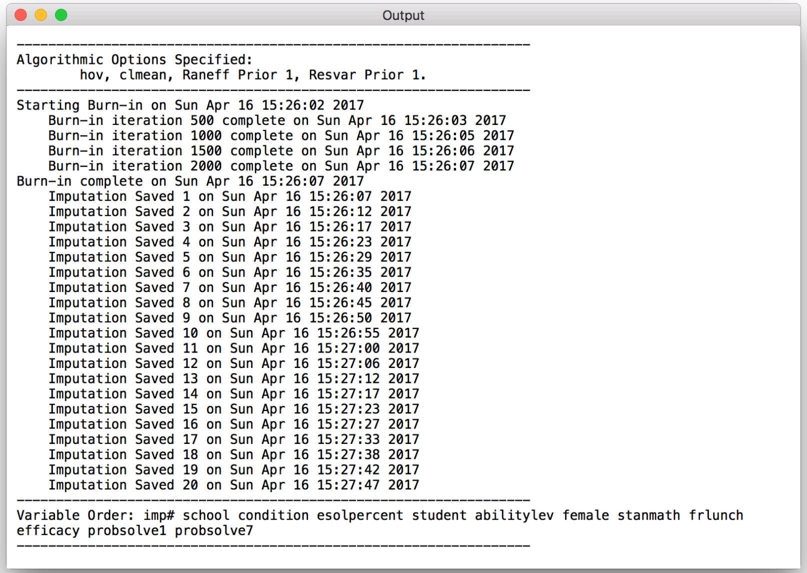
```
DATA: /Users/craig/Desktop/Blimp Examples/Ex2Level.csv;

VARIABLES: school  condition  esolpercent  student  abilitylev  female  stanmath  frlunch
efficacy  probsolve1  probsolve7;

ORDINAL: efficacy;

NOMINAL: condition  abilitylev  female  frlunch;

MISSING: 999;

MODEL: school ~ condition esolpercent abilitylev female stanmath frlunch efficacy probsolve1
probsolve7;

NIMPS: 20;

THIN: 2000;

BURN: 2000;

SEED: 90291;

OUTFILE: /Users/craig/Desktop/Blimp Examples/Imps2Level.csv;

OPTIONS: stacked nopsr csv clmean prior1 hov;
```

Run Program



Save changes to syntax file?

Syntax file must be saved prior to running.



```
-----------------------------------------------------------------
Algorithmic Options Specified:
        hov, clmean, Raneff Prior 1, Resvar Prior 1.
-----------------------------------------------------------------
Starting Burn-in on Sun Apr 16 15:26:02 2017
    Burn-in iteration 500 complete on Sun Apr 16 15:26:03 2017
    Burn-in iteration 1000 complete on Sun Apr 16 15:26:05 2017
    Burn-in iteration 1500 complete on Sun Apr 16 15:26:06 2017
    Burn-in iteration 2000 complete on Sun Apr 16 15:26:07 2017
Burn-in complete on Sun Apr 16 15:26:07 2017
    Imputation Saved 1 on Sun Apr 16 15:26:07 2017
    Imputation Saved 2 on Sun Apr 16 15:26:12 2017
    Imputation Saved 3 on Sun Apr 16 15:26:17 2017
    Imputation Saved 4 on Sun Apr 16 15:26:23 2017
    Imputation Saved 5 on Sun Apr 16 15:26:29 2017
    Imputation Saved 6 on Sun Apr 16 15:26:35 2017
    Imputation Saved 7 on Sun Apr 16 15:26:40 2017
    Imputation Saved 8 on Sun Apr 16 15:26:45 2017
    Imputation Saved 9 on Sun Apr 16 15:26:50 2017
    Imputation Saved 10 on Sun Apr 16 15:26:55 2017
    Imputation Saved 11 on Sun Apr 16 15:27:00 2017
    Imputation Saved 12 on Sun Apr 16 15:27:06 2017
    Imputation Saved 13 on Sun Apr 16 15:27:12 2017
    Imputation Saved 14 on Sun Apr 16 15:27:17 2017
    Imputation Saved 15 on Sun Apr 16 15:27:23 2017
    Imputation Saved 16 on Sun Apr 16 15:27:27 2017
    Imputation Saved 17 on Sun Apr 16 15:27:33 2017
    Imputation Saved 18 on Sun Apr 16 15:27:38 2017
    Imputation Saved 19 on Sun Apr 16 15:27:42 2017
    Imputation Saved 20 on Sun Apr 16 15:27:47 2017
-----------------------------------------------------------------
Variable Order: imp# school condition esolpercent student abilitylev female stanmath frlunch
efficacy probsolve1 probsolve7
-----------------------------------------------------------------
```

Pooling with R Package mitml

```
# Required packages
library(mitml)
library(lme4)

# Read data
imputations <- read.csv("~/desktop/Blimp Examples/Imps2Level.csv", header = F)
names(imputations) <- c("imputation", "school", "condition", “esolpercent",
  "student", "abilitylev", "female", "stanmath", "frlunch", “efficacy”,
  "probsolve1", "probsolve7")
imputations$abilitylev <- factor(imputations$abilitylev)

# Analyze data and pool estimates
model <- "probsolve7 ~ probsolve1 + efficacy + abilitylev + female +
  esolpercent + condition + (1|school)"
implist <-  as.mitml.list(split(imputations, imputations$imputation))
mlm <- with(implist, lmer(model, REML = F))
estimates <- testEstimates(mlm, var.comp = T, df.com = NULL)

# Display estimates
estimates
```

mitml Output

```
Final parameter estimates and inferences obtained from 20 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV
(Intercept)    55.932     4.928    11.349   500.705     0.000     0.242
probsolve1      0.416     0.040    10.330   297.510     0.000     0.338
efficacy        0.721     0.273     2.641   157.466     0.005     0.532
abilitylev2     1.169     1.526     0.766   131.473     0.222     0.613
abilitylev3     2.843     1.680     1.693   185.041     0.046     0.472
female          0.324     0.733     0.442   284.297     0.329     0.349
esolpercent     0.063     0.042     1.525  4350.615     0.064     0.071
condition       4.779     1.931     2.475  2174.122     0.007     0.103

                              Estimate
Intercept~~Intercept|school   18.582
Residual~~Residual            89.179
ICC|school                     0.172

Unadjusted hypothesis test as appropriate in larger samples.
```

Centering Predictors

Centering is performed post-imputation because the means are unknown with missing data

$$x_{ij(cent)}^{(m)} = x_{ij}^{(m)} - x_{c}^{(m)}$$

$$x_{j(cent)}^{(m)} = x_{j}^{(m)} - x_{c}^{(m)}$$

Centering constants (e.g., grand or group mean)

Center variables at imputation-specific constants

Pooling with R Package mitml

```
# Required packages
library(mitml)
library(lme4)

# Read data
imputations <- read.csv("~/Desktop/ex/Imps2Level.csv", header = F)
names(imputations) <- c("imputation", "school", "condition",
"esolpercent", "student",
  "abilitylev", "female", "stanmath", "frlunch", "efficacy",
"probsolve1", "probsolve7")

# Create Dummy codes (Factor 1 is reference)
imputations$abilitylev <- factor(imputations$abilitylev)
dummyCodes <- model.matrix( ~ imputations$abilitylev)
imputations$abilityleveD1 <- dummyCodes[,2]
imputations$abilityleveD2 <- dummyCodes[,3]

# Create imputations as a list
imputationList <- split(imputations, imputations$imputation)
```

Pooling with R Package mitml, Cont.

```
# Grand mean centering
impListCent <- lapply(imputationList,function(dat) {
    # Variables needing centering
    vars <- c("esolpercent", "student", "female", "stanmath",
      "frlunch", "efficacy", "probsolve1","abilityleveD1", "abilityleveD2")
    # Get grand means
    mns <- colMeans(dat[,vars])
    # Center
    dat[,vars] <- sweep(dat[,vars],2,mns)
    # Return data
    return(dat)
})

# Create imputations as mitml List
implistCent <-  as.mitml.list(impListCent)

# Analyze data and pool estimates
model <- "probsolve7 ~ probsolve1 + efficacy + abilitylev + female +
  esolpercent + condition + (1|school)"
mlm <- with(implistCent, lmer(model, REML = F))
estimates <- testEstimates(mlm, var.comp = T, df.com = NULL)
```

Multiple Imputation Significance Tests

Pooling Covariance Matrices

$$\mathrm{cov}_w = \frac{1}{M}\sum_{m=1}^{M} \mathrm{cov}(\hat{\boldsymbol{\theta}})^{(m)}$$

Average covariance matrix

$$\mathrm{cov}_b = \frac{1}{M-1}\sum_{m=1}^{M} (\hat{\boldsymbol{\theta}}^{(m)} - \bar{\boldsymbol{\theta}})(\hat{\boldsymbol{\theta}}^{(m)} - \bar{\boldsymbol{\theta}})^T$$

Variance across imputations

$$\bar{r} = \left(1 + m^{-1}\right)\mathrm{tr}\left(\mathrm{cov}_b \mathrm{cov}_w^{-1}\right) / k$$

Average proportional increase in variance

Wald Test Statistic

Evaluating the Wald statistic to a chi-square (shown below) or *F* distribution gives a *p*-value

$$\mathrm{W} = \frac{\left(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\right)^T \mathrm{cov}_w^{-1} \left(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}_0\right)}{1 + \bar{r}}$$

Wald based on pooled quantities

Inflation factor

Wald Test With mitml

```
# Empty model
model1 <- "probsolve7 ~ (1|school)"
mlm1 <- with(implist, lmer(model1, REML = F))
estimates1 <- testEstimates(mlm1, var.comp = T, df.com = NULL)
estimates1

# Covariates only
model2 <- "probsolve7 ~ probsolve1 + efficacy + abilitylev +
  female + esolpercent + (1|school)"
mlm2 <- with(implist, lmer(model2, REML = F))
estimates2 <- testEstimates(mlm2, var.comp = T, df.com = NULL)
estimates2

# Compare models with Wald test
testModels(mlm2, mlm1, method = "D1")
```

Output

```
Model comparison calculated from 20 imputed data sets.
Combination method: D1

    F.value      df1      df2  p.value      RIV
     28.657        6 1615.839    0.000    0.347

Unadjusted hypothesis test as appropriate in larger samples.
```

First And Second Pass Test Statistics

Pass 1: Average likelihood ratio statistic

$$\bar{T}_1 = \frac{1}{M}\sum_{m=1}^{M} -2l\left(\boldsymbol{\theta}_0^{(t)} | \mathbf{Y}^{(t)}\right) + 2l\left(\boldsymbol{\theta}_1^{(t)} | \mathbf{Y}^{(t)}\right)$$

Pass 2: Average test statistic with likelihood evaluated at the pooled estimates

$$\bar{T}_2 = \frac{1}{M}\sum_{m=1}^{M} -2l\left(\bar{\boldsymbol{\theta}}_0 | \mathbf{Y}^{(t)}\right) + 2l\left(\bar{\boldsymbol{\theta}}_1 | \mathbf{Y}^{(t)}\right)$$

Meng And Rubin (1992) Test Statistic

The LRT can be evaluated against a chi-square (shown below) or *F* distribution

$$\text{LRT} = \frac{\bar{T}_2}{1 + \bar{r}}$$

LRT based on pooled quantities

Inflation factor

$$\bar{r} = \frac{m+1}{k(m-1)}\left(\bar{T}_1 - \bar{T}_2\right)$$

Average proportional increase in variance

Likelihood Ratio Test With mitml

```
# Random intercept model
model1 <- "probsolve7 ~ probsolve1 + efficacy + abilitylev + female +
  esolpercent + condition + (1|school)"
mlm1 <- with(implist, lmer(model1, REML = F))
estimates1 <- testEstimates(mlm1, var.comp = T, df.com = NULL)
estimates1

# Random slope for self-efficacy
model2 <- "probsolve7 ~ probsolve1 + efficacy + abilitylev + female +
  esolpercent + condition + (efficacy|school)"
mlm2 <- with(implist, lmer(model2, REML = F))
estimates2 <- testEstimates(mlm2, var.comp = T, df.com = NULL)
estimates2

# Compare models with Meng and Rubin likelihood ratio test
testModels(mlm2, mlm1, method = "D3")
```

Output

```
Model comparison calculated from 20 imputed data sets.
Combination method: D3

   F.value     df1      df2 p.value     RIV
     0.085       2 786.816   0.918   0.249
```

Three-Level Analysis Example

Motivating Example

Data from a cluster-randomized study investigating a math problem-solving curriculum

29 schools (level-3 units) were randomly assigned to an intervention or control condition

The average number of students (level-2 units) per school was 33.86, with a range of 13 to 61

Seven (approximately) monthly assessments with planned missing data and attrition

Input Data

	Variable	Description	Missing	Metric
Level-3	school	School identifier variable		
	condition	Treatment code (0 = control, 1 = intervention)		Nominal
	esolpercent	Percentage of English as second language	*	Numeric
Level-2	student	Student identifier		
	abilitylev	Ability grouping (3-group classification)	*	Nominal
	female	Female dummy code		Nominal
	stanmath	Standardized math test scores	*	Numeric
	frlunch	Lunch assistance dummy code	*	Nominal
Level-1	wave	Assessment wave		
	time	Months since baseline		Numeric
	condbytime	Condition by time interaction		Numeric
	probsolve	Math problem-solving score	*	Numeric
	efficacy	Math self-efficacy 6-point rating scale	*	Ordinal

Analysis Model

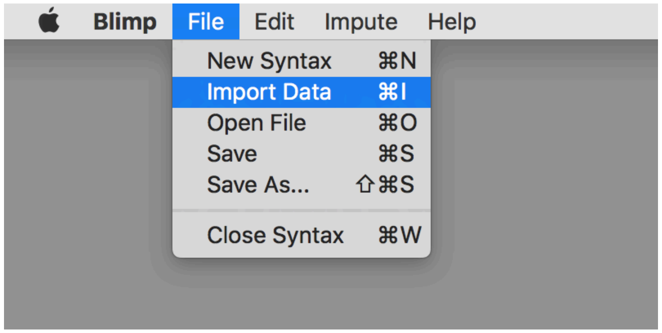
The substantive analysis model examines the intervention by time interaction, controlling for covariates at each level

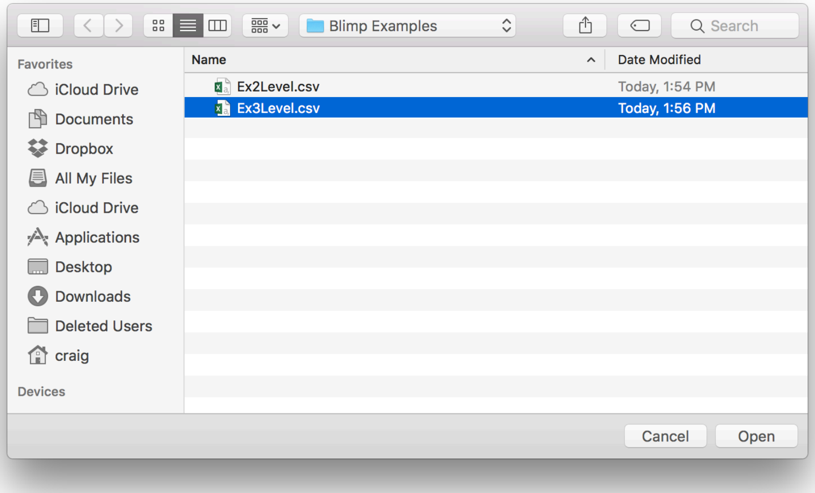
$$probsolve_{ijk} = \gamma_0 + \gamma_1(efficacy_{ijk}) + \gamma_2(time_{ijk}) + \gamma_3(condbytime_{ijk}) + \gamma_4(abilitylev3_{jk}) + \gamma_5(abilitylev3_{jk}) + \gamma_6(female_{jk}) + \gamma_7(esolpercent_k) + \gamma_8(condition_k) + r_{0jk} + r_{1jk}(time_{ijk}) + u_{0k} + u_{1k}(time_{ijk}) + \varepsilon_{ijk}$$

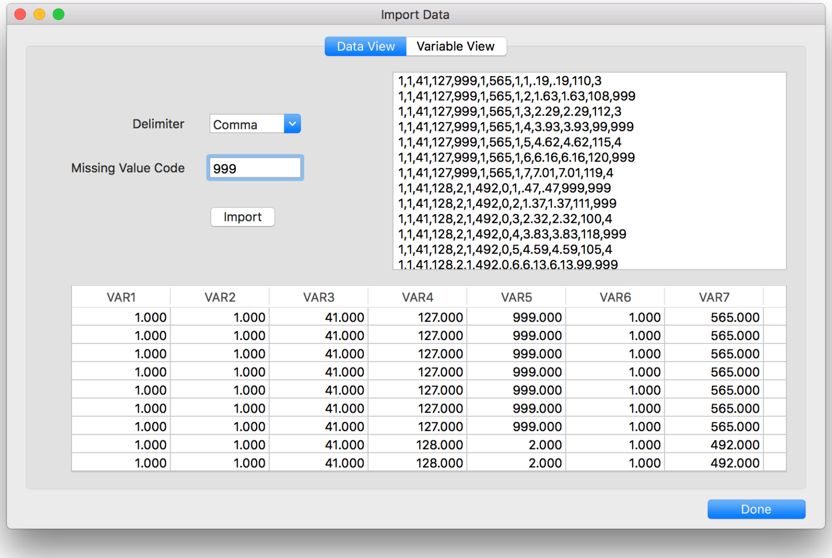
Blimp Syntax

```
DATA: ~/Desktop/Blimp Examples/Ex3Level.csv;
VARIABLES: school condition esolpercent student abilitylev
  female stanmath frlunch wave time condbytime probsolve
  efficacy;
ORDINAL: efficacy;
NOMINAL: condition abilitylev  female  frlunch;
MISSING: 999;
MODEL: student school ~ condition esolpercent abilitylev
  female stanmath frlunch condbytime efficacy
  time:probsolve;
NIMPS: 20;
THIN: 2000;
BURN: 2000;
SEED: 90291;
OUTFILE: ~/Desktop/Blimp Examples/Imps3Level.csv;
OPTIONS: stacked nopsr csv clmean prior1 hov;
```

Import Data





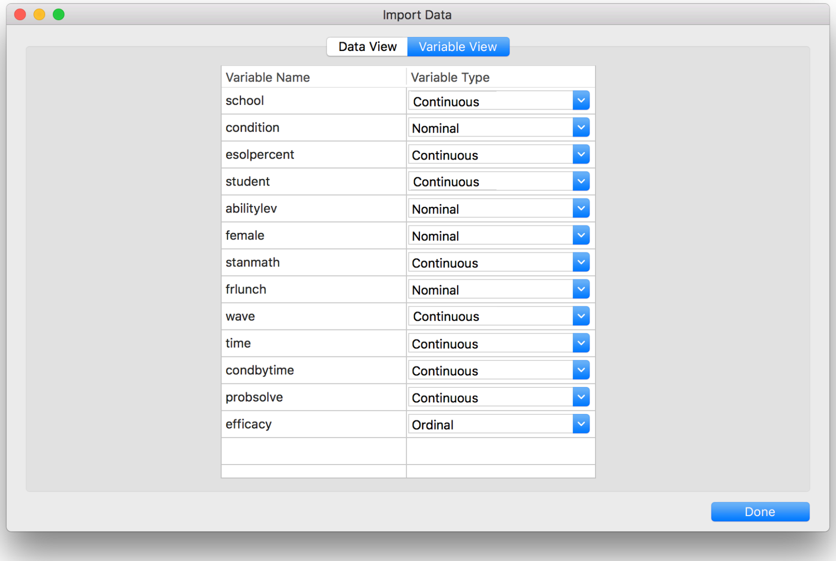


Delimiter Comma

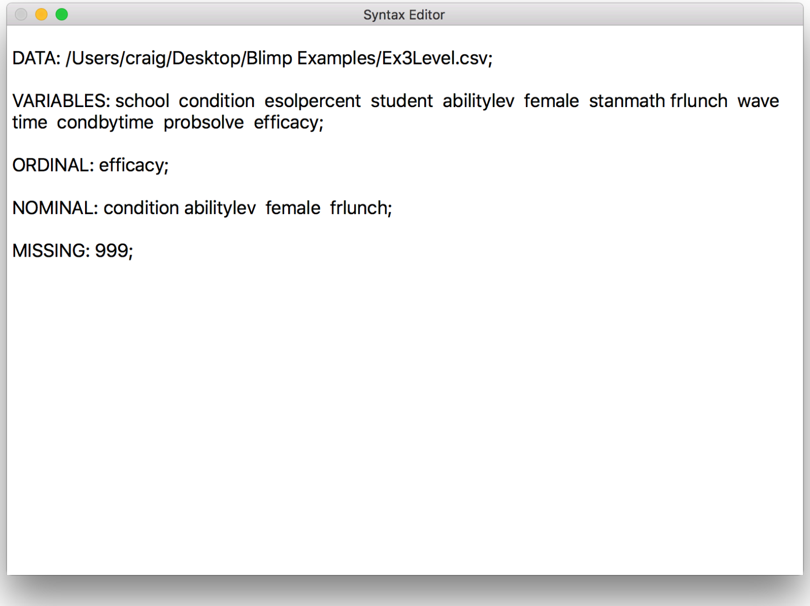
Missing Value Code 999

```
1,1,41,127,999,1,565,1,1,.19,.19,110,3
1,1,41,127,999,1,565,1,2,1.63,1.63,108,999
1,1,41,127,999,1,565,1,3,2.29,2.29,112,3
1,1,41,127,999,1,565,1,4,3.93,3.93,99,999
1,1,41,127,999,1,565,1,5,4.62,4.62,115,4
1,1,41,127,999,1,565,1,6,6.16,6.16,120,999
1,1,41,127,999,1,565,1,7,7.01,7.01,119,4
1,1,41,128,2,1,492,0,1,.47,.47,999,999
1,1,41,128,2,1,492,0,2,1.37,1.37,111,999
1,1,41,128,2,1,492,0,3,2.32,2.32,100,4
1,1,41,128,2,1,492,0,4,3.83,3.83,118,999
1,1,41,128,2,1,492,0,5,4.59,4.59,105,4
1,1,41,128,2,1,492,0,6,6.13,6.13,99,999
```

VAR1	VAR2	VAR3	VAR4	VAR5	VAR6	VAR7
1.000	1.000	41.000	127.000	999.000	1.000	565.000
1.000	1.000	41.000	127.000	999.000	1.000	565.000
1.000	1.000	41.000	127.000	999.000	1.000	565.000
1.000	1.000	41.000	127.000	999.000	1.000	565.000
1.000	1.000	41.000	127.000	999.000	1.000	565.000
1.000	1.000	41.000	127.000	999.000	1.000	565.000
1.000	1.000	41.000	127.000	999.000	1.000	565.000
1.000	1.000	41.000	128.000	2.000	1.000	492.000
1.000	1.000	41.000	128.000	2.000	1.000	492.000

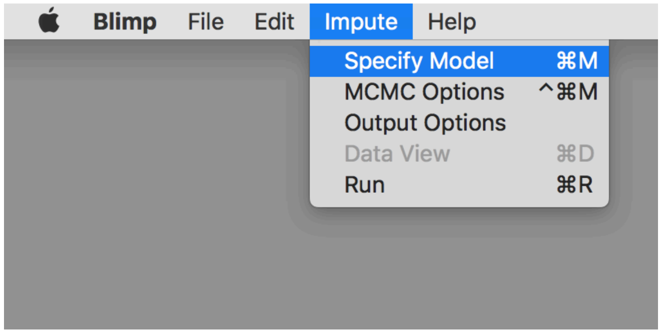


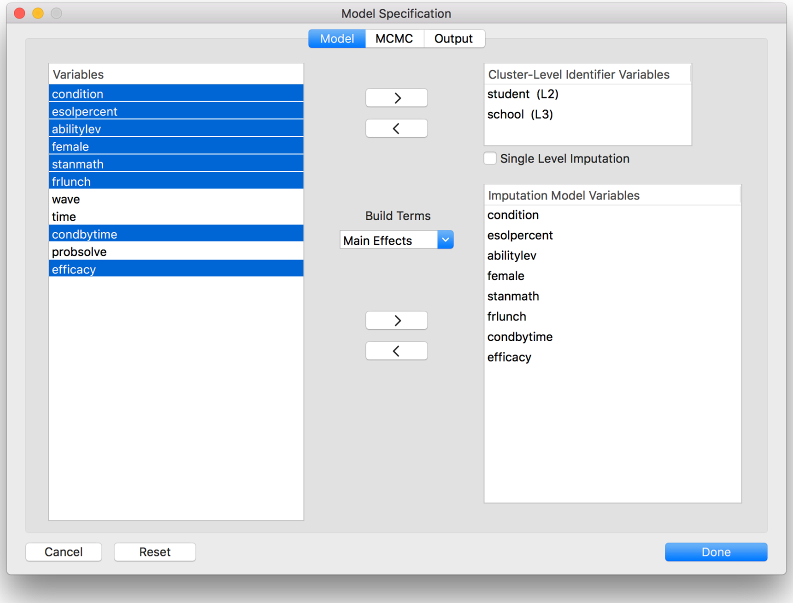
Variable Name	Variable Type
school	Continuous
condition	Nominal
esolpercent	Continuous
student	Continuous
abilitylev	Nominal
female	Nominal
stanmath	Continuous
frlunch	Nominal
wave	Continuous
time	Continuous
condbytime	Continuous
probsolve	Continuous
efficacy	Ordinal



```
DATA: /Users/craig/Desktop/Blimp Examples/Ex3Level.csv;

VARIABLES: school  condition  esolpercent  student  abilitylev  female  stanmath frlunch  wave
time  condbytime  probsolve  efficacy;

ORDINAL: efficacy;

NOMINAL: condition abilitylev  female  frlunch;

MISSING: 999;
```

Specify Imputation Model





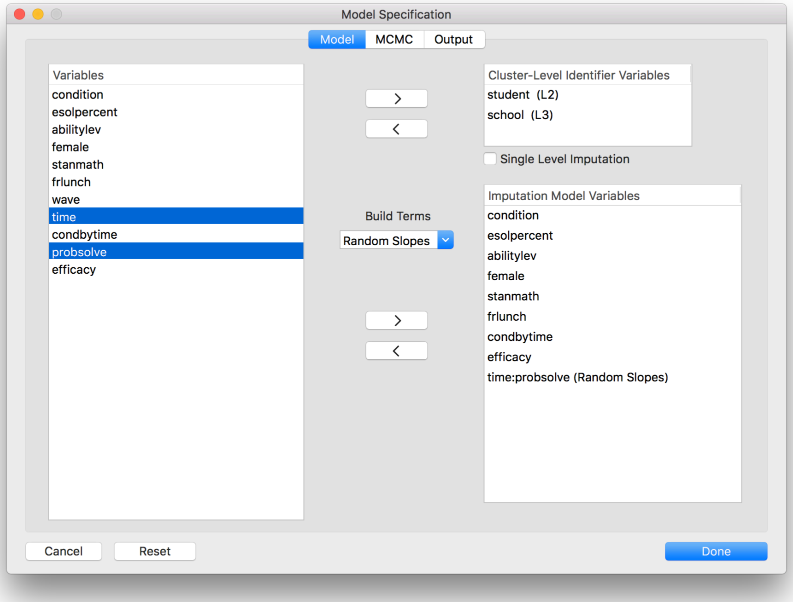
Variables
condition
esolpercent
abilitylev
female
stanmath
frlunch
wave
time
condbytime
probsolve
efficacy

Cluster-Level Identifier Variables
student (L2)
school (L3)

□ Single Level Imputation

Build Terms: Main Effects

Imputation Model Variables
condition
esolpercent
abilitylev
female
stanmath
frlunch
condbytime
efficacy



Variables
condition
esolpercent
abilitylev
female
stanmath
frlunch
wave
time
condbytime
probsolve
efficacy

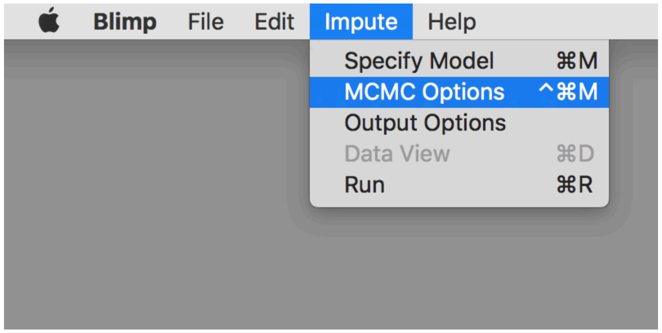
Cluster-Level Identifier Variables
student (L2)
school (L3)

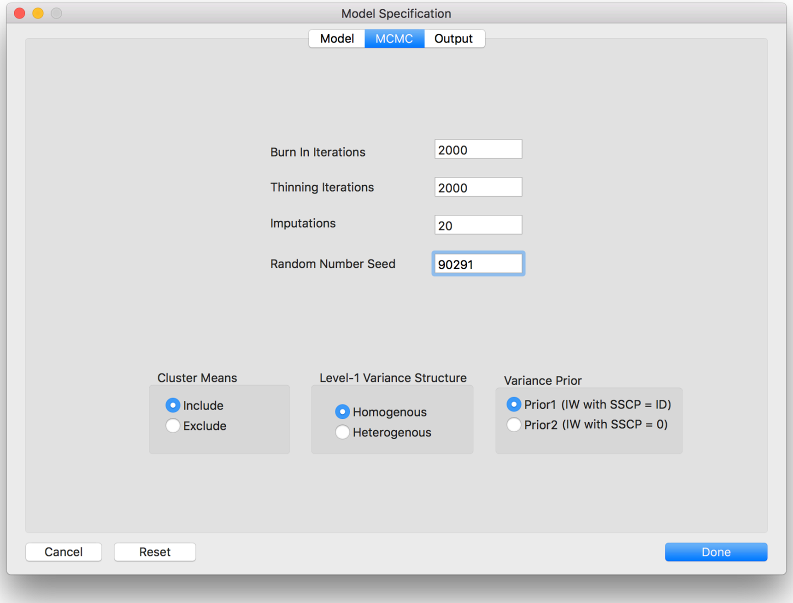
□ Single Level Imputation

Build Terms: Random Slopes

Imputation Model Variables
condition
esolpercent
abilitylev
female
stanmath
frlunch
condbytime
efficacy
time:probsolve (Random Slopes)

Specify Algorithmic Options





Burn In Iterations: 2000

Thinning Iterations: 2000

Imputations: 20

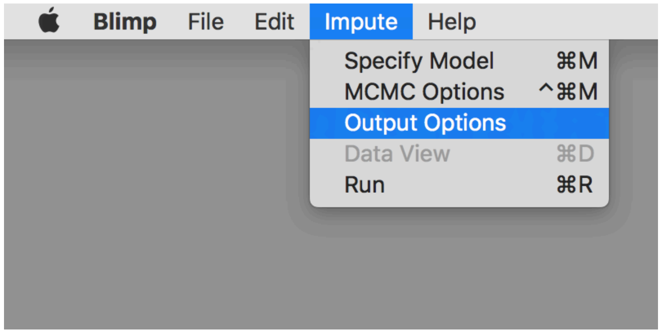
Random Number Seed: 90291

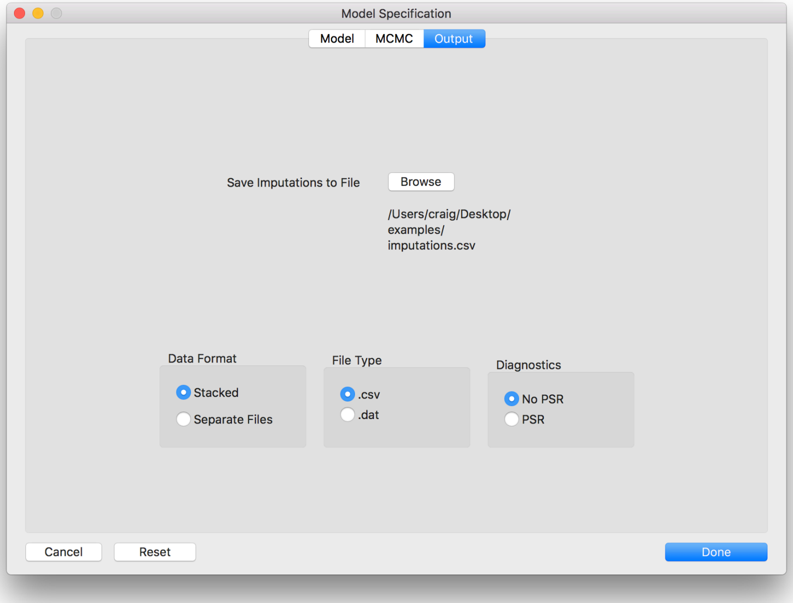
Cluster Means
● Include
○ Exclude

Level-1 Variance Structure
● Homogenous
○ Heterogenous

Variance Prior
● Prior1 (IW with SSCP = ID)
○ Prior2 (IW with SSCP = 0)

Specify Output Options





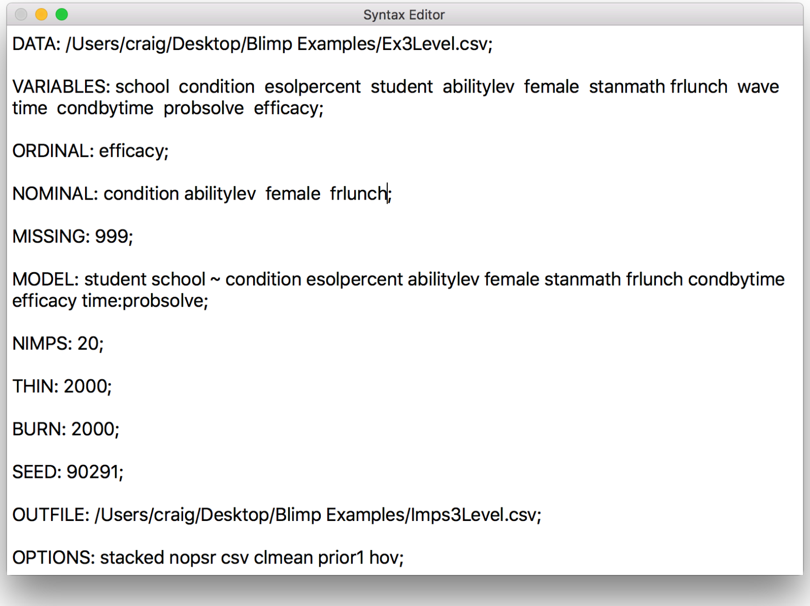
Save Imputations to File

/Users/craig/Desktop/
examples/
imputations.csv

Data Format
● Stacked
○ Separate Files

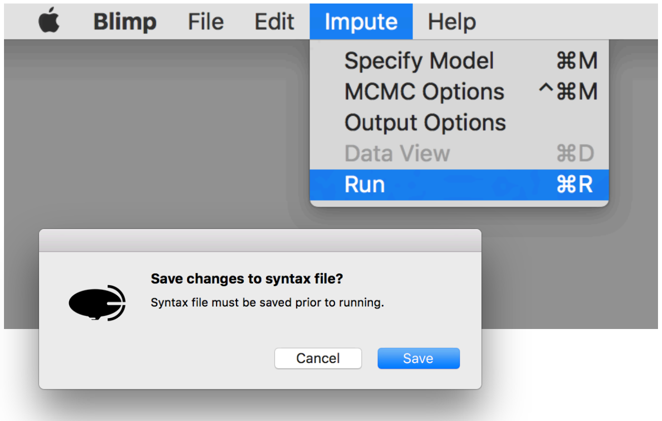
File Type
● .csv
○ .dat

Diagnostics
● No PSR
○ PSR



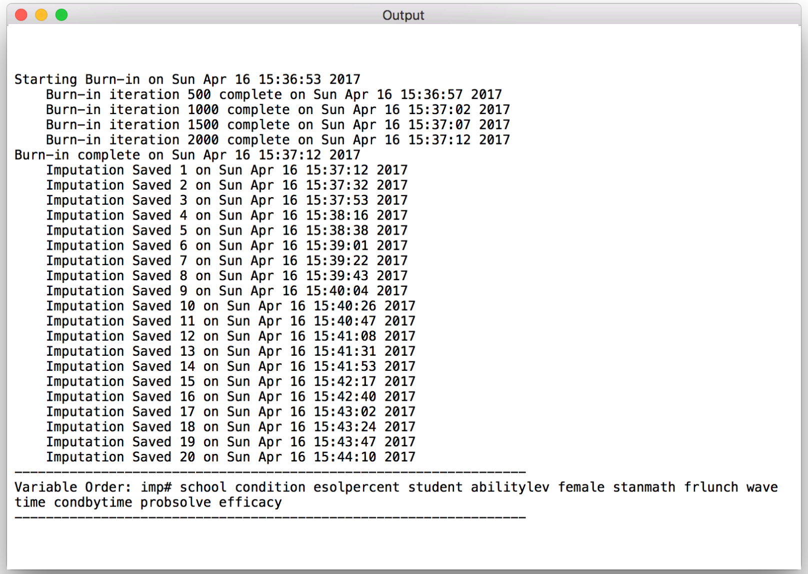
```
DATA: /Users/craig/Desktop/Blimp Examples/Ex3Level.csv;

VARIABLES: school  condition  esolpercent  student  abilitylev  female  stanmath frlunch  wave
time  condbytime  probsolve  efficacy;

ORDINAL: efficacy;

NOMINAL: condition abilitylev  female  frlunch;

MISSING: 999;

MODEL: student school ~ condition esolpercent abilitylev female stanmath frlunch condbytime
efficacy time:probsolve;

NIMPS: 20;

THIN: 2000;

BURN: 2000;

SEED: 90291;

OUTFILE: /Users/craig/Desktop/Blimp Examples/Imps3Level.csv;

OPTIONS: stacked nopsr csv clmean prior1 hov;
```

Run Program



Save changes to syntax file?

Syntax file must be saved prior to running.



```
Starting Burn-in on Sun Apr 16 15:36:53 2017
    Burn-in iteration 500 complete on Sun Apr 16 15:36:57 2017
    Burn-in iteration 1000 complete on Sun Apr 16 15:37:02 2017
    Burn-in iteration 1500 complete on Sun Apr 16 15:37:07 2017
    Burn-in iteration 2000 complete on Sun Apr 16 15:37:12 2017
Burn-in complete on Sun Apr 16 15:37:12 2017
    Imputation Saved 1 on Sun Apr 16 15:37:12 2017
    Imputation Saved 2 on Sun Apr 16 15:37:32 2017
    Imputation Saved 3 on Sun Apr 16 15:37:53 2017
    Imputation Saved 4 on Sun Apr 16 15:38:16 2017
    Imputation Saved 5 on Sun Apr 16 15:38:38 2017
    Imputation Saved 6 on Sun Apr 16 15:39:01 2017
    Imputation Saved 7 on Sun Apr 16 15:39:22 2017
    Imputation Saved 8 on Sun Apr 16 15:39:43 2017
    Imputation Saved 9 on Sun Apr 16 15:40:04 2017
    Imputation Saved 10 on Sun Apr 16 15:40:26 2017
    Imputation Saved 11 on Sun Apr 16 15:40:47 2017
    Imputation Saved 12 on Sun Apr 16 15:41:08 2017
    Imputation Saved 13 on Sun Apr 16 15:41:31 2017
    Imputation Saved 14 on Sun Apr 16 15:41:53 2017
    Imputation Saved 15 on Sun Apr 16 15:42:17 2017
    Imputation Saved 16 on Sun Apr 16 15:42:40 2017
    Imputation Saved 17 on Sun Apr 16 15:43:02 2017
    Imputation Saved 18 on Sun Apr 16 15:43:24 2017
    Imputation Saved 19 on Sun Apr 16 15:43:47 2017
    Imputation Saved 20 on Sun Apr 16 15:44:10 2017
-----------------------------------------------------------------
Variable Order: imp# school condition esolpercent student abilitylev female stanmath frlunch wave
time condbytime probsolve efficacy
-----------------------------------------------------------------
```

Pooling with R Package mitml

```
# Required packages
library(mitml)
library(lme4)

# Read data
imputations <- read.csv("~/desktop/Blimp Examples/Imps3Level.csv", header = F)
names(imputations) <- c("imputation", "school", "condition", “esolpercent”,
  "student", "abilitylev", "female", "stanmath", "frlunch", "wave", “time",
  "condbytime", “probsolve", "efficacy")
imputations$abilitylev <- factor(imputations$abilitylev)

# Analyze data and pool estimates
model <- "probsolve ~ efficacy + time + condbytime + abilitylev + female +
  esolpercent + condition + (time|student:school) + (time|school)"
implist <-  as.mitml.list(split(imputations, imputations$imputation))
mlm <- with(implist, lmer(model, REML = F))
estimates <- testEstimates(mlm, var.comp = T, df.com = NULL)

# Display estimates
estimates
```

mitml Output

```
Final parameter estimates and inferences obtained from 20 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV
(Intercept)    92.715     1.917    48.373   549.605     0.000     0.228
efficacy        0.765     0.144     5.326    56.231     0.000     1.388
time            0.686     0.172     3.985   934.853     0.000     0.166
condbytime      0.549     0.222     2.470  1995.448     0.007     0.108
abilitylev2     0.747     0.886     0.843   321.312     0.200     0.321
abilitylev3     6.974     0.967     7.210   441.810     0.000     0.262
female         -0.530     0.439    -1.207   968.110     0.114     0.163
esolpercent     0.051     0.023     2.194  1003.065     0.014     0.160
condition       0.083     1.085     0.077  2741.808     0.469     0.091
```

mitml Output

```
                                         Estimate
Intercept~~Intercept|student:school        23.532
Intercept~~time|student:school              0.529
time~~time|student:school                   0.131
Intercept~~Intercept|school                 5.038
Intercept~~time|school                     -0.167
time~~time|school                           0.255
Residual~~Residual                         62.353
ICC|school                                  0.274
NA                                          0.075

Unadjusted hypothesis test as appropriate in larger samples.
```

Centering Incomplete Product Terms

Interaction terms can be rescaled to equal the product of deviation score variables

$$xm_{ij(cent)}^{(k)} = xm_{ij}^{(k)} - x_{ij}^{(k)} m_c^{(k)} - m_{ij}^{(k)} x_c^{(k)} + x_c^{(k)} m_c^{(k)}$$

Centering constants (e.g., grand or group mean)

Pooling with R Package mitml

```
# Required packages
library(mitml)
library(lme4)

# Read data
imputations <- read.csv("~/Desktop/ex/Imps3Level.csv", header = F)
names(imputations) <- c("imputation", "school", "condition",
"esolpercent", "student",
  "abilitylev", "female", "stanmath", "frlunch", "wave", "time",
"condbytime", "probsolve",
  "efficacy")

# Create Dummy codes (Factor 1 is reference)
imputations$abilitylev <- factor(imputations$abilitylev)
dummyCodes <- model.matrix( ~ imputations$abilitylev)
imputations$abilityleveD1 <- dummyCodes[,2]
imputations$abilityleveD2 <- dummyCodes[,3]

# Create imputations as a list
imputationList <- split(imputations, imputations$imputation)
```

mitml Output

```
Final parameter estimates and inferences obtained from 20 imputed data sets.

             Estimate Std.Error   t.value        df   p.value       RIV
(Intercept)   101.891     1.361    74.840  1398.955     0.000     0.132
efficacy        0.765     0.144     5.326    56.231     0.000     1.388
time            0.686     0.172     3.985   934.854     0.000     0.166
condbytime      0.549     0.222     2.470  1995.446     0.007     0.108
abilitylev2     0.747     0.886     0.843   321.312     0.200     0.321
abilitylev3     6.974     0.967     7.210   441.809     0.000     0.262
female         -0.530     0.439    -1.207   968.111     0.114     0.163
esolpercent     0.051     0.023     2.194  1003.064     0.014     0.160
condition       3.380     1.462     2.312 20385.340     0.010     0.031
```

Pooling with R Package mitml, Cont.

```
# Centering
impListCent <- lapply(imputationList,function(dat) {
     # Variables needing grand mean centering
     vars <- c("efficacy", "esolpercent", "female","abilityleveD1", "abilityleveD2")
     # Get grand means
     mns <- colMeans(dat[,vars])
     # Grand Mean Center
     dat[,vars] <- sweep(dat[,vars],2,mns)
     ## Center interaction
     # Time centering constant
     timeC <- 6
     # Condition constant
     condC <- 0
     # Center Time
     dat$time <- dat$time - timeC
     # Center Condition
     dat$condition <- dat$condition - condC
     # Center condbytime
     dat$condbytime <- dat$condbytime - (dat$condition*timeC) - (dat$time*condC) + (condC*timeC)
     # Return data
     return(dat)
})

# Analyze data and pool estimates
model <- "probsolve ~ efficacy + time + condbytime + abilitylev + female +
  esolpercent + condition + (time|student:school) + (time|school)"
implist <-  as.mitml.list(impListCent)
```