

simsem: SIMulated Structural Equation Modeling in R

Alexander M. Schoemann
Sunthud Pornprasertmanit
Patrick J. Miller



Monte Carlo Simulations

- Monte Carlo simulations are a popular tool for methodologists with many uses
 - Determine the accuracy of new methods
 - Compare different methods
 - Perform power analyses
 - Determine model fit in SEM

Monte Carlo Simulations

■ General steps in a Monte Carlo Simulation

1. Specify population parameters
2. Create a sample of size N , based on population parameters
3. Analyze sample data from step 2 with chosen statistical method(s).
4. Repeat steps 2 and 3 for each of r replications.

simsem

- A new R package designed to automate Monte Carlo Simulations using SEM
- simsem can:
 - Generate data
 - Modify generated data
 - Analyze data
 - Summarize results
 - Use multiple processors across simulations

simsem Features

■ Data generation

- Currently only continuous data are generated.
 - By default data are generated from a multivariate normal distribution
 - Both manifest and latent variables can have non-normal distributions
- Data can be generated from a covariance matrix and mean vector or through a series of linear equations.

simsem Features

- Data generation (continued)
 - Data can be generated with population misfit
 - Data can be generated with continuously varying parameters
 - Generating and analysis models are specified using LISREL matrices

simsem Features

- Data modification

- Many types of missing data mechanisms can be simulated

- MCAR

- MAR

- Planned missing data designs

- “3” Form Design

- “2” Method Design

simsem Features

■ Data analysis

- All models are fit using lavaan
 - Robust ML estimators are available
 - FIML or multiple imputation can be used when data are missing
 - Equality constraints can be included
- Multiple imputation of missing data is performed with Amelia
 - Data are imputed, analyzed and results are combined with Rubin's Rules for each replication

simsem Features

■ Summarizing Results

- Results from a simulation can be automatically summarized
- Results for each model parameter include:
 - Parameter bias
 - Standard error bias
 - Confidence interval coverage
 - Power

Example 1: Power Analysis

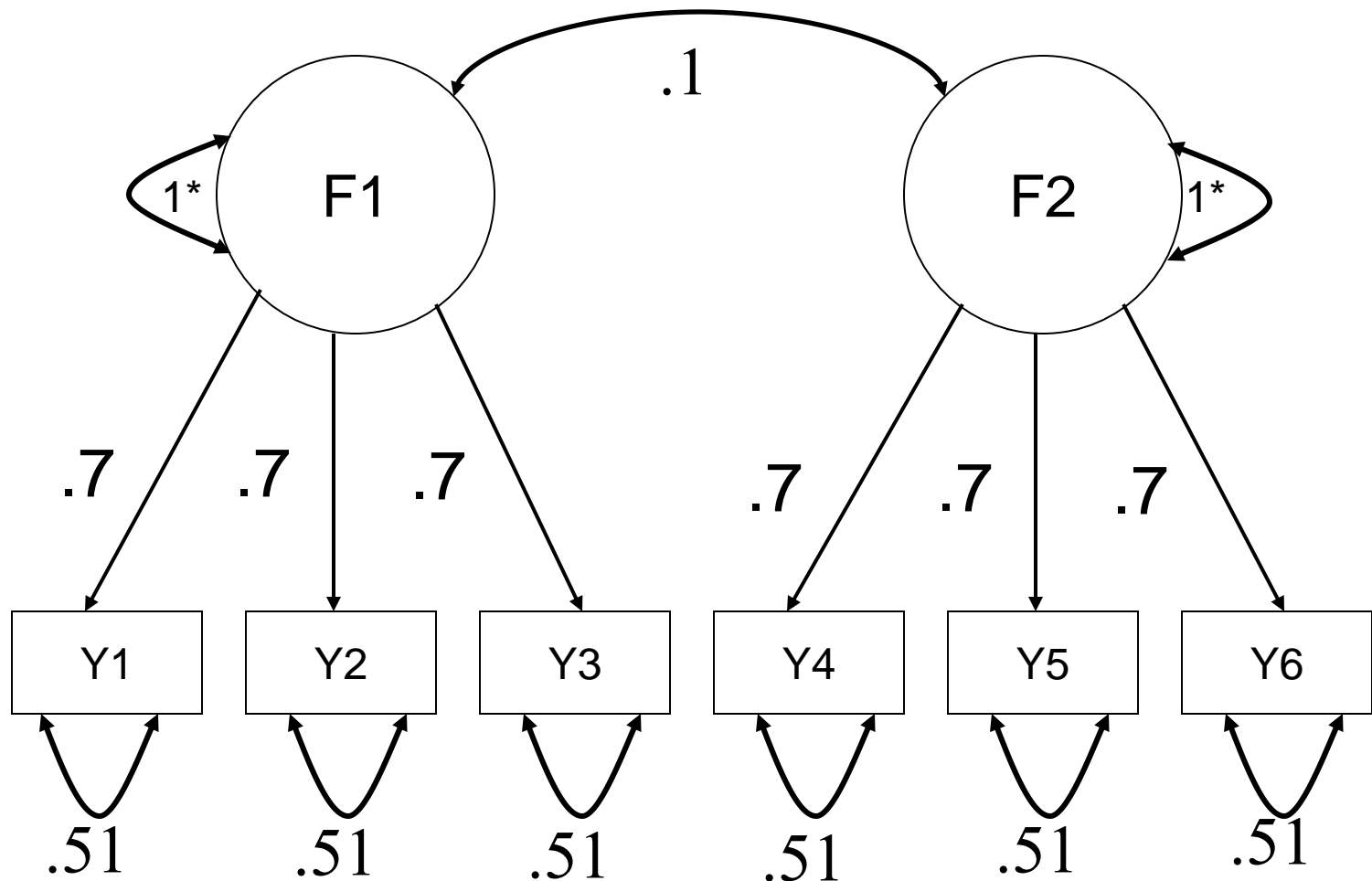
- Given population parameters, what sample size will result in a given level of power (e.g., .80)?
 - Continuously varying sample size approach
 - Specify model and a range of sample sizes
 - Generate 3000 replications varying sample size across replications
 - Record each parameter's significance for each replication (0 not sig., 1 sig.)

Example 1: Power Analysis

- Given population parameters, what sample size will results in a given level of power (e.g., .80)?
 - Use logistic regression to predict a parameter's significance (across all replications) from the sample size of each replication.
 - The predicted probability from the logistic regression at a given N is power for that parameter at that N

$$p = \frac{e^{B_0+B_1N}}{1 + e^{B_0+B_1N}}$$

Example 1: Power Analysis



Example 1: Power Analysis

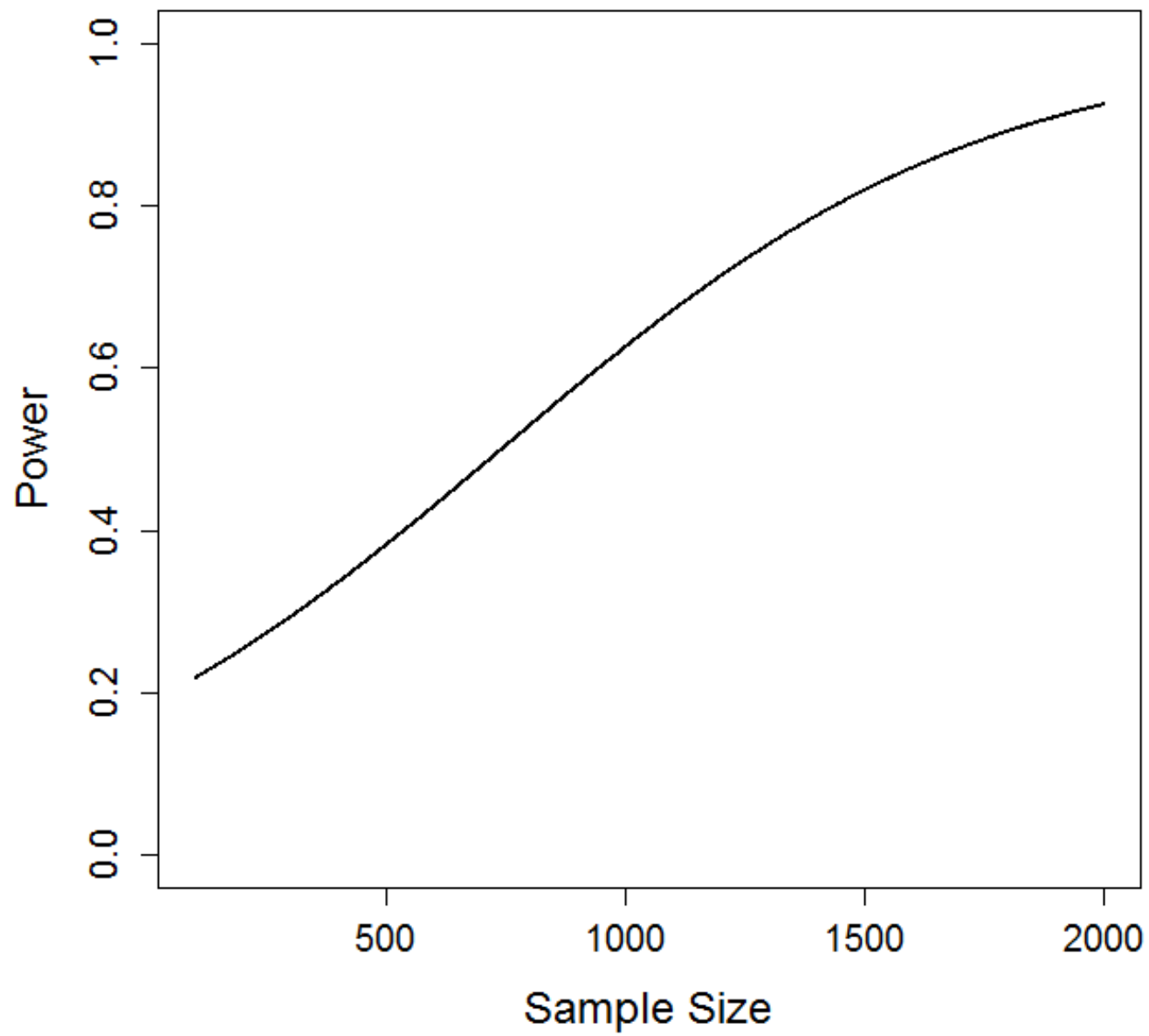
$$\begin{array}{l}
 LY = \begin{array}{cc} 0.7 & 0 \\ 0.7 & 0 \\ 0.7 & 0 \\ 0 & 0.7 \\ 0 & 0.7 \\ 0 & 0.7 \end{array}
 \end{array}$$

$$PS = \begin{array}{cc} 1 & 0.1 \\ 0.1 & 1 \end{array}$$

$$TE = \begin{array}{cccccc} 0.51 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.51 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.51 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.51 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.51 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.51 \end{array}$$

Example 1: Power Analysis

- Results: What sample size results in power for the latent correlation of .80?
 - 3000 replications, randomly varying N between 100-2000
 - $\text{logit}(\text{power}) = \beta_0 + \beta_1 N$
 - Power = .80 when $N = 1436$

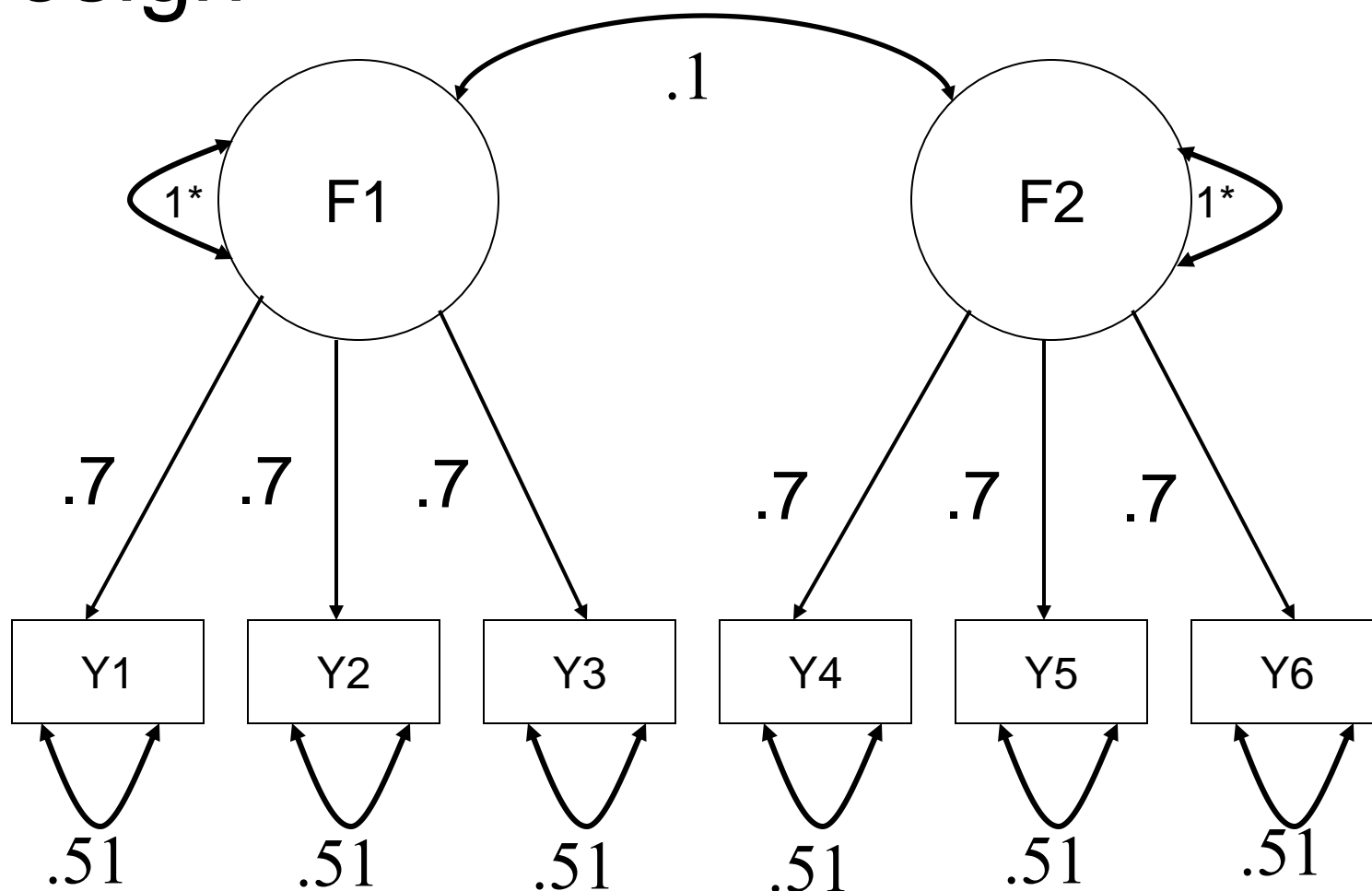


Example 2: Planned Missing Data Design

- Investigate power and bias in a 3 form planned missing data design

| Form | Common Set X | Variable Set A | Variable Set B | Variable Set C |
|------|--------------|----------------|----------------|----------------|
| 1 | ¼ of items | ¼ of items | ¼ of items | Missing |
| 2 | ¼ of items | ¼ of items | Missing | ¼ of items |
| 3 | ¼ of items | Missing | ¼ of items | ¼ of items |

Example 2: Planned Missing Data Design



Example 2: Planned Missing Data Design

- Planned missing design:
 - X block: Y1 and Y4
 - A block: Y2 and Y5
 - B block: Y3
 - C Block: Y4
- Missing data is handled with 5 imputations in Amelia
- $N = 500$
- 1000 replications

Example 2: Results

| | Fit Indices Cutoffs | | | | |
|-------|---------------------|----------|----------|----------|----------|
| | 0.1 | 0.05 | 0.01 | 0.001 | Mean |
| Chi | 14.793 | 18.995 | 27.851 | 43.315 | 5.420 |
| AIC | 7962.256 | 7986.963 | 8049.541 | 8076.073 | 7846.052 |
| BIC | 8042.333 | 8067.041 | 8129.618 | 8156.151 | 7926.130 |
| RMSEA | 0.041 | 0.052 | 0.070 | 0.094 | 0.010 |
| CFI | 0.984 | 0.974 | 0.949 | 0.881 | 0.996 |
| TLI | 0.970 | 0.951 | 0.905 | 0.776 | 1.014 |
| SRMR | 0.041 | 0.044 | 0.050 | 0.056 | 0.032 |

Example 2: Results

===== Parameter Estimates and Standard Errors =====

| | Estimate.Average | Estimate.SD | Average.SE | Power..Not.equal.0. | Std.Est | Std.Est.SD |
|-------|------------------|-------------|------------|---------------------|---------|------------|
| LY1_1 | 0.699 | 0.063 | 0.060 | 1.000 | 0.700 | 0.055 |
| LY2_1 | 0.703 | 0.066 | 0.067 | 1.000 | 0.704 | 0.055 |
| LY3_1 | 0.701 | 0.068 | 0.067 | 1.000 | 0.702 | 0.056 |
| LY4_2 | 0.699 | 0.060 | 0.061 | 1.000 | 0.700 | 0.052 |
| LY5_2 | 0.701 | 0.069 | 0.067 | 1.000 | 0.704 | 0.057 |
| LY6_2 | 0.704 | 0.068 | 0.067 | 1.000 | 0.703 | 0.055 |
| PS2_1 | 0.098 | 0.068 | 0.067 | 0.317 | 0.098 | 0.068 |
| TE1_1 | 0.503 | 0.077 | 0.073 | 0.994 | 0.506 | 0.078 |
| TE2_2 | 0.499 | 0.078 | 0.076 | 1.000 | 0.502 | 0.077 |
| TE3_3 | 0.502 | 0.080 | 0.076 | 0.999 | 0.504 | 0.079 |
| TE4_4 | 0.505 | 0.075 | 0.073 | 0.995 | 0.507 | 0.074 |
| TE5_5 | 0.496 | 0.078 | 0.076 | 1.000 | 0.501 | 0.079 |
| TE6_6 | 0.501 | 0.076 | 0.077 | 0.999 | 0.502 | 0.077 |
| TY1 | 0.000 | 0.045 | 0.045 | 0.052 | 0.000 | 0.045 |
| TY2 | 0.001 | 0.055 | 0.053 | 0.061 | 0.001 | 0.055 |
| TY3 | 0.000 | 0.052 | 0.053 | 0.052 | 0.000 | 0.052 |
| TY4 | 0.000 | 0.044 | 0.045 | 0.051 | 0.000 | 0.044 |
| TY5 | 0.000 | 0.055 | 0.053 | 0.061 | 0.000 | 0.055 |
| TY6 | 0.001 | 0.052 | 0.053 | 0.049 | 0.001 | 0.052 |

Example 2: Results

| | Average.Param | Average.Bias | Coverage | Average.FMI1 | SD.FMI1 | Average.FMI2 | SD.FMI2 |
|-------|---------------|--------------|----------|--------------|---------|--------------|---------|
| LY1_1 | 0.70 | -0.001 | 0.939 | 0.325 | 0.160 | 0.356 | 0.177 |
| LY2_1 | 0.70 | 0.003 | 0.936 | 0.443 | 0.168 | 0.485 | 0.182 |
| LY3_1 | 0.70 | 0.001 | 0.928 | 0.446 | 0.175 | 0.487 | 0.190 |
| LY4_2 | 0.70 | -0.001 | 0.949 | 0.327 | 0.163 | 0.358 | 0.180 |
| LY5_2 | 0.70 | 0.001 | 0.927 | 0.447 | 0.174 | 0.489 | 0.188 |
| LY6_2 | 0.70 | 0.004 | 0.936 | 0.445 | 0.174 | 0.487 | 0.188 |
| PS2_1 | 0.10 | -0.002 | 0.944 | 0.218 | 0.117 | 0.237 | 0.131 |
| TE1_1 | 0.51 | -0.007 | 0.945 | 0.432 | 0.181 | 0.472 | 0.195 |
| TE2_2 | 0.51 | -0.011 | 0.941 | 0.496 | 0.175 | 0.541 | 0.186 |
| TE3_3 | 0.51 | -0.008 | 0.928 | 0.498 | 0.179 | 0.543 | 0.191 |
| TE4_4 | 0.51 | -0.005 | 0.947 | 0.434 | 0.185 | 0.474 | 0.200 |
| TE5_5 | 0.51 | -0.014 | 0.918 | 0.491 | 0.178 | 0.535 | 0.190 |
| TE6_6 | 0.51 | -0.009 | 0.943 | 0.499 | 0.179 | 0.544 | 0.191 |
| TY1 | 0.00 | 0.000 | 0.948 | 0.000 | 0.000 | 0.000 | 0.000 |
| TY2 | 0.00 | 0.001 | 0.939 | 0.265 | 0.135 | 0.290 | 0.151 |
| TY3 | 0.00 | 0.000 | 0.948 | 0.267 | 0.132 | 0.292 | 0.148 |
| TY4 | 0.00 | 0.000 | 0.949 | 0.000 | 0.000 | 0.000 | 0.000 |
| TY5 | 0.00 | 0.000 | 0.939 | 0.273 | 0.133 | 0.299 | 0.149 |
| TY6 | 0.00 | 0.001 | 0.951 | 0.271 | 0.131 | 0.296 | 0.147 |

Some Future Plans

- Multiple group models (coming soon!)
- Categorical indicators
- Multilevel SEM
- Non-linear constraints
- Additional analysis (e.g., OpenMx) and imputation packages (e.g, Mice)
- Latent interactions
- Syntax entry

Also...

- Another R package that may interest R users familiar with SEM
- semTools
 - Useful tools for conducting SEM in R
 - e.g., runMI, imputes missing data, runs each imputed data set, and combines results
 - An open source, community supported package
 - Have an idea for a function? Or a way to improve an existing function? Let us know!

Questions?

- Thanks to
 - Paul Johnson
 - Todd Little



simsem: simsem.org

example code available at: simsem.org

email: schoemann@ku.edu

Partial support provided by NSF grant 1053160 (Wei Wu & Todd D. Little, co-PIs)