

# Monte Carlo Investigations

James H. Steiger

Department of Psychology and Human Development  
Vanderbilt University

# Monte Carlo Investigations

- 1 Introduction
- 2 Monte Carlo Capabilities in Mplus
  - A Basic Monte Carlo Run in Mplus
- 3 Creating a Systematic Monte Carlo Study
  - Generating Input Files
  - Batch Running Input Files
  - Proportion of Successful Convergence
- 4 Externalizing a Monte Carlo Study
  - Saving Monte Carlo Data for External Analysis
  - Analyzing Monte Carlo Data from a Sequence of External Files
- 5 More Realistic Monte Carlo Studies
- 6 Simulating Multivariate Non-Normality
  - The Third-Order Polynomial Transform
  - The Vale-Maurelli Method
- 7 Simulating Non-Perfect Fit
  - The Cudeck-Browne Approach
- 8 Directly Specifying a Population Covariance Matrix

# Introduction

- In this module, we investigate some approaches to Monte Carlo investigation in structural equation modeling.
- There are several reasons for wanting to do Monte Carlo experiments.
- We can use them to examine performance of model estimates, estimate power, and, through the analysis of convergence and estimation failure, gauge the sample size necessary to reduce the probability of iteration failure to an acceptable level.

# Monte Carlo Capabilities in Mplus

- Mplus has some very general Monte Carlo capabilities.
- On the other hand, the capabilities for organizing and analyzing the information produced by a Monte Carlo run is rather limited.
- Fortunately, we have the very flexible and powerful (and free) capabilities of R to come to our rescue.

# Monte Carlo Capabilities in Mplus

- We'll begin by examining a simple run of 1000 replications in a single condition.
- Later, we'll discover how to expand on this to create an entire study.

# Monte Carlo Capabilities in Mplus

## A Basic Monte Carlo Run in Mplus

- Suppose we were interested in the performance of a confirmatory factor analysis model when the data are multivariate normal and the model fits perfectly in the population.
- This is a classic Monte Carlo analysis. A sample condition might involve a population structure with 9 indicator variables, 3 factors, 3 variables per factor, no crossover loadings, and equal loadings of 0.60.
- Here's how we might set this up in Mplus with a sample size of  $n = 100$ .

# Monte Carlo Capabilities in Mplus

## A Basic Monte Carlo Run in Mplus

```
TITLE: MONTE CARLO 9x3 CONSTANT LOADING = 0.60 N = 50
```

```
MONTECARLO:
```

```
NAMES ARE Y1-Y9;
```

```
NOOBSERVATIONS=50;
```

```
NREPS=1000;
```

```
SEED=12345;
```

```
MODEL POPULATION:
```

```
F1 BY Y1-Y3*0.60;
```

```
F2 BY Y4-Y6*0.60;
```

```
F3 BY Y7-Y9*0.60;
```

```
F1-F3@1;
```

```
Y1-Y9*.64;
```

```
F1 WITH F2-F3 *0.00;
```

```
F2 WITH F3*0.00;
```

```
MODEL:
```

```
F1 BY Y1-Y3*0.60;
```

```
F2 BY Y4-Y6*0.60;
```

```
F3 BY Y7-Y9*0.60;
```

```
F1-F3@1;
```

```
Y1-Y9*.64;
```

```
F1 WITH F2-F3 *0.00;
```

```
F2 WITH F3*0.00;
```

# Monte Carlo Capabilities in Mplus

## A Basic Monte Carlo Run in Mplus

- The first section establishes a model that is used to create the statistical population.
- Each parameter must be provided with a start value that is used to generate the population.
- NREPS is the number of Monte Carlo replications, 1000 in this case. Mplus is going to create 1000 samples of size  $n = 50$ , run its estimation routine using the model shown in the second part of the file, and save a summary of the output.
- The second section of our input file shows the model that is actually used to analyze the data. In this case, it is the same model that created the data.
- We include, as starting values, the actual parameter values that were used to create the data. These starting values should work very well, especially with larger sample sizes.
- Let's run our input file with Mplus, and examine the output.



# Monte Carlo Capabilities in Mplus

## A Basic Monte Carlo Run in Mplus

- Note one very important fact at the beginning of the output file.

Number of replications

Requested

1000

Completed

846

Value of seed

12345

- This output tells us that, although we requested 1000 replications, only 846 were completed.
- Iteration evidently failed on 154 out of 1000 replications.
- Examining the output helps explain why.

# Monte Carlo Capabilities in Mplus

## A Basic Monte Carlo Run in Mplus

We'll show output for just the first few replications that failed:

Error messages for each replication (if any)

REPLICATION 1:

```
WARNING: THE RESIDUAL COVARIANCE MATRIX (THETA) IS NOT POSITIVE DEFINITE.
THIS COULD INDICATE A NEGATIVE VARIANCE/RESIDUAL VARIANCE FOR AN OBSERVED
VARIABLE, A CORRELATION GREATER OR EQUAL TO ONE BETWEEN TWO OBSERVED
VARIABLES, OR A LINEAR DEPENDENCY AMONG MORE THAN TWO OBSERVED VARIABLES.
CHECK THE RESULTS SECTION FOR MORE INFORMATION.
PROBLEM INVOLVING VARIABLE Y4.
```

REPLICATION 2:

```
NO CONVERGENCE. NUMBER OF ITERATIONS EXCEEDED.
```

REPLICATION 3:

```
NO CONVERGENCE. NUMBER OF ITERATIONS EXCEEDED.
```

REPLICATION 6:

```
WARNING: THE RESIDUAL COVARIANCE MATRIX (THETA) IS NOT POSITIVE DEFINITE.
THIS COULD INDICATE A NEGATIVE VARIANCE/RESIDUAL VARIANCE FOR AN OBSERVED
VARIABLE, A CORRELATION GREATER OR EQUAL TO ONE BETWEEN TWO OBSERVED
VARIABLES, OR A LINEAR DEPENDENCY AMONG MORE THAN TWO OBSERVED VARIABLES.
CHECK THE RESULTS SECTION FOR MORE INFORMATION.
PROBLEM INVOLVING VARIABLE Y3.
```

# Monte Carlo Capabilities in Mplus

## A Basic Monte Carlo Run in Mplus

- As the output shows, some replications ended in a convergence failure, while others ended with negative residual variances resulting in a residual covariance matrix that was not positive definite.
- Given the small sample size, the question naturally arises — is this kind of result typical of a confirmatory factor analysis?
- If so, what aspects of the analysis will be related to the severity of the problem?

# Creating a Systematic Monte Carlo Study

- We discovered from our original Monte Carlo run that a substantial number of samples failed to iterate to a successful solution.
- This is not a result that we would want to have happen in a real world study.
- The question naturally arises, “What factors are related to the probability of a failed iteration””

## Creating a Systematic Monte Carlo Study

- We could set up a set of Monte Carlo run files, like the one shown in the previous section, and vary a group of parameters systematically across runs.
- A number of potential factors come to mind for such a study. They might include the sample size, the size of the factor loadings (and, concomitantly, the unique variances).
- Other factors include the number of variables, the number of factors, whether or not there are “crossover” loadings, etc.
- Should we choose to examine several of these potential influences, we might end up with hundreds of Monte Carlo runs, or even a thousand or more.

# Creating a Systematic Monte Carlo Study

- R has available a facility for automatically generating model input files.
- This capability, in the R package `MplusAutomation`, can be very useful for developing a Monte Carlo study.
- In this section, we describe how to convert a single Monte Carlo run file into a template that `MplusAutomation` can automatically expand into a large number of model input files, useful in a large Monte Carlo study.

# Creating a Systematic Monte Carlo Study

## Generating Input Files

- The next slide has a “template” file that is used with the `MplusAutomation` package.
- The initial section of the template file establishes some *iterators*, integer variables that will be varied systematically through a range of values.
- In this case, we see two iterators that will be used to vary sample size through 6 values, and size of loading through 4 values.
- Immediately after defining the iterators, we link some numerical values to them. For example, we see that sample sizes will be 50,100,300,400,800, and 1600.
- Following the template section is a typical Monte Carlo input file, with certain values replaced by the variables that will be iterated through.

# Creating a Systematic Monte Carlo Study

## Generating Input Files

```
[[init]]
iterators = ns loading;
ns = 1:6;
loading = 1:4;
n#ns = 50 100 200 400 800 1600;
lambda#loading = 0.20 0.40 0.60 0.80;
vv#loading = .96 .84 .64 .36;
filename = "9x3_[n#ns]_[lambda#loading].inp";
outputDirectory = "D:/MC";
[/init]]
TITLE: MONTE CARLO 9x3 CONSTANT LOADING = [[lambda#loading]] N = [[n#ns]]
MONTECARLO:
NAMES ARE Y1-Y9;
NOBSERVATIONS=[[n#ns]];
NREPS=1000;
SEED=12345;
MODEL POPULATION:
F1 BY Y1-Y3*[[lambda#loading]];
F2 BY Y4-Y6*[[lambda#loading]];
F3 BY Y7-Y9*[[lambda#loading]];
F1-F3@1;
Y1-Y9*[[vv#loading]];
F1 WITH F2-F3 *0.00;
F2 WITH F3*0.00;

MODEL:
F1 BY Y1-Y3*[[lambda#loading]];
F2 BY Y4-Y6*[[lambda#loading]];
F3 BY Y7-Y9*[[lambda#loading]];
F1-F3@1;
Y1-Y9*[[vv#loading]];
F1 WITH F2-F3 *0.00;
F2 WITH F3*0.00;
```



# Creating a Systematic Monte Carlo Study

## Generating Input Files

- To generate many input Monte Carlo files, we simply run the command `createModels` on our template file.
- In a blink of an eye, all the input files will be created in the target directory.
- The next slide shows the contents of the file `9x3_1600_0.80.inp`.
- This file contains commands to run 1000 replications with a sample size of  $n = 1600$ , and a loading of 0.80 for each factor.

# Creating a Systematic Monte Carlo Study

## Generating Input Files

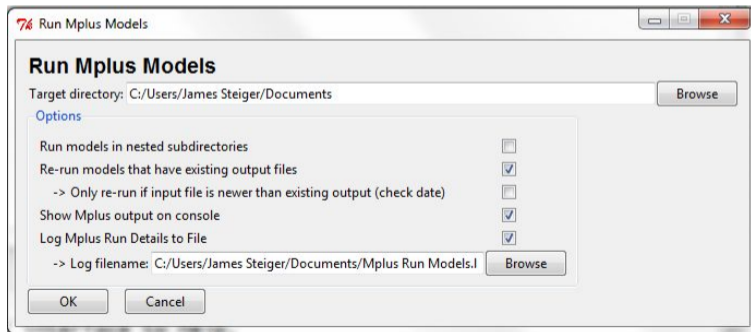
```
TITLE: MONTE CARLO 9x3 CONSTANT LOADING = 0.80 N = 1600
MONTECARLO:
NAMES ARE Y1-Y9;
NOBSERVATIONS=1600;
NREPS=1000;
SEED=12345;
MODEL POPULATION:
F1 BY Y1-Y3*0.80;
F2 BY Y4-Y6*0.80;
F3 BY Y7-Y9*0.80;
F1-F3@1;
Y1-Y9*.36;
F1 WITH F2-F3 *0.00;
F2 WITH F3*0.00;

MODEL:
F1 BY Y1-Y3*0.80;
F2 BY Y4-Y6*0.80;
F3 BY Y7-Y9*0.80;
F1-F3@1;
Y1-Y9*.36;
F1 WITH F2-F3 *0.00;
F2 WITH F3*0.00;
```

# Creating a Systematic Monte Carlo Study

## Batch Running Input Files

- We've generated 24 input files with our template.
- Now we would like to run them. An easy way to do that is to use the `runModels_Interactive()` command.
- This command opens a graphical interface and runs all your model input files for you.



# Creating a Systematic Monte Carlo Study

## Batch Running Input Files

- It took only about 10 minutes for all 24 files to run on one of my faster computers.
- The `MplusAutomation` package has some capabilities for going through output and extracting parameter values. This capability appears to be extremely rudimentary, but still useful in some contexts. It is primarily geared toward extracting actual model estimates and fit statistics from a standard run, not from a Monte Carlo run.
- In this case, we are primarily interested in extracting estimates for the probability of a successful iteration from each of our 24 output files.
- It took me less than 10 minutes to create the table shown on the next slide.

# Creating a Systematic Monte Carlo Study

## Proportion of Successful Convergence

$n$	0.20	0.40	0.60	0.80
50	195	307	846	1000
100	179	469	985	1000
200	201	750	1000	1000
400	265	961	1000	1000
800	435	1000	1000	1000
1600	643	1000	1000	1000

# Creating a Systematic Monte Carlo Study

## Proportion of Successful Convergence

- How would you summarize the results of this little study, and the “prescriptions for practice” that we might take from it?
- What are some areas in which this study fails? Serious flaws? Lack of generality?

# Externalizing a Monte Carlo Study

- In the previous section, we examined how the basic Monte Carlo capabilities of Mplus can be used to generate a study of the performance of a modeling and estimation procedure across a variety of conditions.
- In this section, prompted by some comments and example files provided by Professor Cho, we examine how one might extend these Monte Carlo capabilities in several ways:
  - 1 Analysis of data generated by Mplus using alternative estimation methods not available in Mplus
  - 2 Analysis within Mplus of data generated by an alternative Monte Carlo data generation program

# Externalizing a Monte Carlo Study

## Saving Monte Carlo Data for External Analysis

- The input file on the next slide demonstrates a modification of our previous Monte Carlo file that saves the Monte Carlo data for reanalysis, either by Mplus or by an external program.
- Note that 2 lines have been added and are highlighted in red.
- These lines instruct Mplus to save the files for all replications, and to name the files according to a specific format.



# Externalizing a Monte Carlo Study

## Saving Monte Carlo Data for External Analysis

```
TITLE: MONTE CARLO 9x3 CONSTANT LOADING = 0.60 N = 50
MONTECARLO:
NAMES ARE Y1-Y9;
NOOBSERVATIONS=50;
NREPS=1000;
SEED=12345;
REPSAVE = ALL;           !Save data from ALL replications
SAVE = C:/data/sim_*.DAT; !Save data in files sim1.dat,..,sim1000.dat
MODEL POPULATION:
F1 BY Y1-Y3*0.60;
F2 BY Y4-Y6*0.60;
F3 BY Y7-Y9*0.60;
F1-F3@1;
Y1-Y9*.64;
F1 WITH F2-F3 *0.00;
F2 WITH F3*0.00;
MODEL:
F1 BY Y1-Y3*0.60;
F2 BY Y4-Y6*0.60;
F3 BY Y7-Y9*0.60;
F1-F3@1;
Y1-Y9*.64;
F1 WITH F2-F3 *0.00;
F2 WITH F3*0.00;
```

# Externalizing a Monte Carlo Study

## Analyzing Monte Carlo Data from a Sequence of External Files

- The illustration above shows how to get Mplus to generate data and save each replication in a separate file, possibly for external analysis.
- Mplus can also be instructed to analyze data from a sequence of external files, analyze the data and collect the analysis results within a Monte Carlo analysis summary.
- Here on the next slide is an example from Professor Cho showing how Mplus can analyze a sequence of files.
- Note that a list of file names must be provided in a summary file.
- In this case, Mplus created the external files and the filename list.
- However, if Mplus cannot create data that meets your specifications, you could write an external program in R, save the data for each Monte Carlo replication to individual data files with the same names as those produced by Mplus, and use the Mplus-generated list of file names.
- Of course, using your knowledge of R, you could also program R yourself to generate the file name list!

# Externalizing a Monte Carlo Study

## Analyzing Monte Carlo Data from a Sequence of External Files

TITLE: CFA estimation

DATA: FILE IS sim\_list.dat; !A file containing a list  
!of Data Files to be analyzed.  
TYPE = MONTECARLO; !These data are to be analyzed  
!and compiled as a Monte Carlo Study.

VARIABLE: NAMES ARE Y1-Y9;

MODEL:

F1 BY Y1-Y3\*0.60;

F2 BY Y4-Y6\*0.60;

F3 BY Y7-Y9\*0.60;

F1-F3@1;

Y1-Y9\*.64;

F1 WITH F2-F3 \*0.00;

F2 WITH F3\*0.00;

## More Realistic Monte Carlo Studies

- In assessing the performance of a model-fitting procedure, it is an important initial step to discover whether the model works well when “fundamentals are sound” i.e., the model is correct, the distributional assumptions are correct, and there are no missing data.
- In the previous section, we discovered that even when almost everything is perfect, a simple confirmatory factor analysis model with perfect simple structure, 9 indicator variables, and 3 factors, will not generate correct results very often unless population loadings are large and/or sample size is large.

# More Realistic Monte Carlo Studies

- What happens when things aren't perfect?
- Comparatively few Monte Carlo studies have analyzed how estimation procedures behave when the model doesn't fit perfectly in the population, distributions aren't multivariate normal, or both.
- In one sense, this is surprising: Dozens of statistical experts, from Box to Tukey to Kendall, have stated that model fit is not, in general, perfect, and that it is unrealistic to assume so.
- What's going on?

# More Realistic Monte Carlo Studies

- In one sense, the situation is a natural consequence of the sociology of science and publication priorities.
- The situation in which fundamentals are perfect is relatively easy to specify, simulate, and test.
- The situation in which fundamentals are not perfect opens up a proverbial can of worms.
- Many “Monte Carlo Studies” are actually very inadequate, simply comprising a couple of non-representative situations (or even a simple analysis of one data set) involving perfect fundamentals presented haphazardly during the presentation/promotion of a new and interesting analysis.
- In that context, the author is trying to escape quickly and easily with a publication — he/she is not particularly motivated and/or can't find the time to find conditions under which the new analysis will not work.

# More Realistic Monte Carlo Studies

- Even if the author is an iconoclast and a skeptic with time on his/her hands, there are many issues to be resolved.
- One is complexity. Another is space. Good Monte Carlo investigations that step outside the bounds of “perfect fundamentals” require key judgments about
  - ① Which departures from perfection make the most sense to simulate, and
  - ② How best to present the complex information.
- In the next sections, we discuss methods for simulating multivariate non-normality, and simulating non-perfect models in a reasonable way.

# Simulating Multivariate Non-Normality

- Suppose one wished to simulate continuous multivariate distributions that depart from normality.
- Two primary characteristics on which a continuous distribution may depart from normality are **skewness** and **kurtosis**.
- There are several available methods for simulating multivariate non-normality.
- By far the most popular in the psychometric literature has been the method of Vale and Maurelli(1983, *Psychometrika*), which built on earlier work by Fleishman(1978, *Psychometrika*).



# Simulating Multivariate Non-Normality

- We begin by describing the method of Fleishman(1978) for transforming a standard normal random variable into one with a mean of 0, and standard deviation of 1, and a desired skewness and kurtosis.
- We then show how Vale and Maurelli(1983) adapted this method to generate multivariate non-normal variables with desired correlation matrix and desired marginal skewnesses and kurtoses.

# Simulating Multivariate Non-Normality

## The Third-Order Polynomial Transform

- We begin with some background on moments and cumulants.
- The **second-order central moments around the mean** are

$$\sigma_{ij} = E[(X_i - \mu_i)(X_j - \mu_j)] \quad (1)$$

Correspondingly, the **third- and fourth-order moments** are

$$\sigma_{ijk} = E[(X_i - \mu_i)(X_j - \mu_j)(X_k - \mu_k)] \quad (2)$$

and

$$\sigma_{ijkh} = E[(X_i - \mu_i)(X_j - \mu_j)(X_k - \mu_k)(X_h - \mu_h)] \quad (3)$$

# Simulating Multivariate Non-Normality

## The Third-Order Polynomial Transform

- The second through fourth order **standardized moments** are

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}} \quad (4)$$

$$\rho_{ijk} = \frac{\sigma_{ijk}}{\sqrt{\sigma_{ii}\sigma_{jj}\sigma_{kk}}} \quad (5)$$

$$\rho_{ijkh} = \frac{\sigma_{ijkh}}{\sqrt{\sigma_{ii}\sigma_{jj}\sigma_{kk}\sigma_{hh}}} \quad (6)$$

- The **normalized kurtosis** of variable  $X_j$  is defined as

$$\gamma_{2j} = \rho_{jjjj} - 3 \quad (7)$$

# Simulating Multivariate Non-Normality

## The Third-Order Polynomial Transform

- Note that the moments are invariant under permutation of subscripts, i.e.,  $\rho_{112}$  is the same as  $\rho_{121}$  and  $\rho_{1122}$  is the same as  $\rho_{1212}$  etc.
- Consequently, there are four distinct third-order and five distinct fourth-order moments for a bivariate distribution:  $\rho_{111}$ ,  $\rho_{112}$ ,  $\rho_{122}$ , and  $\rho_{222}$  and  $\rho_{1111}$ ,  $\rho_{1112}$ ,  $\rho_{1122}$ ,  $\rho_{1222}$ , and  $\rho_{2222}$ .
- The **central moments**  $\mu_r$  of a random variable with expected value  $\mu$  are defined as

$$\mu_r = E(X - \mu)^r \quad (8)$$

# Simulating Multivariate Non-Normality

## The Third-Order Polynomial Transform

- Although the moments of a random variable's distribution are very useful for describing it, other functions of the distribution may be more useful in theoretical derivatons.
- In particular, the **cumulants** and their relationship to the moments are of great use.
- Headrick(2002) gives the relationship between cumulants and central moments, and also provides formulas for **normalized cumulants**. A normalized cumulant  $\kappa'_r$  is defined as

$$\kappa'_r = \frac{\kappa_r}{\sqrt{\kappa_2^r}} = \frac{\kappa_r}{\sigma^r} \quad (9)$$

# Simulating Multivariate Non-Normality

## The Third-Order Polynomial Transform

- The first 6 normalized cumulants may be expressed in terms of central moments as follows (note that  $\gamma_1$  and  $\gamma_2$  are commonly employed measures of skewness and kurtosis):

$$\kappa'_1 = 0 \quad (10)$$

$$\kappa'_2 = 1 \quad (11)$$

$$\kappa'_3 = \gamma_1 = \frac{\mu_3}{\sigma^3} \quad (12)$$

$$\kappa'_4 = \gamma_2 = \frac{\mu_4}{\sigma^4} - 3 \quad (13)$$

$$\kappa'_5 = \gamma_3 = \frac{\mu_5}{\sigma^5} - 10\gamma_1 \quad (14)$$

$$\kappa'_6 = \gamma_4 = \frac{\mu_6}{\sigma^6} - 15\gamma_2 - 10\gamma_1^2 - 15 \quad (15)$$

With the sample mean  $\bar{X}$  defined as usual as

$$\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i \quad (16)$$

# Simulating Multivariate Non-Normality

## The Third-Order Polynomial Transform

- Common estimates of the standardized cumulants are

$$\hat{\gamma}_1 = \frac{\sum_{i=1}^N (X_i - \bar{X})^3 / N}{(\sum_{i=1}^N (X_i - \bar{X})^2 / N)^{3/2}} \quad (17)$$

$$\hat{\gamma}_2 = \frac{\sum_{i=1}^N (X_i - \bar{X})^4 / N}{(\sum_{i=1}^N (X_i - \bar{X})^2 / N)^2} - 3 \quad (18)$$

$$\hat{\gamma}_3 = \frac{\sum_{i=1}^N (X_i - \bar{X})^5 / N}{(\sum_{i=1}^N (X_i - \bar{X})^2 / N)^{5/2}} - 10\hat{\gamma}_1 \quad (19)$$

$$\hat{\gamma}_4 = \frac{\sum_{i=1}^N (X_i - \bar{X})^6 / N}{(\sum_{i=1}^N (X_i - \bar{X})^2 / N)^3} - 15\hat{\gamma}_2 - 10\hat{\gamma}_1^2 - 15 \quad (20)$$

# Simulating Multivariate Non-Normality

## The Third-Order Polynomial Transform

- We want to convert a standard normal variable  $Z$  to a nonnormal variable  $Y$  by a non-linear functional transformation  $Y = f(Z)$ .
- Fleishman's (1978) 3<sup>rd</sup> order polynomial transform (PT3), also called the 3<sup>rd</sup> order power method, uses a third-degree polynomial such that

$$Y = a_0 + a_1Z + a_2Z^2 + a_3Z^3 \quad (21)$$

To ensure that the random variable  $Y$  has desired values  $\gamma_1$  and  $\gamma_2$  for skewness and kurtosis, while mean and variance are standardized to 0 and 1, one must solve a set of nonlinear equations for the coefficients  $a_1$ ,  $a_2$ , and  $a_3$  from Equation 21 and later set  $a_0 = -a_2$ :

$$\begin{aligned} \sigma^2 &= a_1^2 + 6a_1a_3 + 2a_2^2 + 15a_3^2 = 1 \\ \gamma_1 &= 2a_2(3a_1^2 + 4a_2^2 + 36a_1a_3 + 135a_3^2) \\ \gamma_2 &= 3a_1^4 + 60a_1^2a_2^2 + 60a_2^4 + 60a_1^3a_3 + 936a_1a_2^2a_3 \\ &\quad + 630a_1^2a_3^2 + 4500a_2^2a_3^2 + 3780a_1a_3^3 + 10395a_3^4 - 3 \end{aligned} \quad (22)$$



# Simulating Multivariate Non-Normality

## The Third-Order Polynomial Transform

- In general, these equations must be solved numerically.
- As an example, a nonnormal  $Y$  with a mean of 0, a variance of 1, a skewness of 0, and a kurtosis of 25, can be constructed as  $Y = 0 + 0.2553Z + 0Z^2 + 0.2038Z^3$ , i.e., with a coefficient vector  $\mathbf{a} = [0.0000, 0.2553, 0.0000, 0.2038]'$ .
- The Fleishman (1978) method was in use for more than 30 years before Kraatz (2011) noted that the coefficients are not uniquely defined: there are several solutions, some of which yield transformations that are monotonic, others non-monotonic.
- The different distributions produced by the different coefficients can have radically different shapes despite having identical means, standard deviations, skewnesses, and kurtoses.

# Simulating Multivariate Non-Normality

## The Third-Order Polynomial Transform

- Not all skewness-kurtosis combinations are possible.
- It has been proven that for any univariate distribution, the range of possible skewness-kurtosis combinations is limited by the equation

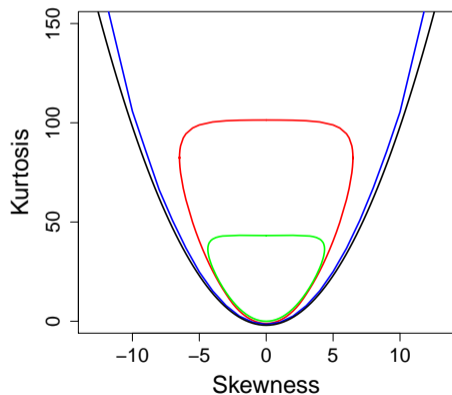
$$\gamma_2 \geq \gamma_1^2 - 2 \quad (23)$$

- The resulting range of valid skewness-kurtosis combinations is visualized as the area above the black line in the skewness-kurtosis plane in Figure 1.

# Simulating Multivariate Non-Normality

## The Third-Order Polynomial Transform

Figure 1 : Skewness-Kurtosis Limitations for PT3 (red), monotonic PT3 (green)  $g$ -and- $h$  distribution (blue), and all Possible Distributions (black)



# Simulating Multivariate Non-Normality

## The Third-Order Polynomial Transform

- The range of available kurtosis values for a given skewness value is further restricted when PT3 is employed. For PT3, kurtosis must satisfy constraints that can be approximated by the following:

$$\gamma_2 > 1.588\gamma_1^2 - 1.139 \quad (24)$$

- Further, kurtosis also has an upper limit: When  $\gamma_1 = 0$ , kurtosis cannot exceed approximately 101.38, and this value of allowable kurtosis will be even lower for  $\gamma_1 > 0$ .
- The range of available skewness-kurtosis combinations for PT3 is approximately bounded by the red continuous line in Figure 1.
- Outside of that range, the equations in 22 do not have a real-valued solution.
- Note that these boundaries were determined numerically by attempting to find solutions to the set of equations in 22 for various values of  $\gamma_1$  and  $\gamma_2$ .
- The set of possible skewness-kurtosis combinations is even smaller for **monotonic** PT3 transformations and enclosed by the green line in Figure 1.

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- We saw in our module on symmetric square roots that it is easy to transform independent normal random variables to have a multivariate normal distribution with any desired covariance matrix.
- One simply linearly combines the independent normal random numbers with any Gram-Factor of the desired  $\Sigma$ .

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- Let  $\mathbf{z}$  be a  $p \times 1$  random vector having a multivariate normal distribution with mean vector  $\mathbf{0}$  and with covariance matrix  $\mathbf{I}$ .
- Let  $\mathbf{L}$  be a unique lower-triangular (Cholesky) factor of  $\mathbf{\Sigma}$ , a  $p \times p$  positive definite covariance matrix such that  $\mathbf{LL}' = \mathbf{\Sigma}$ .
- Then

$$\mathbf{z}^* = \mathbf{Lz} \quad (25)$$

will have a multivariate normal distribution with mean  $\mathbf{0}$  and covariance matrix  $\mathbf{\Sigma}$ .

- Consequently, if the columns of an  $n \times p$  sample data matrix  $\mathbf{Z}$  represent  $n$  observations from a  $MVN(\mathbf{0}, \mathbf{I})$  distribution, then

$$\mathbf{Z}^* = \mathbf{ZL}' \quad (26)$$

will represent  $n$  observations from a  $MVN(\mathbf{0}, \mathbf{\Sigma})$  distribution, where  $\mathbf{\Sigma} = \mathbf{LL}'$ .

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- This very straightforward situation with multivariate normal data is much more complicated when non-normal continuous variates are simulated.
- For example, suppose we create a set of independent nonnormal variables with the desired (marginal) skewnesses and kurtoses using the PT3 method, and *then* use the above approach to correlate them.
- Unfortunately, during the matrix multiplication process, all (except for the first, assuming a Cholesky factor is used) nonnormal variables become linear combinations of the others, so their skewness and kurtosis will be altered by the “central limit effect”, and will no longer have the desired value.
- Conversely, suppose that we first generate multivariate normal random variables with the desired correlations as discussed above, and then apply non-normalizing transforms to the individual variables to produce desired skewnesses and kurtoses. Unfortunately, the correlations between the variables will be altered by the nonlinear transformations applied to them.

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- Fortunately, however, this influence of nonlinear power transformations on the correlations can be calculated and taken into account.
- The final correlation  $\rho_Y$  between two nonnormal variables can be expressed as a function of the **intermediate** correlation  $\rho_Z$  between the two normal variables and their non-normalizing transforms (Li and Hammond, 1975):

$$\rho_Y = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(Z_1)g(Z_2)f_{12}dZ_1dZ_2 \quad (27)$$

where  $f(Z_1)$  is the non-normalizing transform for the first variable,  $g(Z_2)$  is the non-normalizing transform for the second variable, and

$$f_{12} = \frac{1}{2\pi\sqrt{1-\rho_Z^2}} \exp\left(-\frac{Z_1^2 - 2\rho_Z Z_1 Z_2 + Z_2^2}{2(1-\rho_Z^2)}\right) \quad (28)$$

is the standard normal bivariate density.



# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- A popular remedy that takes advantage of Equations 25–26 while successfully generating marginal variates with a desired skewness and kurtosis is to
  - 1 Find an *intermediate* correlation matrix for the random variables  $\mathbf{z}^*$  in Equation 25.
  - 2 Subject the correlated standard normal random scores  $\mathbf{Z}^*$  in Equation 26 to the non-normalizing transformation.
- The intermediate correlation matrix is chosen so that the final correlation matrix — the correlation matrix between the nonnormal random variables, after applying the non-normalizing transform — is as desired.
- This transform-and-calculate (TC) principle, used to extend PT3 to the multivariate case, is demonstrated in detail for the bivariate PT3 in the following slides.

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- The first step of the TC principle applied to PT3 is to find the Fleishman coefficients that are needed to create nonnormal random variables with desired skewnesses and kurtoses.
- Next, one needs to find the required intermediate correlation matrix. As a special case of Equation 27, the correlation  $\rho_Y$  between two nonnormal Fleishman variables  $Y_1$  and  $Y_2$  can be expressed as (Equation 11 in Vale and Maurelli, 1983):

$$\rho_Y = \rho_Z(a_{11}a_{12} + 3a_{11}a_{32} + 3a_{31}a_{12} + 9a_{31}a_{32}) + 2a_{21}a_{22}\rho_Z^2 + 6a_{31}a_{32}\rho_Z^3 \quad (29)$$

where  $\rho_Z$  is the intermediate correlation between two standard normal random variables,  $a_{11}$ ,  $a_{21}$ , and  $a_{31}$  are the transformation coefficients for  $Y_1$ , and  $a_{12}$ ,  $a_{22}$ , and  $a_{32}$  are the transformation coefficients for  $Y_2$ .

- $\rho_Z$  can therefore be found by treating  $\rho_Y$  and the  $a_{ij}$  as known quantities, and numerically solving Equation 29 for the unknown  $\rho_Z$ .

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- Once  $\rho_Z$  is found,  $\Sigma^*$  is created as

$$\Sigma^* = \begin{bmatrix} 1 & \rho_Z \\ \rho_Z & 1 \end{bmatrix} \quad (30)$$

- In the next step, two standard normal random variables are correlated using the intermediate correlation matrix from Equation 30 in the process described in Equations 25–26.
- Finally, 3<sup>rd</sup> order polynomial transformations as in Equation 21 are applied individually to the now correlated standard normal variables, using previously calculated transformation coefficients  $a_{01}$ ,  $a_{11}$ ,  $a_{21}$ ,  $a_{31}$ ,  $a_{02}$ ,  $a_{12}$ ,  $a_{22}$ , and  $a_{32}$ .
- The two variables now have the desired marginal skewness and kurtosis, and also have the desired final correlation  $\rho_Y$ .
- Generalizing to a set of three or more correlated nonnormal variables is straightforward.

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- Here is an example.
- Assume we want to create two nonnormal random variables  $Y_1$  and  $Y_2$  with skewnesses  $\gamma_{11} = \gamma_{12} = 0$ , kurtoses  $\gamma_{21} = \gamma_{22} = 25$ , and final correlation  $\rho_Y = 0.30$ . (This has been a popular choice in the psychometric literature).
- The skewness-kurtosis combination  $\gamma_1 = 0$ ,  $\gamma_2 = 25$  can be produced as  $Y = 0 + 0.2553Z + 0Z^2 + 0.2038Z^3$ , i.e., with a coefficient vector  $\mathbf{a} = [0.0000, 0.2553, 0.0000, 0.2038]'$ .

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- Inserting these coefficients (using the same set for both  $Y_1$  and  $Y_2$ ) into Equation 29, the relationship between final and intermediate correlation is now:

$$\rho_Y = 0.750904\rho_Z + 0.249096\rho_Z^3 \quad (31)$$

- Setting  $\rho_Y = 0.30$ , we can numerically solve for  $\rho_Z$ , obtaining a result of  $\rho_Z = 0.3812$ .
- Next we postmultiply two independent standard normal random variables by the Cholesky decomposition of a correlation matrix with off-diagonal element  $\rho_Z = 0.3812$ .
- Finally, we apply the non-normalizing transforms, changing the normal variables to nonnormal counterparts, and also changing the correlation from 0.3812 to the desired final value of 0.30.

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- The Vale-Maurelli method appears to be straightforward if slightly intricate.
- However, a number of issues emerge in its application, several of which were highlighted by Kraatz (2011):
  - 1 Although a specified skewness-kurtosis combination may not be possible, numerical software might generate a transform formula without an error indication, leading to publication of skewness-kurtosis “conditions” that were not actually achieved.
  - 2 For a given skewness-kurtosis combination for two variables, a given correlation may not be possible, even though the skewness-kurtosis combination is.
  - 3 The range of possible correlations will usually be different for different sets of coefficients yielding identical skewness and kurtosis.
  - 4 In some situations, two different bivariate normal distributions will yield identical correlations between transformed variables.
  - 5 Some simulated distributions have extremely odd shapes, raising questions about their representativeness.

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- As an example of these phenomena, consider the case in which we simulate a bivariate distribution with one normal and one nonnormal marginal.
- Assume  $\gamma_1 = 1$  and  $\gamma_2 = 15$  for the nonnormal distribution.
- The choice of coefficients for the normal distribution is obvious ( $\mathbf{a}_N$ ) and for the nonnormal distribution, we choose  $\mathbf{a}_{31}$  from Equation 32 below:

$$\begin{aligned}
 \mathbf{a}_N &= [ 0 & 1 & 0 & 0 ]' \\
 \mathbf{a}_{31} &= [ -0.1701 & 1.5347 & 0.1701 & -0.3068 ]' \\
 \mathbf{a}_{32} &= [ -0.0773 & 0.4455 & 0.0773 & 0.1570 ]'
 \end{aligned}
 \tag{32}$$

- Substituting the coefficients into Equation 29 yields

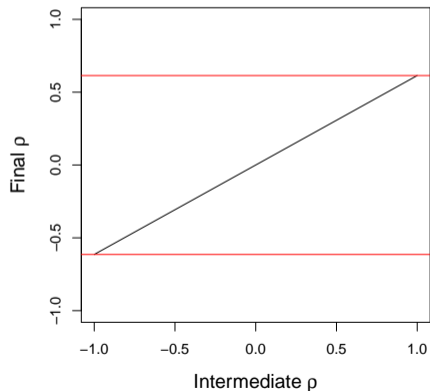
$$\begin{aligned}
 \rho_Y &= \rho_Z(a_{11}a_{12} + 3a_{11}a_{32} + 3a_{31}a_{12} + 9a_{31}a_{32}) + 2a_{21}a_{22}\rho_Z^2 + 6a_{31}a_{32}\rho_Z^3 \\
 &= \rho_Z(a_{12} + 3a_{32}) \\
 &= \rho_Z(1.5347 - 0.9205) \\
 &= 0.6142\rho_Z
 \end{aligned}
 \tag{33}$$

- Figure 2 on the next slide depicts this relationship between final and intermediate correlation.

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

Figure 2 : Relationship Between Intermediate and Final Correlation for a Bivariate PT3 Distribution with  $\gamma_{11} = 0$ ,  $\gamma_{12} = 1$ ,  $\gamma_{21} = 0$  and  $\gamma_{22} = 15$





# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

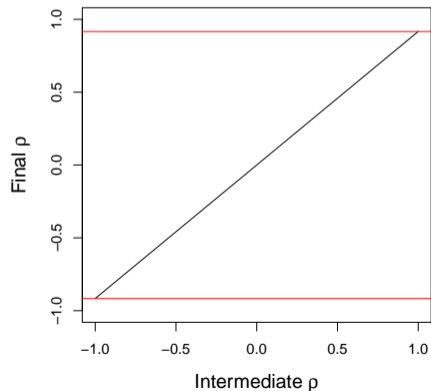
- Inserting the entire possible range of values for  $\rho_Z$  into Equation 33 only produces minimum and maximum values for the final correlation  $\rho_Y$  between  $-0.6142$  and  $0.6142$ .
- For this choice of marginal skewnesses and kurtoses and these sets of coefficients, it is *impossible* to create a final correlation of, say,  $0.80$  between  $Y_1$  and  $Y_2$ .
- Everything else being equal, had we chosen the same coefficients for the normal distribution but  $a_{32}$  for the nonnormal distribution, the relationship between  $\rho_Y$  and  $\rho_Z$  would have been as in Equation 34 and Figure 3:

$$\begin{aligned}\rho_Y &= \rho_Z(a_{12} + 3a_{32}) \\ &= \rho_Z(0.4455 + 0.4710) \\ &= 0.9166\rho_Z\end{aligned}\tag{34}$$

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

Figure 3 : Relationship Between Intermediate and Final Correlation for a Bivariate PT3 Distribution with  $\gamma_{11} = 0$ ,  $\gamma_{12} = 1$ ,  $\gamma_{21} = 0$  and  $\gamma_{22} = 15$



# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- The lesson from the preceding example is that transformation coefficients that produce the same skewness and kurtosis are not equivalent in the range of correlations they can be used to simulate.
- One set of coefficients may allow you to accomplish your objective, while another may not.

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- Multivariate distributions and even bivariate distributions are considerably more complex than univariate ones.
- For example, a bivariate distribution has a total of two 1<sup>st</sup> order moments (the two means), three 2<sup>nd</sup> order moments (two variances and one covariance) and four 3<sup>rd</sup> order moments:  $\rho_{111} = \gamma_{11}$ , skewness of variable 1,  $\rho_{112}$ ,  $\rho_{122}$ , and  $\rho_{222} = \gamma_{12}$ , skewness of variable 2. It also has five 4<sup>th</sup> order moments, six 5<sup>th</sup> order moments, and so forth.
- Hence, any bivariate distribution has  $\sum_{i=1}^4 (i + 1) = (2 + 3 + 4 + 5) = 14$  moments of up to 4<sup>th</sup> order. PT3, PT5, and the *g*-and-*h* distribution provide control over nine of these moments, while leaving the other five moments ( $\rho_{112}$ ,  $\rho_{122}$ ,  $\rho_{1112}$ ,  $\rho_{1122}$ , and  $\rho_{1222}$ ) uncontrolled.
- Of course, any other moments of yet higher order remain altogether uncontrolled as well, except by PT5, which controls univariate 5<sup>th</sup> and 6<sup>th</sup> order moments.
- For any bivariate distribution, there are

$$\frac{M^2 + 3M}{2} \quad (35)$$

moments up to  $M^{\text{th}}$  order.

- For example, if we are interested in the first  $M = 4$  moments, we have  $\sum_{i=1}^4 (i + 1) = (2 + 3 + 4 + 5) = 14$  moments in total. PT3, PT5, and the *g*-and-*h* distribution provide control over nine of these moments: Univariate means, variances, skewnesses, kurtoses, and the bivariate covariance, while leaving the other five moments ( $\rho_{112}$ ,  $\rho_{122}$ ,  $\rho_{1112}$ ,  $\rho_{1122}$ , and  $\rho_{1222}$ ) uncontrolled.

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

- What does this imply in practice? Even with equal skewness and kurtosis, different non-normalizing transforms may lead to univariate distributions with noticeably different shapes.
- We will find that this effect can be dramatically exponentiated for distributions with more than one dimension. We create a bivariate PT3 nIM distribution with  $\gamma_{11} = 2.5$ ,  $\gamma_{21} = 11.5$  for  $Y_1$  and  $\gamma_{12} = 1.4$ ,  $\gamma_{22} = 5.6$  for  $Y_2$ , and  $\rho_Y = .46$ . For  $Y_1$ , the distinctly different coefficient sets are:

$$\begin{aligned} \mathbf{a}_{51} &= [-0.2801 \quad 0.6407 \quad 0.2801 \quad 0.0846 ]' \\ \mathbf{a}_{52} &= [-0.6466 \quad -0.6194 \quad 0.6466 \quad 0.0976 ]' \end{aligned} \quad (36)$$

- For  $Y_2$ , we have

$$\begin{aligned} \mathbf{a}_{61} &= [-0.4281 \quad 1.1984 \quad 0.4281 \quad -0.1770 ]' \\ \mathbf{a}_{62} &= [-0.1594 \quad 0.7400 \quad 0.1594 \quad 0.0726 ]' \end{aligned} \quad (37)$$

- Combining each set for  $Y_1$  with the sets of  $Y_2$ , we obtain the four distributions plotted in Figure 4.

# Simulating Multivariate Non-Normality

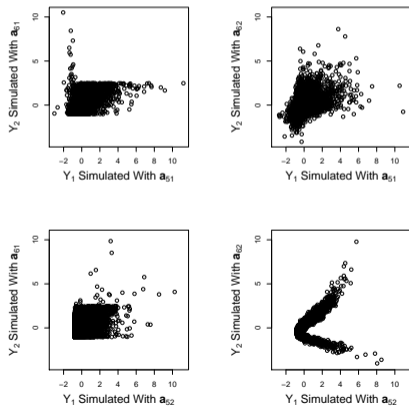
## The Vale-Maurelli Method

- The shapes in Figure 4 are strikingly different from one another, despite all having the same marginal skewnesses and kurtoses and the same correlation *and* all being produced with what has been portrayed in the literature as a single unique method for simulating nonnormal distributions.
- Notice that the only distribution with a “well-behaved” shape is the one constructed from the two transforms  $\mathbf{a}_{51}$  and  $\mathbf{a}_{62}$  (which are both monotonic).
- The distribution created from the two nonmonotonic transforms has a somewhat box-shaped form.
- The distribution in Figure 4(d) almost looks like a mixture of two relatively normal distributions with high correlations in opposite directions.

# Simulating Multivariate Non-Normality

## The Vale-Maurelli Method

Figure 4: Odd-shaped PT3 Distributions



# Simulating Non-Perfect Fit

- Models are only approximations to reality.
- To fit perfectly, models require statistical populations to be confined to a subset of the parameter space. We first learned this fact when diagramming the parameter space for the null hypothesis that  $\mu = 100$  in Psychology 310.
- We learned it again when solving by elimination for the model constraints in our Very Simple Model as a prelude to discussing how Spearman derived the tetrad difference formulae in factor analysis.



# Simulating Non-Perfect Fit

- There is an important hidden lesson in the tetrad differences discussion.
- The expression of a model in terms of  $\Sigma$ -constraints is more fundamental than its expression in terms of data equations.
- Different models (or, if you prefer, data generating systems) can imply identical  $\Sigma$ -constraints.

# Simulating Non-Perfect Fit

- Structural equation models seem “reasonable” when expressed as a system of data-generating equations.
- When expressed in their more fundamental form as a set of  $\Sigma$ -constraints, they seem far less likely to be true.
- For a tetrad difference to be exactly zero would require a remarkable coincidence of correlations, or so it would seem.

# Simulating Non-Perfect Fit

- If a model is “wrong,” when what, indeed, should we be estimating?
- Many Monte Carlo studies have concentrated on a very specific, very artificial situation, in which a more complex model is “right,” and a simpler “wrong” model is missing some of the paths in this more complex model.
- In this case, it seems clear that what we “should” be estimating is the more complex, perfect, “correct” model, so it is easy to calculate how “wrong” certain parameter estimates are when the simpler, incorrect model is fit.
- We could also track how well “model modification indices” are diagnostic of the missing paths in such a situation.

# Simulating Non-Perfect Fit

- Is this “wrong path” approach to simulation of imperfect model fit realistic and reasonable?
- In what ways does it fall short (C.P.)

# Simulating Non-Perfect Fit

## The Cudeck-Browne Approach

- Cudeck and Browne (1992) discussed a method for producing a covariance matrix that has a specific “population discrepancy function” and a known set of minimizing parameters.
- With this method, one could, for example, produce a covariance matrix that, when fit with the method of maximum likelihood and the 9x3 confirmatory factor model discussed earlier, converges to loadings of 0.20, with a discrepancy function of 0.24.
- With this covariance matrix, one could simulate various non-normal distributions using the method of Vale and Maurelli (1983).
- Combining the Cudeck-Browne method with the Vale-Maurelli method would allow one to assess performance across a wide range of distributions, sample sizes, and departures from perfect fit.

# Simulating Non-Perfect Fit

## The Cudeck-Browne Approach

- Suppose we are seeking a covariance matrix such that the maximum likelihood discrepancy function  $F_{ML}(\mathbf{S}, \mathbf{M}(\boldsymbol{\gamma}))$  has a desired value  $\delta$  and a function minimizer of  $\boldsymbol{\gamma} = \boldsymbol{\gamma}_0$ .
- This covariance is constructed in the form  $\boldsymbol{\Sigma} = \mathbf{M}(\boldsymbol{\gamma}) + \mathbf{E}$ .
- As an example, suppose a standardized single common factor model with 10 observed variables has unique variances of

$$\text{diag}(\boldsymbol{\Theta}) = [ 0.65 \quad 0.60 \quad 0.55 \quad 0.50 \quad 0.45 \quad 0.40 \quad 0.35 \quad 0.30 \quad 0.25 \quad 0.20 ]' \quad (38)$$

- Since the common factor pattern  $\boldsymbol{\lambda}$  has only one column, each element  $\lambda_i$  must be equal to  $\sqrt{1 - \theta_{ii}}$ .
- So the first factor loading would be  $\sqrt{1 - 0.65} = \sqrt{0.35} = 0.592$ . The second factor loading would be  $\sqrt{0.40} = 0.632$ , etc.

# Simulating Non-Perfect Fit

## The Cudeck-Browne Approach

- Although the Cudeck-Browne method produces a covariance matrix that has a desired non-zero discrepancy, we can modify the method to yield a desired value of the RMSEA fit statistic.
- Since the population RMSEA is defined as

$$R = \sqrt{\frac{F_{ML}}{\nu}} \quad (39)$$

where  $\nu$  is the model degrees of freedom, we have  $F_{ML} = R^2\nu$ .

- So if we desired an RMSEA of 0.08, recalling that the degrees of freedom for a 10 variable single factor model are

$$\nu = 1/2(p - m)^2 - (p + m) = 1/2(10 - 1)^2 - (10 + 1) = 35 \quad (40)$$

we would need a discrepancy function of

$$F_{ML} = 35 \times 0.08^2 = 36 \times 0.0064 = 0.224 \quad (41)$$

# Simulating Non-Perfect Fit

## The Cudeck-Browne Approach

- The  $10 \times 10$  correlation matrix on the next slide, when analyzed with maximum likelihood and a single factor model, has a minimum discrepancy function of precisely 0.224, with factor loadings and unique variances as given on the previous slide.





# Simulating Non-Perfect Fit

## The Cudeck-Browne Approach

- This matrix is online in a text file called `PopMatrix_08_1.csv`.
- Suppose we wanted to do a brief Monte Carlo study of the performance of a single factor model with this matrix as the population  $\Sigma$ , and multivariate normal data.
- How would we go about doing this? (C.P.)

## Directly Specifying a Population Covariance Matrix

- There are several alternatives, but if we wanted to employ the built-in Monte Carlo capabilities in Mplus without augmenting any of the data that are collected, we would face an immediate inconvenience.
- It would make sense for Mplus to have a `POPULATIONSIGMA` command in its `MODEL POPULATION` section. This way, the user could simply input the matrix from a file or add it to the command section.
- However, it seems that Mplus does not have such a command.

## Directly Specifying a Population Covariance Matrix

- One straightforward approach to modeling a directly specified  $\Sigma$  is to construct a model for it in the MODEL POPULATION: section of the input file.
- Suppose, for example, you wished to model a population covariance matrix of

$$\Sigma = \begin{bmatrix} 1.0 & & \\ 0.2 & 1.0 & \\ 0.2 & 0.3 & 1.0 \end{bmatrix} \quad (43)$$

- We can convey this model to Mplus as follows:

MODEL POPULATION:

Y2 WITH Y1@0.2

Y3 WITH Y1@0.2

Y3 WITH Y2@0.3

Y1@1

Y2@1

Y3@1

## Directly Specifying a Population Covariance Matrix

- Unfortunately, in general such a specification would, for a  $p \times p$  covariance matrix, have  $p(p + 1)/2 + 1$  lines of code.
- So, for example, if we wished to specify the  $10 \times 10$  covariance matrix with the discrepancy function of 0.224, we would need to input 56 lines of code, many of which include a lengthy decimal number.
- Clearly, constructing such code is a job for a computer.

## Directly Specifying a Population Covariance Matrix

- Online is a code file `MplusCovGenerator.R` containing a function `MplusGenerateModelForSigma`.
- This function takes an input covariance matrix in either full form, or as a vector containing the lower triangular elements.
- It generates an output file with the desired commands, which can then be integrated into an Mplus input file for the Monte Carlo run.
- Let's digress to take a look at the function and its use.

# Directly Specifying a Population Covariance Matrix

- The integrated file is online as `MCTEST.inp`.
- Running it, we discover that the mean parameter estimates and the mean RMSEA are right on the population values.