ML Estimation for Multivariate Outcomes; Robust Maximum Likelihood; Empty Multivariate Models

EPSY 905: Multivariate Analysis
Spring 2016
Lecture 6: March 2, 2016
Today’s Class

• Multivariate linear models: an introduction

• An introduction to maximum likelihood estimation for multivariate models
  ➢ Empty models: mean vectors and covariance matrices

• Robust maximum likelihood for MVN outcomes
  ➢ Augmenting likelihood functions for data that aren’t quite MVN

• Measures of model fit
MULTIVARIATE MODELS: AN INTRODUCTION
Multivariate Linear Models

• The next four weeks of lecture are provided to give an overview of multivariate linear models
  ➢ Models for more than one dependent/outcome variable

• Our focus will be on models where the DV is plausibly continuous (so we’ll use error terms that are multivariate normally distributed)
  ➢ Not a necessity – generalized multivariate models are possible
Classical Approaches to Multivariate Linear Models

- In “classical” multivariate textbooks and classes multivariate linear models fall under the names of Multivariate ANOVA (MANOVA) and Multivariate Regression.

- These methods rely upon least squares estimation which:
  - Inadequate with missing data
  - Offers very limited methods of setting covariance matrix structures
  - Does not allow for different sets predictor variables for each outcome
  - Does not give much information about model fit
  - Does not provide adequate model comparison procedures

- The classical methods have been *subsumed* into the modern (likelihood or Bayes-based) multivariate methods.
• We will discuss two large classes of multivariate linear modeling methods:
  - Path analysis models (typically through structural equation modeling software)
  - Linear mixed models (typically through linear models software)

• The theory behind both is identical – the main difference is in software
  - Some software does a lot (Mplus is likely the most complete), but none (as of March 2016) does it all

• We will start with path analysis (via the lavaan package) as the modeling method is more direct but then move to linear mixed models software (via the nlme and lme4 packages) to be complete in our discussion
The “Curse” of Dimensionality

- For **multivariate normal data**: having a quadratically increasing number of parameters as the number of outcomes increases linearly is sometimes called the “curse of dimensionality”

- Having lots of parameters creates a number of problems
  - Estimation issues for small sample sizes
  - Power to detect effects
  - Model fit issues for large numbers of outcomes

- Common solutions to this problem are to impose more limited “structures” on the covariance matrix
  - Many structures exist: discussion of structures is next

- To be used as an analysis model, however, a covariance structure must “fit” **as well as** the saturated/unstructured covariance matrix
  - Discussion of model fit follows discussion of structures
Biggest Difference Between Multivariate and Univariate Models: Model Fit

• In univariate models the “model for the variance” wasn’t much of a model
  - There was one variance term possible and one term estimated
    - A saturated model
  - Model fit was always perfect

• Because of the number of variances/covariances, multivariate models often don’t have saturated models for the variances
  - Therefore, model fit becomes an issue

• Any non-saturated model for the variances must be shown to fit the data** before being used for interpretation
  - ** fit the data has differing standards depending on software type used
## Multivariate Linear Models Software Scorecard

<table>
<thead>
<tr>
<th>Feature</th>
<th>Need</th>
<th>Classical MANOVA/ M-Regression</th>
<th>Path Analysis (lavaan unless stated)</th>
<th>Mixed Models (nlme/lme4 unless stated)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dependent variables can predict other dependent variables simultaneously</td>
<td>Path analysis definition</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Variable-specific modeling options</td>
<td>Precise model development</td>
<td>NO</td>
<td>YES (directly)</td>
<td>YES (with dummy codes)</td>
</tr>
<tr>
<td>“Robust” Maximum Likelihood Estimation</td>
<td>Provides lepto/platykurtic data protection</td>
<td>NO</td>
<td>YES</td>
<td>YES (most without LRT)</td>
</tr>
<tr>
<td>Residual Maximum Likelihood Estimation</td>
<td>Provides unbiased variances, covariance, and (some) path estimates</td>
<td>YES (analogously)</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Ability to Incorporate Random Effects</td>
<td>Additional levels of dependency</td>
<td>NO</td>
<td>YES (two-level nested; Mplus does more)</td>
<td>YES</td>
</tr>
<tr>
<td>Multiple Covariance Structures (beyond Compound Symmetry and Unstructured)</td>
<td>Model parsimony (“Curse” of multidimensionality)</td>
<td>NO</td>
<td>YES (some)</td>
<td>YES (many in R; many more in SAS)</td>
</tr>
<tr>
<td>Approximate Model Fit Indices</td>
<td>Determining How Bad a Model May Be</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Different denominator df methods</td>
<td></td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
</tbody>
</table>
EXAMPLE DATA SET
Today’s Data Example


• Sample of 350 undergraduates (229 women, 121 men)
  - In simulation, 10% of variables were missing (using missing completely at random mechanism)

• Note: simulated data characteristics differ from actual data (some variables extend beyond their official range)
  - Simulated using Multivariate Normal Distribution
    - Some variables had boundaries that simulated data exceeded
  - Results will not match exactly due to missing data and boundaries
Variables of Data Example

- **Gender (1 = male; 0 = female)**
- **Math Self-Efficacy (MSE)**
  - Reported reliability of .91
  - Assesses math confidence of college students
- **Perceived Usefulness of Mathematics (USE)**
  - Reported reliability of .93
- **Math Anxiety (MAS)**
  - Reported reliability ranging from .86 to .90
- **Math Self-Concept (MSC)**
  - Reported reliability of .93 to .95
- **Prior Experience at High School Level (HSL)**
  - Self report of number of years of high school during which students took mathematics courses
- **Prior Experience at College Level (CC)**
  - Self report of courses taken at college level
- **Math Performance (PERF)**
  - Reported reliability of .788
  - 18-item multiple choice instrument (total of correct responses)
Our Destination: Overall Path Model

- High School Math Experience
- Gender
- College Math Experience
- Mathematics Self-Concept
- Mathematics Self-Efficacy
- Direct Effect
- Residual Variance
- Mathematics Performance
- Perceived Usefulness
MAXIMUM LIKELIHOOD BY MVN: USING LAVAAN FOR ESTIMATION
Using MVN Likelihoods in Lavaan

• Lavaan’s default model is a linear (mixed) model that uses ML with the multivariate normal distribution
  ➢ ML is Full Information = All Data Contribute

• You can use lavaan to do analyses for all sorts of linear models including:
  ➢ MANOVA
  ➢ Repeated Measures ANOVA
  ➢ Factor Models

• The MVN assumption is what we will use for most of this course on SEM
  ➢ Later in the semester, we will work with distributions for categorical outcomes
Revisiting Univariate Linear Regression

• We will begin our discussion by starting with perhaps the simplest model we will see: a univariate empty model
  ➢ We will use the PERF variable from our example data

• The empty model is then:

\[ \text{PERF}_i = \beta_0^{\text{PERF}} + \epsilon_i^{\text{PERF}} \]

  ➢ Additionally, \( \epsilon_i^{\text{PERF}} \sim N(0, \sigma_{\epsilon:\text{PERF}}^2) \)
  ➢ So, two parameters are estimated: \( \beta_0^{\text{PERF}} \) and \( \sigma_{\epsilon:\text{PERF}}^2 \)

• Here, the superscript is added to denote these terms are part of the model for the PERF variable
  ➢ We will need these when we get to multivariate models and path analysis
Model Parameter Estimates and Assumptions

• Using lavaan, this model’s syntax is

```r
# Model 1: Univariate empty model for PERF
model01.syntax = "
# Variances:
perf ~ perf
# Means:
perf ~ 1
";

# empty model estimation
model01.fit = sem(model01.syntax, data=math_data, mimic="MPLUS", fixed.x=TRUE, estimator = "MLR")
```

• Interpret the following estimates... $\beta_0$ and $\sigma^2_{e:PERF}$

|                | Estimate | Std.err | Z-value | P(>|z|) |
|----------------|----------|---------|---------|---------|
| Intercepts:    | perf     | 13.966  | 0.174   | 80.397  | 0.000   |
| Variances:     | perf     | 8.751   | 0.756   | 11.581  | 0.000   |

• As we assumed $e_i^{PERF} \sim N(0, \sigma^2_{e:PERF})$ we assume the following about the data:

$$PERF_i \sim N(\beta_0^{PERF}, \sigma^2_{e:PERF})$$

• Using the model estimates, this becomes:

$$PERF_i \sim N(13.966, 8.751)$$
Mostly good agreement between model and data

Also note the sample size:
Adding One More Variable: Multivariate Regression

• We will now move to modeling two variables from our example data that we wish to describe:
  ➢ Mathematics performance (PERF)
  ➢ Perceived usefulness (PERF)

• We will assume these to be continuous variables

• Initially, we will only look at an empty model with these two variables
  ➢ Empty models are baseline models
  ➢ We will use these to show how such models look based on the characteristics of the multivariate normal distribution
  ➢ We will also show the bigger picture when modeling multivariate data: how we must be sure to model the covariance matrix correctly
Multivariate Empty Model: The Notation

• The multivariate model for PERF and USE is given by two regression models, which are estimated simultaneously:

\[ PERF_i = \beta_0^{PERF} + e_i^{PERF} \]
\[ USE_i = \beta_0^{USE} + e_i^{USE} \]

• As there are two variables, the error terms have a joint distribution that will be a multivariate normal:

\[
\begin{bmatrix}
    e_i^{PERF} \\
    e_i^{USE}
\end{bmatrix}
\sim N_2\left( \mathbf{0} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{R} = \begin{bmatrix}
    \sigma_{e:PERF}^2 & \sigma_{e:PERF,USE} \\
    \sigma_{e:PERF,USE} & \sigma_{e:USE}^2
\end{bmatrix} \right)
\]

• Each error term has its own variance but now there is a covariance between error terms

  ➢ We will soon see that the overall \( \mathbf{R} \) matrix structure can be modified
Data Model

• Before showing the syntax and the results, we must first describe how the multivariate empty model implies how our data should look
  
  This will be true for this week...next week we will have Y show up on either side of the equals sign which changes the math a little

• Multivariate model with matrices:

\[
\begin{bmatrix}
PERF_i \\
USE_i
\end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix}
\beta_0^{PERF} \\
\beta_0^{USE}
\end{bmatrix} + \begin{bmatrix} e_i^{PERF} \\
e_i^{USE}
\end{bmatrix}
\]

\[
Y_i = X_i B + e_i
\]

• Using expected values and linear combination rules, we can show that:

\[
\begin{bmatrix}
PERF_i \\
USE_i
\end{bmatrix} \sim N_2 \left( \mu_i = \begin{bmatrix}
\beta_0^{PERF} \\
\beta_0^{USE}
\end{bmatrix}, \sigma^2 = \begin{bmatrix}
\sigma^2_{e:PERF} & \sigma_{e:PERF,USE} \\
\sigma_{e:PERF,USE} & \sigma^2_{e:USE}
\end{bmatrix} \right)
\]

PRE 906, SEM: Multivariate Models and Model Fit
Lavaan Multivariate Regression Model Syntax

\[
\begin{bmatrix}
\text{PERF}_i \\
\text{USE}_i
\end{bmatrix}
\sim N_2\left(
\begin{bmatrix}
\beta_{0,\text{PERF}} \\
\beta_{0,\text{USE}}
\end{bmatrix},
\begin{bmatrix}
\sigma^2_{e:\text{PERF}} & \sigma_{e:\text{PERF,USE}} \\
\sigma_{e:\text{PERF,USE}} & \sigma^2_{e:\text{USE}}
\end{bmatrix}
\right)
\]

`model02.syntax = "
#Variances:
  perf ~~~ perf
  use ~~~ use

#Covariance:
  perf ~~~ use

#Means:
  perf ~~~ 1
  use ~~~ 1``

This covariance matrix is said to be **saturated**: All parameters are estimated.

It is also called an **unstructured** covariance matrix.

No other structure for the covariance matrix can fit better (only as well as)
Multivariate Regression Model Results

• The estimated values:
  - What is the estimated correlation between PERF and USE?

| Covariances: | Estimate | Std.err | Z-value | P>|z|) |
|--------------|----------|---------|---------|------|
| perf ~ use   | 6.847    | 2.850   | 2.403   | 0.016|

| Intercepts:  | Estimate | Std.err | Z-value | P>|z|) |
|--------------|----------|---------|---------|------|
| perf         | 13.959   | 0.174   | 80.442  | 0.000|
| use          | 52.440   | 0.872   | 60.140  | 0.000|

| Variances:   | Estimate | Std.err | Z-value | P>|z|) |
|--------------|----------|---------|---------|------|
| perf         | 8.742    | 0.754   | 11.596  | 0.000|
| use          | 249.245  | 19.212  | 12.973  | 0.000|

• [Side Note] Why is the sample size different from Model 1?

<table>
<thead>
<tr>
<th>Number of observations</th>
<th>Used</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>348</td>
<td>350</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number of missing patterns</th>
<th>3</th>
</tr>
</thead>
</table>
Plotting the Model Estimated Results

\[
\begin{bmatrix}
PERF_i \\
USE_i
\end{bmatrix} \sim N_2
\begin{bmatrix}
13.959 \\
52.440
\end{bmatrix},
\begin{bmatrix}
8.742 & 6.847 \\
6.847 & 249.245
\end{bmatrix}
\]
Comparing Model with Data

Multivariate Regression Model Estimated Density with Data
The Problem with Multivariate Models

- As more dependent variables (outcomes) are added to a multivariate model, the number of parameters needed for a saturated model gets very large:
The “Curse” of Dimensionality

• For multivariate normal data: having a quadratically increasing number of parameters as the number of outcomes increases linearly is sometimes called the “curse of dimensionality”

• Having lots of parameters creates a number of problems
  - Estimation issues for small sample sizes
  - Power to detect effects
  - Model fit issues for large numbers of outcomes

• Common solutions to this problem are to impose more limited “structures” on the covariance matrix
  - Many structures exist: discussion of structures is next

• To be used as an analysis model, however, a covariance structure must “fit” as well as the saturated/unstructured covariance matrix
  - Discussion of model fit follows discussion of structures
ROBUST MAXIMUM LIKELIHOOD:  
AN ESTIMATOR EXCLUSIVE TO PATH ANALYSIS  
MODELING SOFTWARE
Robust Estimation: The Basics

• Robust estimation in ML still assumes the data follow a multivariate normal distribution
  ➢ But that the data have more or less kurtosis than would otherwise be common in a normal distribution

• Kurtosis: measure of the shape of the distribution
  ➢ From Greek word for bulging
  ➢ Can be estimated for data (either marginally for each item or jointly across all items)

• The degree of kurtosis in a data set is related to how incorrect the log-likelihood value will be
  ➢ Leptokurtic data (too-fat tails): $\chi^2$ inflated, SEs too small
  ➢ Platykurtic data (too-thin tails): $\chi^2$ depressed, SEs too large
Visualizing Kurtosis


Black line = Normal Distribution (0 kurtosis)
Robust ML for Non-Normality in lavaan: MLR

- Robust ML can be specified very easily in lavaan:
  - Add estimator = “MLR” to your sem() function call

```r
#empty model estimation
model03.fit = sem(model03.syntax, data=job_data, mimic="MPLUS", fixed.x=TRUE, estimator = "MLR")
```

- The model parameter estimates will all be identical to those found under regular maximum likelihood
  - And...if data are MVN – then no adjustment is made (so we can use MLR for everything!)

- MLR adjusts:
  - Model $\chi^2$ (and associated fit statistics that use it: RMSEA, CFI, TLI) – closely related to Yuan-Bentler $T_2$ (permits MCAR or MAR missing data)
  - Model standard errors: uses Huber-White “sandwich” estimator to adjust standard errors
    - Sandwich estimator found using information matrix of the partial first derivatives to correct information matrix from the partial second derivatives
Adjusted Model Fit Statistics

• Under MLR, model fit statistics are adjusted based on an estimated scaling factor:
  ➢ Scaling factor = 1.000
    • Perfectly MVN data
  ➢ Scaling factor > 1.000
    • Leptokurtosis (too-fat tails; fixes too big $\chi^2$)
  ➢ Scaling factor < 1.000
    • Platykurtosis (too-thin tails; fixes too small $\chi^2$)

• The scaling factor will now show up in all likelihood ratio tests (deviance tests)
  ➢ So you must add it to your calculations
**Adjusted Standard Errors**

- The standard errors for all parameter estimates will be different under MLR
  - Remember, these are used in Wald tests

- If the data show leptokurtosis (too-fat tails):
  - Increases information matrix
  - Fixes too small SEs

- If the data show platykurtosis (too-thin tails):
  - Lowers values in information matrix
  - Fixes too big SEs
Data Analysis Example with MLR

• To demonstrate, we will revisit our analysis of the example data for today’s class using MLR.

• So far, we have estimated two models:
  - Saturated model
  - Independence model

• We compare results of the two estimators (ML v. MLR).

• Because MLR is something does not affect our results if we have MVN data, we can be using MLR for each analysis.
### lavaan Output: Log-likelihoods Under ML and MLR

- **Model 3 Results (Model 1 w/MLR)**

<table>
<thead>
<tr>
<th></th>
<th>Under ML</th>
<th>Under MLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loglikelihood user model (H0)</td>
<td>-124.939</td>
<td>-124.939</td>
</tr>
<tr>
<td>Loglikelihood unrestricted model (H1)</td>
<td>-124.939</td>
<td>-124.939</td>
</tr>
<tr>
<td>Number of free parameters</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Akaike (AIC)</td>
<td>259.877</td>
<td>259.877</td>
</tr>
<tr>
<td>Bayesian (BIC)</td>
<td>264.856</td>
<td>264.856</td>
</tr>
<tr>
<td>Sample-size adjusted Bayesian (BIC)</td>
<td>249.442</td>
<td>249.442</td>
</tr>
</tbody>
</table>

- The actual log-likelihoods are the same
Model 4 Results (Model 2 w/MLR)

<table>
<thead>
<tr>
<th>Loglikelihood and Information Criteria:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loglikelihood user model (H0)</td>
</tr>
<tr>
<td>Scaling correction factor for the MLR correction</td>
</tr>
<tr>
<td>Loglikelihood unrestricted model (H1)</td>
</tr>
<tr>
<td>Scaling correction factor for the MLR correction</td>
</tr>
</tbody>
</table>

• The actual log-likelihoods are the same
  ➢ But, under MLR, the log-likelihood gets re-scaled

• The scaling factor for the restricted model (Model 4) is 1.222 – indicates slightly leptokurtic data

• The scaling factor for the saturated model (Model 3) is 0.9095 – indicates slightly platykurtic data
Adding Scaling Factors to the Analysis

• The MLR-estimated scaling factors are used to rescale the log-likelihoods under LR test model comparisons
  ➢ Extra calculations are needed

• The rescaled LR test is given by:

\[ LR_{RS} = \frac{-2\left(\log L_{\text{restricted}} - \log L_{\text{full}}\right)}{c_{LR}} \]

• The denominator is found by the scaling factors \( c \) and number of parameters \( q \) in each model:

\[ c_{LR} = \left| \frac{(q_{\text{restricted}})(c_{\text{restricted}}) - (q_{\text{full}})(c_{\text{full}})}{(q_{\text{restricted}} - q_{\text{full}})} \right| \]

• Sometimes \( c_{LR} \) can be negative - so take the absolute value
Model Comparison: Independence v. Saturated Model

Loglikelihood and Information Criteria:

<table>
<thead>
<tr>
<th></th>
<th>H0</th>
<th>H1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loglikelihood user model</td>
<td>-148.500</td>
<td>-148.500</td>
</tr>
<tr>
<td>Scaling correction factor for the MLR correction</td>
<td>1.222</td>
<td>0.910</td>
</tr>
</tbody>
</table>

To compare the independence model against the saturated model we must first calculate the scaling factor

- $q_{full} = 5$ – number of parameters in saturated model
- $c_{full} = 0.9095$ – scaling factor from saturated model
- $q_{reduced} = 3$ – number of parameters in one-factor model
- $c_{reduced} = 1.2223$ – scaling factor from one-factor model

The scaling factor for the LR test is then:

$$c_{LR} = \frac{3 \times 1.2223 - 5 \times 0.9095}{3 - 5} = \frac{-0.8806}{-2} = .4403$$
Model Comparison #1: Independence vs. Saturated Model

- The next step is to calculate the re-scaled likelihood ratio test using the original log-likelihoods and the scaling factor:

\[ LR_{RS} = \frac{-2(\log L_{\text{restricted}} - \log L_{\text{full}})}{c_{LR}} \]

\[ = \frac{-2(-148.500 - -124.939)}{.4403} = 107.0225 \]

- Finally, we use the rescaled LR test as we would in any other LR test- compare it to a \( \chi^2 \) with df = difference in number of parameters (here 2)
### Standard Errors/Wald Tests Under MLR

| Covariances: | Estimate | Std.err | Z-value | P(>|z|) |
|-------------|----------|---------|---------|---------|
| IQ ~ perfC  | 19.500   | 9.151   | 2.131   | 0.033   |
| Intercepts: |          |         |         |         |
| IQ          | 100.000  | 3.079   | 32.478  | 0.000   |
| perfC       | 10.350   | 0.584   | 17.714  | 0.000   |
| Variances:  |          |         |         |         |
| IQ          | 189.600  | 59.957  | 3.162   | 0.002   |
| perfC       | 6.828    | 2.159   | 3.162   | 0.002   |

| Covariances: | Estimate | Std.err | Z-value | P(>|z|) |
|-------------|----------|---------|---------|---------|
| IQ ~ perfC  | 19.500   | 6.578   | 2.964   | 0.003   |
| Intercepts: |          |         |         |         |
| IQ          | 100.000  | 3.079   | 32.478  | 0.000   |
| perfC       | 10.350   | 0.584   | 17.714  | 0.000   |
| Variances:  |          |         |         |         |
| IQ          | 189.600  | 56.722  | 3.343   | 0.001   |
| perfC       | 6.828    | 1.831   | 3.729   | 0.000   |

- The SEs of our model under MLR are smaller than the SEs under ML
  - As such, the values of the Wald tests are larger (SEs are the denominator)
MLR: The Take-Home Point

- If you feel you have continuous data that are (tenuously) normally distributed, use MLR
  - Any time you use SEM/CFA/Path Analysis as we have to this point
  - In general, likert-type items with 5 or more categories are treated this way
  - If data aren’t/cannot be considered normal we should still use different distributional assumptions

- If data truly are MVN, then MLR doesn’t adjust anything

- If data are not MVN (but are still continuous), then MLR adjusts the important inferential portions of the results
COVARIANCE STRUCTURES FOR (CONDITIONAL) MULTIVARIATE NORMAL OUTCOMES
• When modeling multivariate outcomes (with the MVN distribution), covariance structures can be placed on virtually any portion of the model itself.

• So far, we have seen the following model:

\[ \mathbf{Y}_i = \mathbf{X}_i \mathbf{B} + \mathbf{e}_i \]

- \( \mathbf{X}_i \) is the matrix of \( p \) predictors for person \( i \)
- \( \mathbf{B} \) is the matrix of regression coefficients (fixed effects)
- \( \mathbf{e}_i \) is the vector of error terms for person \( i \)
  - Assumption: \( \mathbf{e}_i \sim N_p (0, \mathbf{R}) \)

• These specifics lead to the following data assumptions:

\[ \mathbf{Y}_i \sim N_p (\mathbf{X}_i \mathbf{B}, \mathbf{V}_i = \mathbf{R}) \]

• Big picture: covariance matrix of \( \mathbf{Y}_i \) (which I am calling \( \mathbf{V}_i \)) must fit as well as an unstructured matrix of \( \mathbf{Y}_i \).
Covariance Matrices of $Y$: Why $V = R$ Right Now

• Up to this point, we’ve only seen the following covariance matrix notation: $V_i = R$

\[
Y_i = X_i B + e_i \\
Y_i \sim N_p(X_i B, V_i = R)
\]

• This is because our model (an empty multivariate regression model) only has one set of random terms
  - The error terms **assumption**: $e_i \sim N_p(0, R)$
  - $R$ is the covariance matrix of these random terms (the errors)

• When there are more than one set of random terms in a model, the covariance matrix for the data gets more complicated

• **Any** covariance matrix can have a structure
  - Each structure will lead to a different version of $V_i$

• Regardless of the structures imposed, $V_i$ must fit **as well as** a saturated/unstructured model covariance matrix
**Covariance Structures for R: Error Covariance Matrices**

- **The R matrix** is somewhat limited in the structures it can take as the error terms are very general
  - We will demonstrate with our two-variable empty model example
    - Note: Most often you will use UN for two variables as only one covariance is needed*

**Common R matrix structures:**

- **Variance components** (independent variables with same variances):
  - Each variable has the same variance (one variance estimated total)
  - Each pair of variables has covariance = 0 (independent variables)
- **Independent Variables** (independent variables with own variances):
  - Each variable has its own variance estimated (one variance per variable estimated)
  - Each pair of variables has covariance = 0
- **Compound Symmetry**:
  - Each variable has the same variance (one variance estimated total)
  - Each pair of variables has the same covariance (one covariance estimated total)
- **Unstructured**
  - Each variable has its own variance estimated (one variance per variable estimated)
  - Each pair of variables has a covariance estimated
Model 03: Variance Components Structure

\[
\begin{bmatrix}
PERF_i \\
USE_i
\end{bmatrix} \sim N_2\left(\begin{bmatrix}
\beta_0^{PERF} \\
\beta_0^{USE}
\end{bmatrix}, \begin{bmatrix}
\sigma_e^2 & 0 \\
0 & \sigma_e^2
\end{bmatrix}\right)
\]

```
model03.syntax = "
#variances:
  perf ~ (var)*perf
  use ~ (var)*use

#Covariance:
  perf ~ 0*use

#Means:
  perf ~ 1
  use ~ 1"
```
Model 03: Comparing Model with Data

Multivariate Regression Model Estimated Density
Model 04: Independent Variables Structure

$$\begin{bmatrix} \text{PERF}_i \\ \text{USE}_i \end{bmatrix} \sim N_2 \left( \begin{bmatrix} \beta_0^{\text{PERF}} \\ \beta_0^{\text{USE}} \end{bmatrix}, \begin{bmatrix} \sigma_{e:\text{PERF}}^2 & 0 \\ 0 & \sigma_{e:\text{USE}}^2 \end{bmatrix} \right)$$

```r
model04.syntax = "
#Variances:
  perf ~ perf
  use ~ use

#Covariance:
  perf ~ 0*use

#Means:
  perf ~ 1
  use ~ 1"
```

| Covariances: | Estimate | Std.err | Z-value | P(>|z|) |
|--------------|----------|---------|---------|---------|
| perf ~ use   | 0.000    |         |         |         |

| Intercepts:  | Estimate | Std.err | Z-value | P(>|z|) |
|--------------|----------|---------|---------|---------|
| perf         | 13.966   | 0.174   | 80.397  | 0.000   |
| use          | 52.500   | 0.874   | 60.047  | 0.000   |

| Variances:   | Estimate | Std.err | Z-value | P(>|z|) |
|--------------|----------|---------|---------|---------|
| perf         | 8.751    | 0.756   | 11.581  | 0.000   |
| use          | 249.201  | 19.212  | 12.971  | 0.000   |
Model 04: Independent Variables Structure

Multivariate Regression Model Estimated Density
Model 05: Compound Symmetry Structure

\[
\begin{bmatrix}
PERF_i \\
USE_i
\end{bmatrix} \sim N_2\left(\begin{bmatrix}
\beta_0^{PERF} \\
\beta_0^{USE}
\end{bmatrix}, \begin{bmatrix}
\sigma_e^2 & \sigma \\
\sigma & \sigma_e^2
\end{bmatrix}\right)
\]

```
model05.syntax = "
#Variances:
  perf ~ (var)*perf
  use ~ (var)*use

#Covariance:
  perf ~ (cov)*use

#Means:
  perf ~ 1
  use ~ 1"
```

\[
\text{Covariances:}
\begin{array}{cccc}
\text{perf} & \text{use} & \text{Estimate} & \text{Std.err} & \text{Z-value} & \text{P(>|z|)} \\
\text{perf} & \text{use} & (cov) & 7.207 & 3.027 & 2.381 & 0.017 \\
\end{array}
\]

\[
\text{Intercepts:}
\begin{array}{cccc}
\text{perf} & \text{use} & \text{Estimate} & \text{Std.err} & \text{Z-value} & \text{P(>|z|)} \\
\text{perf} & \text{use} & & 13.952 & 0.174 & 80.211 & 0.000 \\
\end{array}
\]

\[
\text{Variances:}
\begin{array}{cccc}
\text{perf} & \text{use} & \text{Estimate} & \text{Std.err} & \text{Z-value} & \text{P(>|z|)} \\
\text{perf} & \text{use} & (var) & 136.004 & 10.219 & 13.308 & 0.000 \\
\end{array}
\]
Model 05: Comparing Model with Data
Model 02: Unstructured/Saturated Structure

\[
\begin{bmatrix}
\text{PERF}_i \\
\text{USE}_i
\end{bmatrix}
\sim N_2\left(\begin{bmatrix}
\beta_0^{\text{PERF}} \\
\beta_0^{\text{USE}}
\end{bmatrix}, \begin{bmatrix}
\sigma_e^{2\text{PERF}} & \sigma_e^{2\text{PERF,USE}} \\
\sigma_e^{2\text{PERF,USE}} & \sigma_e^{2\text{USE}}
\end{bmatrix}\right)
\]

```
model02.syntax = 

#Variances: 
perf ~ ~ perf
use ~ ~ use

#Covariance: 
perf ~ ~ use

#Means: 
perf ~ 1
use ~ 1
```

|              | Estimate | Std.err | Z-value | P(>|z|) |
|--------------|----------|---------|---------|---------|
| Covariances: |          |         |         |         |
| perf ~ use   |          |         |         |         |
| use ~ use    |          |         |         |         |
| Intercepts:  |          |         |         |         |
| perf         | 13.959   | 0.174   | 80.442  | 0.000   |
| use          | 52.440   | 0.872   | 60.140  | 0.000   |
| Variances:   |          |         |         |         |
| perf         | 8.742    | 0.754   | 11.596  | 0.000   |
| use          | 249.245  | 19.212  | 12.973  | 0.000   |
Model 02: Comparing Model with Data

Multivariate Regression Model Estimated Density with Data
WRAPPING UP AND REFOCUSING
Wrapping Up

• Today was a course on ML and “Robust” ML estimation for multivariate outcomes using path analysis

• These topics are important when using lavaan as there are quite a few different estimators in the package
  ➢ ML is not always the default