lavaan and Missing data using MLEs

PSQF 7373 (Spring 2025):

Missing Data Methods
Lecture 04

In This Lecture…

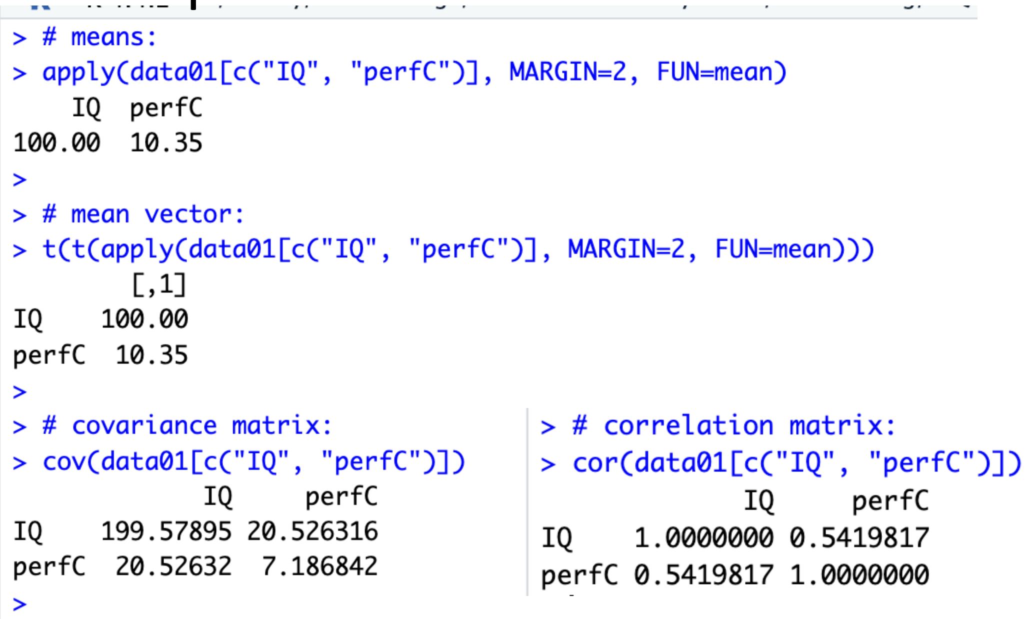
- The Multivariate Normal Distribution

- Multivariate linear models with predictors (using path analysis software/packages)

- Details and terminology from path analysis:
  - Variable naming conventions
  - Software estimation defaults (variables in/out of likelihood)
  - Model comparisons via likelihood ratio tests
  - Measures of absolute and approximate model fit
  - Model modification methods
  - Standardized regression coefficients

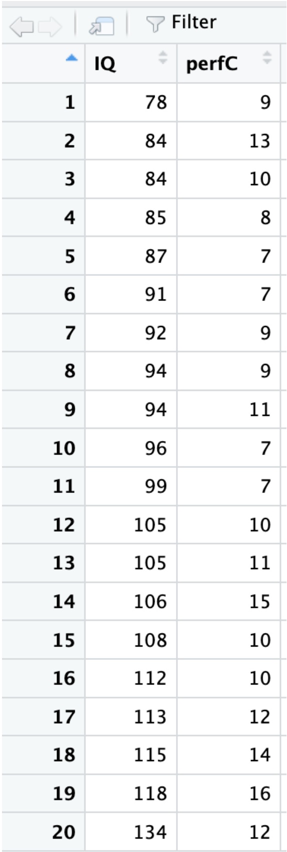
Today’s Example Data #1

- Imagine an employer is looking to hire employees for a job where IQ is important
  - First, we will only use the hypothetical complete data (20) observations so as to show the math behind the estimation calculations
- The employer collects two variables:
  - IQ scores
  - Job performance
- Descriptive Statistics:



```
> # means:
> apply(data01[c("IQ", "perfC")], MARGIN=2, FUN=mean)
    IQ  perfC 
100.00  10.35 
> 
> # mean vector:
> t(t(apply(data01[c("IQ", "perfC")], MARGIN=2, FUN=mean)))
        [,1]
IQ    100.00
perfC  10.35
> 
> # covariance matrix:
> cov(data01[c("IQ", "perfC")])
             IQ     perfC
IQ    199.57895 20.526316
perfC  20.52632  7.186842
> 
```

```
> # correlation matrix:
> cor(data01[c("IQ", "perfC")])
             IQ     perfC
IQ    1.0000000 0.5419817
perfC 0.5419817 1.0000000
```



	IQ	perfC
1	78	9
2	84	13
3	84	10
4	85	8
5	87	7
6	91	7
7	92	9
8	94	9
9	94	11
10	96	7
11	99	7
12	105	10
13	105	11
14	106	15
15	108	10
16	112	10
17	113	12
18	115	14
19	118	16
20	134	12

Multivariate Statistics

- Up to this point in this course, we have focused on the prediction (or modeling) of a single variable
    - Conditional distributions or univariate marginal distributions

- We will need to know about joint distributions to enable us to use lavaan in a manner that will help with missing data

- Path is about exploring **joint distributions**
    - How variables relate to each other simultaneously

- Therefore, we must adapt our conditional distributions to have multiple variables, simultaneously (later, as multiple outcomes)

- We will now look at the joint distributions of two variables $f(x_1, x_2)$ or in matrix form: $f(\mathbf{X})$ (where $\mathbf{X}$ is size N x 2; $f(\mathbf{X})$ gives a scalar/single number)
    - Beginning with two, then moving to anything more than two
    - We will begin by looking at **multivariate descriptive statistics**
        - **Mean vectors and covariance matrices**

Multiple Means: The Mean Vector

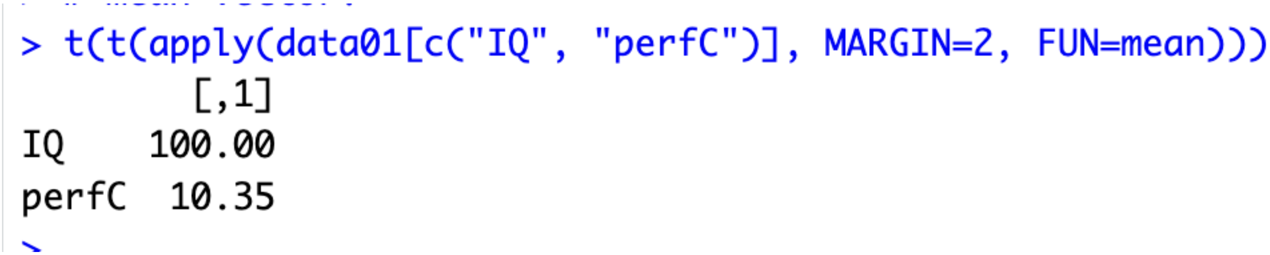
- We can use a vector to describe the set of means for our data

$$\bar{\mathbf{x}} = \frac{1}{N}\mathbf{X}^T\mathbf{1} = \begin{bmatrix}\bar{x}_1\\ \bar{x}_2\\ \vdots\\ \bar{x}_V\end{bmatrix}$$

  - Here **1** is a N x 1 vector of 1s
  - The resulting mean vector is a *v* x 1 vector of means

- For our data: $\bar{\mathbf{x}} = \begin{bmatrix}100.00\\ 10.35\end{bmatrix} = \begin{bmatrix}\bar{x}_{IQ}\\ \bar{x}_{perfC}\end{bmatrix}$

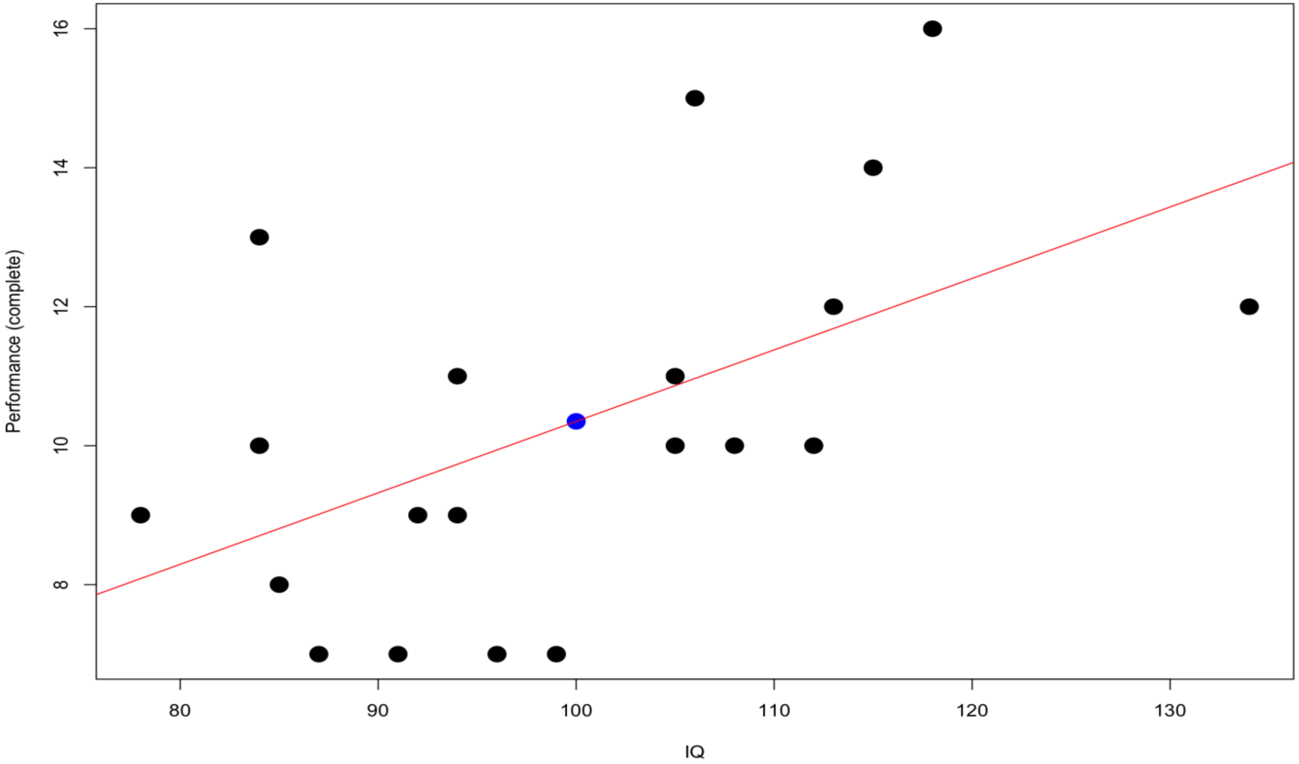
- In R:



```
> t(t(apply(data01[c("IQ", "perfC")], MARGIN=2, FUN=mean)))
        [,1]
IQ    100.00
perfC  10.35
>
```

Mean Vector: Graphically

- The mean vector is the center of the distribution of both variables



Covariance of a Pair of Variables

- The covariance is a measure of the relatedness
  - Expressed in the product of the units of the two variables:

$$s_{x_1 x_2} = \frac{1}{N}\sum_{p=1}^{N}\left(x_{p1} - \bar{x}_1\right)\left(x_{p2} - \bar{x}_2\right)$$

  - The covariance between IQ and PerfC was 20.53 (in IQ-Perfs)
  - The denominator N is the ML version – unbiased is N-1

- Because the units of the covariance are difficult to understand, we more commonly describe association (correlation) between two variables with correlation
  - Covariance divided by the product of each variable's standard deviation

Correlation of a Pair of Variables

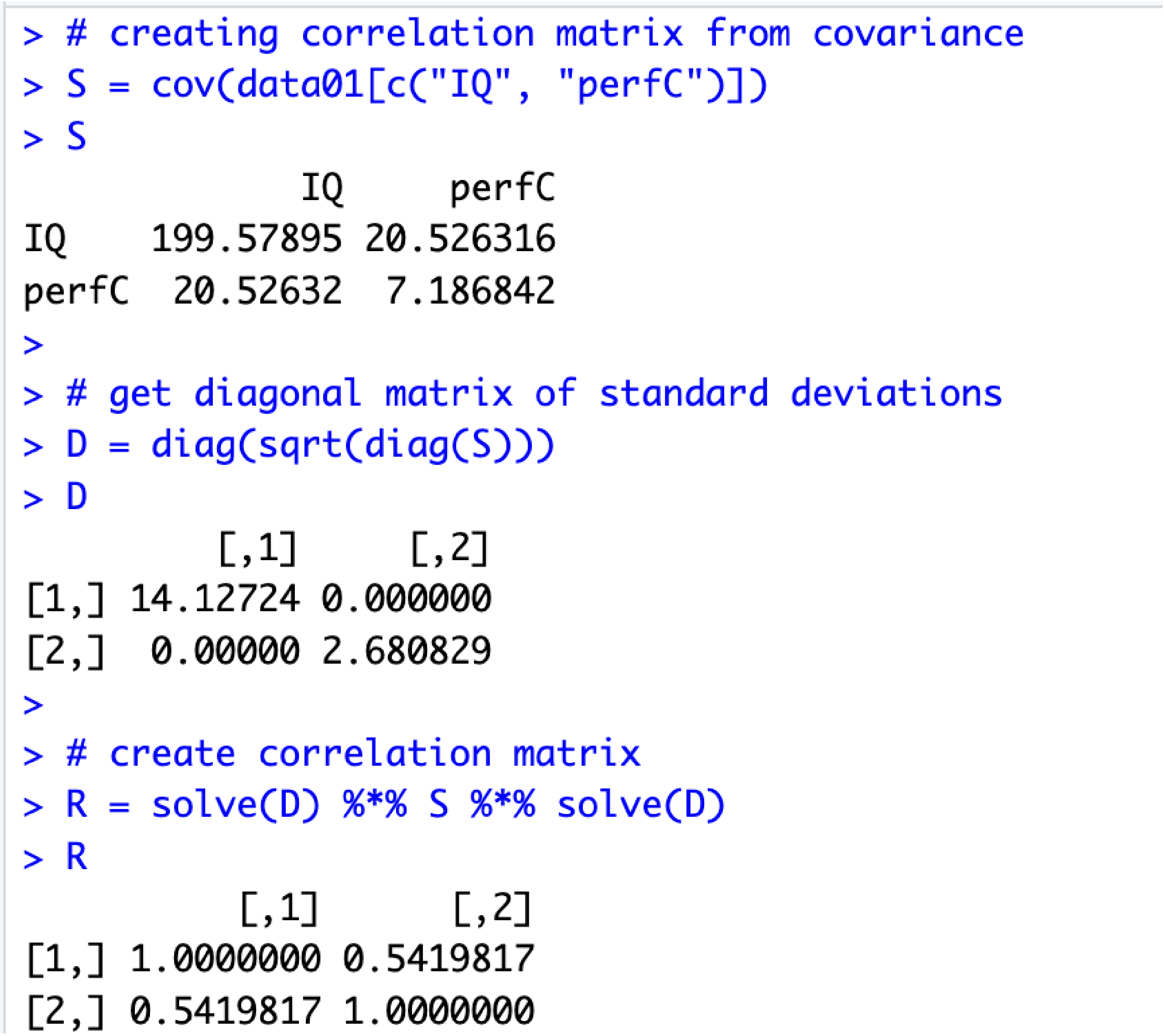
- Correlation is covariance divided by the product of the standard deviation of each variable:

$$r_{x_1 x_2} = \frac{s_{x_1 x_2}}{\sqrt{s_{x_1}^2}\sqrt{s_{x_2}^2}}$$

  - The correlation between IQ and Perf was 0.541

- Correlation is unitless – it only ranges between -1 and 1
  - If x_1 **and** x_2 both had variances of 1, the covariance between them would be a correlation
    - Covariance of standardized variables = correlation

In R:



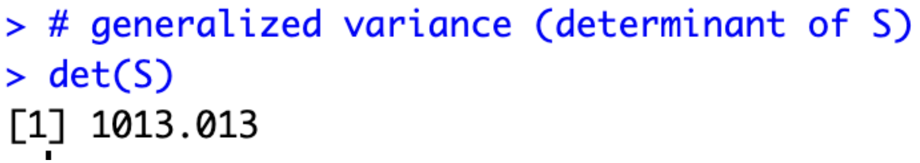
```
> # creating correlation matrix from covariance
> S = cov(data01[c("IQ", "perfC")])
> S
             IQ     perfC
IQ    199.57895 20.526316
perfC  20.52632  7.186842
>
> # get diagonal matrix of standard deviations
> D = diag(sqrt(diag(S)))
> D
         [,1]     [,2]
[1,] 14.12724 0.000000
[2,]  0.00000 2.680829
>
> # create correlation matrix
> R = solve(D) %*% S %*% solve(D)
> R
          [,1]      [,2]
[1,] 1.0000000 0.5419817
[2,] 0.5419817 1.0000000
```

Generalized Variance

- The determinant of the covariance matrix is the **generalized variance**

$$\text{Generalized Sample Variance} = |\mathbf{S}|$$

- It is a measure of spread across all variables
  - Reflecting how much overlap (covariance) in variables occurs in the sample
  - Amount of overlap reduces the generalized sample variance
  - Generalized variance from our example: 1,013.13
  - Generalized variance if zero covariance/correlation: 1,434.342



```
> # generalized variance (determinant of S)
> det(S)
[1] 1013.013
```

- The generalized sample variance is:
  - Largest when variables are uncorrelated
  - Zero when variables form a linear dependency

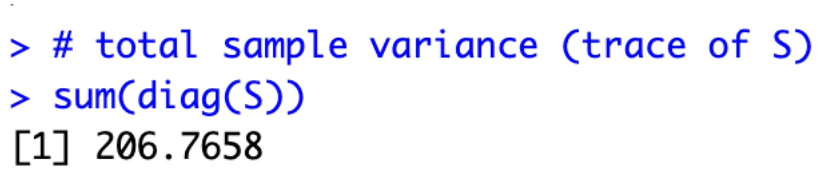
- **In data:**
  - The generalized variance is seldom used descriptively, but shows up more frequently in maximum likelihood functions

Total Sample Variance

- The total sample variance is the sum of the variances of each variable in the sample
  - The sum of the diagonal elements of the sample covariance matrix
  - The trace of the sample covariance matrix

$$Total\ Sample\ Variance = \sum_{v=1}^{V} s_{x_i}^2 = \mathrm{tr}\,\mathbf{S}$$

- Total sample variance for our example:



```
> # total sample variance (trace of S)
> sum(diag(S))
[1] 206.7658
```

- The total sample variance does not take into consideration the covariances among the variables
  - Will not equal zero if linearly dependency exists

- **In data:**
  - The total sample variance is commonly used as the denominator (target) when calculating variance accounted for measures

MULTIVARIATE DISTRIBUTIONS (VARIABLES ≥ 2)

Multivariate Normal Distribution

- The multivariate normal distribution is the generalization of the univariate normal distribution to multiple variables
  - The bivariate normal distribution just shown is part of the MVN

- The MVN provides the relative likelihood of observing **all** *V* variables for a subject *p* simultaneously:

$$\mathbf{x}_p = \begin{bmatrix} x_{p1} & x_{p2} & \cdots & x_{pV} \end{bmatrix}$$

- The multivariate normal density function is:

$$f\left(\mathbf{x}_p\right) = \frac{1}{(2\pi)^{\frac{V}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}}\exp\left[-\frac{\left(\mathbf{x}_p^T - \boldsymbol{\mu}\right)^T \mathbf{\Sigma}^{-1}\left(\mathbf{x}_p^T - \boldsymbol{\mu}\right)}{2}\right]$$

The Multivariate Normal Distribution

$$f\left(\mathbf{x}_p\right) = \frac{1}{(2\pi)^{\frac{V}{2}}|\mathbf{\Sigma}|^{\frac{1}{2}}}\exp\left[-\frac{\left(\mathbf{x}_p^T - \boldsymbol{\mu}\right)^T \mathbf{\Sigma}^{-1}\left(\mathbf{x}_p^T - \boldsymbol{\mu}\right)}{2}\right]$$

- The mean vector is $\boldsymbol{\mu} = \begin{bmatrix} \mu_{x_1} \\ \mu_{x_2} \\ \vdots \\ \mu_{x_V} \end{bmatrix}$
- The covariance matrix is $\mathbf{\Sigma} = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1x_2} & \cdots & \sigma_{x_1x_V} \\ \sigma_{x_1x_2} & \sigma_{x_2}^2 & \cdots & \sigma_{x_2x_V} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{x_1x_V} & \sigma_{x_2x_V} & \cdots & \sigma_{x_V}^2 \end{bmatrix}$
  - ➢ The covariance matrix must be non-singular (invertible)
    - ◆ Technically we call this ”positive semi-definite”, which means the determinant of $\mathbf{\Sigma}$ must be greater than or equal to zero

Comparing Univariate and Multivariate Normal Distributions

- The univariate normal distribution:

$$f(x_p) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

- The univariate normal, rewritten with a little algebra:

$$f(x_p) = \frac{1}{(2\pi)^{\frac{1}{2}}|\sigma^2|^{\frac{1}{2}}} \exp\left[-\frac{(x-\mu)\sigma^{-\frac{1}{2}}(x-\mu)}{2}\right]$$

- The multivariate normal distribution

$$f(\mathbf{x}_p) = \frac{1}{(2\pi)^{\frac{V}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{\left(\mathbf{x}_p^T-\boldsymbol{\mu}\right)^T\boldsymbol{\Sigma}^{-1}\left(\mathbf{x}_p^T-\boldsymbol{\mu}\right)}{2}\right]$$

  - When $V = 1$ (one variable), the MVN is a univariate normal distribution

The Exponent Term

- The term in the exponent (without the $-\frac{1}{2}$) is called the **squared Mahalanobis Distance**

$$d^2(\boldsymbol{x}_p) = \left(\mathbf{x}_p^T - \boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1}\left(\mathbf{x}_p^T - \boldsymbol{\mu}\right)$$

  - Sometimes called the statistical distance
  - Describes how far an observation is from its mean vector, in standardized units
  - Like a multivariate Z score (if data are MVN, is distributed as a χ^2variable with DF = number of variables in X)
  - Can be used to assess if data follow MVN

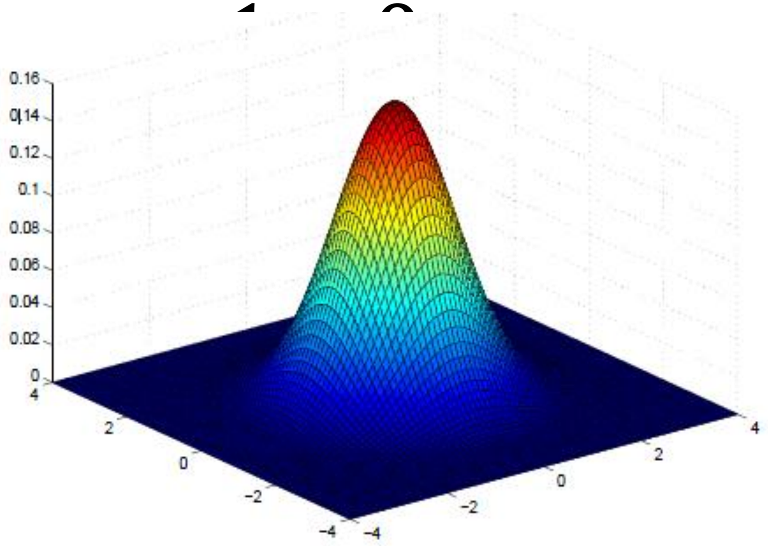
Multivariate Normal Notation

- Standard notation for the multivariate normal distribution of *v* variables is $N_v(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
  - Our example would use a bivariate normal: $N_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

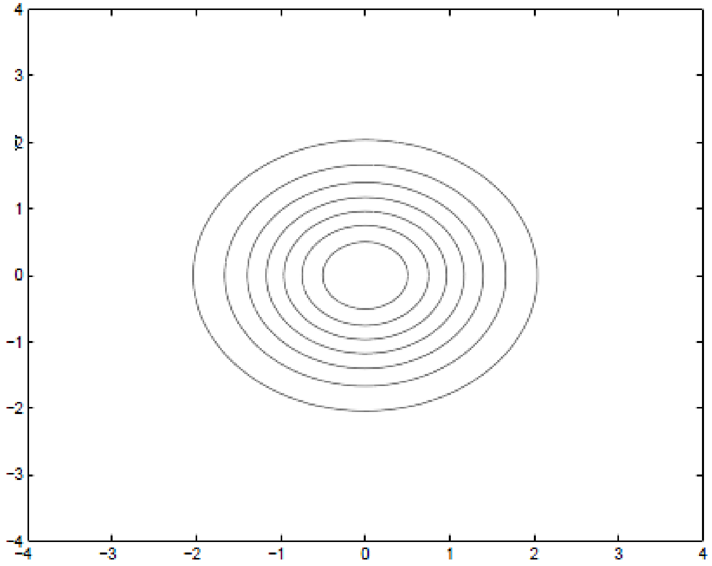
- **<u>In data:</u>**
  - The multivariate normal distribution serves as the basis for most every statistical technique commonly used in the social and educational sciences
    - General linear models (ANOVA, regression, MANOVA)
    - General linear mixed models (HLM/multilevel models)
    - Factor and structural equation models (EFA, CFA, SEM, path models)
    - Multiple imputation for missing data
  - Simply put, the world of commonly used statistics revolves around the multivariate normal distribution
    - Understanding it is the key to understanding many statistical methods

Bivariate Normal Plot #1

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_{x_1} \\ \mu_{x_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} \\ \sigma_{x_1 x_2} & \sigma_{x_2}^2 \end{bmatrix}$$



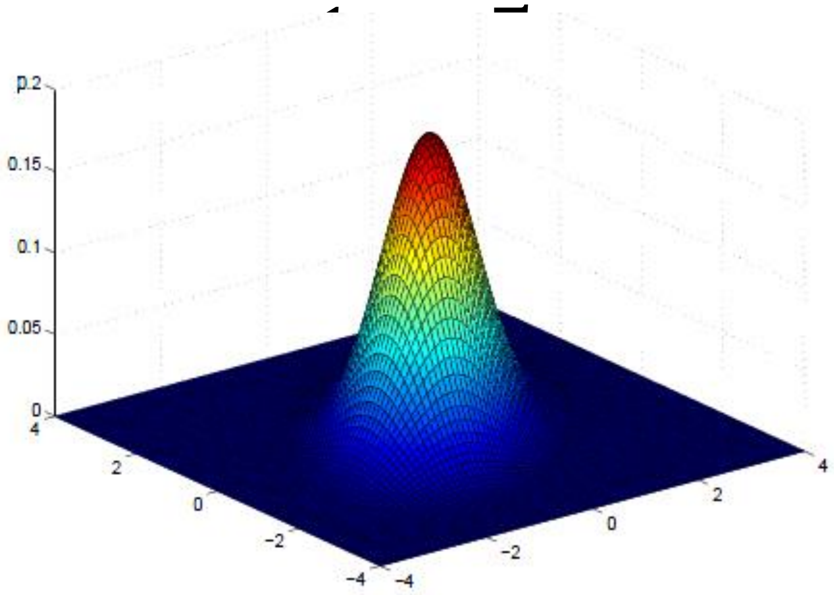
Density Surface (3D)



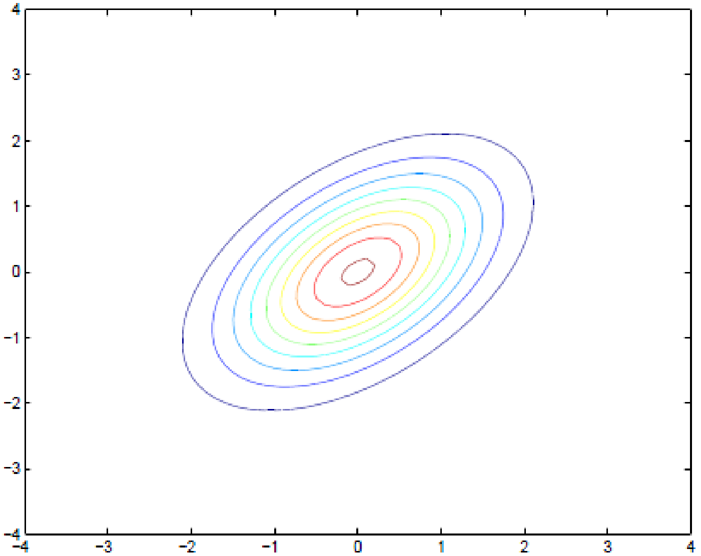
Density Surface (2D): Contour Plot

Bivariate Normal Plot #2 (Multivariate Normal)

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_{x_1} \\ \mu_{x_2} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{x_1}^2 & \sigma_{x_1 x_2} \\ \sigma_{x_1 x_2} & \sigma_{x_2}^2 \end{bmatrix}$$



Density Surface
(3D)



Density Surface
(2D): Contour
Plot

Multivariate Normal Properties

- The multivariate normal distribution has some useful properties that show up in statistical methods

- If $\mathbf{X}$ is distributed multivariate normally:

1. Linear combinations of $\mathbf{X}$ are normally distributed

2. All subsets of $\mathbf{X}$ are multivariate normally distributed

3. A zero covariance between a pair of variables of $\mathbf{X}$ implies that the variables are independent

4. Conditional distributions of $\mathbf{X}$ are multivariate normal

Multivariate Normal Distribution in PROC IML

- To demonstrate how the MVN works, we will now investigate how the PDF provides the likelihood (height) for a given observation:
  - Here we will use the example data and assume the sample mean vector and covariance matrix are known to be the true:

$$\boldsymbol{\mu} = \begin{bmatrix} 100.00 \\ 10.35 \end{bmatrix}; \mathbf{S} = \begin{bmatrix} 199.579 & 20.526 \\ 20.526 & 7.186 \end{bmatrix}$$

- We will compute the likelihood value for several observations (SEE EXAMPLE R SYNTAX FOR HOW THIS WORKS):
  - $\boldsymbol{x}_{2,\cdot} = [84 \quad 13]; f(\boldsymbol{x}) = 0.00042766421$
  - $\boldsymbol{x}_{5,\cdot} = [87 \quad 7]; \log(f(\boldsymbol{x})) = -6.12077$
  - $\boldsymbol{x} = \overline{\boldsymbol{x}} = [100 \quad 10.35]; \log\left(f(\boldsymbol{x})\right) = -5.298$

- Note: this is the height for these observations, not the joint likelihood across all the data
  - We will use the R packaged named lavaan to find the parameters in $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ using maximum likelihood

From Covariance to Correlation

- If we take the SDs (the square root of the diagonal of the covariance matrix) and put them into a diagonal matrix $\mathbf{D}$, the correlation matrix is found by:

$$\mathbf{R} = \mathbf{D}^{-1}\mathbf{S}\mathbf{D}^{-1} = \begin{bmatrix} \frac{s_{x_1}^2}{\sqrt{s_{x_1}^2}\sqrt{s_{x_1}^2}} & \cdots & \frac{s_{x_1x_p}}{\sqrt{s_{x_1}^2}\sqrt{s_{x_V}^2}} \\ \vdots & \ddots & \vdots \\ \frac{s_{x_1x_V}}{\sqrt{s_{x_1}^2}\sqrt{s_{x_V}^2}} & \cdots & \frac{s_{x_V}^2}{\sqrt{s_{x_V}^2}\sqrt{s_{x_V}^2}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \cdots & r_{x_1x_V} \\ \vdots & \ddots & \vdots \\ r_{x_1x_V} & \cdots & 1 \end{bmatrix}$$

Example Covariance Matrix

- For our data, the covariance matrix was:

$$\mathbf{S} = \begin{bmatrix} 199.58 & 20.53 \\ 20.53 & 7.17 \end{bmatrix}$$

- The diagonal matrix $\mathbf{D}$ was:

$$\mathbf{D} = \begin{bmatrix} \sqrt{199.58} & 0 \\ 0 & \sqrt{199.58} \end{bmatrix} = \begin{bmatrix} 14.12 & 0 \\ 0 & 2.68 \end{bmatrix}$$

- The correlation matrix $\mathbf{R}$ was:

$$\mathbf{R} = \mathbf{D}^{-1}\mathbf{S}\mathbf{D}^{-1} = \begin{bmatrix} \frac{1}{14.12} & 0 \\ 0 & \frac{1}{2.68} \end{bmatrix} \begin{bmatrix} 199.58 & 20.53 \\ 20.53 & 7.17 \end{bmatrix} \begin{bmatrix} \frac{1}{14.12} & 0 \\ 0 & \frac{1}{2.68} \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 1.00 & .542 \\ .542 & 1.00 \end{bmatrix}$$

MISSING DATA IN MAXIMUM LIKELIHOOD

Missing Data with Maximum Likelihood

- Handling missing data in maximum likelihood is much more straightforward due to the calculation of the log-likelihood function
  - Each subject contributes a portion due to their observations

- If some of the data are missing, the log-likelihood function uses a reduced form of the MVN distribution
  - Capitalizing on the property of the MVN that subsets of variables from an MVN distribution are also MVN

- The total log-likelihood is then maximized
  - Missing data just are "skipped" – they do not contribute

Each Person's Contribution to the Log-Likelihood

- For a person p, the MVN log-likelihood can be written:

$$\log L_p = -\frac{V}{2}\log(2\pi) - \frac{1}{2}\log\left(\left|\boldsymbol{\Sigma}_{\boldsymbol{p}}\right|\right) - \frac{\left(\boldsymbol{y}_p - \boldsymbol{\mu}_p\right)^T \boldsymbol{\Sigma}_p^{-1}\left(\boldsymbol{y}_p - \boldsymbol{\mu}_p\right)}{2}$$

➢ From our examples with missing data, subjects could either have all of their data…so their input into $\log L_p$ uses:

$$\boldsymbol{y}_p = \begin{bmatrix} y_{p,IQ} \\ y_{p,Perf} \end{bmatrix};$$

$$\boldsymbol{\mu}_p = \mathbf{X}_p\boldsymbol{\beta} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} \mu_{IQ} \\ \mu_{Perf} \end{bmatrix};$$

$$\boldsymbol{\Sigma}_p = \begin{bmatrix} \sigma_{IQ}^2 & \sigma_{IQ,Perf} \\ \sigma_{IQ,Perf} & \sigma_{Perf}^2 \end{bmatrix}$$

➢ …or could be missing the performance variable, yielding:

$$\boldsymbol{y}_p = \left[y_{p,IQ}\right]; \boldsymbol{\mu}_p = \mathbf{X}_p\boldsymbol{\beta} = \begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \left[\beta_0 + \beta_1\right] = \left[\mu_{IQ}\right]; \boldsymbol{\Sigma}_p = \left[\sigma_{IQ}^2\right]$$

Standard Errors with Incomplete Data

- Recall that in maximum likelihood, we derive standard errors based on the information matrix
  - Found through differentiating the likelihood function twice with respect to each parameter

- With missing data, when a score is missing, an observation contributes 0 to any element of the 2nd derivative matrix (Hessian)
  - This process falls under the name of “observed information” – from the data we observe

Observed vs. Expected Information

- In missing data contexts, using expected information requires MCAR to be accurate
  - ➢ Example from textbook: Larger variances

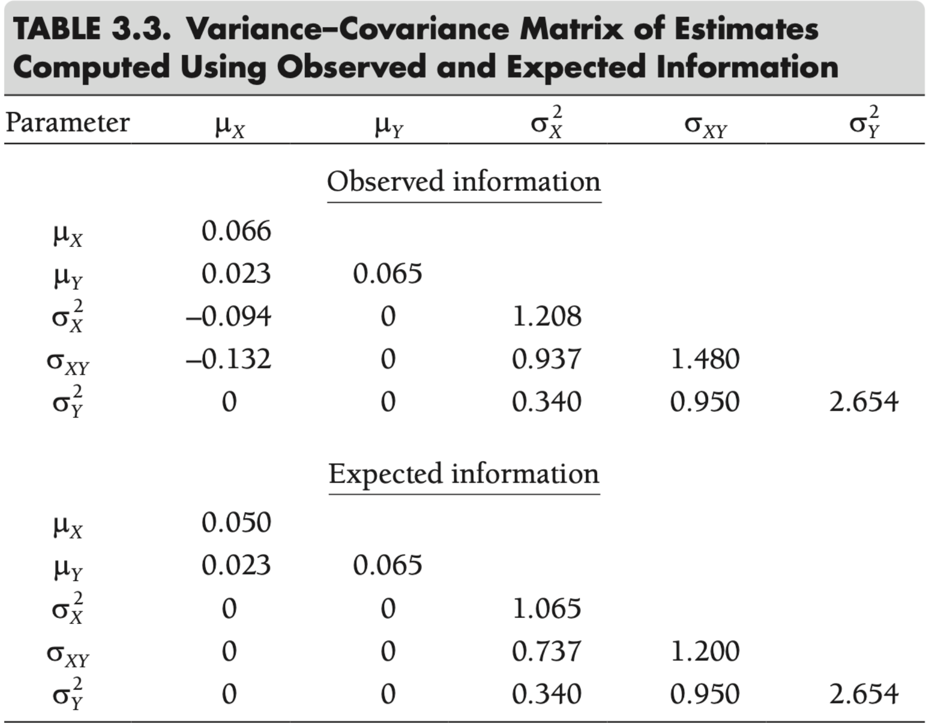


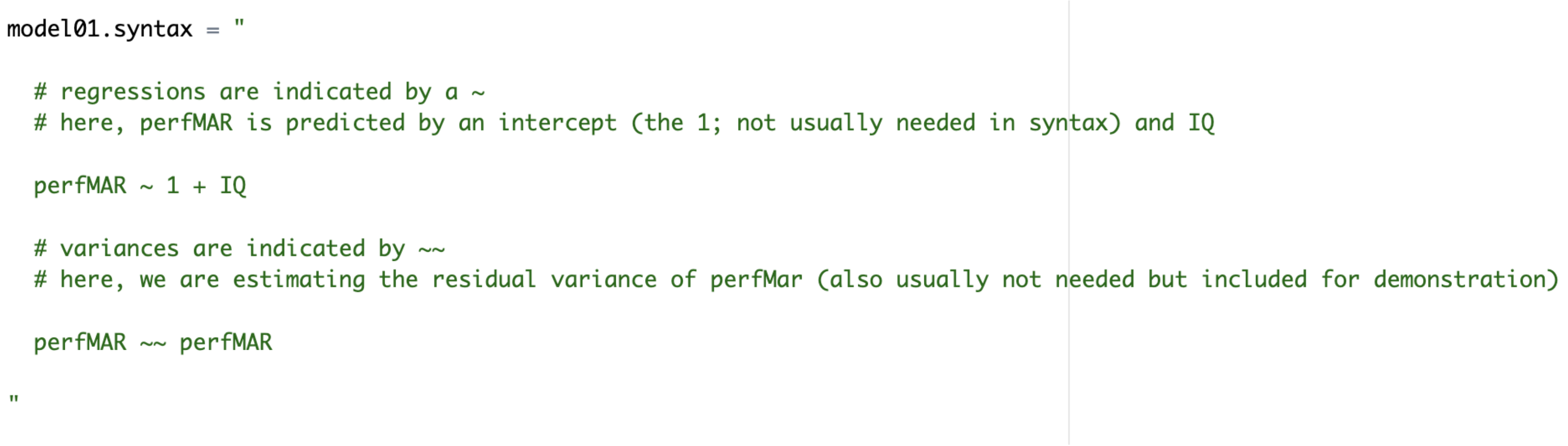
TABLE 3.3. Variance–Covariance Matrix of Estimates Computed Using Observed and Expected Information

Parameter	μ_X	μ_Y	σ_X^2	σ_{XY}	σ_Y^2
			Observed information		
μ_X	0.066				
μ_Y	0.023	0.065			
σ_X^2	–0.094	0	1.208		
σ_{XY}	–0.132	0	0.937	1.480	
σ_Y^2	0	0	0.340	0.950	2.654
			Expected information		
μ_X	0.050				
μ_Y	0.023	0.065			
σ_X^2	0	0	1.065		
σ_{XY}	0	0	0.737	1.200	
σ_Y^2	0	0	0.340	0.950	2.654

Using ML for the Example Data

- We can use lavaan for running multivariate models using maximum likelihood
    - lavaan also uses robust ML (protecting against deviations of normality due to lepto or platykurtic data)
- lavaan uses character strings where a model is specified
    - Then, that syntax is submitted to lavaan using some type of function based on the analysis (sem(), lavaan(), cfa())
- More on lavaan toward the end of this lecture, but for now, let's use it to estimate our parameters using ML

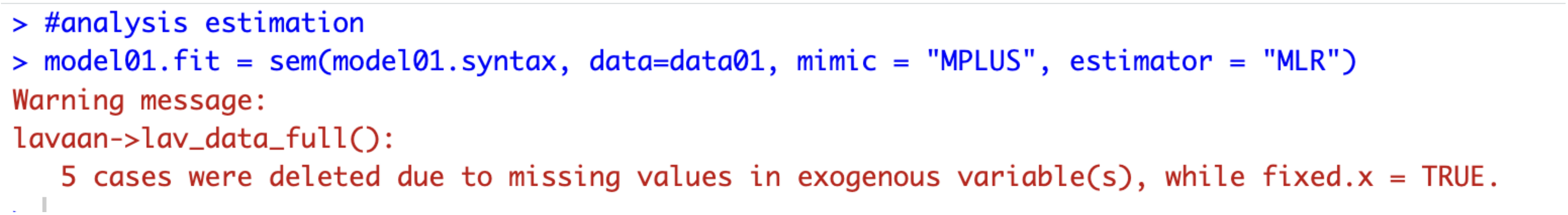
lavaan syntax



```
model01.syntax = "

  # regressions are indicated by a ~
  # here, perfMAR is predicted by an intercept (the 1; not usually needed in syntax) and IQ

  perfMAR ~ 1 + IQ

  # variances are indicated by ~~
  # here, we are estimating the residual variance of perfMar (also usually not needed but included for demonstration)

  perfMAR ~~ perfMAR

"
```

Model Estimation

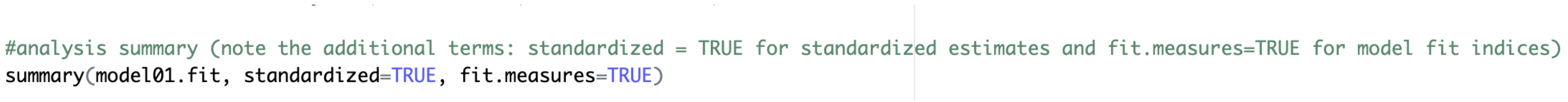
- But…how we've specified the model leads to having an issue with missing data



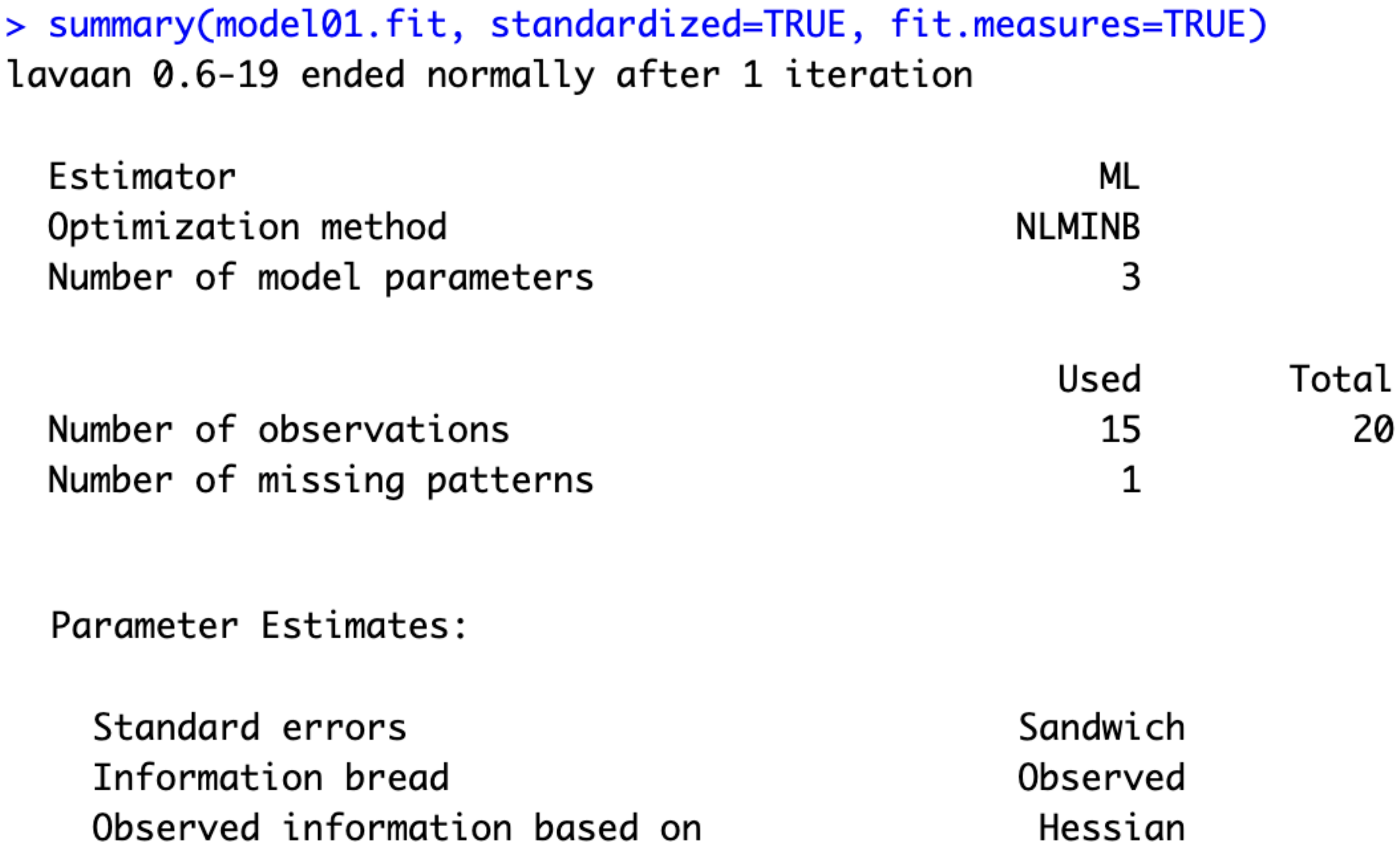
```
> #analysis estimation
> model01.fit = sem(model01.syntax, data=data01, mimic = "MPLUS", estimator = "MLR")
Warning message:
lavaan->lav_data_full():
   5 cases were deleted due to missing values in exogenous variable(s), while fixed.x = TRUE.
```

- For now, we will disregard this message
  - ➢ The rest of the lecture will be about how to build models in lavaan to avoid this message

Lavaan Results: Information from Output

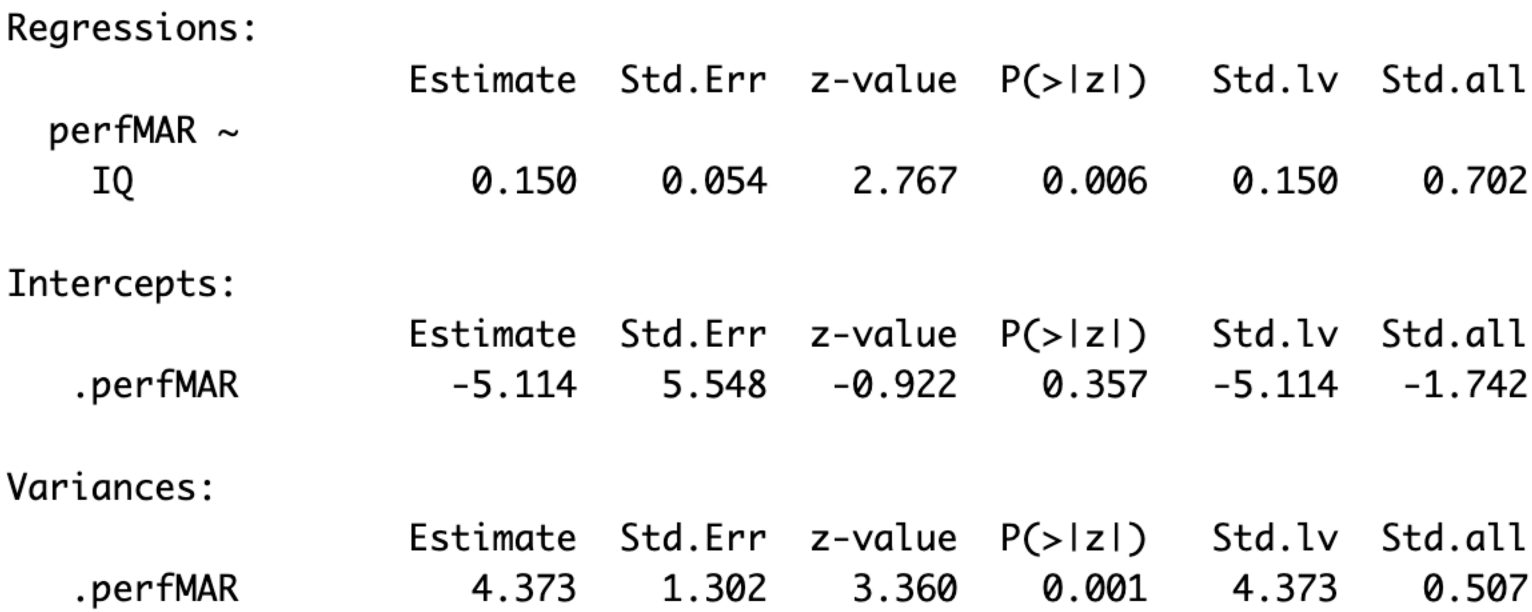


```
#analysis summary (note the additional terms: standardized = TRUE for standardized estimates and fit.measures=TRUE for model fit indices)
summary(model01.fit, standardized=TRUE, fit.measures=TRUE)
```



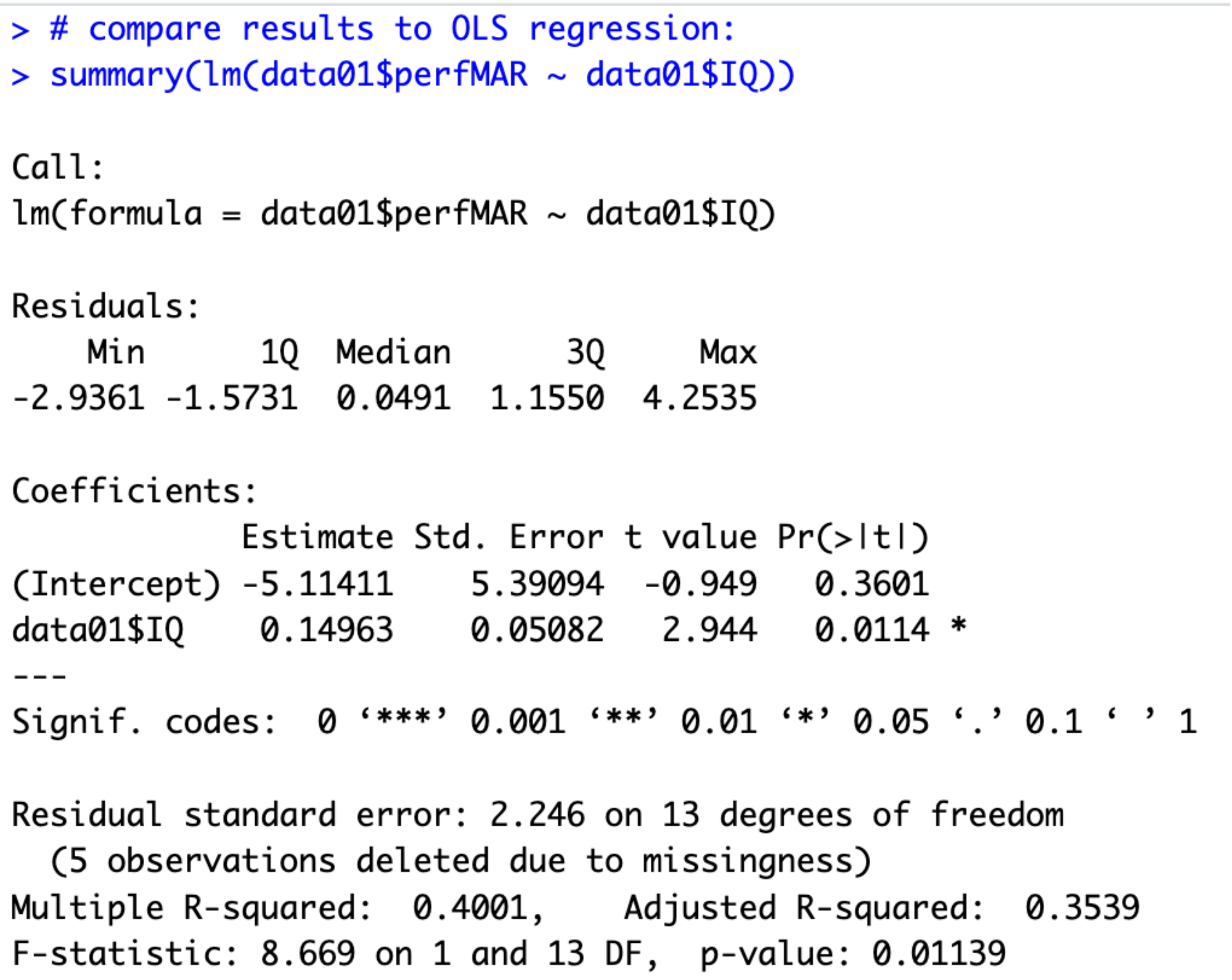
```
> summary(model01.fit, standardized=TRUE, fit.measures=TRUE)
lavaan 0.6-19 ended normally after 1 iteration

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                         3

                                                  Used       Total
  Number of observations                            15          20
  Number of missing patterns                         1


  Parameter Estimates:

    Standard errors                             Sandwich
    Information bread                           Observed
    Observed information based on                Hessian
```

Lavaan Parameter Results



```
Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  perfMAR ~
    IQ                0.150    0.054    2.767    0.006    0.150    0.702

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .perfMAR          -5.114    5.548   -0.922    0.357   -5.114   -1.742

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .perfMAR           4.373    1.302    3.360    0.001    4.373    0.507
```

Comparing lavaan to OLS Regression



```
> # compare results to OLS regression:
> summary(lm(data01$perfMAR ~ data01$IQ))

Call:
lm(formula = data01$perfMAR ~ data01$IQ)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.9361 -1.5731  0.0491  1.1550  4.2535 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -5.11411    5.39094  -0.949   0.3601  
data01$IQ    0.14963    0.05082   2.944   0.0114 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.246 on 13 degrees of freedom
  (5 observations deleted due to missingness)
Multiple R-squared:  0.4001,	Adjusted R-squared:  0.3539 
F-statistic: 8.669 on 1 and 13 DF,  p-value: 0.01139
```

DIFFICULT TO IMPLEMENT METHODS
EM ALGORITHM
FACTORIZATION

Difficult to Methods: EM and Factorization

- Enders describes two methods for estimating models with missing data, the EM algorithm and factorization

- As we will come to see, these methods are difficult to implement and don't easily generalize
  - ➢ But, as a teaching exercise, they do reinforce likelihoods and missing data

From Joint to Conditional/Marginal

- To use either of these methods, we must first note that our sample of two variables has a joint distribution: $f(IQ, perf)$

- As we saw two weeks ago, this joint distribution is equal to the product of a conditional distribution and a marginal distribution:

$$f(Perf|IQ)f(IQ)$$

Model Assumptions

- Now, we need to make some assumptions about the joint distribution to begin

- We can assume IQ and Performance follow a bivariate normal distribution
  - All marginal distributions are normal
  - Conditional distributions are normal

- Then, the distribution of Performance given IQ can be described by a regression

Quantities of Interest

- To show how EM works, we will first focus on the parameters we will estimate in ML

$$\hat{\mu}_X = \frac{1}{N}\sum_{i=1}^{N} X_i \quad \hat{\mu}_Y = \frac{1}{N}\sum_{i=1}^{N} Y_i \tag{3.18}$$

$$\hat{\sigma}_X^2 = \frac{1}{N}\left(\sum_{i=1}^{N} X_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} X_i\right)^2\right) \quad \hat{\sigma}_Y^2 = \frac{1}{N}\left(\sum_{i=1}^{N} Y_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} Y_i\right)^2\right)$$

$$\hat{\sigma}_{XY} = \frac{1}{N}\left(\sum_{i=1}^{N} X_i Y_i - \frac{1}{N}\sum_{i=1}^{N} X_i \sum_{i=1}^{N} Y_i\right)$$

- Here, we see there really are three quantities that we need to calculate to get estimates

E-Step

- The three estimates we need are

$$E\left(X \mid Y, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}\right) = \gamma_0^{(t)} + \gamma_1^{(t)} Y_i \qquad (3.19)$$
$$E\left(X^2 \mid Y, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}\right) = \left(\gamma_0^{(t)} + \gamma_1^{(t)} Y_i\right)^2 + \sigma_{X|Y}^{2(t)}$$
$$E\left(XY \mid Y, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)}\right) = Y_i \times E(X \mid Y) = Y_i\left(\gamma_0^{(t)} + \gamma_1^{(t)} Y_i\right)$$

- Using these we can then obtain new regression parameters

$$\gamma_1^{(t)} = \sigma_{XY}^{(t)} / \sigma_Y^{2(t)} \qquad (3.20)$$
$$\gamma_0^{(t)} = \mu_X^{(t)} - \gamma_1^{(t)} \mu_Y^{(t)}$$
$$\sigma_{X|Y}^{2(t)} = \sigma_X^{2(t)} - \gamma_1^{2(t)} \sigma_Y^{2(t)}$$

M-Step

- With the values of the missing data replaced by the current values of the parameters, we can then calculate quantities in the M-step:

$$\mu_X^{(t+1)} = \frac{1}{N}\left(\sum_{i=1}^{n_C} X_i + \sum_{i=1}^{n_M} E\left(X_i \mid Y_i\right)\right) \tag{3.21}$$

$$\sigma_X^{2(t+1)} = \frac{1}{N}\left(\sum_{i=1}^{n_C} X_i^2 + \sum_{i=1}^{n_M} E\left(X_i^2 \mid Y_i\right) - \frac{1}{N}\left(\sum_{i=1}^{n_C} X_i + \sum_{i=1}^{n_M} E\left(X_i \mid Y_i\right)\right)^2\right)$$

$$\sigma_{XY}^{(t+1)} = \frac{1}{N}\left(\sum_{i=1}^{n_C} X_i Y_i + \sum_{i=1}^{n_M} Y_i E\left(X_i \mid Y_i\right) - \frac{1}{N}\sum_{i=1}^{N} Y_i \left(\sum_{i=1}^{n_C} X_i + \sum_{i=1}^{n_M} E\left(X_i \mid Y_i\right)\right)\right)$$

Example E-M Algorithm: See R Syntax

- The R syntax for this section provides a modified version of Ender's algorithm built for our example data
- As you will see, the difficulty in the code suggests this method is rather limited
- This is also true for the factored regression specifications he discusses

Structural Equation Modeling Framework

- The SEM framework is an alternative to EM or factored regression

- It is far easier to implement—so much so that you may forget that other methods exist

- But, to learn about it, we must first discuss multivariate models

MULTIVARIATE MODELS: AN INTRODUCTION

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
  - ***Subsume:*** include or absorb (something) in something else
  - Meaning: modern methods do what classical methods do (and more)

Contemporary Methods for Estimating Multivariate Linear Models

- We will discuss path analysis models (typically through structural equation modeling and path analysis software)

- Other paradigms exist:
  - Linear mixed models (typically through linear models software)
  - Bayesian networks (frequently not mentioned in social sciences but subsume all we are doing)

- The theory behind each is identical – the main difference is in software
  - Some software does a lot (Mplus is likely the most complete), but none do it all off the shelf

- The frustrating part of each method is that each relies upon different estimation methods
  - So results sometimes lack comparability ‽‽‽

- We will start with path analysis (via the lavaan package)

The Curse of Dimensionality: Shared Across Models

- 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

- For multivariate normal data: having a quadratic increase in the number of parameters as the number of outcomes increases linearly is sometimes called the "curse of dimensionality"

- To be used as an analysis model, however, a covariance structure must "fit" **as well as** the **saturated/unstructured** covariance matrix

Biggest Difference From Univariate Models: Model Fit

- In univariate linear 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

EXAMPLE DATA SET

Today's Data Example

- Data are simulated based on the results reported in:

Pajares, F., & Miller, M. D. (1994). Role of self-efficacy and self-concept beliefs in mathematical problem solving: a path analysis. *Journal of Educational Psychology, 86*, 193-203.

- 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

- Female (sex variable: 0 = male; 1 = 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)

MULTIVARIATE LINEAR MODELS VIA PATH ANALYSIS SOFTWARE AND PACKAGES

Multivariate Regression

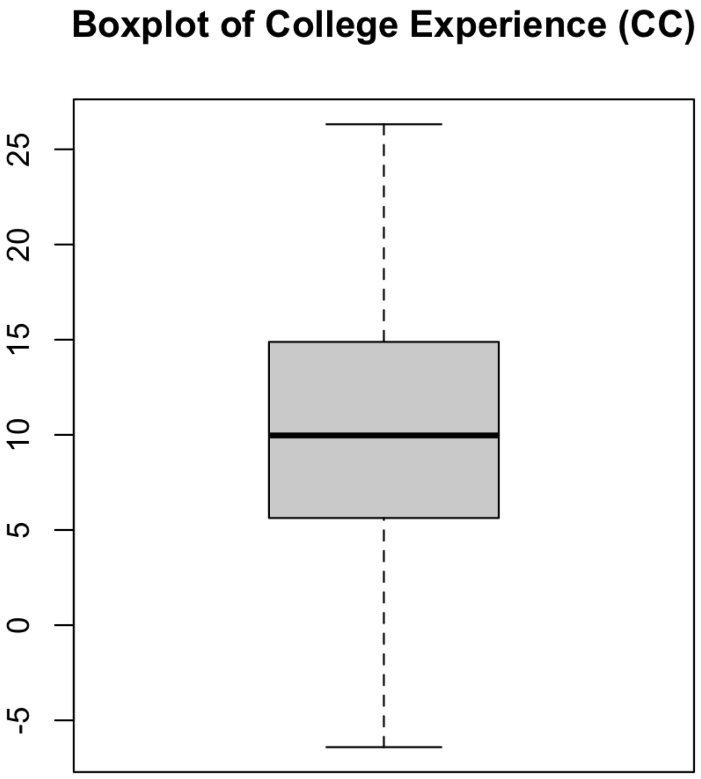
- We begin with a multivariate regression model:
    - Predicting mathematics performance (PERF) with female (F), college math experience (CC), and the interaction between female and college math experience (FxCC)
    - Predicting perceived usefulness (USE) with female (F), college math experience (CC), and the interaction between female and college math experience (FxCC)

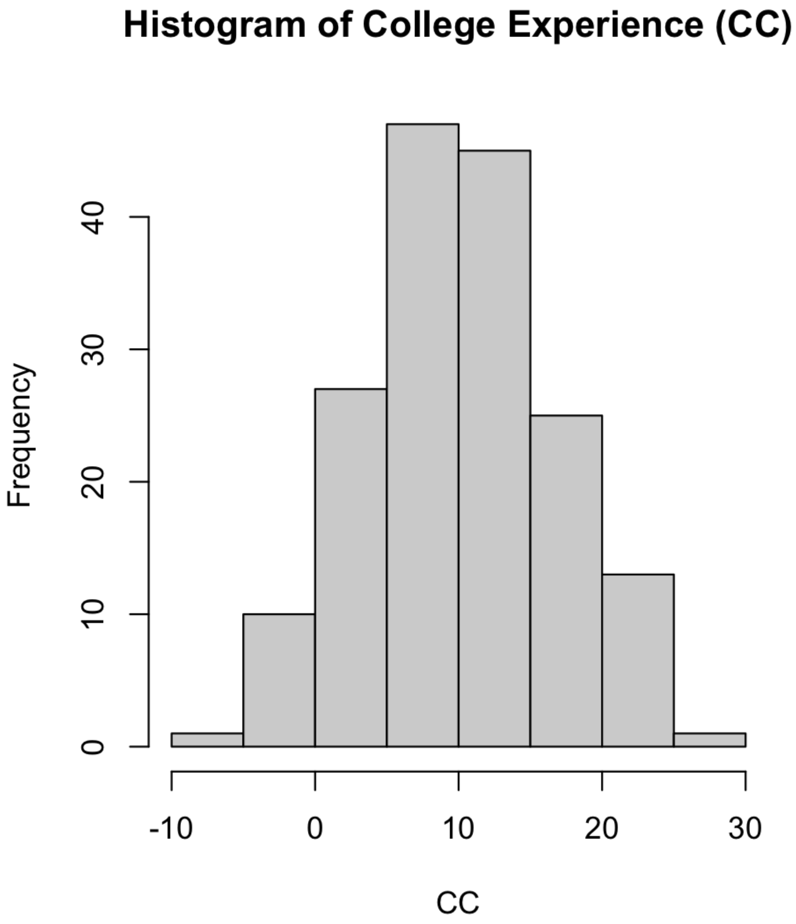
$$PERF_i = \beta_{0,PERF} + \beta_{F,PERF}F_i + \beta_{CC,PERF}CC_i + \beta_{F*CC,PERF}F_iCC_i + e_{i,PERF}$$
$$USE_i = \beta_{0,USE} + \beta_{F,USE}F_i + \beta_{CC,USE}CC_i + \beta_{F*CC,USE}F_iCC_i + e_{i,USE}$$

- We denote the residual for PERF as $e_{i,PERF}$ and the residual for USE as $e_{i,USE}$
    - We also assume the residuals are Multivariate Normal:

$$\begin{bmatrix} e_{i,PERF} \\ e_{i,USE} \end{bmatrix} \sim N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma^2_{e,PERF} & \sigma_{e,PERF,USE} \\ \sigma_{e,PERF,USE} & \sigma^2_{e,USE} \end{bmatrix}\right)$$

Before Continuing: We will Center CC at 10


Boxplot of College Experience (CC)


Histogram of College Experience (CC)

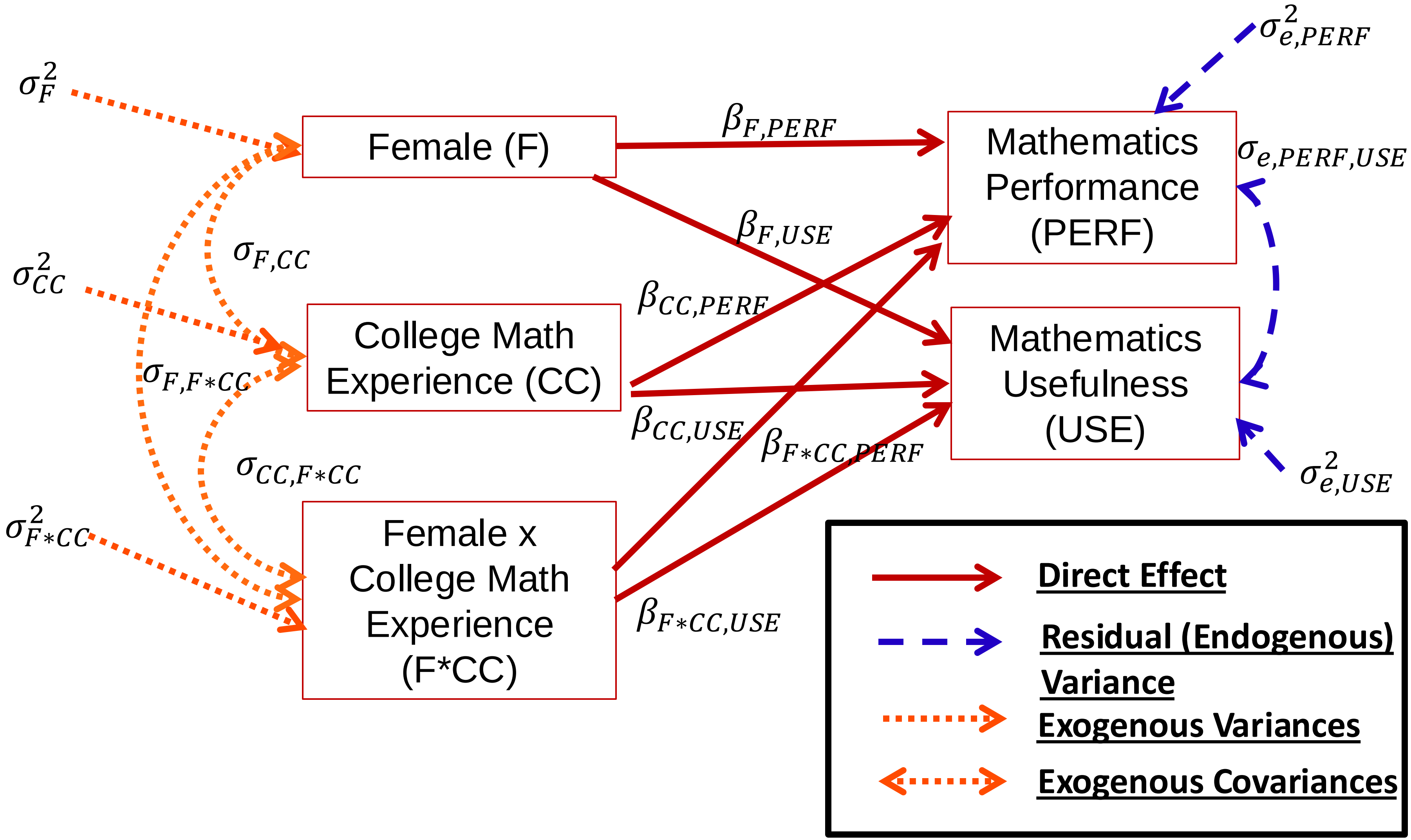
Types of Variables in the Analysis

- An important distinction in path analysis is between endogenous and exogenous variables

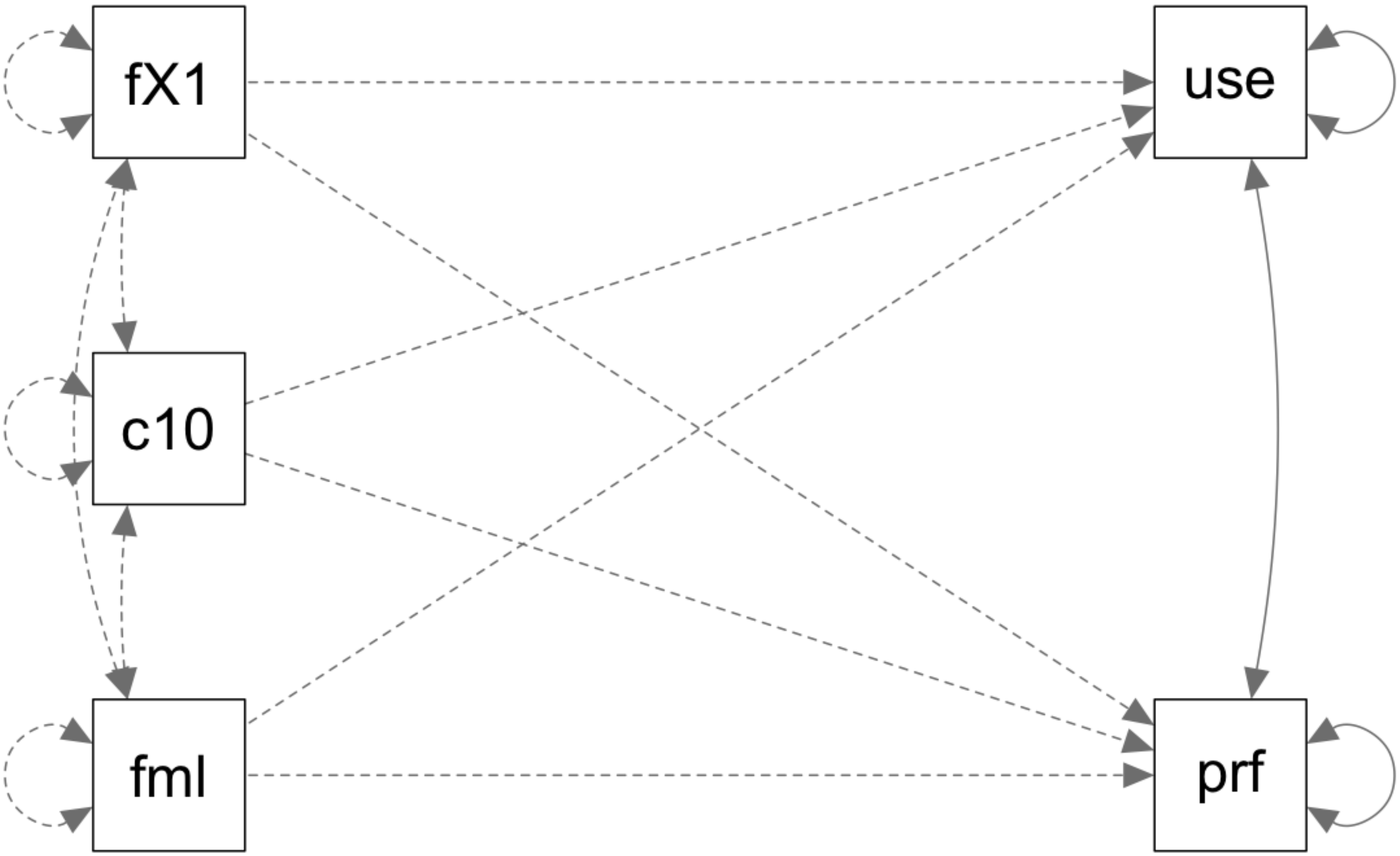
- Endogenous variable(s): variables whose variability ***is explained*** by one or more variables in a model
  - In our example Mathematics Performance (PERF) and Mathematics Usefulness (USE)
  - In univariate linear regression, the **dependent variable** is the only endogenous variable in an analysis

- Exogenous variable(s): variables whose variability ***is not explained*** by any variables in a model
  - In our example Female (F), college experience (CC), and the interaction (FxCC)
  - In linear regression, the **independent variable**(s) are the exogenous variables in the analysis

Multivariate Linear Regression Path Diagram



R’s Version of the Path Diagram

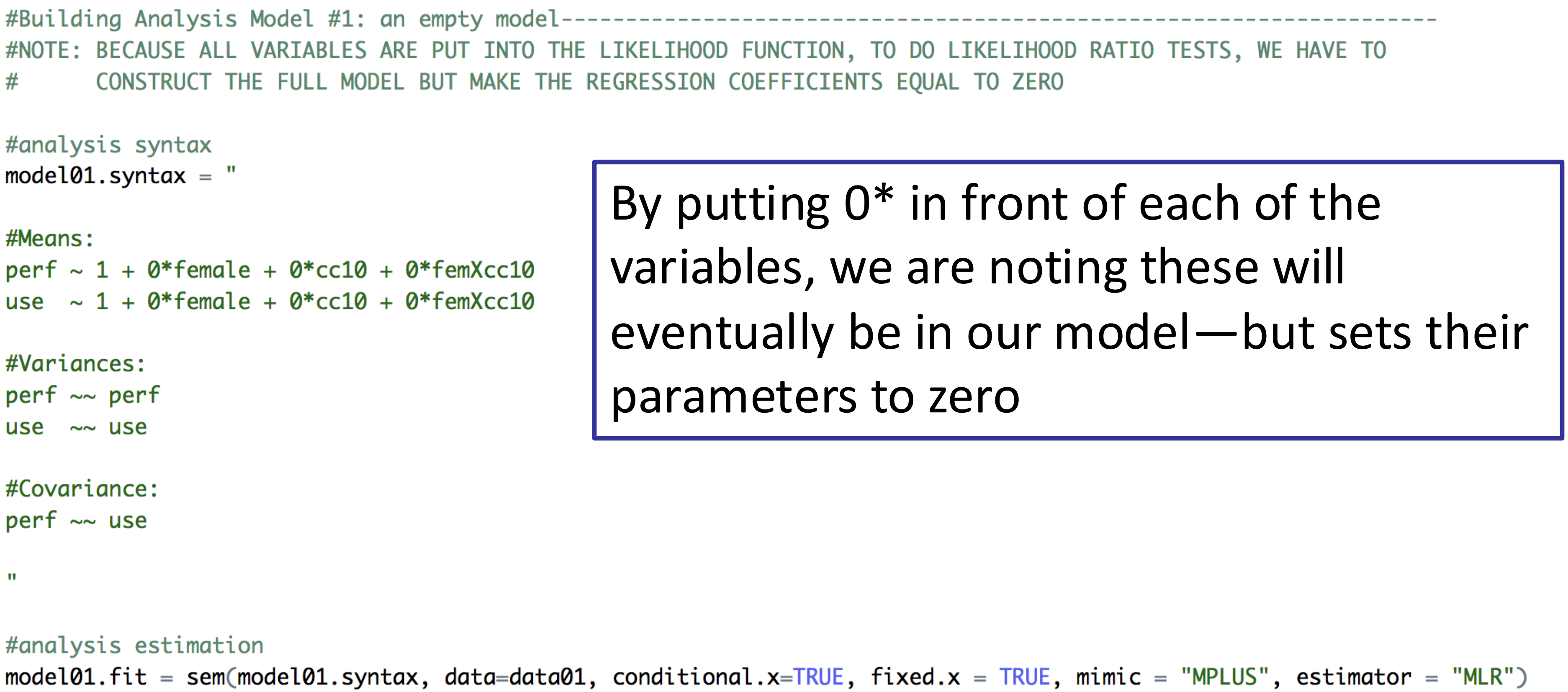


Labeling Variables

- The endogenous (dependent) variables are:
  - Performance (PERF) and Usefulness (USE)

- The exogenous (independent) variables are:
  - Female (F), college experience (CC), and the interaction of Female and college experience (F*CC)

Multivariate Regression in R Using the lavaan Package



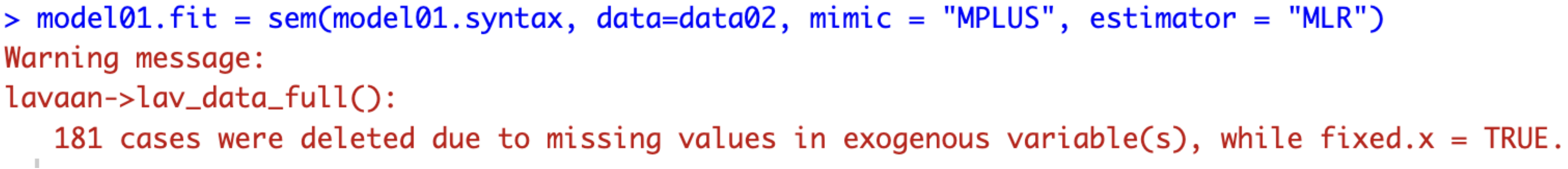
```
#Building Analysis Model #1: an empty model--------------------------------------------------------------------
#NOTE: BECAUSE ALL VARIABLES ARE PUT INTO THE LIKELIHOOD FUNCTION, TO DO LIKELIHOOD RATIO TESTS, WE HAVE TO
#      CONSTRUCT THE FULL MODEL BUT MAKE THE REGRESSION COEFFICIENTS EQUAL TO ZERO

#analysis syntax
model01.syntax = "

#Means:
perf ~ 1 + 0*female + 0*cc10 + 0*femXcc10
use  ~ 1 + 0*female + 0*cc10 + 0*femXcc10

#Variances:
perf ~~ perf
use  ~~ use

#Covariance:
perf ~~ use

"

#analysis estimation
model01.fit = sem(model01.syntax, data=data01, conditional.x=TRUE, fixed.x = TRUE, mimic = "MPLUS", estimator = "MLR")
```

By putting 0* in front of each of the variables, we are noting these will eventually be in our model—but sets their parameters to zero

- **<u>A note about path analysis software:</u>**
  - Most packages put all variables into the likelihood function (Mplus does not)
  - So, you must start with all variables in the model for LRTs

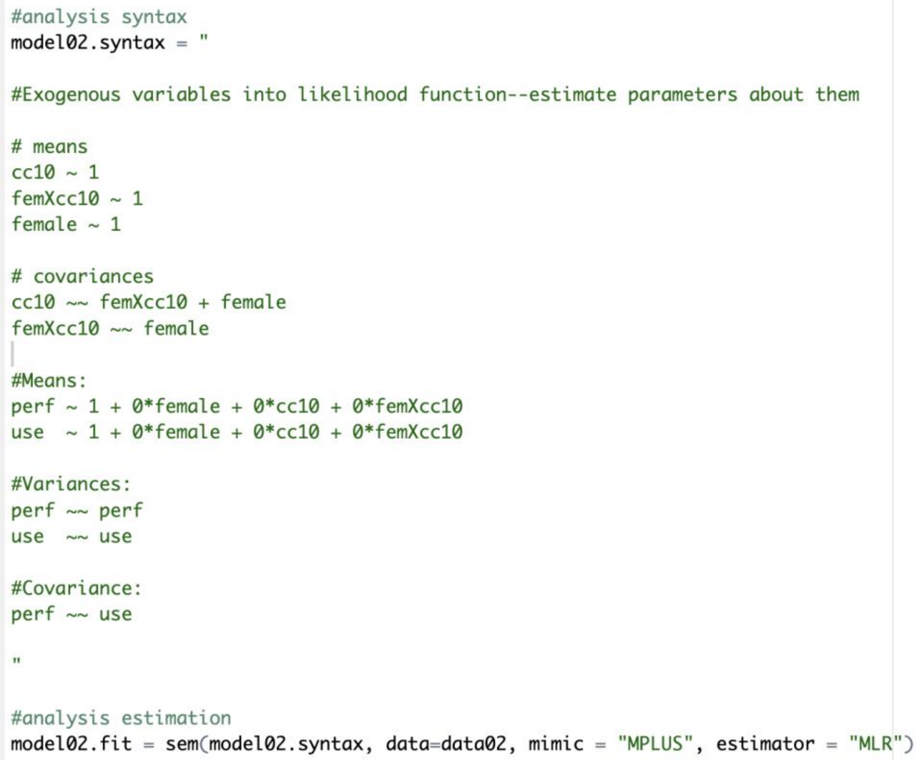
An Issue: The Missing Data

- Upon trying to run the initial empty model, we are told the following



```
> model01.fit = sem(model01.syntax, data=data02, mimic = "MPLUS", estimator = "MLR")
Warning message:
lavaan->lav_data_full():
   181 cases were deleted due to missing values in exogenous variable(s), while fixed.x = TRUE.
```

- This is a remedy:



```
#analysis syntax
model02.syntax = "

#Exogenous variables into likelihood function--estimate parameters about them

# means
cc10 ~ 1
femXcc10 ~ 1
female ~ 1

# covariances
cc10 ~~ femXcc10 + female
femXcc10 ~~ female

#Means:
perf ~ 1 + 0*female + 0*cc10 + 0*femXcc10
use  ~ 1 + 0*female + 0*cc10 + 0*femXcc10

#Variances:
perf ~~ perf
use  ~~ use

#Covariance:
perf ~~ use

"

#analysis estimation
model02.fit = sem(model02.syntax, data=data02, mimic = "MPLUS", estimator = "MLR")
```

Multivariate Regression Model Parameters

- Lavaan considers all five variables to be part of a multivariate normal distribution, so the unstructured (saturated) model has a total of 20 parameters:
  - 5 means
  - 5 variances
  - 10 covariances (5-choose-2 or 5*(5-1)/2))

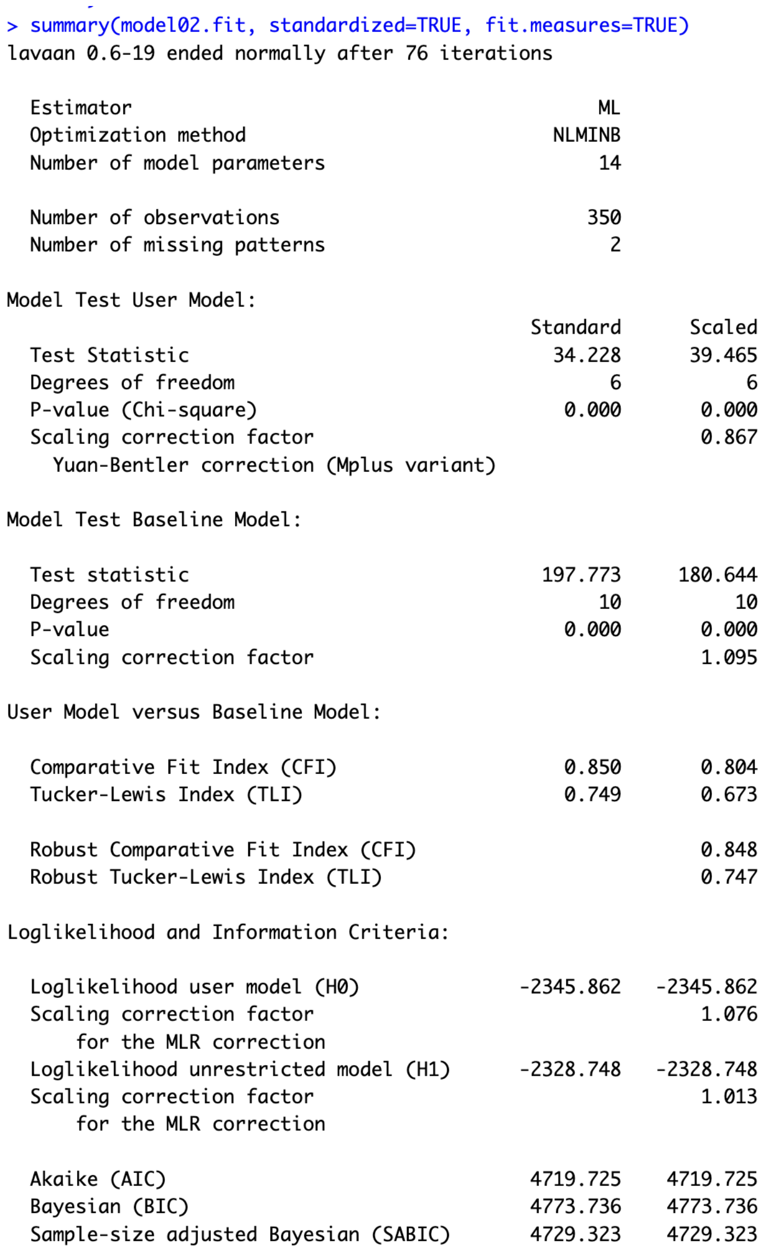
- The model itself has 11 parameters:
  - 5 intercepts
  - 0 regression slopes (but we'll add these next)
  - 2 residual variances
  - 1 residual covariance
  - 3 exogenous variances
  - 3 exogenous covariances

- Lavaan will estimate two models for each analysis: H0 (your model) and H1 (saturated model)

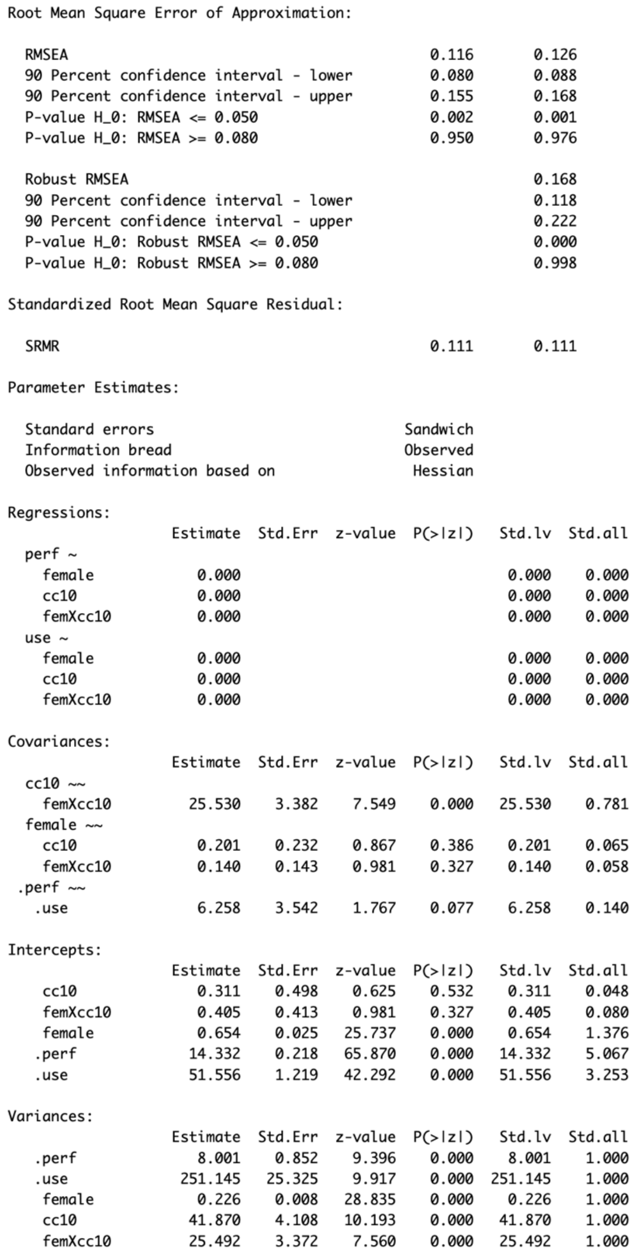
- Degrees of DF in path models come from comparing the saturated model number of parameters with the parameters estimated
  - Parameters available 20 – 14 parameters estimated = 6 df

- Therefore, this model will not fit perfectly – model fit statistics will be available

Output from Lavaan: Summary Statement

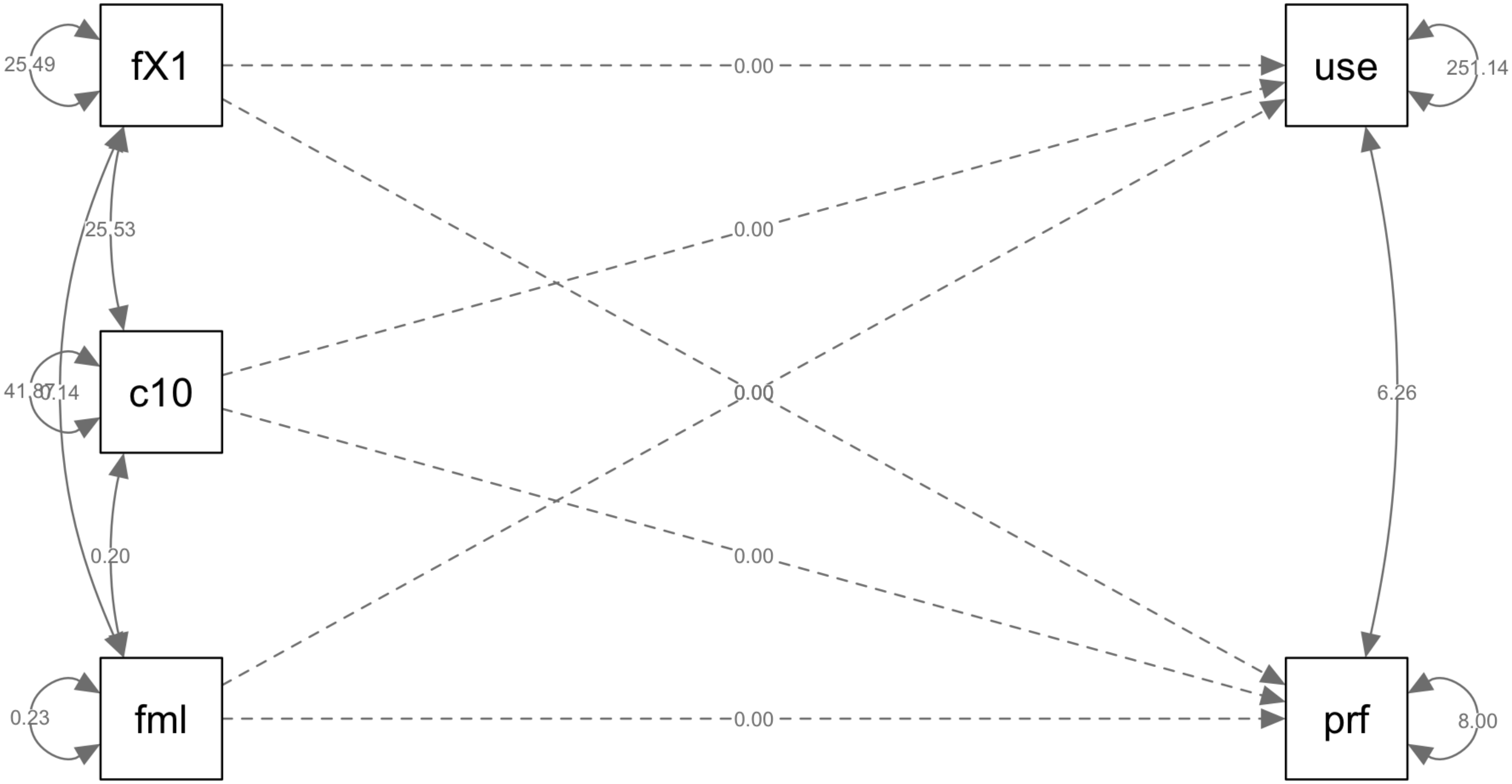


```
> summary(model02.fit, standardized=TRUE, fit.measures=TRUE)
lavaan 0.6-19 ended normally after 76 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        14

  Number of observations                           350
  Number of missing patterns                         2

Model Test User Model:
                                              Standard      Scaled
  Test Statistic                                34.228      39.465
  Degrees of freedom                                 6           6
  P-value (Chi-square)                           0.000       0.000
  Scaling correction factor                                  0.867
    Yuan-Bentler correction (Mplus variant)

Model Test Baseline Model:

  Test statistic                               197.773     180.644
  Degrees of freedom                                10          10
  P-value                                        0.000       0.000
  Scaling correction factor                                  1.095

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.850       0.804
  Tucker-Lewis Index (TLI)                       0.749       0.673

  Robust Comparative Fit Index (CFI)                         0.848
  Robust Tucker-Lewis Index (TLI)                            0.747

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -2345.862   -2345.862
  Scaling correction factor                                  1.076
      for the MLR correction
  Loglikelihood unrestricted model (H1)      -2328.748   -2328.748
  Scaling correction factor                                  1.013
      for the MLR correction

  Akaike (AIC)                                4719.725    4719.725
  Bayesian (BIC)                              4773.736    4773.736
  Sample-size adjusted Bayesian (SABIC)       4729.323    4729.323
```

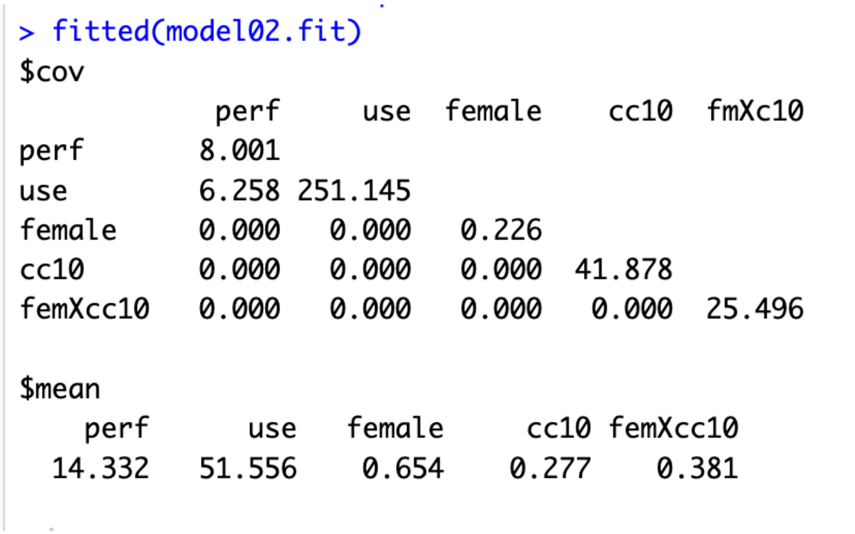


```
Root Mean Square Error of Approximation:

  RMSEA                                          0.116       0.126
  90 Percent confidence interval - lower         0.080       0.088
  90 Percent confidence interval - upper         0.155       0.168
  P-value H_0: RMSEA <= 0.050                    0.002       0.001
  P-value H_0: RMSEA >= 0.080                    0.950       0.976

  Robust RMSEA                                               0.168
  90 Percent confidence interval - lower                     0.118
  90 Percent confidence interval - upper                     0.222
  P-value H_0: Robust RMSEA <= 0.050                         0.000
  P-value H_0: Robust RMSEA >= 0.080                         0.998

Standardized Root Mean Square Residual:

  SRMR                                           0.111       0.111

Parameter Estimates:

  Standard errors                             Sandwich
  Information bread                           Observed
  Observed information based on                Hessian

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  perf ~
    female            0.000                               0.000    0.000
    cc10              0.000                               0.000    0.000
    femXcc10          0.000                               0.000    0.000
  use ~
    female            0.000                               0.000    0.000
    cc10              0.000                               0.000    0.000
    femXcc10          0.000                               0.000    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  cc10 ~~
    femXcc10         25.530    3.382    7.549    0.000   25.530    0.781
  female ~~
    cc10              0.201    0.232    0.867    0.386    0.201    0.065
    femXcc10          0.140    0.143    0.981    0.327    0.140    0.058
 .perf ~~
   .use               6.258    3.542    1.767    0.077    6.258    0.140

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
    cc10              0.311    0.498    0.625    0.532    0.311    0.048
    femXcc10          0.405    0.413    0.981    0.327    0.405    0.080
    female            0.654    0.025   25.737    0.000    0.654    1.376
   .perf             14.332    0.218   65.870    0.000   14.332    5.067
   .use              51.556    1.219   42.292    0.000   51.556    3.253

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
   .perf              8.001    0.852    9.396    0.000    8.001    1.000
   .use             251.145   25.325    9.917    0.000  251.145    1.000
    female            0.226    0.008   28.835    0.000    0.226    1.000
    cc10             41.870    4.108   10.193    0.000   41.870    1.000
    femXcc10         25.492    3.372    7.560    0.000   25.492    1.000
```

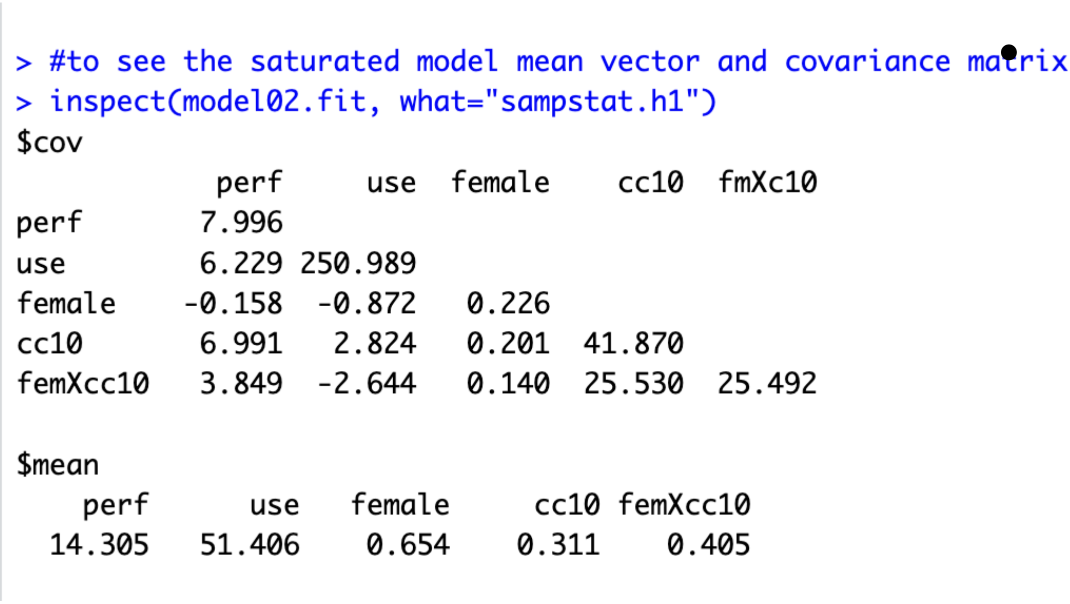
Path Diagram with Numbers Shown



Output from lavaan: “Fitted” and Saturated Covariance Matrix



```
> fitted(model02.fit)
$cov
           perf     use  female    cc10  fmXc10
perf      8.001
use       6.258 251.145
female    0.000   0.000   0.226
cc10      0.000   0.000   0.000  41.878
femXcc10  0.000   0.000   0.000   0.000  25.496

$mean
    perf      use   female     cc10 femXcc10
  14.332   51.556    0.654    0.277    0.381
```



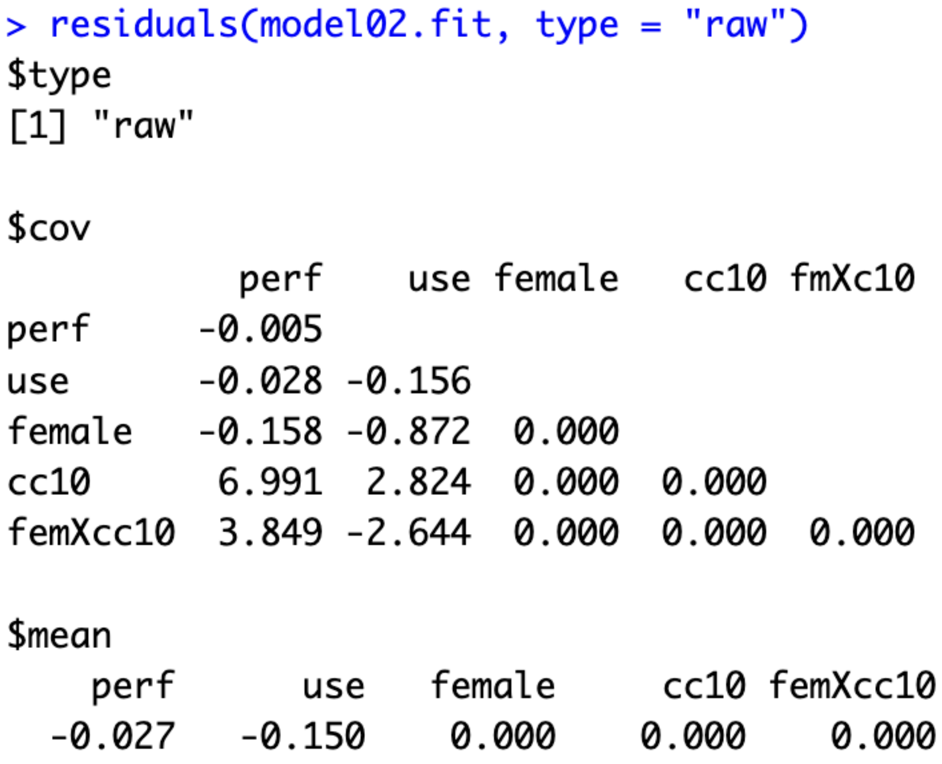
```
> #to see the saturated model mean vector and covariance matrix
> inspect(model02.fit, what="sampstat.h1")
$cov
           perf     use  female    cc10  fmXc10
perf      7.996
use       6.229 250.989
female   -0.158  -0.872   0.226
cc10      6.991   2.824   0.201  41.870
femXcc10  3.849  -2.644   0.140  25.530  25.492

$mean
    perf      use   female     cc10 femXcc10
  14.305   51.406    0.654    0.311    0.405
```

- The fitted covariance matrix shows you what the model implies the variances and covariances should be

- Model parameters provide the endogenous parameters

- The lower matrix is the saturated model matrix

Output from lavaan: Residual Covariance Matrices



```
> residuals(model02.fit, type = "raw")
$type
[1] "raw"

$cov
          perf    use female   cc10 fmXc10
perf    -0.005
use     -0.028 -0.156
female  -0.158 -0.872  0.000
cc10     6.991  2.824  0.000  0.000
femXcc10 3.849 -2.644  0.000  0.000  0.000

$mean
    perf       use    female     cc10 femXcc10
  -0.027    -0.150     0.000    0.000    0.000
```

The “raw” residuals are the difference between the model implied covariance matrix and the H1 (saturated model) covariance matrix/mean vector

METHODS OF EXAMINING MODEL FIT

Methods of Model Fit

- Model-data fit is of utmost concern when building models with multivariate outcomes

- If a model does not fit the data:
  - Parameter estimates may be biased
  - Standard errors of estimates may be biased
  - Inferences made from the model may be wrong
  - If the saturated model fit is wrong, then the LRTs will be inaccurate

- Examining model fit is the first step in multivariate models

- That said, not all “good-fitting” models are useful…
  - …model fit just allows you to talk about your model…there may be nothing of significance (statistically or practically) in your results, though

Types of Model Fit Information

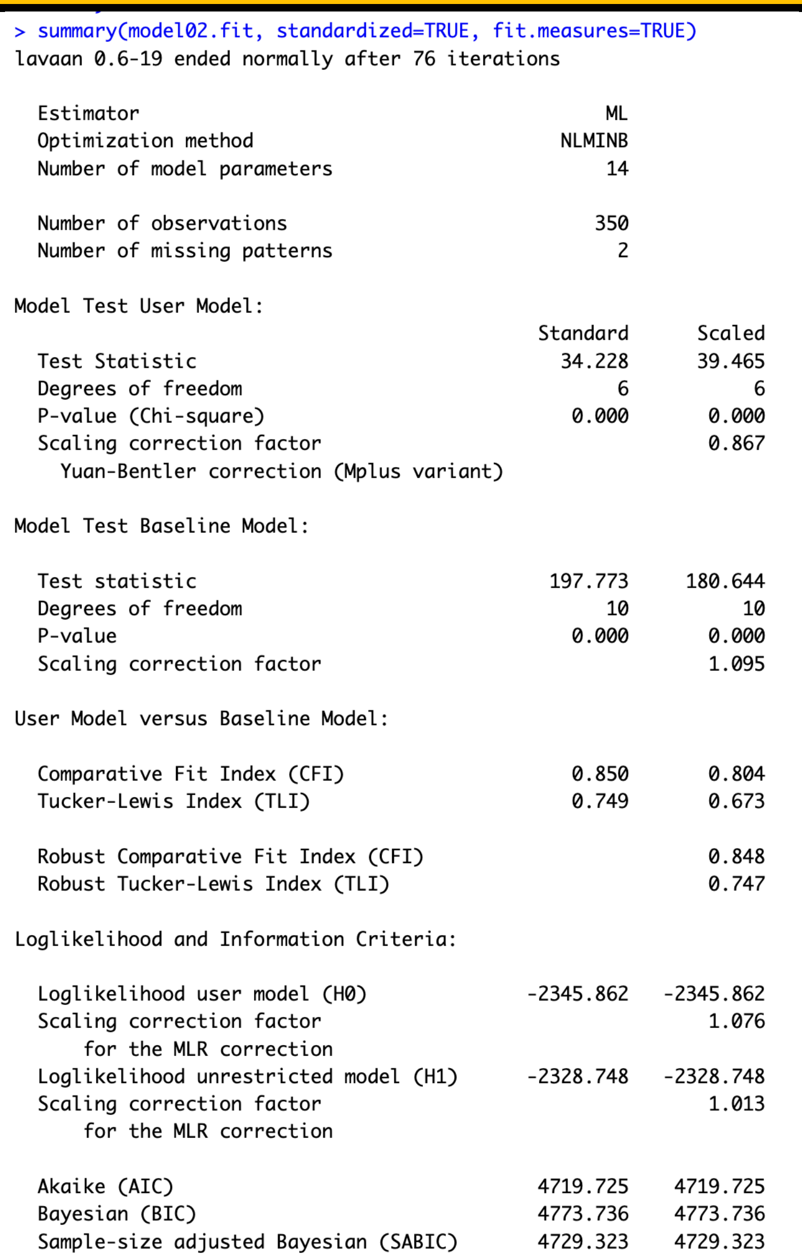
- Model fit information for models where outcomes are conditionally MVN* come in several types, but all are based on the premise that any model mean and covariance structure must fit as well as the saturated mean vector and covariance matrix model

    ***If model outcomes are not conditionally MVN, model fit is very different**

- All possible models/structures **are nested within** the saturated mean vector and covariance matrix model
    - Most model fit statistics come from comparing any model/structure with the saturated model

- Indices shown first are called “global” model fit indices
    - Report fit of model globally (as opposed to locally for specific parameters)

Example lavaan Model Fit Output



```
> summary(model02.fit, standardized=TRUE, fit.measures=TRUE)
lavaan 0.6-19 ended normally after 76 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        14

  Number of observations                           350
  Number of missing patterns                         2

Model Test User Model:
                                              Standard      Scaled
  Test Statistic                                34.228      39.465
  Degrees of freedom                                 6           6
  P-value (Chi-square)                           0.000       0.000
  Scaling correction factor                                  0.867
    Yuan-Bentler correction (Mplus variant)

Model Test Baseline Model:

  Test statistic                               197.773     180.644
  Degrees of freedom                                10          10
  P-value                                        0.000       0.000
  Scaling correction factor                                  1.095

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.850       0.804
  Tucker-Lewis Index (TLI)                       0.749       0.673

  Robust Comparative Fit Index (CFI)                         0.848
  Robust Tucker-Lewis Index (TLI)                            0.747

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -2345.862   -2345.862
  Scaling correction factor                                  1.076
      for the MLR correction
  Loglikelihood unrestricted model (H1)      -2328.748   -2328.748
  Scaling correction factor                                  1.013
      for the MLR correction

  Akaike (AIC)                                4719.725    4719.725
  Bayesian (BIC)                              4773.736    4773.736
  Sample-size adjusted Bayesian (SABIC)       4729.323    4729.323
```

The fit.measures=TRUE Model Fit Statistics

- **Unlabeled section**
  - **Likelihood ratio test versus the saturated model**
  - **Testing if your model fits as well as the saturated model**
- Model test baseline model
  - Likelihood ratio test pitting the saturated model against the independent variables model
  - Testing whether any variables have non-zero covariances (significant correlations)
- User model versus baseline model
  - CFI
  - TLI
- Loglikelihood and Information Criteria
  - Likelihood ratio tests (nested models)
  - Information criteria comparisons (non-nested models)
- Root Mean Square Error of Approximation
  - How far off a model is from the saturated model, per degree of freedom
- Standardized Root Mean Square Residual
  - How far off a model's correlations are from the saturated model correlations

Indices of Global Model Fit

- Primary: obtained model **χ^2** (from Model test baseline model) – here we use the MLR rescaled **χ^2** from the “Robust” Column
  - χ^2 is evaluated based on model df (difference in parameters between your CFA model and the saturated model)
  - Tests null hypothesis that ***this*** model (H_0) fits equally to ***saturated model*** (H_1) so significance is undesirable (smaller χ^2, bigger p-value is better)
    - Means saturated model is estimated **automatically** for each model analyzed
  - Just using χ^2 is insufficient, however:
    - Distribution doesn’t behave like a true χ^2 if sample sizes are small (or, if not using MLR, if items are non-normally distributed)
    - Obtained χ^2 depends largely on sample size
    - Some mention this is an unreasonable null hypothesis (perfect fit??)

- Because of these issues, alternative measures of fit are usually used in conjunction with the χ^2 test of model fit
  - Absolute Fit Indices (besides χ^2)
  - Parsimony-Corrected; Comparative (Incremental) Fit Indices

Chi-Square Test of Model Fit

- The Chi-Square Test of Model Fit provides a likelihood ratio test comparing the current model to the **saturated (unstructured) model**:
  - The value is -2 times the difference in log-likelihoods (rescaled if MLR)
  - The degrees of freedom is the difference in the number of estimated model parameters
  - The p-value is from the Chi-square distribution
- **If this test has a significant p-value:**
  - The current model (H_0) is rejected – the model fit is significantly worse than the full model
  - In latent variable models, this test is usually ignored
    - Said to be overly sensitive
- **If this test does not have a significant p-value:**
  - The current model (H_0) is not rejected – **fits equivalently to full model**

Where the Saturated Model Test Comes From

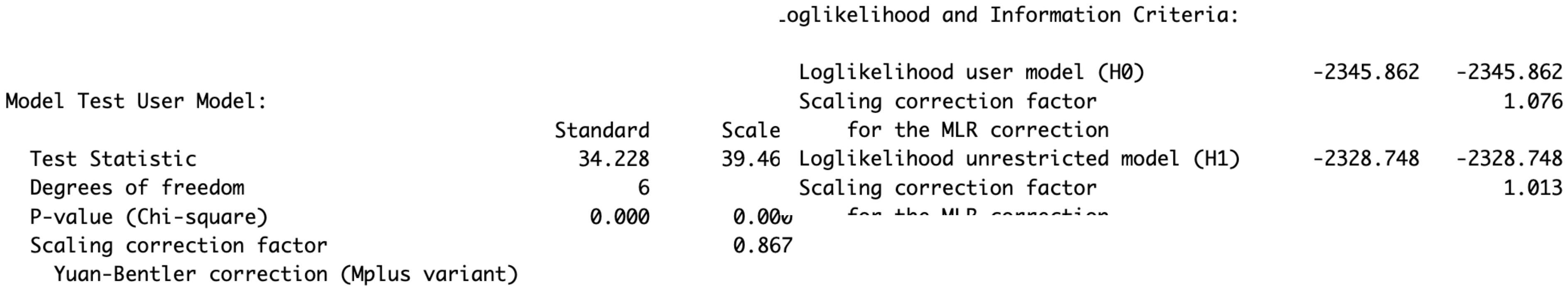
- The saturated model LRT comes from a likelihood ratio test of the current model with the saturated model

- If using MLR (Robust method), then this LRT is rescaled based on the estimated scaling factors of both models

- This same information can be obtained from:
  - Loglikelihood model output section
  - anova() function comparing fit for current and saturated models

Calculating the LRT for Global Fit Test

- From the lavaan output:

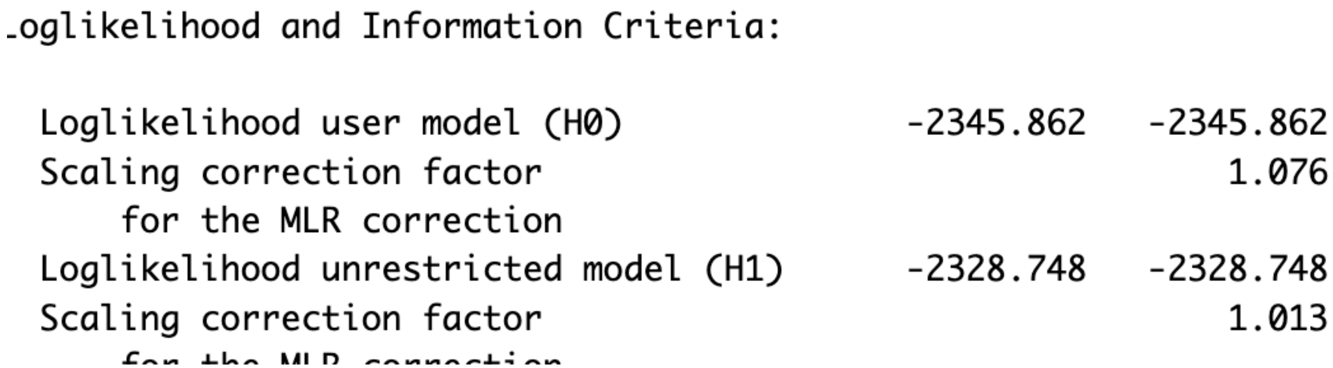


```
Model Test User Model:
                                               Standard      Scale
  Test Statistic                                 34.228      39.46
  Degrees of freedom                                  6
  P-value (Chi-square)                            0.000       0.000
  Scaling correction factor                                   0.867
    Yuan-Bentler correction (Mplus variant)
```

```
.oglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -2345.862   -2345.862
  Scaling correction factor                                  1.076
      for the MLR correction
  Loglikelihood unrestricted model (H1)      -2328.748   -2328.748
  Scaling correction factor                                  1.013
      for the MLR correction
```

- Conclusion: this model fit significantly worse than the saturated model
  - And it should—especially if any of our predictors have non-zero betas

Saturated Model LRT and Loglikelihood Output



```
.oglikelihood and Information Criteria:

   Loglikelihood user model (H0)             -2345.862   -2345.862
   Scaling correction factor                               1.076
       for the MLR correction
   Loglikelihood unrestricted model (H1)     -2328.748   -2328.748
   Scaling correction factor                               1.013
       for the MLR correction
```

- If the loglikelihoods of the current model (“User model” or H_0) are equal to the loglikelihoods of the saturated model (“Unrestriced model” or H_1), then you are running a model that is equivalent to the saturated model
  - ➢ **<u>No other model fit will be available or useful</u>**

The fit.measures=TRUE Model Fit Statistics

- ~~Unlabeled section~~
  - ~~Likelihood ratio test versus the saturated model~~
  - ~~Testing if your model fits as well as the saturated model~~

- **Model test baseline model**
  - **Likelihood ratio test pitting the saturated model against the independent variables model**
  - **Testing whether any variables have non-zero covariances (significant correlations)**

- User model versus baseline model
  - CFI
  - TLI

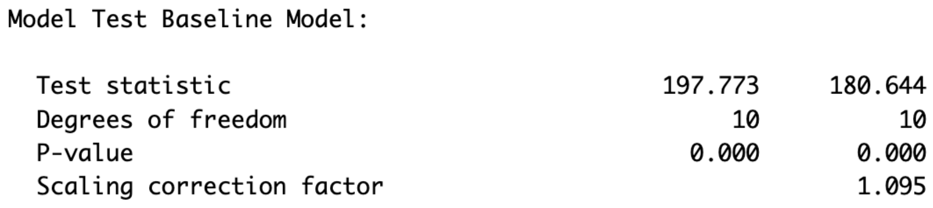
- Loglikelihood and Information Criteria
  - Likelihood ratio tests (nested models)
  - Information criteria comparisons (non-nested models)

- Root Mean Square Error of Approximation
  - How far off a model is from the saturated model, per degree of freedom

- Standardized Root Mean Square Residual
  - How far off a model's correlations are from the saturated model correlations

Model Test Baseline Model

- The “model test baseline model” section provides a LRT:
  - Comparing the saturated (unstructured) model with an independent variables model (called the baseline model)



```
Model Test Baseline Model:

  Test statistic                               197.773     180.644
  Degrees of freedom                                10          10
  P-value                                        0.000       0.000
  Scaling correction factor                                  1.095
```

- Here, the “null” model is the baseline (the independent variables model)
  - If the test is significant, this means that at least one (and likely more than one) variable has a significant covariance (and correlation)

  - If the test is not significant, this means that the independence model is appropriate
    - This is not likely to happen
    - But if it does, there are virtually no other models that will be significant

- Not often reported as it is likely variables are correlated

The fit.measures=TRUE Model Fit Statistics

- ~~Unlabeled section~~
  - ~~Likelihood ratio test versus the saturated model~~
  - ~~Testing if your model fits as well as the saturated model~~

- ~~Model test baseline model~~
  - ~~Likelihood ratio test pitting the saturated model against the independent variables model~~
  - ~~Testing whether any variables have non-zero covariances (significant correlations)~~

- **User model versus baseline model**
  - **CFI**
  - **TLI**

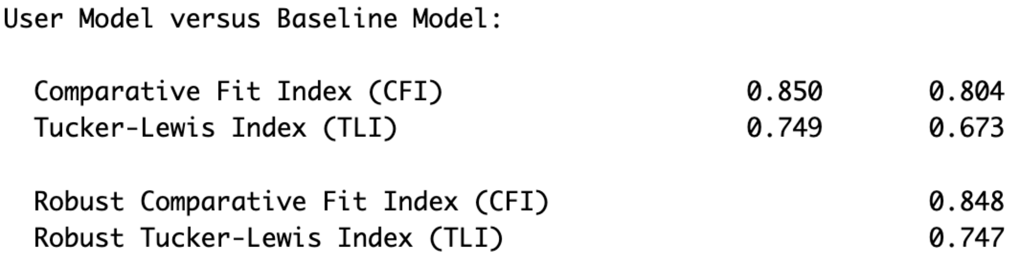
- Loglikelihood and Information Criteria
  - Likelihood ratio tests (nested models)
  - Information criteria comparisons (non-nested models)

- Root Mean Square Error of Approximation
  - How far off a model is from the saturated model, per degree of freedom

- Standardized Root Mean Square Residual
  - How far off a model’s correlations are from the saturated model correlations

User Model Versus Baseline Model Section

- The “User model versus baseline model” section provides two additional measures of model fit comparing the current (user) model to the baseline (independent variables) model



```
User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    0.850       0.804
  Tucker-Lewis Index (TLI)                       0.749       0.673

  Robust Comparative Fit Index (CFI)                         0.848
  Robust Tucker-Lewis Index (TLI)                            0.747
```

- CFI stands for Comparative Fit Index
  - Higher is better (above .95 indicates good fit)

- TLI stands for Tucker Lewis Index
  - Higher is better (above .95 indicates good fit)

Comparative (Incremental) Fit Indices

- Fit evaluated relative to a 'null' model (of 0 covariances)
  - Relative to that, your model should be great!

T = target (current/estimated) model
N = null (baseline/independent variables) model

- **CFI: Comparative Fit Index**
  - Based on idea of the chi-square non-centrality parameter: (χ^2 – df)
  - $CFI = 1 - \frac{\max(\chi^2_T - df_T, 0)}{\max(\chi^2_T - df_T, \chi^2_N - df_N, 0)}$
  - From 0 to 1: bigger is better, > .90 = "acceptable", > .95 = "good"

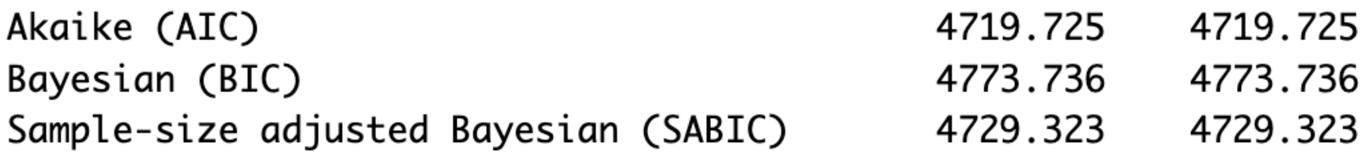
- **TLI: Tucker-Lewis Index** (= **Non-Normed Fit Index**)
  - $TLI = \frac{\frac{\chi^2_N}{df_N} - \frac{\chi^2_T}{df_T}}{\frac{\chi^2_N}{df_N} - 1}$
  - From <0 to >1, bigger is better, >.95 = "good"

The fit.measures=TRUE Model Fit Statistics

- ~~Unlabeled section~~
  - ~~Likelihood ratio test versus the saturated model~~
  - ~~Testing if your model fits as well as the saturated model~~
- ~~Model test baseline model~~
  - ~~Likelihood ratio test pitting the saturated model against the independent variables model~~
  - ~~Testing whether any variables have non-zero covariances (significant correlations)~~
- ~~User model versus baseline model~~
  - ~~CFI~~
  - ~~TLI~~
- **Loglikelihood and Information Criteria**
  - **Likelihood ratio tests (nested models)**
  - **Information criteria comparisons (non-nested models)**
- Root Mean Square Error of Approximation
  - How far off a model is from the saturated model, per degree of freedom
- Standardized Root Mean Square Residual
  - How far off a model's correlations are from the saturated model correlations

Comparing Information Criteria

- Information criteria are relative tests of fit



```
Akaike (AIC)                              4719.725    4719.725
Bayesian (BIC)                            4773.736    4773.736
Sample-size adjusted Bayesian (SABIC)     4729.323    4729.323
```

- The are calculated based on the log-likelihood of the model, factoring in a penalty for number of parameters (plus other things)

- They should never be used to compare nested models
  - The likelihood ratio test is the most powerful test statistic to use for nested models

- When comparing non-nested models, first choose a statistic
  - AIC, BIC, or Sample-size Adjusted BIC are what are given by default

- The preferred model is the one with the lowest value of that statistic

The fit.measures=TRUE Model Fit Statistics

- ~~Unlabeled section~~
  - ~~Likelihood ratio test versus the saturated model~~
  - ~~Testing if your model fits as well as the saturated model~~

- ~~Model test baseline model~~
  - ~~Likelihood ratio test pitting the saturated model against the independent variables model~~
  - ~~Testing whether any variables have non-zero covariances (significant correlations)~~

- ~~User model versus baseline model~~
  - ~~CFI~~
  - ~~TLI~~

- ~~Loglikelihood and Information Criteria~~
  - ~~Likelihood ratio tests (nested models)~~
  - ~~Information criteria comparisons (non-nested models)~~

- **Root Mean Square Error of Approximation**
  - **How far off a model is from the saturated model, per degree of freedom**

- Standardized Root Mean Square Residual
  - How far off a model's correlations are from the saturated model correlations

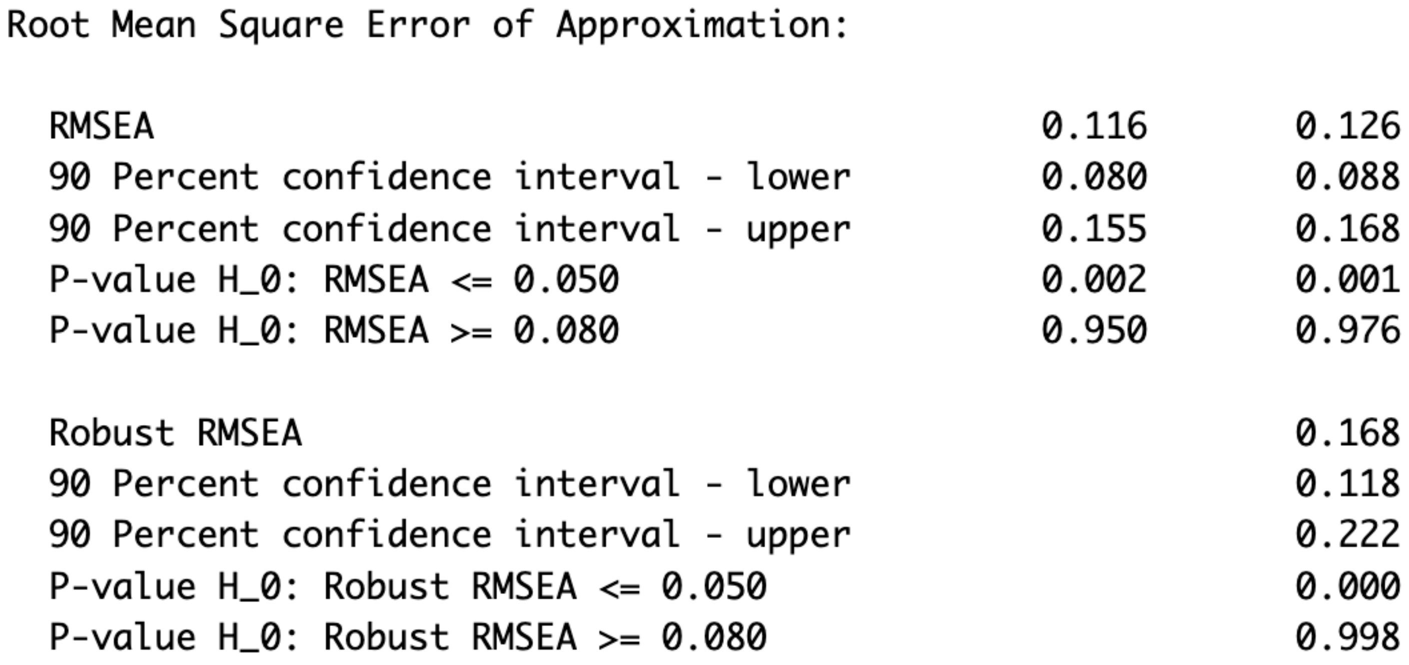
Indices of Global Model Fit

Parsimony-Corrected: **RMSEA**

- **Root Mean Square Error of Approximation**
- Uses comparison with CFA model and saturated model
  - χ^2 listed here from first part of lavaan output
- Relies on a non-centrality parameter (NCP)
  - Indexes how far off your model is → χ^2 distribution shoved over
  - NCP → $d = (\chi^2 – df) / (N-1)$ Then, $RMSEA = SQRT(d/df)$
    - df is difference between # parameters in CFA model and saturated model
  - RMSEA ranges from 0 to 1; smaller is better
    - < .05 or .06 = “good”, .05 to .08 = “acceptable”, .08 to .10 = “mediocre”, and >.10 = “unacceptable”
  - In addition to point estimate, get 90% confidence interval
  - RMSEA penalizes for model complexity – it’s discrepancy in fit per df left in model (but not sensitive to N, although CI can be)
  - Test of “close fit”: null hypothesis that RMSEA ≤ .05

RMSEA (Root Mean Square Error of Approximation)

- The RMSEA is an index of model fit where 0 indicates perfect fit (smaller is better):



```
Root Mean Square Error of Approximation:

  RMSEA                                          0.116       0.126
  90 Percent confidence interval - lower         0.080       0.088
  90 Percent confidence interval - upper         0.155       0.168
  P-value H_0: RMSEA <= 0.050                    0.002       0.001
  P-value H_0: RMSEA >= 0.080                    0.950       0.976

  Robust RMSEA                                               0.168
  90 Percent confidence interval - lower                     0.118
  90 Percent confidence interval - upper                     0.222
  P-value H_0: Robust RMSEA <= 0.050                         0.000
  P-value H_0: Robust RMSEA >= 0.080                         0.998
```

- The goal is a model with an RMSEA less than .05
  - Although there is some flexibility

- The result above indicates our model fits poorly (RMSEA of .0088)

The fit.measures=TRUE Model Fit Statistics

- ~~Unlabeled section~~
  - ~~Likelihood ratio test versus the saturated model~~
  - ~~Testing if your model fits as well as the saturated model~~

- ~~Model test baseline model~~
  - ~~Likelihood ratio test pitting the saturated model against the independent variables model~~
  - ~~Testing whether any variables have non-zero covariances (significant correlations)~~

- ~~User model versus baseline model~~
  - ~~CFI~~
  - ~~TLI~~

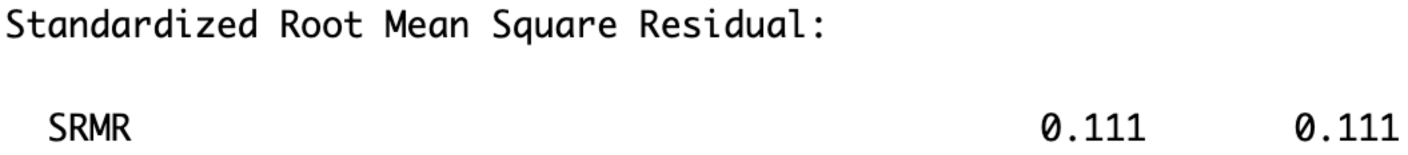
- ~~Loglikelihood and Information Criteria~~
  - ~~Likelihood ratio tests (nested models)~~
  - ~~Information criteria comparisons (non-nested models)~~

- ~~Root Mean Square Error of Approximation~~
  - ~~How far off a model is from the saturated model, per degree of freedom~~

- **Standardized Root Mean Square Residual**
  - **How far off a model's correlations are from the saturated model correlations**

Standardized Root Mean Squared Residual

- The SRMR (standardized root mean square residual) provides the average standardized difference between:
  - The estimated covariance matrix of the saturated model
  - The estimated covariance matrix of the current model



```
Standardized Root Mean Square Residual:

  SRMR                                   0.111       0.111
```

- Lower is better (some suggest less than 0.08)

LOCAL MODEL FIT MEASURES

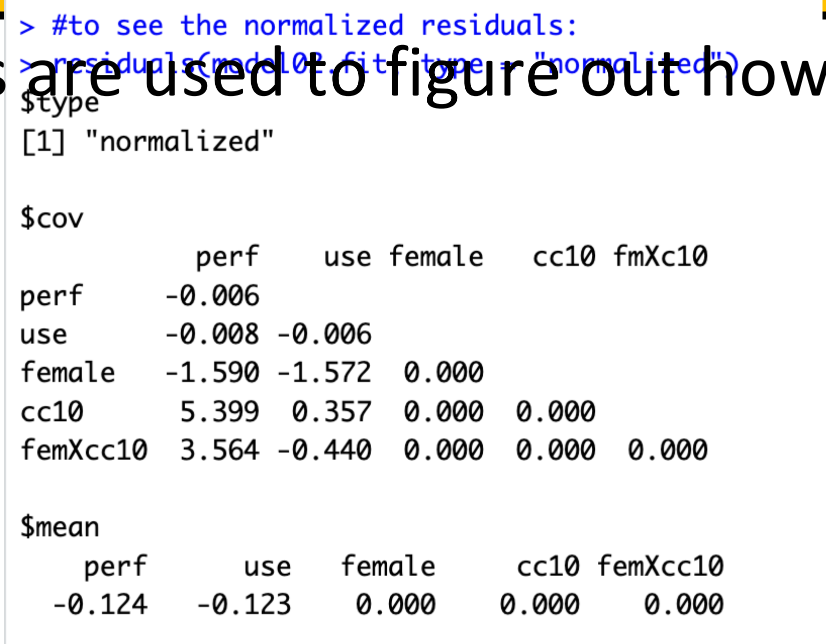
"Local" Measures of Model (Mis)Fit

- Local measures of model (mis)fit are statistics that point to the location (typically of a covariance matrix) where a model may not fit well
  - As opposed to "global" measures that indicate a model fit overall

- Local measures of model (mis)fit are typically of two types:
  - Residual covariance matrices (unstandardized, standardized, or normalized)
    - The difference between the model's estimated covariance matrix and the saturated model's estimated covariance matrix
    - These were used for the SRMR
  - Model "modification indices"
    - 1-degree of freedom hypothesis tests for the improvement of the model LRT if one more parameter was allowed to be estimated

Residual Covariance Matrices

- Residual covariance matrices are used to figure out how to best improve model misfit



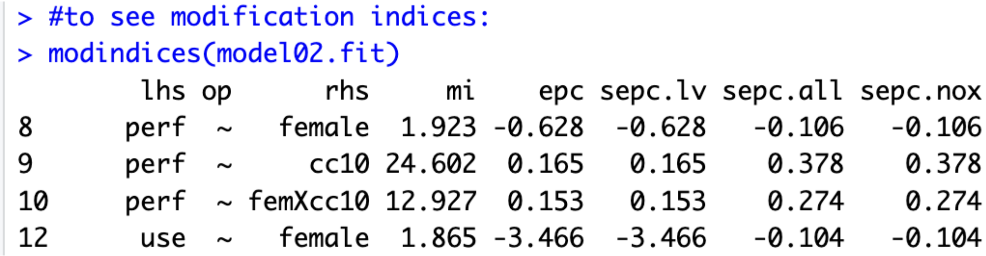
```
> #to see the normalized residuals:
> residuals(model02.fit, type = "normalized")
$type
[1] "normalized"

$cov
           perf    use female   cc10 fmXc10
perf     -0.006                            
use      -0.008 -0.006                     
female   -1.590 -1.572  0.000              
cc10      5.399  0.357  0.000  0.000       
femXcc10  3.564 -0.440  0.000  0.000  0.000

$mean
    perf      use   female     cc10 femXcc10 
  -0.124   -0.123    0.000    0.000    0.000 
```

- The “raw” or “unstandardized” residual covariance matrix for the model literally takes the difference between model implied and saturated model covariance matrices

- I often prefer “normalized” versions of these matrices
  - We can inspect the normalized residual covariance matrix (like z-scores) to see where our biggest misfit occurs

Modification Indices: More Help for Fit

- As we used Maximum Likelihood to estimate our model, another useful feature is that of the modification indices
  - Modification indices, also called Score or LaGrangian Multiplier tests, attempt to suggest the change in the log-likelihood for adding a given model parameter (larger values indicate a better fit for adding the parameter)

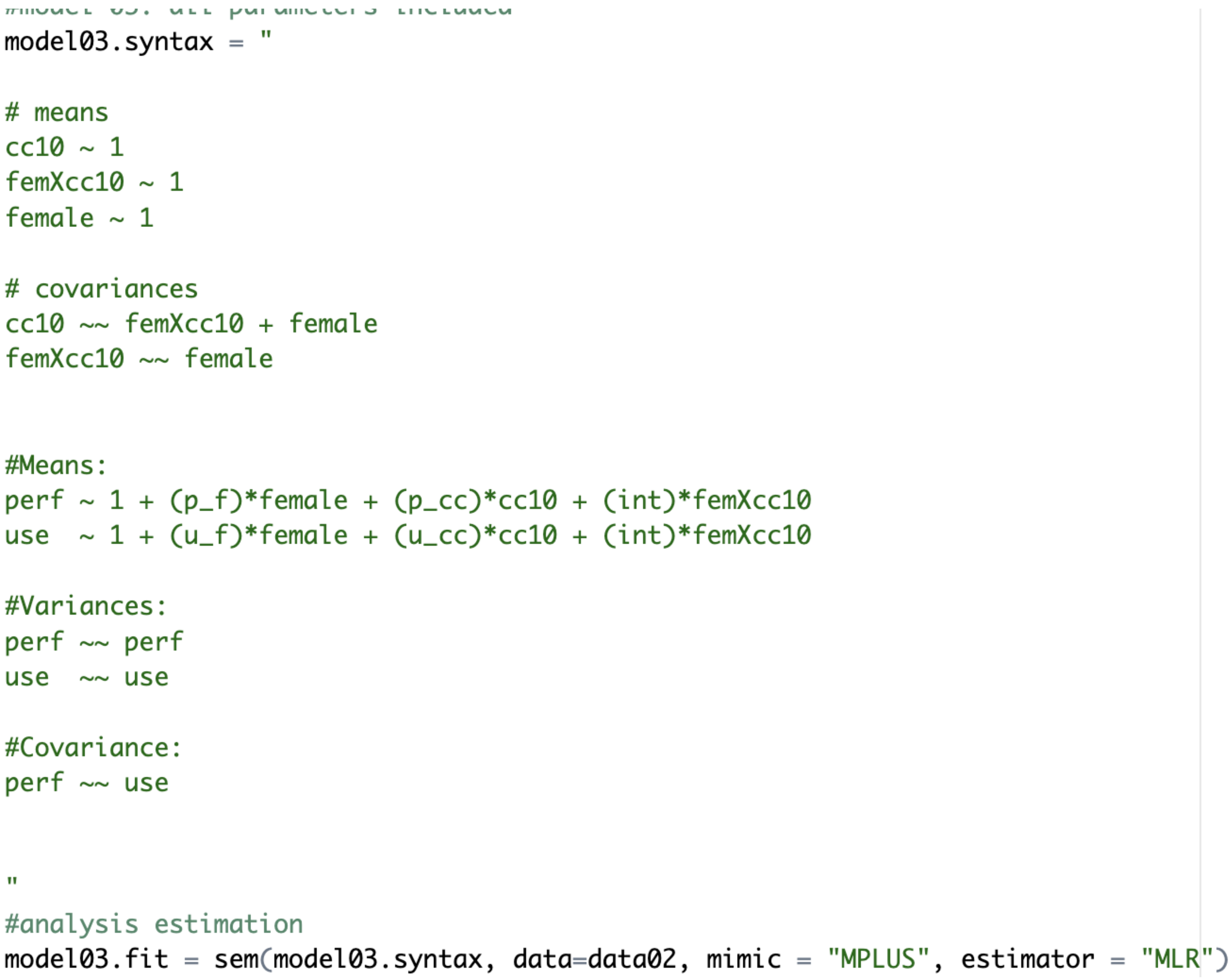


```
> #to see modification indices:
> modindices(model02.fit)
        lhs op      rhs     mi    epc sepc.lv sepc.all sepc.nox
8      perf  ~   female  1.923 -0.628  -0.628   -0.106   -0.106
9      perf  ~     cc10 24.602  0.165   0.165    0.378    0.378
10     perf  ~ femXcc10 12.927  0.153   0.153    0.274    0.274
12      use  ~   female  1.865 -3.466  -3.466   -0.104   -0.104
```

- mi : the expected value of the LRT of the current model and a model where this parameter was added
- epc column: expected value of the parameter in the model where this parameter was added

ADDING PREDICTORS TO THE MODEL

Adding Predictors: Removing Zero Values from Parameters



```
#model 03: all parameters included
model03.syntax = "

# means
cc10 ~ 1
femXcc10 ~ 1
female ~ 1

# covariances
cc10 ~~ femXcc10 + female
femXcc10 ~~ female


#Means:
perf ~ 1 + (p_f)*female + (p_cc)*cc10 + (int)*femXcc10
use  ~ 1 + (u_f)*female + (u_cc)*cc10 + (int)*femXcc10

#Variances:
perf ~~ perf
use  ~~ use

#Covariance:
perf ~~ use


"

#analysis estimation
model03.fit = sem(model03.syntax, data=data02, mimic = "MPLUS", estimator = "MLR")
```

First Question: Which Model “Fits” Better?

- After adding the predictors (estimating their betas) to the model, we must first ask which model fits better

- A likelihood ratio test (LRT) can be performed comparing model02 (with predictors) and model01 (without)

- **Which model is the null model?**

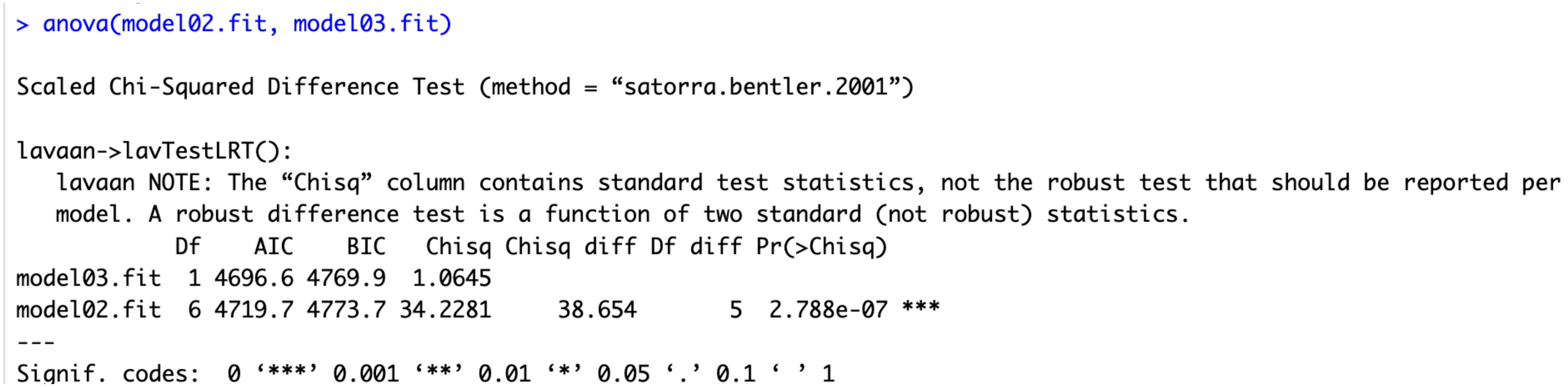
- **Which model is the alternative model?**

- **What is the null hypothesis?**

- **What is the alterative hypothesis?**

LRT With Scaled Chi-Squares

- R makes the scaled Chi-square LRT easy…use the anova() function and it will rescale the Chi-squares automatically

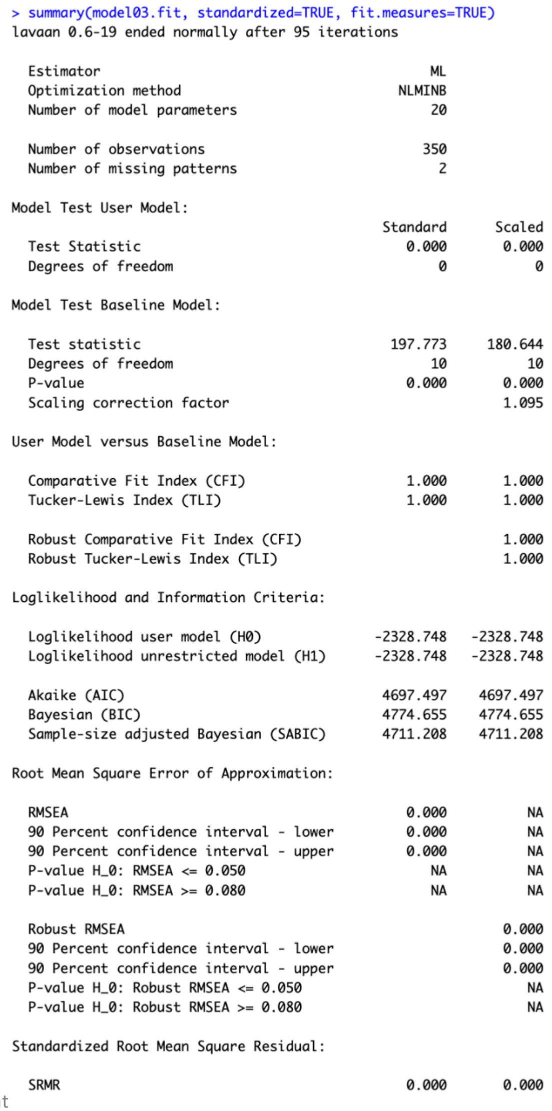


```
> anova(model02.fit, model03.fit)

Scaled Chi-Squared Difference Test (method = “satorra.bentler.2001”)

lavaan->lavTestLRT():
   lavaan NOTE: The “Chisq” column contains standard test statistics, not the robust test that should be reported per
   model. A robust difference test is a function of two standard (not robust) statistics.
            Df    AIC    BIC   Chisq Chisq diff Df diff Pr(>Chisq)
model03.fit  1 4696.6 4769.9  1.0645
model02.fit  6 4719.7 4773.7 34.2281     38.654       5  2.788e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- Here we see that we reject model01 (the null model)
- So we conclude that **at least** one beta value was significantly different from zero

Step 2: Inspect Model Fit

- Next we inspect the model fit of model03:

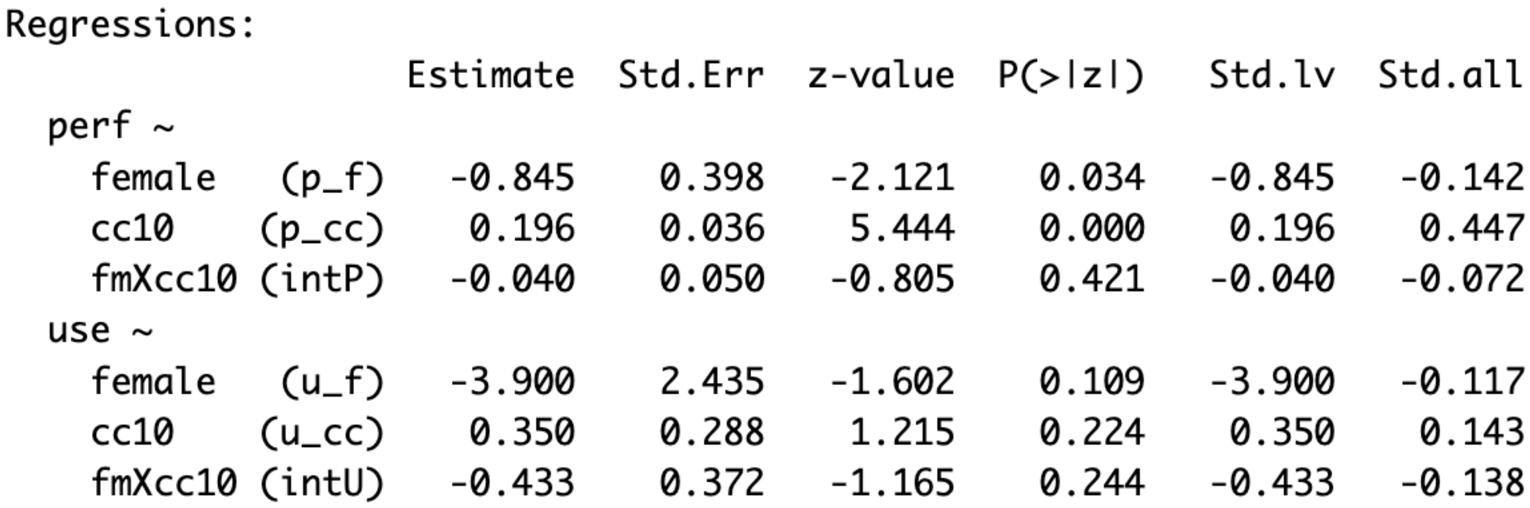


```
> summary(model03.fit, standardized=TRUE, fit.measures=TRUE)
lavaan 0.6-19 ended normally after 95 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        20

  Number of observations                           350
  Number of missing patterns                         2

Model Test User Model:
                                              Standard      Scaled
  Test Statistic                                 0.000       0.000
  Degrees of freedom                                 0           0

Model Test Baseline Model:

  Test statistic                               197.773     180.644
  Degrees of freedom                                10          10
  P-value                                        0.000       0.000
  Scaling correction factor                                  1.095

User Model versus Baseline Model:

  Comparative Fit Index (CFI)                    1.000       1.000
  Tucker-Lewis Index (TLI)                       1.000       1.000

  Robust Comparative Fit Index (CFI)                         1.000
  Robust Tucker-Lewis Index (TLI)                            1.000

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -2328.748   -2328.748
  Loglikelihood unrestricted model (H1)      -2328.748   -2328.748

  Akaike (AIC)                                4697.497    4697.497
  Bayesian (BIC)                              4774.655    4774.655
  Sample-size adjusted Bayesian (SABIC)       4711.208    4711.208

Root Mean Square Error of Approximation:

  RMSEA                                          0.000          NA
  90 Percent confidence interval - lower         0.000          NA
  90 Percent confidence interval - upper         0.000          NA
  P-value H_0: RMSEA <= 0.050                       NA          NA
  P-value H_0: RMSEA >= 0.080                       NA          NA

  Robust RMSEA                                               0.000
  90 Percent confidence interval - lower                     0.000
  90 Percent confidence interval - upper                     0.000
  P-value H_0: Robust RMSEA <= 0.050                            NA
  P-value H_0: Robust RMSEA >= 0.080                            NA

Standardized Root Mean Square Residual:

  SRMR                                           0.000       0.000
```

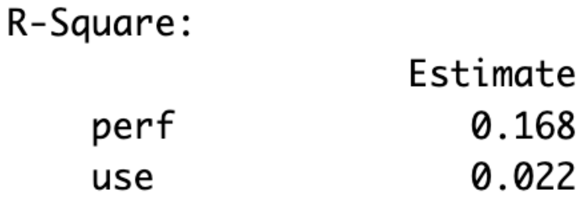
- Model03 has the same log-likelihood as the saturated model…so it is equivalent to the saturated model
  - Therefore it fits perfectly!

- Any path model where **all** exogenous variables predict **all** endogenous variables **AND** all covariances between endogenous variables are estimated is the saturated model

Up Next: Inspect Parameters and Make Interpretations



```
Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)   Std.lv  Std.all
  perf ~
    female    (p_f)  -0.845    0.398   -2.121    0.034   -0.845   -0.142
    cc10     (p_cc)   0.196    0.036    5.444    0.000    0.196    0.447
    fmXcc10 (intP)   -0.040    0.050   -0.805    0.421   -0.040   -0.072
  use ~
    female    (u_f)  -3.900    2.435   -1.602    0.109   -3.900   -0.117
    cc10     (u_cc)   0.350    0.288    1.215    0.224    0.350    0.143
    fmXcc10 (intU)   -0.433    0.372   -1.165    0.244   -0.433   -0.138
```



```
R-Square:
                   Estimate
    perf              0.168
    use               0.022
```

Questions to Answer about this Model

- What is the effect of college experience on usefulness for males?

- What is the effect of college experience on usefulness for females?

- What is the difference between males and females ratings of usefulness when college experience = 10?

- How did the difference between males and females ratings change for each additional hour of college experience?

Questions to Answer about this Model

- What is the effect of college experience on performance for males?

- What is the effect of college experience on performance for females?

- What is the difference between males and females performance when college experience = 10?

- How did the difference between males and females performance change for each additional hour of college experience?

WRAPPING UP

Multivariate Linear Models with Predictors

- In this lecture we discussed the basics of multivariate linear models with predictors
  - Model specification/identification
  - Model estimation
  - Model fit (necessary, but not sufficient)
  - Model modification and re-estimation
  - Final model parameter interpretation

- There is a lot to the analysis – but what is important to remember is the over-arching principal of multivariate analyses: covariance between variables is important
  - Path models imply very specific covariance structures
  - The validity of the results hinge upon accurately finding an approximation to the covariance matrix

Where We Go Next

- The SEM framework allows us to implement full ML estimation with missing data for multivariate linear models

- Next, we will cover how to integrate auxiliary variables into the SEM framework
  - ➢ But first, we must describe path analysis in more detail