Introduction to Mixed Models for Multivariate Regression

EPSY 905: Fundamentals of Multivariate Modeling
Online Lecture #15

THE UNIVERSITY OF KANSAS

In This Lecture…

- Multivariate regression via mixed models
- Comparing and contrasting path analysis with mixed models
  - Differences in model fit measures
  - Differences in software estimation methods
  - Model comparisons via multivariate Wald tests (instead of LRTs)
  - How to compute R^2

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.

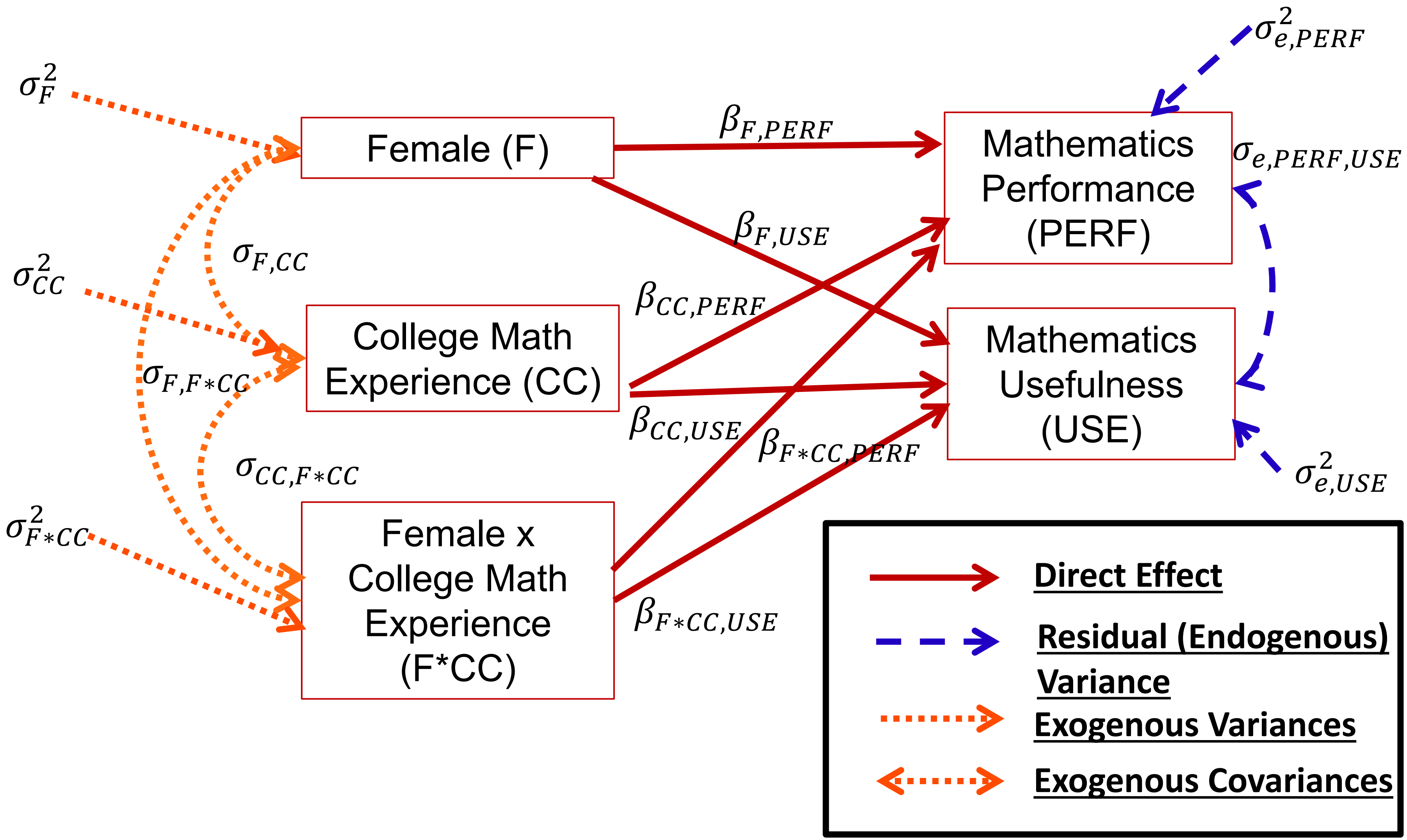
- Sample of 350 undergraduates (229 women, 121 men)
  - In simulation, 10% of variables were missing (using missing completely at random mechanism)

- 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 (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)

Multivariate Linear Regression Path Diagram



The Big Picture

- Mixed models are used for many types of analyses:
  - Analogous to MANOVA and M-Regression (so repeated measures analyses)
  - Multilevel models for clustered, longitudinal, and crossed-effects data
- The biggest difference between mixed models and path analysis software is in the assumed distribution of the exogenous variables
  - Mixed models: no distribution assumed
  - Path analysis: most software assumes multivariate normal
    - Mplus does not by default
  - This affects how missing data are managed – mixed models cannot have any missing IVs
- Mixed models also do not allow endogenous variables to predict other endogenous variables
  - No indirect effects are possible from a single analysis (multiple analyses needed)
- Mixed models software also often needs variables to be stored in so-called "stacked" or long-format data (one row per DV)
  - We used wide-format data for lavaan (one row per person)

Multivariate Linear Models Software Scorecard

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

MULTIVARIATE REGRESSION/ANOVA/ANCOVA VIA PATH ANALYSIS

Multivariate Regression

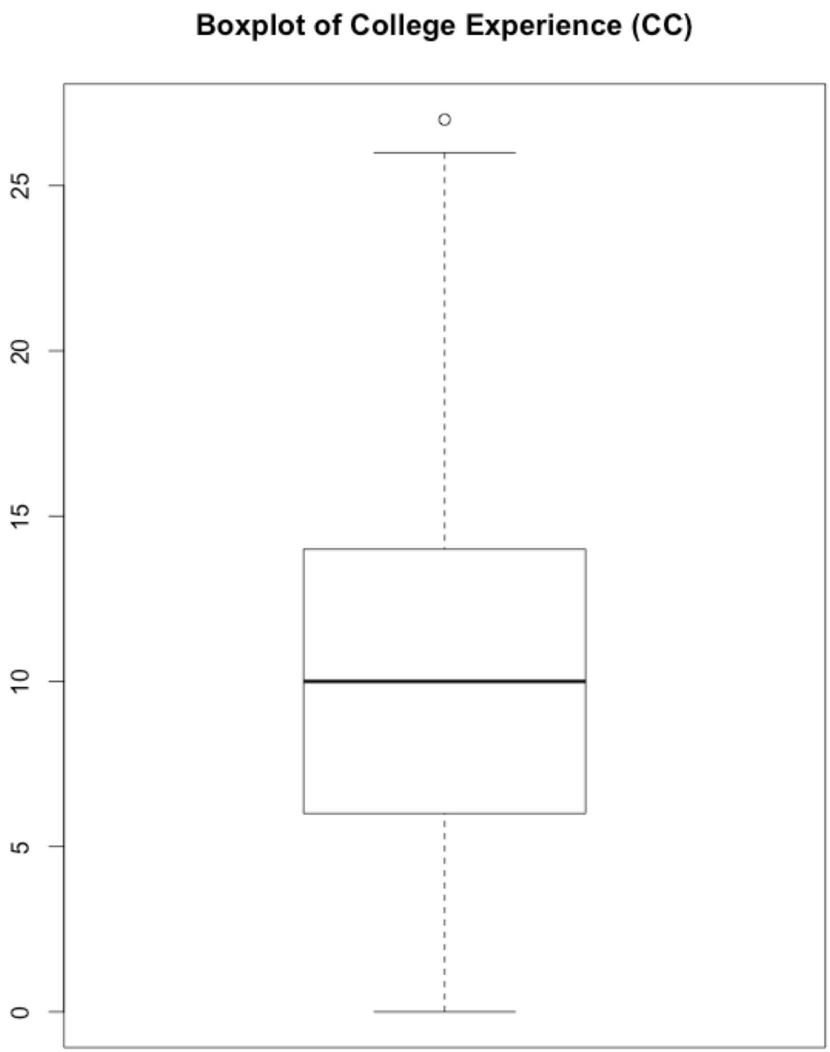
- Before we dive into mixed models, we will begin with a multivariate regression model:
  - Predicting mathematics performance (PERF) with female (F), college math experience (CC), and the interaction between female and college math experience (FxCC)
  - Predicting perceived usefulness (USE) with female (F), college math experience (CC), and the interaction between female and college math experience (FxCC)

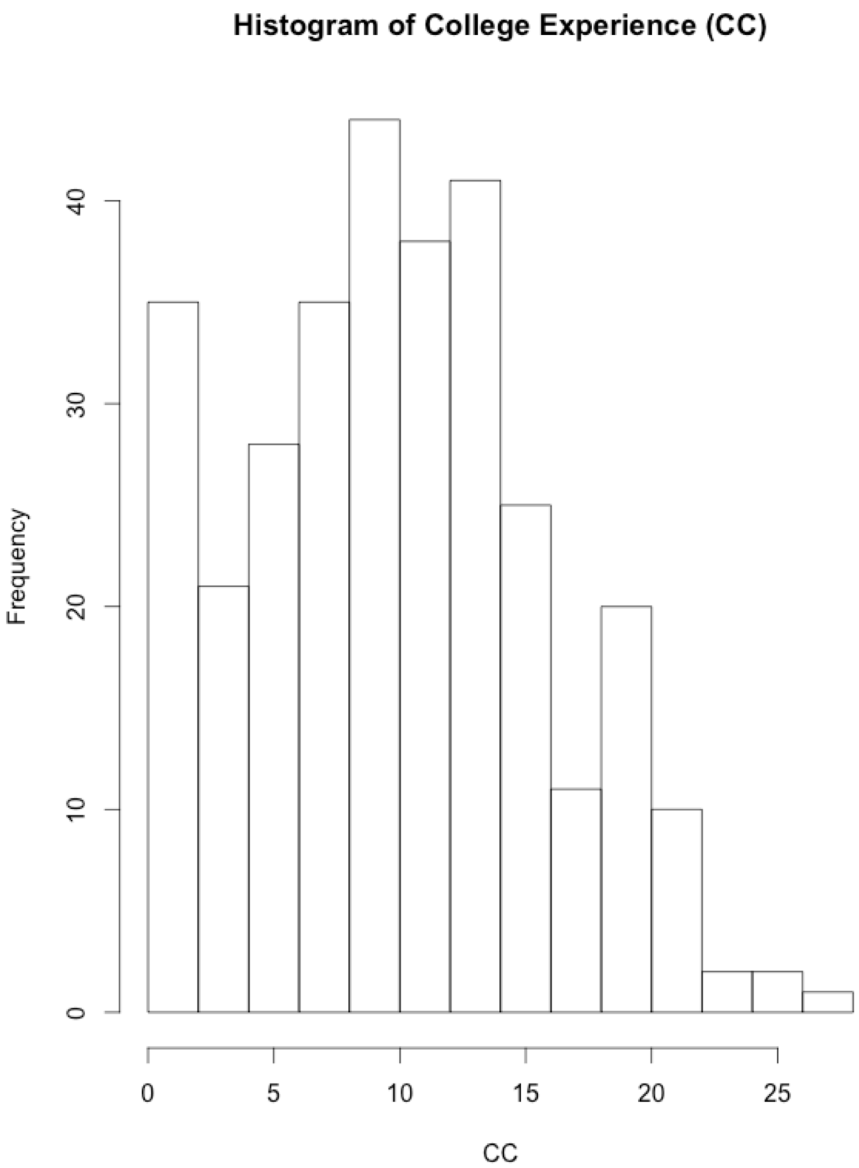
$$PERF_i = \beta_{0,PERF} + \beta_{F,PERF}F_i + \beta_{CC,PERF}CC_i + \beta_{F*CC,PERF}F_i * CC_i + e_{i,PERF}$$
$$USE_i = \beta_{0,USE} + \beta_{F,USE}F_i + \beta_{CC,USE}CC_i + \beta_{F*CC,USE}F_i * CC_i + e_{i,USE}$$

- 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





Types of Variables in the Analysis

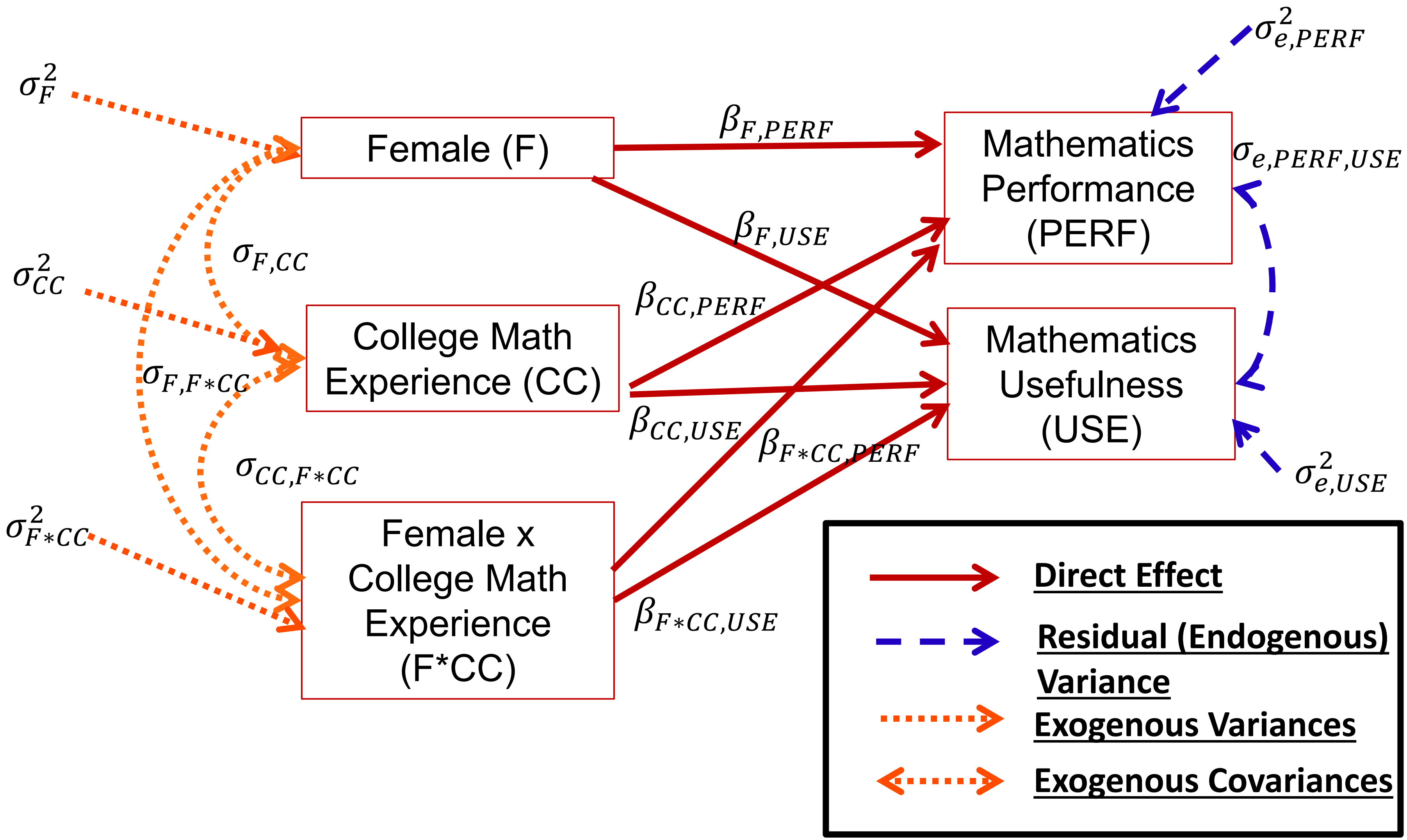
- An important distinction in path analysis is between endogenous and exogenous variables

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

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

- These distinctions are not commonly used in mixed models:
  - Endogenous variables are called dependent variables or outcomes
  - Exogenous variables are called independent variables or predictors

Multivariate Linear Regression Path Diagram



CONVERTING DATA FROM WIDE- TO LONG-FORMAT USING THE RESHAPE FUNCTION

First Step: Convert Data from Wide to Long

- Original wide-format data (all DVs for a person on one row)

	id	perf	use	female	cc10
1	1	14	44	1	-1
2	2	12	77	0	-8
3	3	NA	64	1	2
4	4	19	71	0	10
5	5	12	48	0	5

- Reshape commands:

```
#omit all variables but those in the analysis:
data02 = data01[c("id", "perf", "use", "female", "cc10")]

#create long-format data set using reshape command
data02_long = reshape(data02, varying = c("perf", "use"), v.names = "score",
                      timevar = "DV", times = c("perf", "use"),
                      direction = "long")
```

- Resulting data:

	id	female	cc10	DV	score
1.perf	1	1	-1	perf	14
1.use	1	1	-1	use	44
2.perf	2	0	-8	perf	12
2.use	2	0	-8	use	77
3.perf	3	1	2	perf	NA
3.use	3	1	2	use	64
4.perf	4	0	10	perf	19
4.use	4	0	10	use	71
5.perf	5	0	5	perf	12

THE UNIVERSITY OF KANSAS

Second Step: Remove any Missing Values

- Because all variables are not put into the likelihood function, any missing values (on any DV or IV) are not permitted
  - For IVs, omitting variables has serious implications for the analysis (assumes data are missing completely at random)
  - For DVs, because the DVs are part of the likelihood, they can be missing without problem (well without problems that aren't easy to handle)

- So, we must remove all missing values from all variables:

```
#catch #1: missing values must be removed from DV and all predictors:
data03_long = data02_long[which(is.na(data02_long$score) == FALSE &
                                is.na(data02_long$cc10)  == FALSE &
                                is.na(data02_long$female)  == FALSE),]
```

Third Step: Create Dummy Code for DV

- In stacked data, the dependent variable values are all put into one column (variable) of the data set

- As mixed model software has one spot for a single outcome, we trick the software into modeling more than one by using a dummy coded variable that is part of our analysis, making each effect conditional on the correct DV

```
#create dummy coded variable for indicating which DV is in use:
data03_long$dPerf = 0
data03_long[which(data03_long$DV == "perf"),]$dPerf = 1
```

- We've decided to put USE as the reference variable
  - Like reference group, but now for DV

BUILDING THE "EMPTY" MULTIVARIATE MODEL IN MIXED MODELS SOFTWARE

Multivariate Regression from the Path Analysis Method

- If we were to put our empty model into a path analysis framework, we would end up with the following:

$$PERF_i = \beta_{0,PERF} + e_{i,PERF}$$
$$USE_i = \beta_{0,USE} + e_{i,USE}$$

- 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:

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

Building the Empty Model: Not So Empty

- A multivariate model using mixed model software uses the dummy code for DV to make all effects conditional on the specific DV in the model
  - Here I use the symbol δ to represent each fixed effect in the multivariate model from the mixed model perspective
  - I will compare/contrast these with the symbols β from the fixed effects in path analysis
- For instance, our empty model is thus:

$$Y_{i,DV} = \delta_0 + \delta_1 dPerf_i + e_{i,DV}$$

- The prediction is conditional on the value of dPerf:
  - When dPerf = 0 → DV = “Use” → $USE_i = \delta_0 + e_{i,USE}$
  - When dPerf = 1 → DV = “Perf” → $PERF_i = \delta_0 + \delta_1 * 1 + e_{i,Perf}$

Estimating the Empty Model

- From the nlme library, we use the gls() function
  - Be sure the library is installed and loaded before trying this!

```
#create empty model using REML estimation to attempt to mirror initial analysis:
model01_mixed = gls(model = score ~ 1 + dPerf, method = "REML", data = data03_long,
                    correlation=corSymm(form=~1|id), weights=varIdent(form = ~1|DV))
```

- model = score ~ 1 + dPerf:
  - DV ~ → put the name of the DV on the left of the ~
  - 1 → Unnecessary but indicates the intercept is estimated
  - dPerf → The rest of the IVs go between plus signs after the ~
- method = "REML" → Don't change (more details next)
- correlation = corSymm(form = ~1|id)
  - Provides estimates of all unique correlations
  - Needs id variable name after | for program to know which data comes from which person
- weights = varIdent(form = ~1|DV)
  - Estimates a different (residual) variance for each DV
  - With correlation line ensures an unstructured model is estimated

Empty Model Results: Fixed Effects and "Model Fit"

```
> summary(model01_mixed)
Generalized least squares fit by REML
  Model: score ~ 1 + dPerf
  Data: data03_long
       AIC      BIC    logLik
  3774.313 3795.871 -1882.156

Correlation Structure: General
 Formula: ~1 | id
 Parameter estimate(s):
 Correlation:
  1
2 0.136
Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | DV
 Parameter estimates:
    perf      use
1.000000 5.337397

Coefficients:
                Value Std.Error   t-value p-value
(Intercept)  52.40986 0.9437950  55.53098       0
dPerf       -38.46901 0.9399708 -40.92575       0

 Correlation:
      (Intr)
dPerf -0.98

Standardized residuals:
        Min          Q1         Med          Q3         Max
-3.24789265 -0.64196261  0.01956532  0.68109325  2.99644101

Residual standard error: 3.023304
Degrees of freedom: 553 total; 551 residual
```

Empty Model Results: Residual Covariance Matrix

- The residual covariance matrix comes from the getVarCov() function:

```
> getVarCov(model01_mixed)
Marginal variance covariance matrix
       [,1]     [,2]
[1,] 9.1404   6.6311
[2,] 6.6311 260.3900
  Standard Deviations: 3.0233 16.137
```

Mapping Multivariate Mixed Models onto Path Models

- To compare this result with the path analyses we conducted previously, we'll have to use this data set
  - Omit the same observations

- So, we'll need to take our long-format data and reshape it into wide-format:

```
#Comparison with Path Model Version: removed NA values ----------------------------------------
data03_wide = reshape(data03_long, timevar = "DV", idvar = c("id", "female", "cc10"), direction = "wide")

#create DVs with original names:
data03_wide$perf = data03_wide$score.perf
data03_wide$use = data03_wide$score.use

#creating interation between male and cc:-----------------------------------------------------
data03_wide$femaleXcc10 = data03_wide$female*data03_wide$cc10
```

Lavaan Model Syntax

```
model01_mirror.syntax = "

#Means:
perf ~ 1
use  ~ 1

#Variances:
perf ~~ perf
use  ~~ use

#Covariance:
perf ~~ use

"

model01_path_noNA.fit = sem(model01_mirror.syntax, data=data03_wide,
                            conditional.x=TRUE, fixed.x = TRUE, mimic = "MPLUS", estimator = "MLR")

summary(model01_path_noNA.fit, fit.measures = TRUE, standardized = TRUE)
```

Comparing and Contrasting Results: Model Fit

```
> summary(model01_mixed)
Generalized least squares fit by REML
  Model: score ~ 1 + dPerf
  Data: data03_long
       AIC      BIC    logLik
  3774.313 3795.871 -1882.156
```

```
> summary(model01_path_noNA.fit, fit.measures = TRUE, standardized = TRUE)
lavaan (0.5-20) converged normally after  29 iterations

  Number of observations                           311

  Number of missing patterns                         3

  Estimator                                         ML      Robust
  Minimum Function Test Statistic                0.000       0.000
  Degrees of freedom                                 0           0
  Scaling correction factor                                     NA
    for the Yuan-Bentler correction (Mplus variant)

Model test baseline model:

  Minimum Function Test Statistic                4.798       4.574
  Degrees of freedom                                 1           1
  P-value                                        0.028       0.032

User model versus baseline model:

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

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)              -1882.250   -1882.250
  Loglikelihood unrestricted model (H1)      -1882.250   -1882.250

  Number of free parameters                          5           5
  Akaike (AIC)                                3774.500    3774.500
  Bayesian (BIC)                              3793.199    3793.199
  Sample-size adjusted Bayesian (BIC)         3777.341    3777.341

Root Mean Square Error of Approximation:

  RMSEA                                          0.000       0.000
  90 Percent Confidence Interval          0.000  0.000       0.000  0.000
  P-value RMSEA <= 0.05                          1.000       1.000

Standardized Root Mean Square Residual:

  SRMR                                           0.000       0.000
```

Comparing and Contrasting Results: Parameter Estimates

```
Correlation Structure: General
 Formula: ~1 | id
 Parameter estimate(s):
 Correlation:
  1
2 0.136
Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | DV
 Parameter estimates:
    perf      use
1.000000 5.337397

Coefficients:
                Value Std.Error   t-value p-value
(Intercept)  52.40986 0.9437950  55.53098       0
dPerf       -38.46901 0.9399708 -40.92575       0
```

```
Covariances:
                   Estimate  Std.Err  Z-value  P(>|z|)   Std.lv  Std.all
  perf ~~
    use               6.610    3.049    2.168    0.030    6.610    0.136

Intercepts:
                   Estimate  Std.Err  Z-value  P(>|z|)   Std.lv  Std.all
    perf             13.941    0.187   74.660    0.000   13.941    4.620
    use              52.410    0.940   55.750    0.000   52.410    3.253

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)   Std.lv  Std.all
    perf              9.105    0.827   11.006    0.000    9.105    1.000
    use             259.498   20.968   12.376    0.000  259.498    1.000
```

```
> getVarCov(model01_mixed)
Marginal variance covariance matrix
       [,1]     [,2]
[1,] 9.1404   6.6311
[2,] 6.6311 260.3900
  Standard Deviations: 3.0233 16.137
```

```
#create mean for Usefulness (sum of intercept plus main effect of dPerf)
estimate01 = matrix(c(1,1), nrow = 1); rownames(estimate01) = "Mean Usefulness"
estimate02 = matrix(c(1,0), nrow = 1); rownames(estimate02) = "Mean Performance"

estimates = rbind(estimate01, estimate02)
effects = glht(model = model01_mixed, linfct = estimates)
summary(effects)

	 Simultaneous Tests for General Linear Hypotheses

Fit: gls(model = score ~ 1 + dPerf, data = data03_long, correlation = corSymm(form = ~1 |
    id), weights = varIdent(form = ~1 | DV), method = "REML")

Linear Hypotheses:
                        Estimate Std. Error z value Pr(>|z|)
Mean Usefulness == 0     13.9408     0.1869   74.61   <1e-10 ***
Mean Performance == 0    52.4099     0.9438   55.53   <1e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```


THE UNIVERSITY OF KANSAS

RESIDUAL MAXIMUM LIKELIHOOD ESTIMATION (UNBIASED VARIANCE COMPONENTS)

Residual Maximum Likelihood Estimation

- The ML estimator is nice, but the variance estimate is downward biased (too small)
  - Remember – it divides by N for the residual covariance matrix
- In small samples, this is likely to lead to biased estimates and incorrect p-values
  - The variance goes into the SE, which goes into the Wald test, which dictates the p-value for the beta
- Instead, another maximum likelihood technique has been developed: Residual Maximum Likelihood (REML)
  - Maximizes the likelihood of the residuals rather than the data
  - Has unbiased estimates of the residual covariance matrix
  - Is the default method of estimation for most mixed model estimation packages
- There is one catch to REML: you cannot use a LRT to compare nested models with differing fixed effects
  - Because the algorithm uses residuals, if the residuals change, the likelihood changes
  - Residuals come from the fixed effects → if fixed effects are different, then residuals change, causing the likelihood to change
  - Can use multivariate Wald test for fixed effects
- Don't mix ML and REML for the same analysis

ADDING PREDICTORS TO THE MODEL

Adding Predictors To The Model

- Adding predictors to the model is similar to adding predictors in regular regression models
  - No 0* terms in syntax to remove

- By using REML we cannot compare models using likelihood ratio tests
  - REML LRTs must have same fixed effects
  - Adding predictors adds new fixed effects to the empty model

- We are predicting each DV with female, cc10, and female*cc10

THE UNIVERSITY OF KANSAS

Model with Predictors: Syntax

```
#model 02: all predictors included -----------------------------------------------------------------
model02_mixed = gls(model = score ~ 1 + dPerf  + female + dPerf*female +
                      cc10 + dPerf*cc10 + female*cc10 + dPerf*female*cc10,
                    method = "REML", data = data03_long, correlation=corSymm(form=~1|id),
                    weights=varIdent(form = ~1|DV))
```

<u>Interpret the Parameters:</u>

β_0:

$\beta_{dPerf}(dPerf_i)$:

$\beta_{Female}(Female_i)$:

$\beta_{dPerf*Female}(dPerf_i Female_i)$:

$\beta_{CC10}(CC10_i)$:

$\beta_{dPerf*CC10}(dPerf_i CC10_i)$:

$\beta_{Female*CC10}(CC10_i Female_i)$:

$\beta_{dPerf*Female*CC10}(dPerf_i CC10_i Female_i)$:

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) cannot be performed as we are using REML

```
> anova(model01_mixed, model02_mixed)
              Model df      AIC      BIC    logLik   Test  L.Ratio p-value
model01_mixed     1  5 3774.313 3795.871 -1882.156
model02_mixed     2 11 3771.308 3818.617 -1874.654 1 vs 2 15.00473  0.0202
Warning message:
In nlme::anova.lme(object = model01_mixed, model02_mixed) :
  fitted objects with different fixed effects. REML comparisons are not meaningful.
```

- **Which model is the null model?**
  - **Model01**
- **Which model is the alternative model?**
  - **Model02**
- **What is the null hypothesis?**
  - H_0:
- **What is the alterative hypothesis?**

THE UNIVERSITY OF KANSAS

Syntax for Multivariate Wald Test for Model Fit

```
#create a specific vector for each beta weight: will use later to create specific parameters for each DV prediction
intercept =             matrix(c(1, 0, 0, 0, 0, 0, 0, 0), nrow = 1)
rownames(intercept) = "Intercept"
dPerf =                 matrix(c(0, 1, 0, 0, 0, 0, 0, 0), nrow = 1)
rownames(dPerf) = "dPerf"
female =                  matrix(c(0, 0, 1, 0, 0, 0, 0, 0), nrow = 1)
rownames(female) = "female"
cc10 =                  matrix(c(0, 0, 0, 1, 0, 0, 0, 0), nrow = 1)
rownames(cc10) = "cc10"
dPerf_female =            matrix(c(0, 0, 0, 0, 1, 0, 0, 0), nrow = 1)
rownames(dPerf_female) = "dPerf:female"
dPerf_cc10 =            matrix(c(0, 0, 0, 0, 0, 1, 0, 0), nrow = 1)
rownames(dPerf_cc10) = "dPerf:cc10"
female_cc10  =            matrix(c(0, 0, 0, 0, 0, 0, 1, 0), nrow = 1)
rownames(female_cc10) = "female:cc10"
dPerf_female_cc10 =       matrix(c(0, 0, 0, 0, 0, 0, 0, 1), nrow = 1)
rownames(dPerf_female_cc10) = "dPerf:female:cc10"

#analog to LRT: test all parameters simultaneously (a multi-DF contrast)
overall_sig = rbind(female, cc10, dPerf_female, dPerf_cc10, female_cc10, dPerf_female_cc10)
effects = glht(model = model02_mixed, linfct = overall_sig)

#note: R's package do not give you an F-test [BAD R, BAD!]
summary(effects, test=Ftest())
```

Results for Model Fit

- Note: R's nlme function doesn't do a good job with df.residual and provides a Chi-square test
  - SAS is way more advanced on this as there are multiple options

```
         General Linear Hypotheses

Linear Hypotheses:
                                                    Estimate
Intercept for Use == 0                              51.79933
Simple main effect for Female predicting Use == 0    1.83570
Simple main effect for CC10 predicting Use == 0      0.19525
Interaction of Female and CC10 predicting Use == 0   0.26018
Intercept for Perf == 0                             13.68949
Simple main effect for Female predicting Perf == 0   0.65832
Simple main effect for CC10 predicting Perf == 0     0.09871
Interaction of Female and CC10 predicting Perf == 0  0.09377

Global Test:
  Chisq DF Pr(>Chisq)
1  8328  8          0
Warning message:
In test(object) :
  'df.residual' is not available for 'model' a Chisq test is performed instead of the requested F test.
>
```

- Also note there are 6 degrees of freedom (one for each additional beta weight in the model)

THE UNIVERSITY OF KANSAS

Up Next: Inspect Parameters and Make Interpretations

- Using the summary() function for the model:

```
Coefficients:
                       Value Std.Error   t-value p-value
(Intercept)         51.79933 1.1750554  44.08246  0.0000
dPerf              -38.10983 1.1751688 -32.42924  0.0000
female               1.83570 2.0060709   0.91507  0.3606
cc10                 0.19525 0.1981999   0.98510  0.3250
dPerf:female        -1.17738 2.0068176  -0.58669  0.5577
dPerf:cc10          -0.09653 0.1979034  -0.48778  0.6259
female:cc10          0.26018 0.3527298   0.73761  0.4611
dPerf:female:cc10   -0.16641 0.3529074  -0.47155  0.6374
```

- But, there is a way to directly code these beta weights into path model analogs using the glht() function

THE UNIVERSITY OF KANSAS

Direct Betas using Linear Combinations and glht()

```
#creating direct betas vectors for prediction of usefuleness (dPerf = 0)
beta_use_0 = intercept; rownames(beta_use_0) = "Intercept for Use"
beta_use_female = female; rownames(beta_use_female) = "Simple main effect for Female predicting Use"
beta_use_cc10 = cc10; rownames(beta_use_cc10) = "Simple main effect for CC10 predicting Use"
beta_use_femaleXcc10 = female_cc10; rownames(beta_use_femaleXcc10) = "Interaction of Female and CC10 predicting Use"

beta_perf_0 = intercept + dPerf; rownames(beta_perf_0) = "Intercept for Perf"
beta_perf_female = female + dPerf_male; rownames(beta_perf_female) = "Simple main effect for Female predicting Perf"
beta_perf_cc10 = cc10 + dPerf_cc10; rownames(beta_perf_cc10) = "Simple main effect for CC10 predicting Perf"
beta_perf_femaleXcc10 = female_cc10 + dPerf_female_cc10; rownames(beta_perf_femaleXcc10) = "Interaction of Female and CC10 predicting Perf"

beta = rbind(beta_use_0, beta_use_female, beta_use_cc10, beta_use_femaleXcc10, beta_perf_0, beta_perf_female,
             beta_perf_cc10, beta_perf_femaleXcc10)
effects = glht(model = model02_mixed, linfct = beta)
summary(effects)
```

```
	 Simultaneous Tests for General Linear Hypotheses

Fit: gls(model = score ~ 1 + dPerf + female + dPerf * female + cc10 + 
    dPerf * cc10 + female * cc10 + dPerf * female * cc10, data = data03_long, 
    correlation = corSymm(form = ~1 | id), weights = varIdent(form = ~1 | 
        DV), method = "REML")

Linear Hypotheses:
                                                 Estimate Std. Error z value Pr(>|z|)    
Intercept for Use == 0                           51.79933    1.17506  44.082   <1e-04 ***
Simple main effect for Female predicting Use == 0  1.83570    2.00607   0.915   0.9575    
Simple main effect for CC10 predicting Use == 0    0.19525    0.19820   0.985   0.9372    
Interaction of Female and CC10 predicting Use == 0  0.26018    0.35273   0.738   0.9879    
Intercept for Perf == 0                          13.68949    0.22375  61.183   <1e-04 ***
Simple main effect for Female predicting Perf == 0  0.65832    0.38378   1.715   0.4733    
Simple main effect for CC10 predicting Perf == 0    0.09871    0.03543   2.786   0.0397 *  
Interaction of Female and CC10 predicting Perf == 0  0.09377    0.06716   1.396   0.7129    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```

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?

Model R-squared

- To determine the model R-squared, we have to compare the variance/covariance matrix from model01 and model02 and make the statistics ourselves:

```
> #R-squared for model02 vs model01 ------------------------------------------------------------------------------
> #model01 residual covariance matrix:
> Vmodel01 = getVarCov(model01_mixed)
> Vmodel01
Marginal variance covariance matrix
       [,1]     [,2]
[1,] 9.1404   6.6311
[2,] 6.6311 260.3900
  Standard Deviations: 3.0233 16.137
>
> #model02 residual covariance matrix:
> Vmodel02 = getVarCov(model02_mixed)
> Vmodel02
Marginal variance covariance matrix
       [,1]     [,2]
[1,] 8.5491   5.0582
[2,] 5.0582 259.5200
  Standard Deviations: 2.9239 16.11
>
> #Rsquare for Performance:
> (Vmodel01[1,1] - Vmodel02[1,1])/Vmodel01[1,1]
[1] 0.06468645
>
> #Rsquare for Usefulness:
> (Vmodel01[2,2] - Vmodel02[2,2])/Vmodel01[2,2]
[1] 0.003341706
```


THE UNIVERSITY OF KANSAS

WRAPPING UP

Differences Between Mixed and Path Model Results

- Things we get directly from path models that we do not get directly in mixed models:
  - Tests for approximate model fit
  - Scaled Chi-square for some types of non-normal data
  - Standardized parameter coefficients
  - Tests for indirect effects
  - R-squared statistics

- Things we get directly in mixed models that we do not get in path models:
  - REML (unbiased estimates of variances/covariances)

Wrapping Up

- In this lecture we discussed the basics of mixed model analyses for multivariate models
  - Model specification/identification
  - Model estimation
  - 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
  - Mixed models imply very specific covariance structures
  - The validity of the results still hinge upon accurately finding an approximation to the covariance matrix