Latent Variable Models in Education

A. Alexander Beaujean
Alex_Beaujean@baylor.edu

Latent Variable Models in Education

Fall 2012

Latent Variables

Table of Contents

Confirmatory Factor Analysis

- Single Factor Model
- Two Factor Model
  - Structure Coefficients
  - Model Fit

Latent Variable Identification

- Scaling the Latent Variable
- Number of Indicators for a Latent Variable
- Latent Variable Correlations
- Loading Estimation
- Empirical Underidentification

Talk Outline

Confirmatory Factor Analysis

Single Factor Model

Two Factor Model

Confirmatory Factor Analysis

- Structural equation modeling consists of two parts.
  - The structural model (which is basically the path models).
  - The measurement model, which is the latent variable (factor analysis) component.

Confirmatory Factor Analysis

- The purpose of factor analysis is to understand the underlying structure that produced a covariance matrix.
- These underling structures are *factors* (aka common factors)
  - For this course, unless stated differently, factors will be synonymous with common factors
- Factors are latent variables: they cannot be observed or measured directly

Confirmatory Factor Analysis

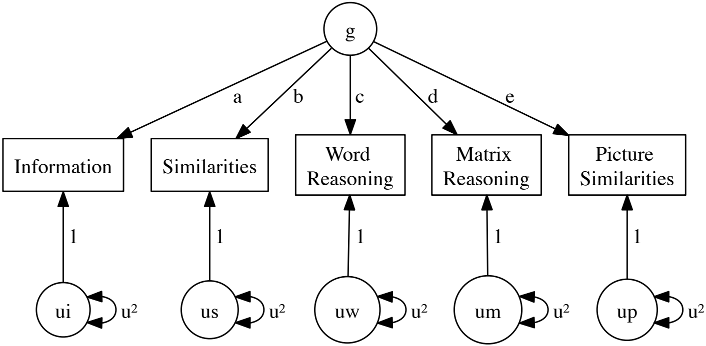
- The thought behind factor analysis is that there are a small number of factors within a given domain
- These factors influence the manifest (i.e., observable) variables (MVs) and hence produce the covariance among the variable.
- Thus variation (or covariation) in the factors "causes" variation in the manifest variables, and covariation in the manifest variables is due to dependence of those variables on one or more factors.
- Factor analysis is the method used to understand the nature of those factors
  - (and, sometimes, identify the number of the factors that produce the observed (co)variation and variation in the manifest variables)

Talk Outline

Confirmatory Factor Analysis

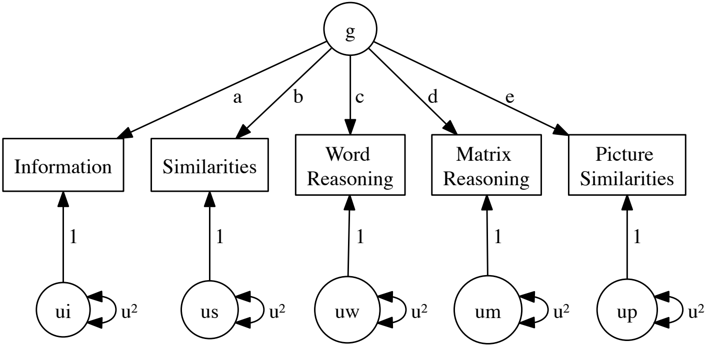
- Single Factor Model
- Two Factor Model

- An example of a simple factor analysis of some of the the Wechsler Intelligence Scale for Children-Fourth Edition subscales
- It has one common factor (g) and five MVs.

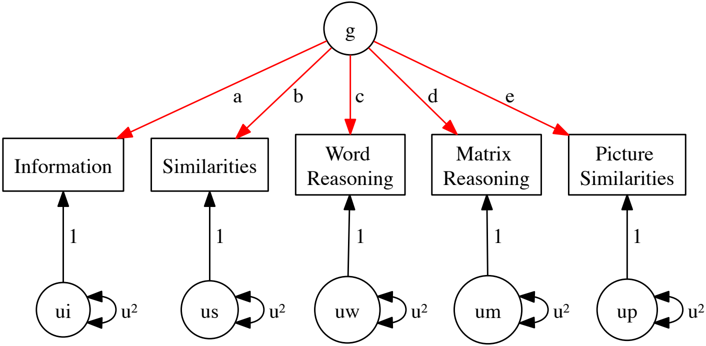


Simple Factor Analysis Model

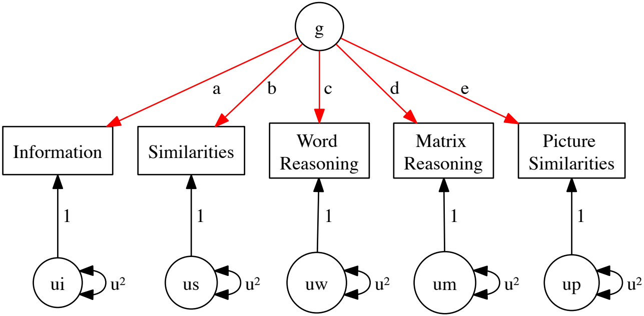
- How many parameters are there (total) to estimate?



- The amount that common factors influence the MVs is measured by *factor loadings* (or *factor pattern coefficients*)
- These are akin to regression coefficients in multiple regression.
- a, b, c, d and e are all factor loadings.



- Can obtain something akin to an R^2 for each MV: find all the "legitimate" paths that paths that go from a MV to its exogenous (latent) variables and return back to the MV.
- For example, go from the Information MV to *g* and then back to Information only through a (twice), thus the amount of variance of the Information MV that *g* explains is a^2.



- In factor analysis, the amount of variance a (common) LV explains of a MV is called the *communality* (h^2).
- From the simple factor example, $h^2 = \frac{\text{VAR}[g]}{\text{VAR}[\text{Information}]}$
- Conversely, the *uniqueness* the amount of variance in the MV not explained by the (common) factors.
- In the simple factor example, the uniqueness of the Information variable is $1 - a^2$.

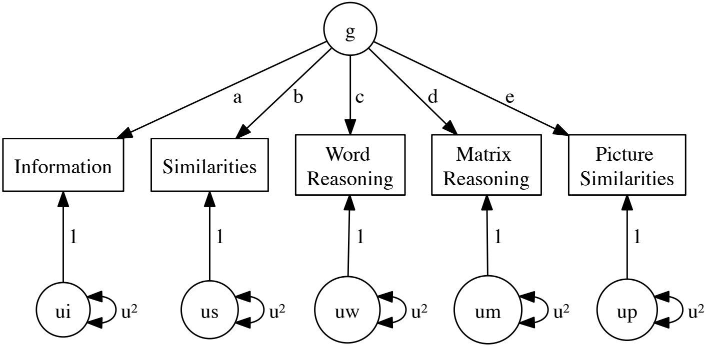
- Let's have some data for the figure,

Correlations for the WISC-IV data

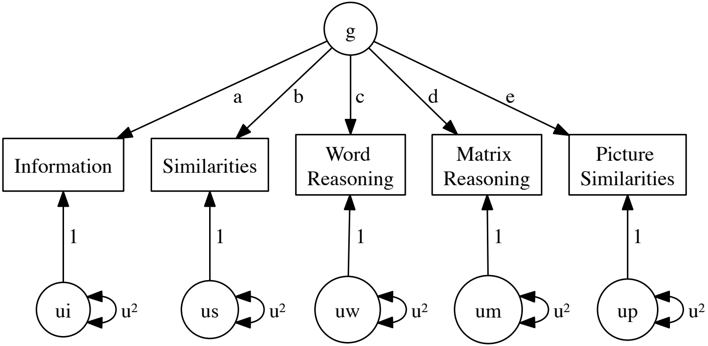
	Info	Sim	Word Reas	Matrix Reas	Picture Sim
inss	1.00	0.72	0.64	0.51	0.37
siss	0.72	1.00	0.63	0.48	0.38
wrss	0.64	0.63	1.00	0.37	0.38
mrss	0.51	0.48	0.37	1.00	0.38
psss	0.37	0.38	0.38	0.38	1.00

- How much unique "information" is in this matrix?
- $5 \times 6/2 = 15$

- Review: How many parameters are there (total) to estimate?



- Review: How many parameters are there (total) to estimate?



- 10

- Analyze the data in R, using lavaan
- First, specify the model

```
> WiscIV.model<-'
g =~ a*inss + b*siss + c*wrss + d*mrss + e*psss
'
```

- Notice that the factor loadings are labeled to match the diagram
  - Not required, but may make it easier to interpret the output

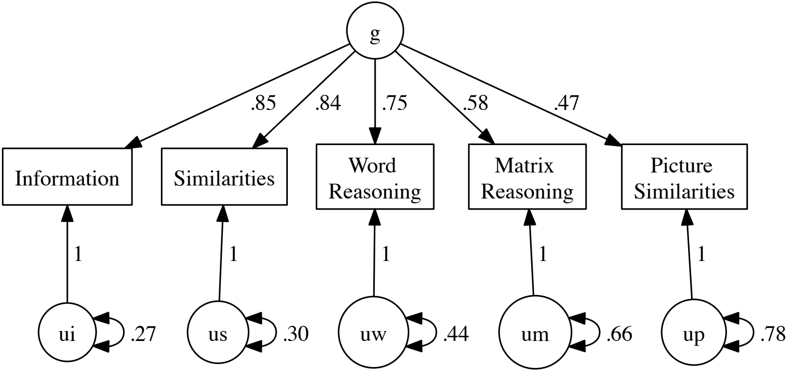
- Next, estimate the parameters and double check the df

```
> WiscIV.fit<-cfa(WiscIV.model, std.lv=TRUE, sample.cov=WiscIV.cor
    , sample.nobs=550)
> summary(WiscIV.fit)
lavaan (0.5-7) converged normally after  14 iterations

  Number of observations                           550

  Estimator                                         ML
  Minimum Function Chi-square                   26.496
  Degrees of freedom                                 5
  P-value                                        0.000
```

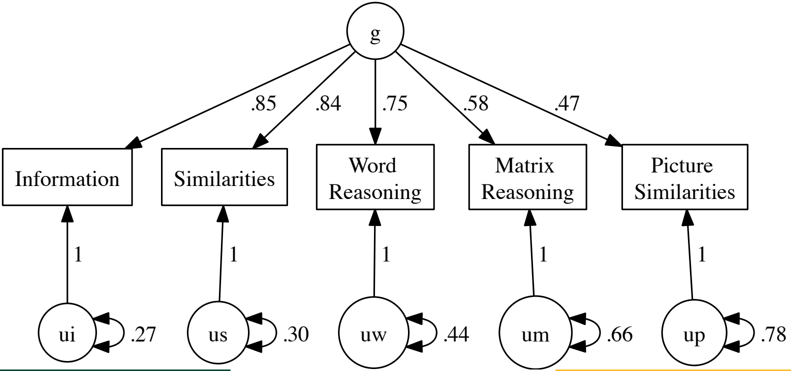
- Last, obtain the parameter estimates

```
> parameterEstimates(WiscIV.fit, ci=FALSE)[,1:5]
    lhs op  rhs label   est
1     g =~ inss     a 0.854
2     g =~ siss     b 0.838
3     g =~ wrss     c 0.745
4     g =~ mrss     d 0.580
5     g =~ psss     e 0.466
6  inss ~~ inss       0.269
7  siss ~~ siss       0.295
8  wrss ~~ wrss       0.443
9  mrss ~~ mrss       0.662
10 psss ~~ psss       0.781
11    g ~~    g       1.000
```

- The communality for Information, a^2, is $.854^2 = .729 = .73$
- Thus, the uniqueness is $1 - a^2 = 1 - .73 = .27$.



- Calculate the implied correlations using Wright's Rules
  - COR[Information,Similarities] $= ab$
  - Plugging in parameter estimates, the reproduced correlation is $(.85)(.84) = .71$, only $.01$ off form the sample correlation of $.72$.



- Obtain implied covariances (correlations)

```
> fitted(WiscIV.fit)
$cov
     inss  siss  wrss  mrss  psss
inss 0.998
siss 0.716 0.998
wrss 0.636 0.625 0.998
mrss 0.495 0.486 0.432 0.998
psss 0.398 0.391 0.347 0.270 0.998

$mean
inss siss wrss mrss psss
   0    0    0    0    0
```

- Obtain residual covariances (correlations)

```
> residuals(WiscIV.fit, type="raw")
$cov
      inss   siss   wrss   mrss   psss
inss  0.000
siss  0.000  0.000
wrss  0.005  0.007  0.000
mrss  0.014 -0.004 -0.059  0.000
psss -0.033 -0.014  0.028  0.109  0.000

$mean
inss siss wrss mrss psss
   0    0    0    0    0
```

Reproduced (Lower) and Residual (Upper) Correlation Matrices

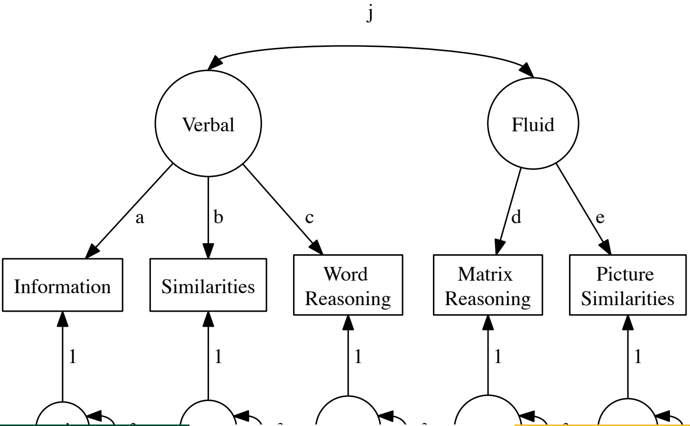
	Info	Sim	Word Reas	Matrix Reas	Picture Sim
inss	1.00	-0.00	0.00	0.01	-0.03
siss	0.72	1.00	0.01	-0.00	-0.01
wrss	0.64	0.62	1.00	-0.06	0.03
mrss	0.50	0.49	0.43	1.00	0.11
psss	0.40	0.39	0.35	0.27	1.00

Talk Outline

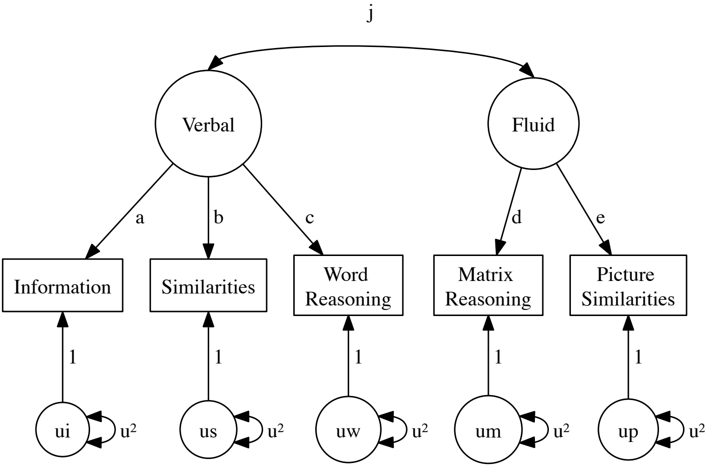
Confirmatory Factor Analysis

- Single Factor Model
- Two Factor Model

- Lets say that we goofed and we should have should have specified a two factor model (Fluid and Verbal abilities) instead of a one factor model



- Now, how many parameters are there (total) to estimate?



Confirmatory Factor Analysis

Two Factor Model

- Analyze the data in `R`, using `lavaan`
- Specify the model, estimate the parameters, and double check the df

```
> WiscIV.model2<-'
V =~ a*inss + b*siss + c*wrss
F=~ d*mrss + e*psss
V~~j*F
'
> WiscIV.fit2<-cfa(WiscIV.model2, std.lv=TRUE, sample.cov=WiscIV.
    cor, sample.nobs=550)
> summary(WiscIV.fit2)
lavaan (0.5-7) converged normally after  18 iterations

  Number of observations                           550

  Estimator                                         ML
  Minimum Function Chi-square                   12.207
  Degrees of freedom                                 4
  P-value                                        0.016
```

Confirmatory Factor Analysis

Two Factor Model

- Obtain parameter estimates

```
> parameterEstimates(WiscIV.fit2, ci=FALSE)[,1:5]
    lhs op  rhs label   est
1     V =~ inss     a 0.857
2     V =~ siss     b 0.840
3     V =~ wrss     c 0.746
4     F =~ mrss     d 0.692
5     F =~ psss     e 0.548
6     V ~~    F     j 0.821
7  inss ~~ inss       0.264
8  siss ~~ siss       0.293
9  wrss ~~ wrss       0.442
10 mrss ~~ mrss       0.519
11 psss ~~ psss       0.698
12    V ~~    V       1.000
13    F ~~    F       1.000
```

- These are the factor loadings (pattern coefficients)
- and (co) variances of MV and LV

- Structure coefficients
  - Correlation between a latent variable and a manifest variable

Talk Outline

Confirmatory Factor Analysis

Single Factor Model

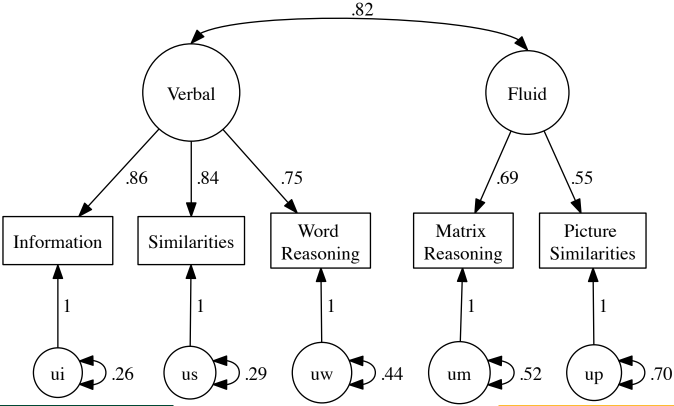
Two Factor Model

Structure Coefficients

Model Fit

- Structure coefficients
  - Correlation between a latent variable and a manifest variable
- To obtain the factor structure coefficients, you can either use
  - Wright's Rules or
  - Matrix multiplication

- Structure coefficients
  - Using Wright's rules, trace the path from the MV to the factor.
  - For example the correlation between Information and the Fluid factor is: $aj = (.86)(.82) = .71$.



- Structure coefficients
  - Using matrix multiplication
  - Post multiply the factor loading matrix $\underset{(5\times 2)}{\mathbf{\Lambda}}$ by the factor correlation matrix $\underset{(2\times 2)}{\mathbf{\Phi}}$, i.e., $\underset{(5\times 2)}{\mathbf{\Lambda\Phi}}$
  - $\underset{(5\times 2)}{\mathbf{\Lambda}} = \begin{bmatrix} .86 & 0 \\ .84 & 0 \\ .75 & 0 \\ 0 & .69 \\ 0 & .55 \end{bmatrix}$, & $\underset{(2\times 2)}{\mathbf{\Phi}} = \begin{bmatrix} 1 & .82 \\ .82 & 1 \end{bmatrix}$
  - $\underset{(5\times 2)}{\mathbf{\Lambda\Phi}} = \begin{bmatrix} .86 & .71 \\ .84 & .69 \\ .75 & .62 \\ .57 & .69 \\ .45 & .55 \end{bmatrix}$
- Notice there are no “0” structure coefficients, even though there were “0” pattern coefficients

Confirmatory Factor Analysis

Two Factor Model: Structure Coefficients

- Structure coefficients
  - Using matrix multiplication
  - In R

```
> load.matrix<-matrix(c(.86,.84,.75,0,0,0,0,0,.69,.55), ncol=2)
> #name the loading matrix columns and rows
> colnames(load.matrix)<-c("Verbal", "Fluid") ; rownames(load.matrix)<-c("inss", "siss",
    "wrss", "mrss", "psss")
> fac.cor<-matrix(c(1,.82,.82,1), ncol=2)
> #name the factor correlation matrix columns and rows
> rownames(fac.cor)<-colnames(fac.cor)<-c("Verbal", "Fluid")
> round(load.matrix%*%fac.cor,2) #Factor Structure coefficients
     Verbal Fluid
inss   0.86  0.71
siss   0.84  0.69
wrss   0.75  0.62
mrss   0.57  0.69
psss   0.45  0.55
```

Talk Outline

Confirmatory Factor Analysis

Single Factor Model

Two Factor Model

Structure Coefficients

Model Fit

- To compare the models, we can estimate the model fit statistics.

```
> fitMeasures(WiscIV.fit)
chisq                   df              pvalue    baseline.chisq
26.496                 5.000              0.000         1072.572
 baseline.df   baseline.pvalue
       10.000            0.000
cfi                tli              logl  unrestricted.logl              npar
0.980              0.960         -3376.540         -3363.293             10.000
aic
6773.081
bic                 ntotal              bic2             rmsea
6816.180           550.000           6784.436            0.088
rmsea.ci.lower    rmsea.ci.upper
     0.057             0.123
rmsea.pvalue       srmr        srmr_nomean
0.023              0.034              0.034
```

- To compare the models, we can estimate the model fit statistics.

```
> fitMeasures(WiscIV.fit2)
chisq                  df              pvalue    baseline.chisq
12.207               4.000               0.016          1072.572
baseline.df   baseline.pvalue
      10.000             0.000
cfi                tli              logl  unrestricted.logl             npar
0.992              0.981         -3369.396         -3363.293            11.000
aic
6760.793
bic             ntotal              bic2                rmsea
6808.202          550.000             6773.283            0.061
rmsea.ci.lower     rmsea.ci.upper
         0.024              0.102
rmsea.pvalue       srmr       srmr_nomean
0.268              0.018              0.018
```

- The fit indices converge in indicating that both models fit the data relatively well
- but the 2-factor model fits better than the 1-factor model

Talk Outline

Latent Variable Identification

- Scaling the Latent Variable
- Number of Indicators for a Latent Variable
- Latent Variable Correlations
- Loading Estimation
- Empirical Underidentification

Latent Variable Identification

- A latent variable model (or more generally, a structural equation model) is (usually) identified if you can express the model's parameters as independent functions of the elements of the covariance matrix.
- This works fine with a simple model, but with more complex models this becomes a tedious chore. Fortunately, there are some "rules of thumb" that are often sufficient.
- We will just discuss the rules for factor analysis.[1]
  - There are additional rules when a structural model is involved as well.

[1] Adapted from Kenny, Kashy, and Bolger (1988)

Talk Outline

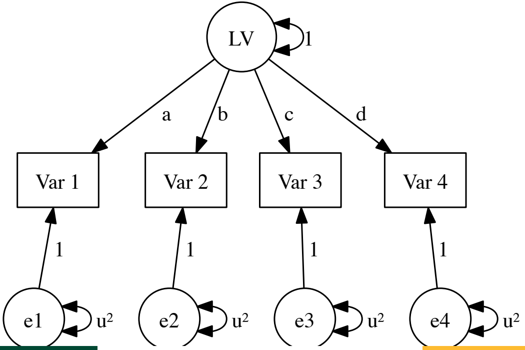
Latent Variable Identification

- Scaling the Latent Variable
- Number of Indicators for a Latent Variable
- Latent Variable Correlations
- Loading Estimation
- Empirical Underidentification

- Latent variables are not measurable, so there are no units by which to measure them, and the model is not identified
- Thus, we have to set their scale.
- This can be done in one of two ways.[2]

[2] There is a third if the LV is part of a structural model.

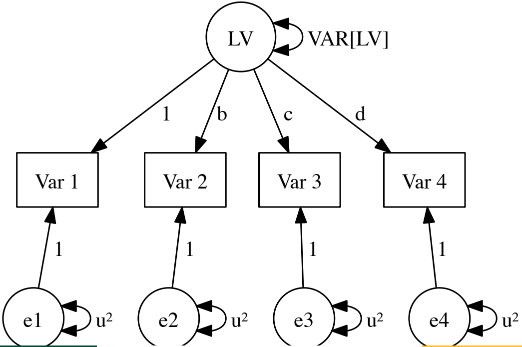
- This can be done in one of two ways:
  1. Set the latent variable's scale to 1 and mean to 0
     - This, in effect, makes it a standardized variable (i.e, on a Z-scale)
     - Moreover, if the indictor variables are standardized (or input a correlation matrix), this makes the factor loadings standardized regression weights.



Latent Variable Identification

Scaling the Latent Variable

- This can be done in one of two ways:
  2. Constrain a single factor loading for each latent variable to an arbitrary value (usually unity)
     - This gives the LV the same measurement unit as the MV
     - The variable whose loading is constrained is a *marker variable*
     - Usually want marker variable to be a “good representative” of the LV



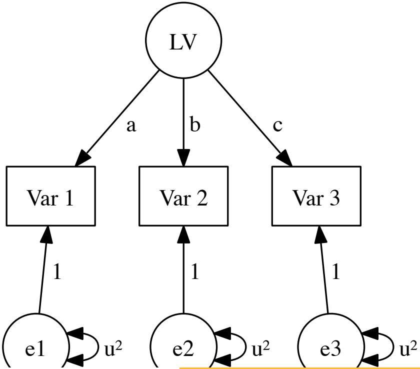
Talk Outline

Latent Variable Identification

- Scaling the Latent Variable
- Number of Indicators for a Latent Variable
- Latent Variable Correlations
- Loading Estimation
- Empirical Underidentification

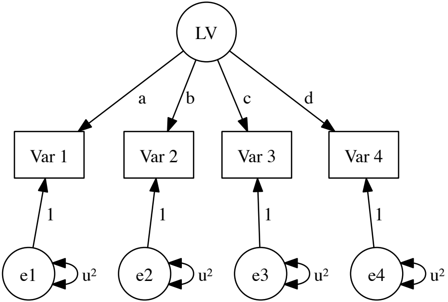
- If a latent variable has at least ___four___ indicators and none of the residual variances are correlated, then there should not be a problem with identification.
  - Why four?

	X_1	X_2	X_3
X_1	σ^2_{11}		
X_2	σ^2_{12}	σ^2_{22}	
X_3	σ^2_{13}	σ^2_{23}	σ^2_{33}



- If a latent variable has at least *four* indicators and none of the residual variances are correlated, then there should not be a problem with identification.
    - Why four?

	X_1	X_2	X_3	X_4
X_1	σ^2_{11}			
X_2	σ^2_{12}	σ^2_{22}		
X_3	σ^2_{13}	σ^2_{23}	σ^2_{33}	
X_4	σ^2_{14}	σ^2_{24}	σ^2_{34}	σ^2_{44}



- If you cannot have at least four indicators, the the model can still be identified if:
  - The construct has at least three indicators *or*
  - The construct has at least two indicators *or*
  - The construct has one indicator

- The construct has at least three indicators (and the error variances are uncorrelated)
  - Just identified

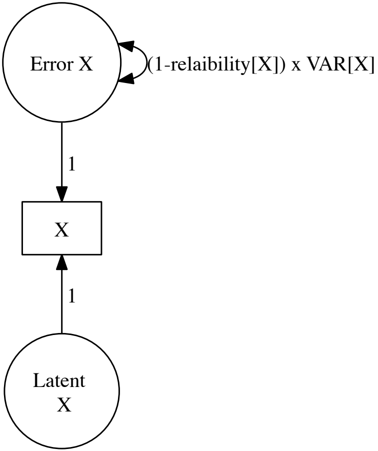
- The construct has at least two indicators (and the error variances are uncorrelated) *and*
  - The indicators' loadings are set equal to each other

Latent Variable Identification

Number of Indicators for a Latent Variable

- The construct has one indicator *and either*
  - Its error variance is fixed to some value
    - Usually zero (which means there is perfect reliability) *or*
    - $1 - r_{XX'}\sigma^2_X$ (where $r_{XX'}$ and σ^2_X are a variable's reliability and variance, respectively),
  - Constrain the loading and error variance, and estimate the variable's variance.
- This will be discussed more in the *psychometrics* lecture.

- Single indicator latent variable



Talk Outline

Latent Variable Identification

Scaling the Latent Variable

Number of Indicators for a Latent Variable

Latent Variable Correlations

Loading Estimation

Empirical Underidentification

Latent Variable Identification

Latent Variable Correlations

- If there is more than one latent variable, then for every pair of latent variables, either
  - There is, at least, one indicator that does not have a correlated measurement error with an indicator from another latent variable, *or*
  - The correlation between the pair of constructs is constrained to a specified value.

Talk Outline

Latent Variable Identification

- Scaling the Latent Variable
- Number of Indicators for a Latent Variable
- Latent Variable Correlations
- Loading Estimation
- Empirical Underidentification

- For every indicator, there must be at least one other indicator (of the same LV or a different LV) that does not have a correlated measurement error.

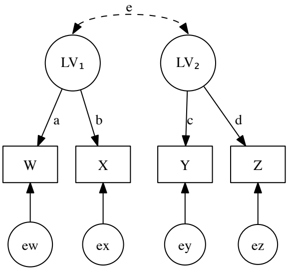
Talk Outline

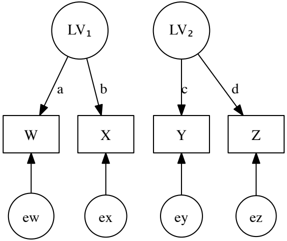
Latent Variable Identification

- Scaling the Latent Variable
- Number of Indicators for a Latent Variable
- Latent Variable Correlations
- Loading Estimation
- Empirical Underidentification

- *Empirical Underidentification* is when a model *should* be identified based on its structure, but it is not identified based on the sample data

- Empirical underidentification example
- As long as $|e| > 0$, the model on the left is identified because there are 10 pieces of information and 9 parameters to estimate.
- If $e = 0$, then the model is then two separate latent variable models (model on right)
  - For both models, there is $2 \times 3/2 = 3$ pieces of information, but 4 parameters to estimate, making them underidentified.





Mediation

Table of Contents

Mediation

Example

Mediation Effect Sizes

Structural Equation Modeling Examples

Single Group

Preparing Data

McIver, Carmines, & Zeller's (1980) Police Attitudes

κ^2

Talk Outline

Mediation

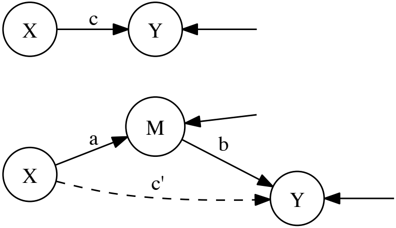
Example

Mediation Effect Sizes

Mediation

- Mediation models investigate how or why two (or more) variables are related.
- Mediation is when one (or more) variables explains the reason why two (or more variables) are related.

Mediation



- First, there is a relationship (via *c*) between variables X (exogenous) and Y (endogenous).
- Then, M is put into the model and is related to both X (via *a*) and Y (via *b*).
- After M was put into the model, then the relationship between X and Y dwindles (i.e., $c' < c$).

Talk Outline

Mediation

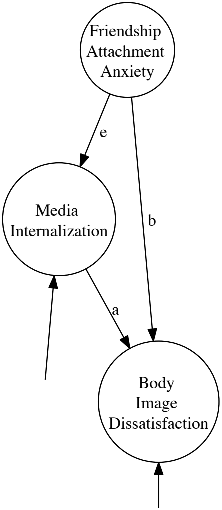
Example

Mediation Effect Sizes

Mediation

Example from Patton, Benedict, and Beaujean (submitted).

- *Media internalization* (awareness and attitudes toward prevailing sociocultural standards of attractiveness) was hypothesized to mediate the positive association between *attachment anxiety in friendships* and *body image dissatisfaction*.



Mediation Model

Mediation

- Their latent variables were each defined using item parcels.
- This involves making "subscales" from the instrument's items to make three or four (homogenous) continuous indicators.[3]

[3] See T. D. Little, Cunningham, Shahar, and Widaman (2002)

Mediation

	ECRF_P1	ECRF_P2	ECRF_P3	SATAQ_P1	SATAQ_P2	SATAQ_P3
ECRF_P1	49.20	43.73	41.18	16.92	15.75	17.74
ECRF_P2	43.73	55.93	44.54	17.54	16.00	17.78
ECRF_P3	41.18	44.54	56.96	17.71	16.31	17.95
SATAQ_P1	16.92	17.54	17.71	45.30	43.11	42.70
SATAQ_P2	15.75	16.00	16.31	43.11	48.10	43.82
SATAQ_P3	17.74	17.78	17.95	42.70	43.82	46.21
BSQ_P1	26.48	27.86	32.40	60.30	60.25	60.54
BSQ_P2	24.27	27.22	32.48	55.65	54.69	55.96
BSQ_P3	30.87	32.47	35.58	62.21	60.63	61.90

	BSQ_P1	BSQ_P2	BSQ_P3
ECRF_P1	26.48	24.27	30.87
ECRF_P2	27.86	27.22	32.47
ECRF_P3	32.40	32.48	35.58
SATAQ_P1	60.30	55.65	62.21
SATAQ_P2	60.25	54.69	60.63
SATAQ_P3	60.54	55.96	61.90
BSQ_P1	157.93	144.06	156.74
BSQ_P2	144.06	147.90	151.43
BSQ_P3	156.74	151.43	172.72

Mediation

- First, we need to test the measurement model part of the SEM.

```
#Measuremetn Model
MediationMeasurement.model<-'
#Measurement Models
AtchAnx =~ ECRF_P1 + ECRF_P2 + ECRF_P3
MediaInt =~ SATAQ_P1 + SATAQ_P2 + SATAQ_P3
BodImmDis =~ BSQ_P1 + BSQ_P2 + BSQ_P3
'

MediationMeasurement.fit<-cfa(MediationMeasurement.model, sample.cov=Mediation.cov,
    sample.nobs=321)
summary(MediationMeasurement.fit, fit.measures=TRUE, standardized=TRUE, rsquare=TRUE)
```

Mediation

- Selected results

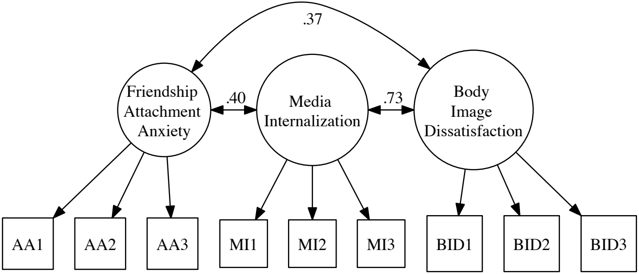
```
lavaan (0.5-9) converged normally after 147 iterations

  Number of observations                           321

  Estimator                                         ML
  Minimum Function Chi-square                   48.350
  Degrees of freedom                                24
  P-value                                        0.002

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.994
  Tucker-Lewis Index (TLI)                       0.991

Root Mean Square Error of Approximation:

  RMSEA                                          0.056
  90 Percent Confidence Interval          0.033  0.079
  P-value RMSEA <= 0.05                          0.303

Standardized Root Mean Square Residual:

  SRMR                                           0.019
```

Mediation

```
                        Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  AtchAnx =~
    ECRF_P1                1.000                               6.353    0.907
    ECRF_P2                1.078    0.044   24.476    0.000    6.850    0.917
    ECRF_P3                1.020    0.047   21.906    0.000    6.482    0.860
  MediaInt =~
    SATAQ_P1               1.000                               6.480    0.964
    SATAQ_P2               1.023    0.024   42.599    0.000    6.628    0.957
    SATAQ_P3               1.016    0.022   46.142    0.000    6.580    0.970
  BodImmDis =~
    BSQ_P1                 1.000                              12.207    0.973
    BSQ_P2                 0.964    0.019   50.198    0.000   11.765    0.969
    BSQ_P3                 1.050    0.020   53.673    0.000   12.815    0.977

Covariances:
  AtchAnx ~~
    MediaInt              16.310    2.618    6.229    0.000    0.396    0.396
    BodImmDis             28.275    4.859    5.819    0.000    0.365    0.365
  MediaInt ~~
    BodImmDis             58.057    5.643   10.289    0.000    0.734    0.734
```

- The indicators for each latent variable are pretty equivalent
- There is a relationship between attachment anxiety and body image dissatisfaction (path b, $r = .37$)

Mediation


.37
Friendship Attachment Anxiety
.40
Media Internalization
.73
Body Image Dissatisfaction
AA1
AA2
AA3
MI1
MI2
MI3
BID1
BID2
BID3

Mediation

- Specify the structural model

```
MediationStructural.model<-'
#Measurement Models
AtchAnx =~ ECRF_P1 + ECRF_P2 + ECRF_P3
MediaInt =~ SATAQ_P1 + SATAQ_P2 + SATAQ_P3
BodImmDis =~ BSQ_P1 + BSQ_P2 + BSQ_P3

#Structural Models
BodImmDis ~ a*MediaInt  + b*AtchAnx
MediaInt ~ e*AtchAnx
'

MediationStructural.fit<-sem(MediationStructural.model, sample.cov=Mediation.cov, sample.
    nobs=321)
summary(MediationStructural.fit, fit.measures=TRUE, standardized=TRUE, rsquare=TRUE)
```

- Selected results

```
lavaan (0.5-9) converged normally after 144 iterations

  Number of observations                           321

  Estimator                                         ML
  Minimum Function Chi-square                   48.350
  Degrees of freedom                                24
  P-value                                        0.002

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.994
  Tucker-Lewis Index (TLI)                       0.991

Root Mean Square Error of Approximation:

  RMSEA                                          0.056
  90 Percent Confidence Interval          0.033  0.079
  P-value RMSEA <= 0.05                          0.303

Standardized Root Mean Square Residual:

  SRMR                                           0.019
```

- Why are these the same as the correlation model?

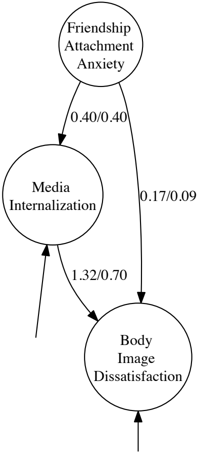
Mediation

- Selected results

```
                      Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Regressions:
  BodImmDis ~
    MediaInt   (a)       1.317    0.085   15.551    0.000    0.699    0.699
    AtchAnx    (b)       0.168    0.085    1.969    0.049    0.088    0.088
  MediaInt ~
    AtchAnx    (e)       0.404    0.057    7.144    0.000    0.396    0.396

R-Square:

    MediaInt             0.157
    BodImmDis            0.545
```

- Why are these *not* the same as the correlation model?

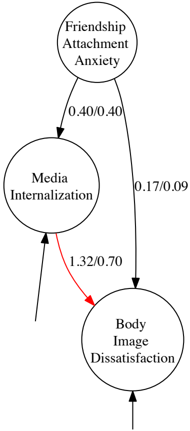
Mediation



Unstandardized/Standardized Coefficients

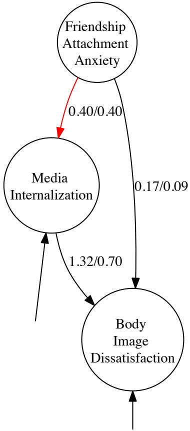
Mediation

- Media internalization is strongly related to body image dissatisfaction (path *a*, $b = .70$)



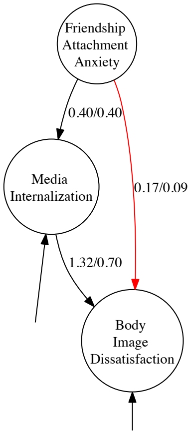
Mediation

- Media internalization is strongly related to body image dissatisfaction (path *a*, $b = .70$)
- Media internalization is moderately related to attachment anxiety (path *e*, $b = .40$)



Mediation

- Media internalization is strongly related to body image dissatisfaction (path *a*, $b = .70$)
- Media internalization is moderately related to attachment anxiety (path *e*, $b = .40$)
- The attachment anxiety-body image dissatisfaction relationship (path *b*), dwindles to almost 0 ($b = .09$) in the presence of these variables



Mediation

A table showing the effects of the model

Relationship	Direct Effect	Indirect Effect	Total
Anxiety $\rightarrow$ Body Image Dissatisfaction	.09	$.40 \times .70 = .28$	$.28 + .09 = .37$
Anxiety $\rightarrow$ Media Internalization	.40	–	.40
Media Internalization $\rightarrow$ Body Image Dissatisfaction	.70	–	.70

Mediation

- Compare the full model to a model where we remove path b.

```
MediationStructural2.model<-'
#Measurement Models
AtchAnx =~ ECRF_P1 + ECRF_P2 + ECRF_P3
MediaInt =~ SATAQ_P1 + SATAQ_P2 + SATAQ_P3
BodImmDis =~ BSQ_P1 + BSQ_P2 + BSQ_P3

#Structural Models
BodImmDis ~  a*MediaInt
MediaInt ~   e*AtchAnx
'
```

Mediation

- Selected results

```
lavaan (0.5-9) converged normally after 117 iterations

  Number of observations                           321

  Estimator                                         ML
  Minimum Function Chi-square                   52.194
  Degrees of freedom                                25
  P-value                                        0.001

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.993
  Tucker-Lewis Index (TLI)                       0.990

Root Mean Square Error of Approximation:

  RMSEA                                          0.058
  90 Percent Confidence Interval          0.036  0.080
  P-value RMSEA <= 0.05                          0.251

Standardized Root Mean Square Residual:

  SRMR                                           0.035
```

Mediation

- Selected results

```
                         Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Regressions:
  BodImmDis ~
    MediaInt   (a)      1.385    0.078   17.663    0.000    0.735    0.735
  MediaInt ~
    AtchAnx    (e)      0.407    0.056    7.204    0.000    0.399    0.399

R-Square:

    MediaInt            0.159
    BodImmDis           0.540
```

Talk Outline

Mediation

Example

Mediation Effect Sizes

- There are multiple measures of mediation effects
- See Preacher and Kelley (2011)

Index of Mediation

$$ab_{cs} = ab\frac{S_x}{S_y}$$

where
ab is the (unstandardized) indirect effect from the predictor to the outcome
S_x is the standard deviation of the predictor, and S_y is the standard deviation of the outcome

- the outcome changes by ab_{cs} standard deviations for every 1 SD increase in predictor indirectly via the mediator.
- Can (probably) be used to compare indirect effects across populations or studies when variables use different metrics in each population.

$R^2_{\text{Y.Mediated}}$

$$R^2_{4.5} \text{ or } R^2_{\text{Y.Mediated}} = r^2_{YM} - (R^2_{Y,MX} - r^2_{YX})$$

where
r^2 is the squared correlation
R^2 is the squared multiple correlation
M is the mediating variable,
Y is the outcome variable, and
X is the predictor variable.

- Overlap of the variances of X and Y that also overlaps with the variance of M
- The variance in Y that is common to both X and M but that can be attributed to neither alone

κ^2

$$\kappa^2 = \frac{ab}{\mathscr{M}(ab)}$$

where
ab is the (unstandardized) indirect effect from the predictor to the outcome
$\mathscr{M}$ an operator that return the most extreme possible observable value of the argument parameter with the same sign as the corresponding sample parameter estimate, and $\mathscr{M}(ab) = \mathscr{M}(a)\mathscr{M}(b)$

- The proportion of the maximum possible indirect effect that could have occurred, had the effects been as large as the design and data permitted.

$\mathscr{M}()$

$\mathscr{M}()$ is an operator that returns the most extreme possible observable value of the argument parameter *with the same sign as* the corresponding sample parameter estimate

$$\mathscr{M}(ab) = \mathscr{M}(a)\mathscr{M}(b)$$

where
ab is the (unstandardized) indirect effect from the predictor to the outcome
$\mathscr{M}(a) =$ most extreme value, *with the same sign as a*, in:

$$\left\{ \frac{\sigma_{YM}\sigma_{YX} \pm \sqrt{\sigma^2_M\sigma^2_Y - \sigma^2_{YM}}\sqrt{\sigma^2_X\sigma^2_Y - \sigma^2_{YX}}}{\sigma^2_X\sigma^2_Y} \right\}$$

$\mathscr{M}()$

and

$\mathscr{M}(b)$ = most extreme value, *with the same sign as b*, in:

$$\left\{ \pm \frac{\sqrt{\sigma_X^2 \sigma_Y^2 - \sigma_{YX}^2}}{\sqrt{\sigma_X^2 \sigma_M^2 - \sigma_{MX}^2}} \right\}$$

For the Example study

- Index of Mediation
  - $a = 0.40$, $b = 1.32$, $S_y = 8.23$, and $S_x = 6.35$
  - $\frac{(.40)(1.32)6.35}{8.23} = .41$
  - Body image dissatisfaction increases $.41$ standard deviations for every 1 SD increase in friendship attachment anxiety indirectly via the media internalization.
- $R^2_{\text{Y.Mediated}}$
  - $r^2_{YM} = 0.73^2$, $r^2_{YX} = .37^2$, and $R^2_{Y,MX} = 0.55$
  - $R^2_{\text{Y.Mediated}} = .53 - (.55 - .14) = .12$
  - 12% of the variance in body image dissatisfaction is explained by the friendship attachment anxiety and media internalization together.

Mediation

For the Example study

- κ^2
  - $a = .40$, $b = 1.32$, $\mathscr{M}(a) = .92$, and $\mathscr{M}(b) = 1.75$
  - $\frac{(.40)(1.32)}{(.92)(1.75)} = .30$
  - The indirect effect from friendship attachment anxiety to body image dissatisfaction through media internalization is about 30% as large as it could possibly be.

Talk Outline

Structural Equation Modeling Examples

- Single Group
- κ^2

Talk Outline

Structural Equation Modeling Examples

Single Group

κ^2

Structural Equation Modeling Examples

Single Group

Preparing Data

McIver, Carmines, & Zeller's (1980) Police Attitudes

κ^2

- Loehlins book CD has many of the correlation/covariance matrices in a `.txt` file (`DataMatrices.txt`).
- I copied data into a separate `.txt` files and named them according to the dataset.
- We can read in that data instead of inputting it manually.
- To do so
  - Read in the data as a matrix using `R`'s `matrix()` function
  - Name the rows and columns of the matrix using the `rownames()` and `colnames()` functions, respectively.

SEM Examples
Single Group: Preparing Data

- I suggest placing a copy of the files on your computer.
- Say they are located in a folder called *Loehlin*, in the `/Users/` directory (i.e., all files are in `/Users/alex_beaujean/Loehlin`).
- Can either:
  - Specify the name of the directory for each call, or
  - Point `R`'s working directory to that location using the `setwd()` function.

- My text file (`MaruyamaMcGarvey.txt`) looks like this:

```
1.00 .56 .17 .17 .16 .06 .16 .01 -.07 -.02 .05 .10 .10
 .56  1.00 .10 .30 .21 .15 .21  -.04  -.05  -.01 .04 .10 .17
 .17 .10  1.00 .19  -.04 .00 .28  -.04 .00 .04 .02  -.04  -.03
 .17 .30 .19  1.00 .50 .29 .40 .01 .13 .21 .28 .23 .32
 .16 .21  -.04 .50  1.00 .28 .19 .12 .27 .27 .24 .18 .40
 .06 .15 .00 .29 .28  1.00 .32 .10 .16 .14 .08 .09 .14
 .16 .21 .28 .40 .19 .32  1.00  -.06  -.07 .08 .13 .17 .17
 .01 -.04  -.04 .01 .12 .10  -.06  1.00 .42 .18 .07 .02 .08
-.07 -.05 .00 .13 .27 .16  -.07 .42  1.00 .31 .15 .08 .17
-.02 -.01 .04 .21 .27 .14 .08 .18 .31  1.00 .25 .08 .33
.05 .04 .02 .28 .24 .08 .13 .07 .15 .25  1.00 .59 .55
.10 .10 -.04 .23 .18 .09 .17 .02 .08 .08 .59  1.00 .49
.10 .17 -.03 .32 .40 .14 .17 .08 .17 .33 .55 .49  1.00
```

SEM Examples

Single Group: Preparing Data

- Now read in the text file

```
> MaruyamaMcGarvey.data<-matrix(scan(file="/Users/ Loehlin/MaruyamaMcGarvey.txt"), ncol
    =13)
> rownames(MaruyamaMcGarvey.data) <-colnames(MaruyamaMcGarvey.data) <-c("SEI","EDH","RP",
    "VACH","VGR","RAV","PEA","FEV","MEV","TEV","SP","PP","WP")
```

- Or

```
> setwd("/Users/Loehlin") #You will need to change this for your computer
> MaruyamaMcGarvey.data<-matrix(scan(file="MaruyamaMcGarvey.txt"),ncol=13)
> rownames(MaruyamaMcGarvey.data) <-colnames(MaruyamaMcGarvey.data) <-c("SEI","EDH","RP",
    "VACH","VGR","RAV","PEA","FEV","MEV","TEV","SP","PP","WP")
```

- The alternative is to type the covariance matrix directly into `R` .
- Since they are symmetric matrices, make use of the `diag()`, `upper.tri()` and `lower.tri()` functions.
- By default, `R` assumes entering the matrix data by columns.

- The following code will create a correlation matrix titled `CorM` that consists of the correlations among four variables.

```
> CorM<-diag(4) #4 x 4 Diagonal matrix
> CorM[lower.tri(CorM, diag=FALSE)]<-c(.85, .84, .68, .61, .59, .41) #lower triangle of
    matrix, order is by columns
> CorM[upper.tri(CorM, diag=FALSE)] <- CorM[lower.tri(CorM)] #make matrix full
```

- Names to the variables (rows/columns) using the `rownames()` and `colnames()` functions.

```
> CorMNames<-c("Var1", "Var2", "Var3", "Var4") #Names of the variables
> rownames(CorM)<-colnames(CorM)<-CorMNames #Gives row and column names
```

```
> CorM

     Var1 Var2 Var3 Var4
Var1 1.00 0.85 0.84 0.61
Var2 0.85 1.00 0.68 0.59
Var3 0.84 0.61 1.00 0.41
Var4 0.68 0.59 0.41 1.00
```

Talk Outline

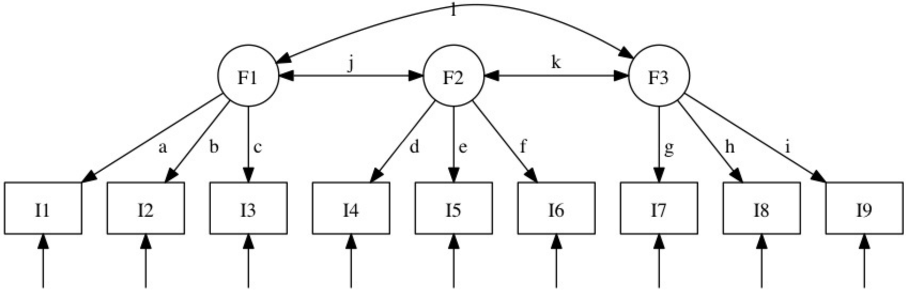
Structural Equation Modeling Examples

- Single Group
  - Preparing Data
  - McIver, Carmines, & Zeller's (1980) Police Attitudes

κ^2

SEM Examples

Single Group: Police Attitudes



McIver et al. (1980) Police Attitudes Model (Loehlin, 2004, (Figure 3.4))

- Enter the Data

```
rownames(McIver.data) <-colnames(McIver.data) <-c("PS", "RE", "RT", "HO", "CO", "ET", "BU
    ", "VA", "RO")
```

- Specify the Model

```
#Latent variable structure
F1 =~ a*PS + b*RE + c*RT
F2 =~ d*HO + e*CO + f*ET
F3 =~ g*BU + h*VA + i*RO

#Variances
F1 ~~ j*F2
F1 ~~ k*F3
F2 ~~ l*F3
'
```

- Estimate the model and obtain the fit statistics:

```
> fitMeasures(model.3.4.fit, fit.measure=c("chisq", "df", "pvalue", "rmsea"))

  chisq       df   pvalue    rmsea
226.232   24.000    0.000    0.028
```

SEM Examples

Single Group: Police Attitudes

- The `parameterEstimates()` function will return both the factor pattern coefficients and factor correlations

```
> parameterEstimates(model.3.4.fit, ci=FALSE)
   lhs op rhs label    est    se       z pvalue
1   F1 =~  PS     a  0.742 0.010  73.585      0
2   F1 =~  RE     b  0.653 0.010  64.521      0
3   F1 =~  RT     c  0.565 0.010  55.097      0
4   F2 =~  HO     d  0.750 0.010  76.609      0
5   F2 =~  CO     e  0.681 0.010  69.090      0
6   F2 =~  ET     f  0.650 0.010  65.741      0
7   F3 =~  BU     g  0.796 0.010  79.401      0
8   F3 =~  VA     h  0.725 0.010  72.595      0
9   F3 =~  RO     i  0.590 0.010  59.223      0
10  F1 ~~  F2     j  0.619 0.010  62.306      0
11  F1 ~~  F3     k -0.407 0.011 -35.435      0
12  F2 ~~  F3     l -0.239 0.012 -19.746      0
13  PS ~~  PS        0.450 0.011  41.854      0
14  RE ~~  RE        0.574 0.011  54.152      0
15  RT ~~  RT        0.681 0.011  61.955      0
16  HO ~~  HO        0.437 0.010  42.866      0
17  CO ~~  CO        0.536 0.010  52.893      0
18  ET ~~  ET        0.577 0.010  56.357      0
19  BU ~~  BU        0.366 0.011  33.118      0
20  VA ~~  VA        0.475 0.010  45.409      0
21  RO ~~  RO        0.652 0.011  61.910      0
```

- The factor pattern coefficients have the =~ operation, and the factor correlations use the ~ operation, e.g., `F1 ~ F2`.
- To obtain the communalities, use matrix algebra-based procure (or *rsquare=TRUE* argument in `summary()` function)

```
> fig3.4.Parms<-inspect(model.3.4.fit, "parameter.estimates")
> diag(fig3.4.Parms$lambda %*% t(fig3.4.Parms$lambda)) #communalities

  PS   RE   RT   HO   CO   ET   BU   VA   RO
0.55 0.43 0.32 0.56 0.46 0.42 0.63 0.53 0.35
```

- The `resid()` function will return the residual covariances and means

```
> resid(model.3.4.fit) # residual correlations and means (means not used in this example
    , so it returns a vector of 0s)
$cov
   PS     RE     RT     HO     CO     ET     BU     VA     RO
PS  0.000
RE  0.016  0.000
RT -0.009 -0.019  0.000
HO -0.015 -0.013  0.038  0.000
CO -0.033 -0.015  0.032  0.009  0.000
ET  0.001  0.007  0.063 -0.008 -0.003  0.000
BU  0.000  0.021  0.013  0.013  0.019 -0.026  0.000
VA -0.011  0.002  0.006  0.020  0.028 -0.017  0.003  0.000
RO -0.022 -0.023 -0.005 -0.044 -0.004 -0.038  0.000 -0.007  0.000

$mean
PS RE RT HO CO ET BU VA RO
 0  0  0  0  0  0  0  0  0
```

- Loehlin specifies two alternative models for this data.
- Revised Model 1

```
model.3.4.rev1<-'
#Latent variable structure
F1 =~ a*PS + b*RE + c*RT
F2 =~ d*HO + e*CO + f*ET + RT
F3 =~ g*BU + h*VA + i*RO

#Variances
F1 ~~ j*F2
F1 ~~ k*F3
F2 ~~ l*F3
'

> model.3.4.rev1.fit<-cfa(model.3.4.rev1, sample.cov=McIver.data, sample.nobs=11000, std.
    lv=TRUE)
```

- Revised Model 2

```
model.3.4.rev2<-'
#Latent variable structure
F1 =~ a*PS + b*RE + c*RT
F2 =~ d*HO + e*CO + f*ET + RT
F3 =~ g*BU + h*VA + i*RO

#Variances
F1 ~~ j*F2
F1 ~~ k*F3
F2 ~~ l*F3

#covariances
HO ~~ RO
RE ~~ BU
'

> model.3.4.rev2.fit<-cfa(model.3.4.rev2, sample.cov=McIver.data, sample.nobs=11000, std.
    lv=TRUE)
```

- To compare the three models using the χ^2 test, we can use the `anova()` function

```
> anova(model.3.4.rev2.fit, model.3.4.rev1.fit, model.3.4.fit)
Chi Square Difference Test

                 Df    AIC    BIC   Chisq Chisq diff Df diff Pr(>Chisq)
model3.4rev2.fit 21 256938 257113  83.611
model3.4rev1.fit 23 256978 257139 127.317     43.707       2   3.23e-10 ***
model3.4fit      24 257075 257228 226.232     98.915       1  < 2.2e-16 ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "." 0.1 " " 1
```

Talk Outline

Structural Equation Modeling Examples

- Single Group
- κ^2

κ^2 (Preacher & Kelley, 2011)

$$\kappa^2 = \frac{ab}{\mathscr{M}(ab)}$$

where

ab is the (unstandardized) indirect effect from the predictor to the outcome
$\mathscr{M}$ an operator that return the most extreme possible observable value of the argument parameter with the same sign as the corresponding sample parameter estimate, and $\mathscr{M}(ab) = \mathscr{M}(a)\mathscr{M}(b)$

κ^2

$$\kappa^2 = \frac{ab}{\mathscr{M}(ab)}$$

- The proportion of the maximum possible indirect effect that could have occurred, had the effects been as large as the design and data permitted.

$\mathscr{M}()$

$\mathscr{M}()$ is an operator that returns the most extreme possible observable value of the argument parameter *with the same sign as* the corresponding sample parameter estimate

$$\mathscr{M}(ab) = \mathscr{M}(a)\mathscr{M}(b)$$

where

ab is the (unstandardized) indirect effect from the predictor to the outcome

$\mathscr{M}(a) =$ most extreme value, *with the same sign as a*, in:

$$\left\{ \frac{\sigma_{YM}\sigma_{YX} \pm \sqrt{\sigma^2_M\sigma^2_Y - \sigma^2_{YM}}\sqrt{\sigma^2_X\sigma^2_Y - \sigma^2_{YX}}}{\sigma^2_X\sigma^2_Y} \right\}$$

SEM Examples

κ^2

$\mathscr{M}()$

and

$\mathscr{M}(b)$ = most extreme value, *with the same sign as b*, in:

$$\left\{ \pm \frac{\sqrt{\sigma_X^2 \sigma_Y^2 - \sigma_{YX}^2}}{\sqrt{\sigma_X^2 \sigma_M^2 - \sigma_{MX}^2}} \right\}$$

SEM Examples

κ^2

- You could calculate this by hand every time you have data to analyze
  - or, you would write an `R` function once and use it for all subsequent calculations

```
kappa2<-function(S,samp.size){
#require lavaan
 if (!require(lavaan))
stop("You must have lavaan installed to use kappa2")
#S needs to be a covariance matrix in the order of X,M,Y
 if (!(is.matrix(S)))
        stop("Data should be a matrix")
if (length(samp.size)!=1)
        stop("sample size should be a single number")
colnames(S)<-rownames(S)<-c("X", "M", "Y")

mediation.model<-'
Y ~  c*X  + b*M
M ~ a*X
'
mediation.fit<-sem(mediation.model, sample.cov=S, sample.nobs=samp.size)
parm.est<-parameterEstimates(mediation.fit)
#Path coefficients
a<-parm.est[which(parm.est$label=="a"), "est"]
b<-parm.est[which(parm.est$label=="b"), "est"]
ab<-a*b

#original vcov
Smx<-S[2,1]
Syx<-S[3,1]
Sym<-S[3,2]
Sx2<-S[1,1]
Sm2<-S[2,2]
```

SEM Examples (cont.)

κ^2

```
Sy2<-S[3,3]

####Effect size
# max a
maxa1<-(Sym*Syx - sqrt(Sm2*Sy2-Sym^2)*                sqrt(Sx2*Sy2 - Syx^2))/(Sx2*Sy2)
maxa2<-(Sym*Syx + sqrt(Sm2*Sy2-Sym^2)*                sqrt(Sx2*Sy2 - Syx^2))/(Sx2*Sy2)
maxa<-ifelse(sign(a)==sign(maxa1), maxa1, maxa2)

# max b
maxb1<-sqrt(Sx2*Sy2 - Syx^2)/sqrt(Sx2*Sm2 - Smx^2)
maxb2<- -1*sqrt(Sx2*Sy2 - Syx^2)/sqrt(Sx2*Sm2 - Smx^2)
maxb<-ifelse(sign(b)==sign(maxb1), maxb1, maxb2)

#max a * max b
maxab<- maxa * maxb

#kappa^2
kappa2<- ab/maxab
#max a * max b
maxab<- maxa * maxb

#kappa^2
kappa2<- a*b/maxab
list(a=a, b=b, ab=ab, maxa=maxa, maxb=maxb, maxab=maxab, kappa2=kappa2)
}
```

SEM Examples

κ^2

	VAC (*X*)	ATD (*M*)	DVB (*Y*)
VAC (*X*)	2.268	*.291*	*−.190*
ATD (*M*)	0.662	2.276	*−.493*
DVB (*Y*)	−0.087	−0.226	0.092
M	7.158	5.893	1.649

Note. Numbers on the diagonal are variances, those below the diagonal are covariances, and those above the diagonal (italicized) are correlations. VAC = (higher) achievement values; ATD = (more intolerant) attitude toward deviance; DVB = (more) deviant behavior.

Preacher and Kelley (2011) Data (from Jessor and Jessor's [1991] *Socialization of problem behavior in youth* study)

SEM Examples (cont.)

κ^2

```
> S<-matrix(c(2.2683, 0.6615,  -0.0869,
0.6615, 2.2764, -0.2259,
-0.0869, -0.2259, 0.0922), ncol=3)
> colnames(S)<-rownames(S)<-c("X", "M", "Y")
> S
        X       M       Y
X  2.2683  0.6615 -0.0869
M  0.6615  2.2764 -0.2259
Y -0.0869 -0.2259  0.0922
```

SEM Examples

κ^2

```
> kappa2(S,100)
$a
[1] 0.2916281

$b
[1] -0.09626046

$ab
[1] -0.02807225

$maxa
[1] 0.9495158

$maxb
[1] -0.2065304

$maxab
[1] -0.1961039

$kappa2
[1] 0.1431499
```

$$k^2 = \hat{\kappa}^2 = \frac{\hat{a}\hat{b}}{\mathfrak{M}(\hat{a}\hat{b})} = \frac{-.0281}{-.1961} = .143, \qquad (44)$$

Preacher & Kelley's (2011) κ^2

Multiple Groups: Invariance

Table of Contents

Invariance

Example

Talk Outline

Invariance

Example

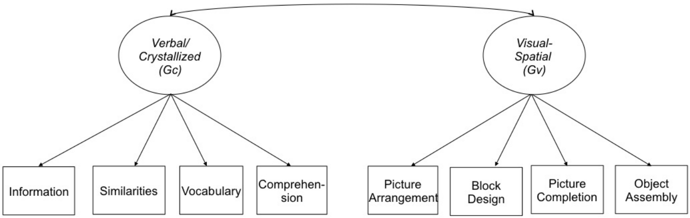
Talk Outline

Invariance

- Example

- Taken from Beaujean, Freeman, Youngstrom, and Carlson (2012)
- Question: Is structure of cognitive ability the same in youths with and without manic symptoms?
- Sample: 81 youths with manic symptoms; 200 youth from WISC-III norming sample (age 9)

Invariance
Example



```
library(lavaan)
##Manic data
manic.means
<-c(10.09,12.07, 10.25, 9.96, 10.90, 11.24, 10.30, 10.44)
manic.sd
<-c(3.06, 3.53, 3.18, 2.85, 2.49, 3.95, 3.35, 3.13)
manic.cor<-matrix(, nrow=8, ncol=8)
manic.cor[lower.tri(manic.cor, diag=TRUE)]<-c(1.00, 0.72,
0.66, 0.65, 0.40, 0.29, 0.36, 0.38, 1.00, 0.78, 0.74,
0.56, 0.35, 0.46, 0.42, 1.00, 0.75, 0.57, 0.39, 0.49,
0.49, 1.00, 0.58, 0.46, 0.33, 0.40, 1.00, 0.52, 0.43,
0.49, 1.00, 0.47, 0.47, 1.00, 0.62, 1.00)
manic.cor[upper.tri(manic.cor, diag=TRUE)]
<-t(manic.cor)[upper.tri(manic.cor, diag=TRUE)]
dimnames(manic.cor)<-list(c("Info", "Sim", "Vocab",
"Comp", "PicComp", "PicArr", "BlkDsgn", "ObjAsmb"),
c("Info", "Sim", "Vocab", "Comp", "PicComp", "PicArr",
"BlkDsgn", "ObjAsmb"))
manic.cov<-cor2cov(manic.cor,manic.sd)
```

```
#Normng Data
norming.means
<-c(10.10, 10.30, 9.80, 10.10, 10.10, 10.10, 9.90, 10.20)
norming.sd
<-c(3.10, 2.90, 3.00, 2.90, 3.20, 3.30, 3.40, 3.30)
norming.cor<-matrix(, nrow=8, ncol=8)
norming.cor[lower.tri(norming.cor, diag=TRUE)]<-c(1.00, 0.65,
0.68, 0.49, 0.45, 0.34, 0.50, 0.42, 1.00, 0.72, 0.53,
0.49, 0.31, 0.50, 0.48, 1.00, 0.58, 0.47, 0.30, 0.40,
0.44, 1.00, 0.40, 0.30, 0.33, 0.33, 1.00, 0.36, 0.48,
0.47, 1.00, 0.32, 0.33, 1.00, 0.59, 1.00)
norming.cor[upper.tri(norming.cor, diag=TRUE)]
<-t(norming.cor)[upper.tri(norming.cor, diag=TRUE)]
dimnames(norming.cor)<-list(c("Info", "Sim", "Vocab",
"Comp", "PicComp", "PicArr", "BlkDsgn", "ObjAsmb"),
c("Info", "Sim", "Vocab", "Comp", "PicComp", "PicArr",
"BlkDsgn", "ObjAsmb"))
norming.cov<-cor2cov(norming.cor,norming.sd)
```

```
#Function to calculate Mcdonald's NCI
Mc<-function (object, digits=3){
fit <- inspect(object, "fit") #lavaan's default output
chisq = unlist(fit["chisq"])# unlist(fit["chisq"]) #model Chi-square
df <- unlist(fit["df"]) #model df
n <- object@SampleStats@ntotal
ncp <- max(chisq - df,0)  #non-centrality parameter
d<- ncp/(n-1)  #scaled non-centrality parameter
Mc = exp((d)*-.5)  #McDonald's non-centrality index
Mc
}
```

Invariance

Example

```
> manic.model<-'
+ gc =~ Info + Sim + Vocab + Comp
+ gv =~ PicComp + PicArr + BlkDsgn + ObjAsmb
+ '
> manic.fit<-cfa(manic.model, sample.cov=manic.cov, sample.nobs=81, sample.mean=manic.
     means, meanstructure=TRUE)
> fitMeasures(manic.fit, c("chisq", "df", "cfi", "rmsea", "srmr"))
 chisq     df    cfi  rmsea   srmr
29.188 19.000  0.971  0.081  0.047
> Mc(manic.fit)
[1] 0.9383099
```

```
> norming.model<-'
+ gc =~ Info + Sim + Vocab + Comp
+ gv =~ PicComp + PicArr + BlkDsgn + ObjAsmb
+ '
> norming.fit<-cfa(norming.model, sample.cov=norming.cov, sample.nobs=200, sample.mean=
    norming.means, meanstructure=TRUE)
> # summary(norming.fit, fit.measures=TRUE )
> fitMeasures(norming.fit, c("chisq", "df", "cfi", "rmsea", "srmr"))
 chisq     df    cfi  rmsea   srmr
24.211 19.000  0.992  0.037  0.029
> Mc(norming.fit)
[1] 0.986993
```

Invariance

Example

```
#Combine the data sets into a single list

combined.cor<-list(manic=manic.cor, norming=norming.cor)
combined.cov<-list(manic=manic.cov, norming=norming.cov)
combined.n<-list(manic=81, norming=200)
combined.means<-list(manic=manic.means, norming=norming.means)

combined.model<-'
gc =~ Info + Sim + Vocab + Comp
gv =~ PicComp + PicArr + BlkDsgn + ObjAsmb
'
```

Invariance

Example

```
> # Configural Invariance
> configural.fit<-cfa(combined.model, sample.cov=combined.cov, sample.nobs=combined.n,
    sample.mean=combined.means)
> fitMeasures(configural.fit, c("chisq", "df", "cfi", "rmsea", "srmr"))
 chisq     df    cfi  rmsea   srmr
53.399 38.000  0.985  0.054  0.038
> Mc(configural.fit)
[1] 0.9728769
```

```
> # Metric Invariance
> metric.fit<-cfa(combined.model, sample.cov=combined.cov, sample.nobs=combined.n, sample.mean=combined.means, group.equal=c("loadings"))
> fitMeasures(metric.fit, c("chisq", "df", "cfi", "rmsea", "srmr"))
 chisq     df    cfi  rmsea   srmr
66.012 44.000  0.979  0.060  0.061
> Mc(metric.fit)
[1] 0.9614559
```

```
# Scalar Invariance
> scalar.fit<-cfa(combined.model, sample.cov=combined.cov, sample.nobs=combined.n, sample
    .mean=combined.means, group.equal=c("loadings", "intercepts"))
> fitMeasures(scalar.fit, c("chisq", "df", "cfi", "rmsea", "srmr"))
  chisq      df     cfi   rmsea    srmr
109.107  50.000   0.942   0.092   0.064
> Mc(scalar.fit)
[1] 0.8998313
```

```
> #Scalar Invariance 2: Allow intercepts for Similarities subtest to be freely estimated
> scalar.fit2<-cfa(combined.model, sample.cov=combined.cov, sample.nobs=combined.n,
    sample.mean=combined.means, group.equal=c("loadings", "intercepts"), group.partial=c
    ("Sim~1"))
> fitMeasures(scalar.fit2, c("chisq", "df", "cfi", "rmsea", "srmr"))
 chisq     df    cfi  rmsea   srmr
75.943 49.000  0.974  0.063  0.058
> Mc(scalar.fit2)
[1] 0.9530259
```

```
> #Strict Invariance
> strict.fit<-cfa(combined.model, sample.cov=combined.cov, sample.nobs=combined.n, sample
    .mean=combined.means, group.equal=c("loadings", "intercepts", "residuals"), group.
    partial=c("Sim~1"))
> fitMeasures(strict.fit, c("chisq", "df", "cfi", "rmsea", "srmr"))
 chisq     df    cfi  rmsea   srmr
94.901 57.000  0.963  0.069  0.070
> Mc(strict.fit)
[1] 0.9345593
```

Invariance

Example

```
> #Strict Invariance 2: Allow residual variances for Picture Comp, Comprehen, and Pict
    Arrangement subtests to be freely estimated
> strict.fit2<-cfa(combined.model, sample.cov=combined.cov, sample.nobs=combined.n,
    sample.mean=combined.means, group.equal=c("loadings", "intercepts", "residuals"),
    group.partial=c("Sim~1", "PicComp~~PicComp", "Comp~~Comp", "PicArr~~PicArr"))
> fitMeasures(strict.fit2, c("chisq", "df", "cfi", "rmsea", "srmr"))
 chisq     df    cfi  rmsea   srmr
76.930 54.000  0.978  0.055  0.059
> Mc(strict.fit2)
[1] 0.9598809
```

Invariance

Example

```
> #Factor Variances
> factor.var.fit<-cfa(combined.model, sample.cov=combined.cov, sample.nobs=combined.n,
    sample.mean=combined.means, group.equal=c("loadings", "intercepts", "residuals", "lv
    .variances"), group.partial=c("Sim~1", "PicComp~~PicComp", "Comp~~Comp", "PicArr
    ~~PicArr"))
> fitMeasures(factor.var.fit, c("chisq", "df", "cfi", "rmsea", "srmr"))
 chisq     df    cfi  rmsea   srmr
80.030 56.000  0.977  0.055  0.071
> Mc(factor.var.fit)
[1] 0.9579975
```

Invariance

Example

```
> #Factor Covariances
> factor.covar.fit<-cfa(combined.model, sample.cov=combined.cov, sample.nobs=combined.n,
    sample.mean=combined.means, group.equal=c("loadings", "intercepts", "residuals", "lv
    .variances", "lv.covariances"), group.partial=c("Sim~1", "PicComp~~PicComp", "Comp
    ~~Comp", "PicArr~~PicArr"))
> fitMeasures(factor.covar.fit, c("chisq", "df", "cfi", "rmsea", "srmr"))
 chisq     df    cfi  rmsea   srmr
80.415 57.000  0.977  0.054  0.070
> Mc(factor.covar.fit)
[1] 0.9590489
```

```
> #Factor Means
> factor.means.fit<-cfa(combined.model, sample.cov=combined.cov, sample.nobs=combined.n,
    sample.mean=combined.means, group.equal=c("loadings", "intercepts", "residuals", "lv
    .variances", "lv.covariances", "means"), group.partial=c("Sim~1", "PicComp
    ~~PicComp", "Comp~~Comp", "PicArr~~PicArr"))
> fitMeasures(factor.means.fit, c("chisq", "df", "cfi", "rmsea", "srmr"))
 chisq     df    cfi  rmsea   srmr
83.999 59.000  0.976  0.055  0.074
> Mc(factor.means.fit)
[1] 0.9563401
```

```
> summary(factor.means.fit, standardized=TRUE)
lavaan (0.5-9) converged normally after  81 iterations

  Number of observations per group
  manic                                             81
  norming                                          200

  Estimator                                         ML
  Minimum Function Chi-square                   83.999
  Degrees of freedom                                59
  P-value                                        0.018

Chi-square for each group:

  manic                                         49.617
  norming                                       34.382

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard
```

Invariance

Example

```
Group 1 [manic]:

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  gc =~
    Info              1.000                               2.396    0.778
    Sim               1.105    0.073   15.235    0.000    2.646    0.858
    Vocab             1.099    0.072   15.343    0.000    2.633    0.864
    Comp              0.863    0.069   12.532    0.000    2.067    0.794
  gv =~
    PicComp           1.000                               2.030    0.766
    PicArr            0.906    0.118    7.668    0.000    1.839    0.496
    BlkDsgn           1.214    0.120   10.133    0.000    2.464    0.729
    ObjAsmb           1.171    0.115   10.169    0.000    2.376    0.733

Covariances:
  gc ~~
    gv                3.683    0.503    7.325    0.000    0.757    0.757
```

Invariance

Example

```
Intercepts:
    Info              10.097    0.184   54.997    0.000   10.097    3.281
    Sim               11.902    0.253   47.054    0.000   11.902    3.858
    Vocab              9.930    0.182   54.599    0.000    9.930    3.257
    Comp              10.011    0.170   58.800    0.000   10.011    3.844
    PicComp           10.396    0.175   59.305    0.000   10.396    3.923
    PicArr            10.394    0.208   49.873    0.000   10.394    2.804
    BlkDsgn           10.015    0.202   49.692    0.000   10.015    2.964
    ObjAsmb           10.269    0.193   53.091    0.000   10.269    3.167
    gc                 0.000                               0.000    0.000
    gv                 0.000                               0.000    0.000

Variances:
    Info               3.732    0.381                      3.732    0.394
    Sim                2.512    0.312                      2.512    0.264
    Vocab              2.359    0.301                      2.359    0.254
    Comp               2.511    0.473                      2.511    0.370
    PicComp            2.902    0.613                      2.902    0.413
    PicArr            10.359    1.725                     10.359    0.754
    BlkDsgn            5.346    0.604                      5.346    0.468
    ObjAsmb            4.866    0.554                      4.866    0.463
    gc                 5.740    0.765                      1.000    1.000
    gv                 4.119    0.685                      1.000    1.000
```

Invariance

Example

```
Group 2 [norming]:

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  gc =~
    Info              1.000                               2.396    0.778
    Sim               1.105    0.073   15.235    0.000    2.646    0.858
    Vocab             1.099    0.072   15.343    0.000    2.633    0.864
    Comp              0.863    0.069   12.532    0.000    2.067    0.685
  gv =~
    PicComp           1.000                               2.030    0.650
    PicArr            0.906    0.118    7.668    0.000    1.839    0.538
    BlkDsgn           1.214    0.120   10.133    0.000    2.464    0.729
    ObjAsmb           1.171    0.115   10.169    0.000    2.376    0.733

Covariances:
  gc ~~
    gv                3.683    0.503    7.325    0.000    0.757    0.757
```

Invariance

Example

```
Intercepts:
    Info              10.097     0.184    54.997     0.000    10.097     3.281
    Sim               10.368     0.197    52.636     0.000    10.368     3.361
    Vocab              9.930     0.182    54.599     0.000     9.930     3.257
    Comp              10.011     0.170    58.800     0.000    10.011     3.319
    PicComp           10.396     0.175    59.305     0.000    10.396     3.328
    PicArr            10.394     0.208    49.873     0.000    10.394     3.039
    BlkDsgn           10.015     0.202    49.692     0.000    10.015     2.964
    ObjAsmb           10.269     0.193    53.091     0.000    10.269     3.167
    gc                 0.000                                   0.000     0.000
    gv                 0.000                                   0.000     0.000

Variances:
    Info               3.732     0.381                         3.732     0.394
    Sim                2.512     0.312                         2.512     0.264
    Vocab              2.359     0.301                         2.359     0.254
    Comp               4.828     0.534                         4.828     0.531
    PicComp            5.641     0.667                         5.641     0.578
    PicArr             8.318     0.912                         8.318     0.711
    BlkDsgn            5.346     0.604                         5.346     0.468
    ObjAsmb            4.866     0.554                         4.866     0.463
    gc                 5.740     0.765                         1.000     1.000
    gv                 4.119     0.685                         1.000     1.000
```

Invariance
Example

```
#test all invariance steps at once
library(semTools)
measurementInvariance(combined.model, sample.cov=combined.cov, sample.nobs=combined.n,
    sample.mean=combined.means)
```

```
Measurement invariance tests:

Model 1: configural invariance:
    chisq        df     pvalue        cfi      rmsea        bic
   53.399    38.000      0.050      0.985      0.054  10675.018

Model 2: weak invariance (equal loadings):
    chisq        df     pvalue        cfi      rmsea        bic
   66.012    44.000      0.017      0.979      0.060  10653.801

[Model 1 versus model 2]
  delta.chisq       delta.df delta.p.value      delta.cfi
       12.613          6.000         0.050          0.006

Model 3: strong invariance (equal loadings + intercepts):
    chisq        df     pvalue        cfi      rmsea        bic
  109.107    50.000      0.000      0.942      0.092  10753.280

[Model 1 versus model 3]
  delta.chisq       delta.df delta.p.value      delta.cfi
       55.708         12.000         0.000          0.043

[Model 2 versus model 3]
  delta.chisq       delta.df delta.p.value      delta.cfi
       43.095          6.000         0.000          0.036
```

```
Model 4: equal loadings + intercepts + means:
    chisq         df     pvalue        cfi      rmsea        bic
  112.413    52.000      0.000      0.941      0.091 10745.310

[Model 1 versus model 4]
  delta.chisq       delta.df delta.p.value     delta.cfi
       59.015         14.000         0.000         0.044

[Model 3 versus model 4]
  delta.chisq       delta.df delta.p.value     delta.cfi
        3.307          2.000         0.191         0.001
```

Multiple Groups: Behavior Genetic Models

Table of Contents

Behavior Genetic Models

- Behavior Genetic Theory
- Example 1: MZ and DZ Twins
- Example 2: Multiple Familial Relationships

Talk Outline

Behavior Genetic Models

- Behavior Genetic Theory
- Example 1: MZ and DZ Twins
- Example 2: Multiple Familial Relationships

Talk Outline

Behavior Genetic Models

- Behavior Genetic Theory
- Example 1: MZ and DZ Twins
- Example 2: Multiple Familial Relationships

- Behavior genetics (BG) is the study of the genetic and environmental influences on psychological traits
- Traditionally, these analysis have used "natural experiments" to study these relationships, such as twins reared in differing environments and adoption studies, as well as the general study of how similar/different siblings are to each other
- With the advent of more powerful computers and data analysis programs, however (especially the *Mx* program and its translation into the R language via *OpenMx*), the field has moved to using latent variable models for much of its analyses

- For the "classic" BG model, we will assume there is just one phenotype measured in both MZ and DZ twins
- The genetic influence on the the phenotype can be decomposed into
  - Additive effects of alleles at various loci,
  - Dominance effects of alleles at various loci, and
  - Epistatic interactions between loci.
- Often with human samples, epistatic and dominance effects are confounded, so are lumped into a single *non-additive* genetic effects category.

- We can decompose the environmental influence on the phenotype into
  - Effects due to a *shared* environment, such as being raised by the same parents in the same house (aka "between-family" effects); and
  - Effects due to an *unshared* environment, such as having different peers or attending different schools (aka "within-family" effects).
    - These unshared effects also include random environmental events, such as getting into automobile accident, as well as random measurement events (i.e, measurement error).

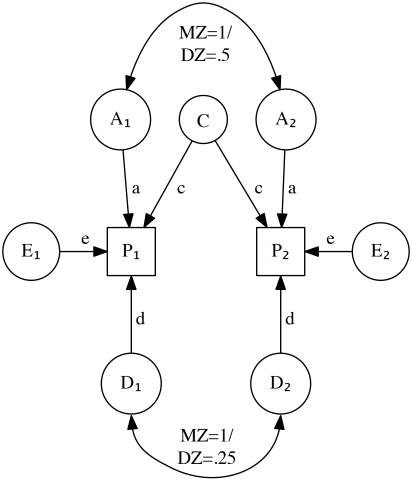
- For relatives *i* and *j*, their phenotypes, P_i and P_j, are assumed to be a linear functions of the additive genetic influence (A_i and A_j), non-additive influence (D_i and D_j), shared environmental influence $(C)i$ and C_j) and unshared environmental variance (E_i and E_j).
- Thus

$$\begin{aligned} P_1 &= a_1 A_1 + d_1 D_1 + c_1 C_1 + e_1 E_1 \\ P_2 &= a_2 A_2 + d_2 D_2 + c_2 C_2 + e_2 E_2 \end{aligned} \tag{1}$$

- For the same set of twins, we would not expect the influences do differ.
- That is, we would not expect, say, the heritability estimates or the shared environmental influence estimates to differ.
- Thus, we can simplify Equation 1 to

$$\begin{aligned} P_1 &= aA_1 + dD_1 + cC_1 + eE_1 \\ P_2 &= aA_2 + dD_2 + cC_2 + eE_2 \end{aligned} \tag{2}$$

- If twins (no matter what the zygosity) are reared together, the shared environment influence (C) is going to be the same.
- Likewise, by its definition, the non-shared environment is going to be completely different.
- From genetic theory, we know that for MZ twins, the genetic influence (i.e., A and D) on a trait will be the same for both twins.
- For DZ twins, on average, the additive genetic influence will only be $\frac{1}{2}$ the same and the non-additive genetic influence will be be $\frac{1}{4}$ the same.



ACDE Model

- For MZ and DZ twins reared together, the expected correlations on the measured phenotype are

$$_{MZ}r_{P_1,P_2} = a^2 + d^2 + c^2 \tag{3}$$

and

$$_{DZ}r_{P_1,P_2} = 0.5a^2 + 0.25d^2 + c^2 \tag{4}$$

respectively.

- The variance for the trait is

$$\sigma_P^2 = a^2 + d^2 + c^2 + e^2 \tag{5}$$

- Together, these three equations represent four unknown parameters (*a, b, c* and *d*), but use input from only three known statistics (${}_{MZ}r_{P_1,P_2}$, ${}_{DZ}r_{P_1,P_2}$, and σ^2_P).
- Thus, we can only estimate three of the four parameters.
- As it turns out, with just MZ and DZ twins reared together in the sample, *c* and *d* are confounded, so either c or d can be estimated within the same model.
  - To estimate the *c* and *d* parameters within the same model would require additional data (e.g., twins separated at birth, relatives of twins).

- The c or d parameters do not necessarily need to be estimated, if the hypothesis is that only additive genetic components and/or random environmental events are influencing the phenotype.
- With twin designs it is typical to test the the following series of models that postulate different genetic and environmental components are influencing behavior: ACE, ADE, AE, CE and E.

Talk Outline

Behavior Genetic Models

Behavior Genetic Theory

Example 1: MZ and DZ Twins

Example 2: Multiple Familial Relationships

- In `lavaan`, to input the data from multiple groups we need to combine the data from the separate groups into a single `list`.
- In `R`, a list is an ordered (and possibly named) collection of objects, gathered under one name.

- As an example, say we obtained BMI scores in a set of female monozygotic (MZ) and dizygotic (DZ) twins.[4]

	Twin 1	Twin 2
MZF T1	.725	.589
MZF T2	.589	.792
DZF T1	.779	.246
DZF T2	.246	.837

MZ n: 534; DZ n: 328

[4] Data taken from Neale and Maes (1992)

- First, we need to enter the covariance matrices for the MZ and DZ twins, separately.

```
#Young Female MZ Twins
MZFY<-matrix(c(.725,.589,.589,.792),nrow=2)
rownames(MZFY)<-c("P1", "P2")
colnames(MZFY)<-c("P1", "P2")
#Young Female DZ Twins
DZFY<-matrix(c(.779,.246,.246,.837),nrow=2)
rownames(DZFY)<-c("P1", "P2")
colnames(DZFY)<-c("P1", "P2")
```

- Next we need to combine the covariances and *n*s

```
bmi.cov<-list(MZ=MZFY,DZ=DZFY)
bmi.n<-list(MZ=534,DZ=328)
```

- ADE Model

```
bmi.ade.model<-'
#Genetic Model
A1=~ NA*P1 + c(a,a)*P1 + start(c(.5,.5))*P1
A2=~ NA*P2 + c(a,a)*P2 + start(c(.5,.5))*P2
D1 =~ NA*P1 + c(d,d)*P1
D2 =~ NA*P2 + c(d,d)*P2
#Variances
A1 ~~ 1*A1
A2 ~~ 1*A2
D1 ~~ 1*D1
D2 ~~ 1*D2
P1~~c(e2,e2)*P1
P2~~c(e2,e2)*P2
#covariances
A1 ~~ c(1,.5)*A2
A1 ~~ 0*D1 + 0*D2
A2 ~~ 0*D1 + 0*D2
D1 ~~ c(1,.25)*D2
'
```

- ACE Model

```
bmi.ace.model<-'
#Genetic Model
A1=~ NA*P1 + c(a,a)*P1 + start(c(.5,.5))*P1
A2=~ NA*P2 + c(a,a)*P2 + start(c(.5,.5))*P2
C1 =~ NA*P1 + c(c,c)*P1
C2 =~ NA*P2 + c(c,c)*P2
#Variances
A1 ~~ 1*A1
A2 ~~ 1*A2
C1 ~~ 1*C1
C2 ~~ 1*C2
P1~~c(e2,e2)*P1
P2~~c(e2,e2)*P2
#covariances
A1 ~~ c(1,.5)*A2
A1 ~~ 0*C1 + 0*C2
A2 ~~ 0*C1 + 0*C2
C1 ~~ c(1,1)*C2
'
```

- CE Model

```
bmi.ce.model<-'
#Genetic Model
C1 =~ NA*P1 + c(c,c)*P1
C2 =~ NA*P2 + c(c,c)*P2
#Variances
C1 ~~ 1*C1
C2 ~~ 1*C2
P1~~c(e2,e2)*P1
P2~~c(e2,e2)*P2
#covariances
C1 ~~ c(1,1)*C2
'
```

- AE Model

```
bmi.ae.model<-'
#Genetic Model
A1=~ NA*P1 + c(a,a)*P1 + start(c(.5,.5))*P1
A2=~ NA*P2 + c(a,a)*P2 + start(c(.5,.5))*P2
#Variances
A1 ~~ 1*A1
A2 ~~ 1*A2
P1~~c(e2,e2)*P1
P2~~c(e2,e2)*P2
#covariances
A1 ~~ c(1,.5)*A2
'
```

```
> fit.m<-c("chisq", "df", "pvalue", "aic", "rmsea", "srmr")
> bmi.ade.fit<-cfa(bmi.ade.model, sample.cov=bmi.cov, sample.nobs=bmi.n)
> fitMeasures(bmi.ade.fit, fit.m)
   chisq        df    pvalue       aic     rmsea      srmr
   3.704     3.000     0.295  3934.811     0.023     0.045
```

```
> bmi.ace.fit<-cfa(bmi.ace.model, sample.cov=bmi.cov, sample.nobs=bmi.n)
> fitMeasures(bmi.ace.fit, fit.m)
   chisq       df   pvalue      aic    rmsea     srmr
   8.040    3.000    0.045 3939.146    0.062    0.058
```

```
> bmi.ce.fit<-cfa(bmi.ce.model, sample.cov=bmi.cov, sample.nobs=bmi.n)
> fitMeasures(bmi.ce.fit, fit.m)
   chisq       df   pvalue      aic    rmsea     srmr
 160.372    4.000    0.000 4089.478    0.301    0.127
```

```
> bmi.ae.fit<-cfa(bmi.ace.model, sample.cov=bmi.cov, sample.nobs=bmi.n)
> fitMeasures(bmi.ae.fit, fit.m)
   chisq        df    pvalue       aic     rmsea      srmr
   8.040     3.000     0.045  3939.146     0.062     0.058
```

- The best fitting model is ADE model

```
> summary(bmi.ade.fit, standardized=TRUE)
Group 1 [MZ]:

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  A1 =~
    P1         (a)    0.562    0.139    4.053    0.000    0.562    0.636
  A2 =~
    P2         (a)    0.562    0.139    4.053    0.000    0.562    0.636
  D1 =~
    P1         (d)    0.543    0.140    3.874    0.000    0.543    0.615
  D2 =~
    P2         (d)    0.543    0.140    3.874    0.000    0.543    0.615

Covariances:
  A1 ~~
    A2                1.000                               1.000    1.000
    D1                0.000                               0.000    0.000
    D2                0.000                               0.000    0.000
  A2 ~~
    D1                0.000                               0.000    0.000
    D2                0.000                               0.000    0.000
  D1 ~~
    D2                1.000                               1.000    1.000
```

Behavior Genetic Models

Example 1: MZ and DZ Twins

```
Variances:
    A1                1.000                                 1.000    1.000
    A2                1.000                                 1.000    1.000
    D1                1.000                                 1.000    1.000
    D2                1.000                                 1.000    1.000
    P1       (e2)     0.170    0.010                        0.170    0.218
    P2       (e2)     0.170    0.010                        0.170    0.218
```

- $a^2 = .636^2 = .405$, $e^2 = .218$, $d^2 = .615^2 = .378$, & $c^2 = 0$.
- Random environment accounts for a relatively modest proportion of the total variation in BMI, 21.8%.
- Narrow heritability accounts for 40.5% of the total variance and $\frac{.405}{.405+.378} \times 100 =$ 51.7% of the broad heritability variance.

Talk Outline

Behavior Genetic Models

Behavior Genetic Theory

Example 1: MZ and DZ Twins

Example 2: Multiple Familial Relationships

```
> MZ1R<-matrix(c(1,.47,.47,1),nrow=2)
> dimnames(MZ1R)<-list(c("T1", "T2"),c("T1", "T2"))
> DZ2R<-matrix(c(1,.00,.00,1),nrow=2)
> dimnames(DZ2R)<-list(c("T1", "T2"),c("T1", "T2"))
> MZ3R<-matrix(c(1,.45,.45,1),nrow=2)
> dimnames(MZ3R)<-list(c("T1", "T2"),c("T1", "T2"))
> DZ4R<-matrix(c(1,.08,.08,1),nrow=2)
> dimnames(DZ4R)<-list(c("T1", "T2"),c("T1", "T2"))
> PA5R<-matrix(c(1,.07,.07,1),nrow=2)
> dimnames(PA5R)<-list(c("T1", "T2"),c("T1", "T2"))
> PA6R<-matrix(c(1,-.03,-.03,1),nrow=2)
> dimnames(PA6R)<-list(c("T1", "T2"),c("T1", "T2"))
> PN7R<-matrix(c(1,.22,.22,1),nrow=2)
> dimnames(PN7R)<-list(c("T1", "T2"),c("T1", "T2"))
> PN8R<-matrix(c(1,.13,.13,1),nrow=2)
> dimnames(PN8R)<-list(c("T1", "T2"),c("T1", "T2"))
> AS9R<-matrix(c(1,-.05,-.05,1),nrow=2)
> dimnames(AS9R)<-list(c("T1", "T2"),c("T1", "T2"))
> AS10R<-matrix(c(1,-.21,-.21,1),nrow=2)
> dimnames(AS10R)<-list(c("T1", "T2"),c("T1", "T2"))
```

```
> sib.cor<-list(MZ1=MZ1R,DZ2=DZ2R,MZ3=MZ3R,DZ4=DZ4R, PA5=PA5R, PA6=PA6R, PN7=PN7R, PN8=
    PN8R, AS9=AS9R, AS10=AS10R)
> sib.n<-list(MZ1=45,DZ2=34,MZ3=102,DZ4=119,PA5=257,PA6=271, PA7=56, PA8=54, PA9=48, PA10
    =80)
> sib.cor
$MZ1
     T1   T2
T1 1.00 0.47
T2 0.47 1.00

$DZ2
   T1 T2
T1  1  0
T2  0  1

$MZ3
     T1   T2
T1 1.00 0.45
T2 0.45 1.00

$DZ4
     T1   T2
T1 1.00 0.08
T2 0.08 1.00

$PA5
     T1   T2
T1 1.00 0.07
```

```
T2  0.07 1.00

$PA6
       T1    T2
T1  1.00 -0.03
T2 -0.03  1.00

$PN7
      T1   T2
T1 1.00 0.22
T2 0.22 1.00

$PN8
      T1   T2
T1 1.00 0.13
T2 0.13 1.00

$AS9
       T1    T2
T1  1.00 -0.05
T2 -0.05  1.00

$AS10
       T1    T2
T1  1.00 -0.21
T2 -0.21  1.00
```

The naming scheme I used for the data correlation matrix objects is: MZ; monozygotic twin, DZ: dizygotic twin, PA: parent-adopted child, PN: Parent-natural child, AS: Adopted siblings. Within each data matrix, I named the columns and rows `T1` and `T2`. The numbers match those in Loehlin's Table 4.12.

- Model 1: All *r*s equal (null model)

```
#All rs equal (null)
BG.model.1<-'
#Latent Variables
S1 =~ .87*T1
S2 =~ .87*T2

#covariances
S1 ~~ S2
'
```

- Model 2: h only

```
#h only
BG.model.2<-'
#Latent Variables
S1 =~ .87*T1
S2 =~ .87*T2

#Genetic Model
G1 =~ NA*S1 + c(h,h,h,h,h,h,h,h,h,h)*S1
G2 =~ NA*S2 + c(h,h,h,h,h,h,h,h,h,h)*S2

#Variances
G1 ~~ 1*G1
G2 ~~ 1*G2
S1 ~~ S1
S2 ~~ S2

#covariances
G1 ~~ c(1,.5,1,.5,0,0,.5,.5,0,0)*G2
'
```

- Model 3: $h + c$

```
# h + c model
BG.model.3<-'
#Latent Variables
S1 =~ .87*T1
S2 =~ .87*T2

#Genetic Model
G1 =~ NA*S1 + c(h,h,h,h,h,h,h,h,h,h)*S1
G2 =~ NA*S2 + c(h,h,h,h,h,h,h,h,h,h)*S2
C =~ NA*S1 + c(c,c,c,c,c,c,c,c,c,c)*S1 + c(c,c,c,c,c,c,c,c,c,c)*S2

#Variances
G1 ~~ 1*G1
G2 ~~ 1*G2
C ~~ 1*C
S1 ~~ S1
S2 ~~ S2

#covariances
G1 ~~ c(1,.5,1,.5,0,0,.5,.5,0,0)*G2
G1 ~~ 0*C
G2 ~~ 0*C
'
```

Behavior Genetic Models

Example 2: Multiple Familial Relationships

- Model 4: $h + d$

```
# h + d
BG.model.4<-'
#Latent Variables
S1 =~ .87*T1
S2 =~ .87*T2

#Genetic Model
G1 =~ NA*S1 + c(h,h,h,h,h,h,h,h,h,h)*S1
G2 =~ NA*S2 + c(h,h,h,h,h,h,h,h,h,h)*S2
D1 =~ NA*S1 + c(d,d,d,d,d,d,d,d,d,d)*S1
D2 =~ NA*S2 + c(d,d,d,d,d,d,d,d,d,d)*S2

#Variances
G1 ~~ 1*G1
G2 ~~ 1*G2
D1 ~~ 1*D1
D2 ~~ 1*D2
S1 ~~ S1
S2 ~~ S2

#covariances
G1 ~~ c(1,.5,1,.5,0,0,.5,.5,0,0)*G2
D1 ~~ c(1,.25,1,.25,0,0,0,0,0,0)*D2
D1 ~~ 0*G1
```

Behavior Genetic Models (cont.)

Example 2: Multiple Familial Relationships

```
D1 ~~ 0*G2
D2 ~~ 0*G1
D2 ~~ 0*G2
'
```

- Model 5: $h + c_1 + c_2$ (MZ twins shared environment is different from other relationship)

```
# h + c1 + c2
BG.model.5<-'
#Latent Variables
S1 =~ .87*T1
S2 =~ .87*T2

#Genetic Model
G1 =~ NA*S1 + c(h,h,h,h,h,h,h,h,h,h)*S1
G2 =~ NA*S2 + c(h,h,h,h,h,h,h,h,h,h)*S2
C =~ NA*S1 + c(c1,c2,c1,c2,c2,c2,c2,c2,c2,c2)*S1 + c(c1,c2,c1,c2,c2,c2,c2,c2,c2,c2)*S2

#Variances
G1 ~~ 1*G1
G2 ~~ 1*G2
C ~~ 1*C
S1 ~~ S1
S2 ~~ S2

#covariances
G1 ~~ c(1,.5,1,.5,0,0,.5,.5,0,0)*G2
G1 ~~ 0*C
G2 ~~ 0*C
'
```

- Model 6: $h + c_1 + c_2 + c_3$ (parent-child, siblings, & MZ twins)

```
#h + c1 + c2 + c3
BG.model.6<-'
#Latent Variables
S1 =~ .87*T1
S2 =~ .87*T2

#Genetic Model
G1 =~ NA*S1 + c(h,h,h,h,h,h,h,h,h,h)*S1
G2 =~ NA*S2 + c(h,h,h,h,h,h,h,h,h,h)*S2
C =~ NA*S1 + c(c1,c2,c1,c2,c3,c3,c3,c3,c2,c2)*S1 + c(c1,c2,c1,c2,c3,c3,c3,c3,c2,c2)*S2

#Variances
G1 ~~ 1*G1
G2 ~~ 1*G2
C ~~ 1*C
S1 ~~ S1
S2 ~~ S2

#covariances
G1 ~~ c(1,.5,1,.5,0,0,.5,.5,0,0)*G2
G1 ~~ 0*C
G2 ~~ 0*C
'
```

```
> fitMeasures(BG.model.1fit, fit.measures=c("chisq", "df", "aic", "rmsea"))
  chisq        df       aic     rmsea
 35.818     9.000 6064.013     0.167
> fitMeasures(BG.model.2fit, fit.measures=c("chisq", "df", "aic", "rmsea"))
  chisq        df       aic     rmsea
  8.737     9.000 6036.932     0.000
> fitMeasures(BG.model.3fit, fit.measures=c("chisq", "df", "aic", "rmsea"))
  chisq        df       aic     rmsea
  8.737     8.000 6038.932     0.029
> fitMeasures(BG.model.4fit, fit.measures=c("chisq", "df", "aic", "rmsea"))
  chisq        df       aic     rmsea
  7.657     8.000 6037.852     0.000
> fitMeasures(BG.model.5fit, fit.measures=c("chisq", "df", "aic", "rmsea"))
  chisq        df       aic     rmsea
  6.457     7.000 6038.652     0.000
> fitMeasures(BG.model.6fit, fit.measures=c("chisq", "df", "aic", "rmsea"))
  chisq        df       aic     rmsea
  5.954     6.000 6040.149     0.000
```

- Model 6's χ^2 is different from that reported in Loehlin, likely because the c^2 estimates for the non MZ siblings was not estimated well

- Model 2 seems to fit the best

```
> summary(BG.model.2fit)
lavaan (0.5-9) converged normally after  72 iterations

  Number of observations per group
  MZ1                                               45
  DZ2                                               34
  MZ3                                              102
  DZ4                                              119
  PA5                                              257
  PA6                                              271
  PN7                                               56
  PN8                                               54
  AS9                                               48
  AS10                                              80

  Estimator                                         ML
  Minimum Function Chi-square                    8.737
  Degrees of freedom                                 9
  P-value                                        0.462

Chi-square for each group:

  MZ1                                            0.262
  DZ2                                            1.203
```

Behavior Genetic Models (cont.)

Example 2: Multiple Familial Relationships

```
  MZ3                                         0.389
  DZ4                                         1.401
  PA5                                         1.262
  PA6                                         0.244
  PN7                                         0.036
  PN8                                         0.211
  AS9                                         0.120
  AS10                                        3.608
Group 1 [MZ1]:

                   Estimate  Std.err  Z-value  P(>|z|)
Latent variables:
  S1 =~
    T1                0.870
  S2 =~
    T2                0.870
  G1 =~
    S1        (h)     0.710    0.065   10.923    0.000
  G2 =~
    S2        (h)     0.710    0.065   10.923    0.000

Covariances:
  G1 ~~
    G2                1.000

Variances:
    G1                1.000
    G2                1.000
    S1                0.715    0.218
```

```
    S2                   0.715    0.218
    T1                   0.000
    T2                   0.000
```

- Because VAR[G] = 1 and the “data” was standardized, the unstandardized estimate is actually standardized. Thus, $h^2 = .71^2 = .50$ (i.e., accounts for approximately 50% of the total variance)

Higher Order Factor Models

Table of Contents

Higher Order Factor Models

- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
  - Schmid-Leiman Transformation
  - Bifactor Rotation
- Examples
  - First-Order Factor Model
  - Higher-Order Factor Model
  - Hierarchical Factor Model
  - EFA Models

Talk Outline

Higher Order Factor Models

- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
- Examples

Talk Outline

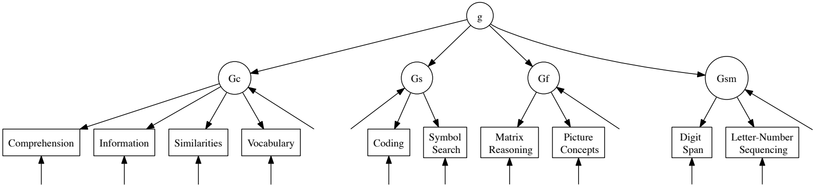
Higher Order Factor Models

- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
- Examples

Higher Order Factor Models

Second-Order Factor Models

- The models considered thus far have only estimated factors that directly influence the subtests.
- An extension of this kind of model is a higher-order model (Rindskopf & Rose, 1988), which specifies
  1. there are factors that directly influence the MVs (i.e, first-order factors), and
  2. there are factors that directly influence factors (i.e., second-order factors).



Example of Higher Order Factor Model

- In a higher-order factor model,
  - the covariance of the first-order factors is accounted for by a second-order factor that represents a higher-order construct.[5]
  - the first-order factor's variance are comprised of two components:
    1. variance explained by the second-order factor, and
    2. variance that is independent of the second-order factor (i.e., residual variance).
       - The latter component is represented by specific (residual) factors that explain individual differences in the first-order factors *over and above* the second-order factor.
  - In most higher-order models, the specific factors are uncorrelated with the higher order factor and among themselves.

[5] There could be more than one second-order factor.

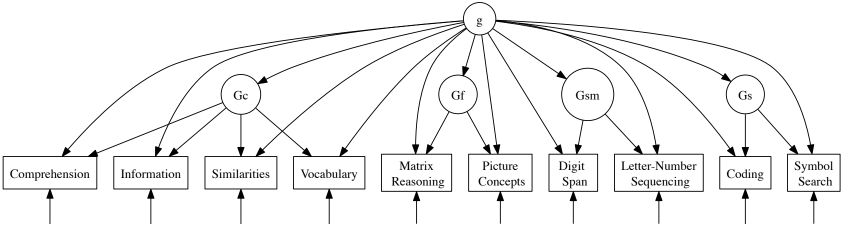
- Chen, West, and Sousa (2006) write that higher-order models could be applicable when
  - the lower-order factors are substantially correlated with each other, and
  - there is a higher-order factor that is hypothesized to account for the relationship among the lower-order factors.

- With higher order factor models, the influence of the second-order factor on the manifest variables is mediated by the first-order variables.
- Thus, the second-order factor influences all the manifest variables, but it does so only indirectly.

- Using Wright's rules, you can estimate the direct impact of the second order and first-order factors on the MVs.
- The factor loadings of the MVs on the second-order factor are computed by multiplying the factor loading of each MV on the corresponding first-order factor by the factor loading of this first-order factor on the second-order factor.
- The loadings of the MVs on a first-order factor can be computed by multiplying the factor loading of each MV on the corresponding first-order factor by the standard deviation of the corresponding specific factor.

- Check model fit
  - Fit indices
  - Model nested within the first-order factor model (i.e., $\Delta\chi^2$).
  - One other way to see how well this model fit the data is to inspect the residual correlations of the first-order factors (i.e., the difference between the model-implied correlations among the first-order constructs and the corresponding correlations in the first-order factor model).
  - Examine the amount of variance in the first-order factors explained by the second order factor.

- An alternative (full) higher-order model is one that has with direct effects from the second-order factor to every MV, *over and above* the second-order effect on the first-order factors.
- This model is equivalent to a hierarchal model, though, which will be discussed later.



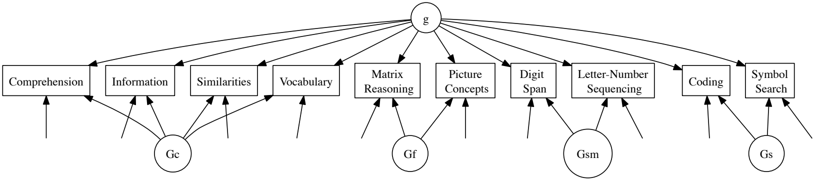
Full Higher Order Factor Model

Talk Outline

Higher Order Factor Models

- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
- Examples

- An alternative (generalization) to the higher-order model is a hierarchical model (AKA bi-factor model, nested-factor model).
- The hierarchal model specifies that all the factors are first-order factors, only some of these first order factors are more general than others.
- The hierarchical and higher-order models' interpretation are similar.
  - The second-order factor in the higher-order model corresponds to the general factor in the hierarchical model.
  - The disturbances of the first-order factors in the higher-order model are similar to the domain specific factors in the hierarchical model.
  - In the hierarchical model, the general factor and the domain specific factors are assumed to be orthogonal, just as with the higher-order model where the second-order factor and the disturbances (unique factors) are defined to be orthogonal.



Example of Hierarchical Model

- Notice that all the factors are first-order, but that g is uncorrelated with the other factors.

- Chen et al. (2006) write that hierarchal models could be applicable when
  - there is a general factor that is hypothesized to account for the commonality of the items;
  - there are multiple domain specific factors, each of which is hypothesized to account for the unique influence of the specific domain *over and above* the general factor; or
  - interest is in the domain specific factors as well as the common factor.

- Chen et al. (2006) note some possible advantages of the hierarchical model over second-order models
  1. A hierarchical model can be used as a less restricted baseline model to which a second-order model can be compared.
  2. Hierarchical models can be used to study the role of domain specific factors that are independent of the general factor.
  3. The hierarchical model allows for the direct examination of the strength of the relationship between the first-order factors and their associated MVs via the factor loadings; these relationships cannot be directly tested in the second-order factor model as the first-order factors are represented by disturbances of the first-order factors.
  4. The hierarchical model can be useful in testing whether a MV of the first-order factors predict external variables, over and above the general factor, as the domain specific factors are directly represented as independent factors;

- Chen et al. (2006) note some possible advantages of the hierarchical model over second-order models
  5. The hierarchical model allows for the testing of measurement invariance of the domain specific factors, in addition to the general factor in different groups, whereas with the second-order model, only the second-order factor can be directly tested for invariance between groups, as the domain specific factors are represented by disturbances.
  6. Likewise, in the hierarchical model, latent mean differences in both the general and domain specific factors can be compared across different groups (assuming at least scalar invariance), as opposed to the second-order model where only the second-order latent means can be directly compared.

Talk Outline

Higher Order Factor Models

- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
- Examples

Talk Outline

Higher Order Factor Models

- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
  - Schmid-Leiman Transformation
  - Bifactor Rotation
- Examples

- Schmid and Leiman (1957) developed a transformation of the higher-order factor model to yield uncorrelated first-order factors that represent both the second-order and the first-order constructs.
- This transformation of the factor loadings makes them reflect the incremental influence of both general and specific abilities on the indicator variable.
- As this procedure just transforms the higher order factor model, the "fit" of both models will be identical (Yung, Thissen, & McLeod, 1999).

Higher Order Factor Models

Exploratory Factor Models: Schmid-Leiman Transformation

- The hierarchal model and second-order model are mathematically equivalent *only* using the S-L method, because it imposes these two proportionality constraints:
    1. The factor loadings of the general factor in the hierarchal model must be the product of the corresponding lower-order factor loadings and the second-order factor loadings in the second-order models; and
    2. The ratio of the general factor loading to its corresponding first-order factor loading is the same within each domain specific factor.

- The S-L transformation can be used to estimate the direct impact of the second-order factor and the first-order variables on the MVs using Wright's rules.
- For the second-order factor loadings, multiply the factor loading of each MV on the corresponding first-order factor by the factor loading of the first-order factor on the second-order factor.
- To compute the loadings of the MVs on a first-order factor, multiply the first-order factor loading by the standard deviation of the corresponding first-order factor.
- This would be tedious to do by hand for each indicator, so we can follow the steps outlined in Gorsuch (1983) for an EFA (which can be applied to a second-order CFA).

1. Do EFA of the *m* MVs, extracting *p* factors with an oblique rotation.
   1.1 Save the $m \times p$ first-order factor loading matrix, $\mathbf{\Lambda}_1$.
   1.2 Save the $p \times p$ first-order inter-factor correlation matrix, $\mathbf{\Phi}_1$.
2. Using $\mathbf{\Phi}_1$, do an EFA and extract the second order factor.
   2.1 Save the $p \times 1$ second-order factor loading matrix, $\mathbf{\Lambda}_2$.
   2.2 Save the $p \times 1$ second-order factor uniquenesses, $\mathbf{u}_2^2$.
3. Create a $p \times p$ diagonal matrix using the square root of $\mathbf{u}_2^2$, $\mathbf{d}^* = \text{tr}(\mathbf{u}_2)\mathbf{I}$.
4. Create an $p \times p + 1$ augmented matrix, $\mathbf{A}$, where $\mathbf{A} = \left[\begin{array}{c|c} \mathbf{\Lambda}_2 & \mathbf{d}^* \end{array} \right]$.
5. Get a $m \times p + 1$ matrix of factor loadings for the second- and first-order factors by multiplying $\mathbf{\Lambda}_1$ by $\mathbf{A}$, $\mathbf{\Lambda}_{SL} = \mathbf{\Lambda}_1 \mathbf{A}$.

- For a CFA,

1. Fit a second-order factor model, which will give $\mathbf{\Lambda}_1$, $\mathbf{\Phi}_1$, $\mathbf{\Lambda}_2$, and $\mathbf{u}_2^2$.
2. Create a $p \times p$ diagonal matrix using the square root of $\mathbf{u}_2^2$, $\mathbf{d}^* = \text{tr}(\mathbf{u}_2)\mathbf{I}$.
3. Create an $p \times p + 1$ augmented matrix, $\mathbf{A}$, where $\mathbf{A} = \left[\begin{array}{c|c} \mathbf{\Lambda}_2 & \mathbf{d}^* \end{array} \right]$.
4. Get a $m \times p + 1$ matrix of factor loadings for the second- and first-order factors by multiplying $\mathbf{\Lambda}_1$ by $\mathbf{A}$, $\mathbf{\Lambda}_{SL} = \mathbf{\Lambda}_1\mathbf{A}$.

- The S-L transformation orthogonalizes the the relationship between the higher-order and lower-order factors.
  - First the highest order factor solution is determined, then the next highest order is determined based on the variance orthogonal to the highest order, etc.
- The S-L factors are proportionality constrained.
  - This constraint affects the proportion of variance in the MVs explained by second-order and first-order factors.
  - Specifically, for a given set of MVs, the ratios of variance attributable to the respective first-order factor to variance attributable to the second-order factor are constrained to be the same.

- The S-L factors are proportionality constrained.
  - For example, say the loadings on the first order factor $\mathbf{F}_1^1$ for MV1 is .762 and for MV2 is .846.
  - Say the loading of ${}_1\mathbf{F}_1$ on the second-order factor ${}_2\mathbf{F}_1$ is .818.
  - The factor loadings of the MVs on ${}_2\mathbf{F}_1$ are $.762 \times .818 = .623$ and $.846 \times .818 = .692$ for MV1 and MV2, respectively.
  - The variance ratio for the MV1 is $\frac{.762^2}{.623^2} = 1.50$ and for MV2 is $\frac{.846^2}{.692^2} = 1.50$.

Talk Outline

Higher Order Factor Models

- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
  - Schmid-Leiman Transformation
  - Bifactor Rotation
- Examples

Higher Order Factor Models

Exploratory Factor Models: Bifactor Rotation

- Jennrich and Bentler (2011) developed an alternative to the S-L rotation, the *bi-factor* rotation.
- This is a rotation criterion that loads on the first factor and encourages perfect cluster structure (i.e., no cross-loadings) for the loadings on the remaining factors.
- Exploratory bi-factor analysis (EFBA) is a more direct and "satisfactory" approach to bi-factor model building than using the S-L rotation.
- A $m \times p$ loading matrix $\mathbf{\Lambda}$ has bi-factor structure if each row of $\mathbf{\Lambda}$ has, at most, one nonzero element in its last pâĹŠ1 columns.
- EBFA is simply standard exploratory factor analysis using a bi-factor rotation criterion.

Talk Outline

Higher Order Factor Models

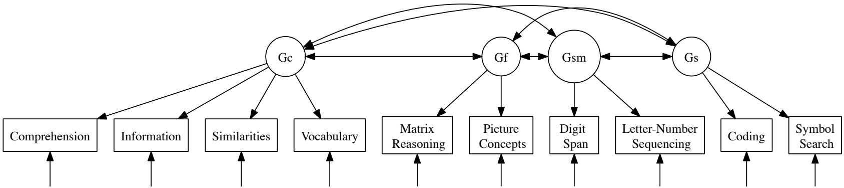
- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
- Examples

Talk Outline

Higher Order Factor Models

- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
- Examples
  - First-Order Factor Model
  - Higher-Order Factor Model
  - Hierarchical Factor Model
  - EFA Models

- Before specifying a higher order model, it's beneficial to examine a first-order (four-factor) model to see what the correlations are among the factors
- The higher-order factor model is nested in the first-order factor model.
- Thus, it is possible to test whether the higher-order factor fully accounts for the covariance among the first-order variables



Four Factor Model of Cognitive Ability from the WISC-IV

- Model specification

```
WISC.fourFactor.model<-'
Gc =~ Comprehension + Information +  Similarities + Vocabulary
Gf =~ Matrix.Reasoning + Picture.Concepts
Gsm =~  Digit.Span + Letter.Number
Gs =~ Coding + Symbol.Search
'
```

```
> WISC.fourFactor.fit<-cfa(model=WISC.fourFactor.model, sample.cov=WiscIV.cov, sample.
    nobs=550)
> summary(WISC.fourFactor.fit, fit.measure=TRUE, standardized=TRUE)

lavaan (0.5-9) converged normally after  90 iterations

  Number of observations                           550

  Estimator                                         ML
  Minimum Function Chi-square                   51.634
  Degrees of freedom                                29
  P-value                                        0.006

Chi-square test baseline model:

  Minimum Function Chi-square                 2552.014
  Degrees of freedom                                45
  P-value                                        0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.991
  Tucker-Lewis Index (TLI)                       0.986

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)             -12564.289
  Loglikelihood unrestricted model (H1)     -12538.472
```

Higher Order Factor Models (cont.)

Examples: First-Order Factor Model

```

  Number of free parameters                            26
  Akaike (AIC)                                  25180.578
  Bayesian (BIC)                                25292.636
  Sample-size adjusted Bayesian (BIC)           25210.101

Root Mean Square Error of Approximation:

  RMSEA                                             0.038
  90 Percent Confidence Interval          0.020     0.054
  P-value RMSEA <= 0.05                             0.885

Standardized Root Mean Square Residual:

  SRMR                                              0.020

Parameter estimates:

  Information                                    Expected
  Standard Errors                                Standard

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  Gc =~
    Comprehension     1.000                                2.192    0.762
    Information       1.160    0.056   20.783    0.000    2.544    0.846
    Similarities      1.167    0.056   20.758    0.000    2.558    0.845
    Vocabulary        1.218    0.056   21.835    0.000    2.670    0.885
  Gf =~
```

Higher Order Factor Models (cont.)

Examples: First-Order Factor Model

```
    Matrix.Resnng       1.000                                         1.999     0.692
    Pictur.Cncpts       0.839     0.078    10.771     0.000    1.677     0.563
  Gsm =~
    Digit.Span          1.000                                         1.968     0.661
    Letter.Number       1.171     0.086    13.562     0.000    2.305     0.772
  Gs =~
    Coding              1.000                                         1.778     0.601
    Symbol.Search       1.429     0.139    10.275     0.000    2.541     0.815

Covariances:
  Gc ~~
    Gf                  3.412     0.337    10.129     0.000    0.779     0.779
    Gsm                 3.302     0.336     9.824     0.000    0.765     0.765
    Gs                  2.181     0.290     7.511     0.000    0.560     0.560
  Gf ~~
    Gsm                 3.243     0.352     9.215     0.000    0.824     0.824
    Gs                  2.505     0.329     7.604     0.000    0.705     0.705
  Gsm ~~
    Gs                  2.474     0.324     7.626     0.000    0.707     0.707

Variances:
    Comprehension       3.474     0.240                        3.474     0.420
    Information         2.573     0.204                        2.573     0.284
    Similarities        2.621     0.207                        2.621     0.286
    Vocabulary          1.977     0.181                        1.977     0.217
    Matrix.Resnng       4.340     0.424                        4.340     0.521
    Pictur.Cncpts       6.052     0.434                        6.052     0.683
    Digit.Span          4.991     0.377                        4.991     0.563
    Letter.Number       3.611     0.381                        3.611     0.405
```

```
    Coding              5.585    0.428                            5.585    0.639
    Symbol.Search       3.261    0.574                            3.261    0.336
    Gc                  4.806    0.468                            1.000    1.000
    Gf                  3.997    0.544                            1.000    1.000
    Gsm                 3.873    0.497                            1.000    1.000
    Gs                  3.161    0.484                            1.000    1.000
```

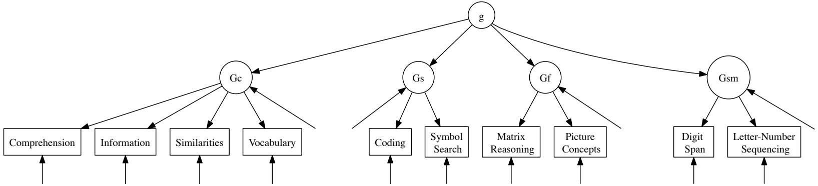
- Notice the lack of `std.lv=TRUE` argument in the `cfa()` function.
- This is equivalent to including the argument `std.lv=FALSE` because that is the default value for the function.

Talk Outline

Higher Order Factor Models

- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
- Examples
  - First-Order Factor Model
  - Higher-Order Factor Model
  - Hierarchical Factor Model
  - EFA Models

- A typical higher-order model for cognitive abilities data is one that posits that g is the sole reason why the the first-order factors are correlated with each other (Carroll, 1993). Such a model is shown below.



Higher Order Factor Model of Cognitive Ability from the WISC-IV

```
1 WISC.higherOrder.model<-'
2 gc =~ Comprehension + Information +  Similarities + Vocabulary
3 gf =~ Matrix.Reasoning + Picture.Concepts
4 gsm =~  Digit.Span + Letter.Number
5 gs =~ Coding + Symbol.Search
6
7 g=~ NA*gf + gc  + gsm + gs
8 g~~ 1*g
9 '
```

- Notice the the `NA*` in front of the `gf` term in line 7, which estimates this loading instead of constraining it to one (the default).
- The trade-off is that you must constrain g's variance to one, which is done with the `g~~1*g` term on line 8.
- This allows the estimation of the (residual) variances of the first-order factors instead of constraining all the latent variables' variances to be unity with the `std.lv=TRUE` argument.

```
> WISC.higherOrder.fit<-cfa(model=WISC.higherOrder.model, sample.cov=WiscIV.cov, sample.
    nobs=550)
> summary(WISC.higherOrder.fit, fit.measure=TRUE, standardized=TRUE, rsquare=TRUE)
lavaan (0.5-9) converged normally after  56 iterations

  Number of observations                           550

  Estimator                                         ML
  Minimum Function Chi-square                   57.592
  Degrees of freedom                                31
  P-value                                        0.003

Chi-square test baseline model:

  Minimum Function Chi-square                 2552.014
  Degrees of freedom                                45
  P-value                                        0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.989
  Tucker-Lewis Index (TLI)                       0.985

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)             -12567.268
  Loglikelihood unrestricted model (H1)     -12538.472

```

Higher Order Factor Models (cont.)

Examples: Higher-Order Factor Model

```
  Number of free parameters                          24
  Akaike (AIC)                                25182.537
  Bayesian (BIC)                              25285.975
  Sample-size adjusted Bayesian (BIC)         25209.788

Root Mean Square Error of Approximation:

  RMSEA                                           0.039
  90 Percent Confidence Interval          0.023   0.055
  P-value RMSEA <= 0.05                           0.856

Standardized Root Mean Square Residual:

  SRMR                                            0.023

Parameter estimates:

  Information                                  Expected
  Standard Errors                              Standard

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  Gc =~
    Comprehension     1.000                               2.194    0.762
    Information       1.160    0.056   20.813    0.000    2.545    0.846
    Similarities      1.165    0.056   20.753    0.000    2.555    0.844
    Vocabulary        1.217    0.056   21.857    0.000    2.670    0.885
  Gf =~
    Matrix.Resnng     1.000                               1.990    0.689
```

Higher Order Factor Models (cont.)

Examples: Higher-Order Factor Model

```
    Pictur.Cncpts       0.847     0.079    10.754     0.000     1.685     0.566
  Gsm =~
    Digit.Span          1.000                                   1.967     0.661
    Letter.Number       1.172     0.087    13.505     0.000     2.306     0.772
  Gs =~
    Coding              1.000                                   1.771     0.599
    Symbol.Search       1.441     0.142    10.141     0.000     2.551     0.818
  g =~
    Gf                  1.851     0.122    15.173     0.000     0.930     0.930
    Gc                  1.794     0.112    16.023     0.000     0.818     0.818
    Gsm                 1.822     0.130    14.070     0.000     0.927     0.927
    Gs                  1.293     0.133     9.709     0.000     0.730     0.730

Variances:
    g                   1.000                                   1.000     1.000
    Comprehension       3.467     0.240                         3.467     0.419
    Information         2.567     0.203                         2.567     0.284
    Similarities        2.634     0.208                         2.634     0.287
    Vocabulary          1.976     0.181                         1.976     0.217
    Matrix.Resnng       4.377     0.424                         4.377     0.525
    Pictur.Cncpts       6.026     0.435                         6.026     0.680
    Digit.Span          4.996     0.378                         4.996     0.564
    Letter.Number       3.606     0.382                         3.606     0.404
    Coding              5.610     0.430                         5.610     0.641
    Symbol.Search       3.210     0.584                         3.210     0.330
    Gc                  1.596     0.226                         0.332     0.332
    Gf                  0.533     0.340                         0.135     0.135
    Gsm                 0.547     0.241                         0.141     0.141
    Gs                  1.464     0.263                         0.467     0.467
```

```

R-Square:

    Comprehension      0.581
    Information        0.716
    Similarities       0.713
    Vocabulary         0.783
    Matrix.Reasoning      0.475
    Picture.Concepts      0.320
    Digit.Span         0.436
    Letter.Number      0.596
    Coding             0.359
    Symbol.Search      0.670
    Gc                 0.668
    Gf                 0.865
    Gsm                0.859
    Gs                 0.533
```

- The factor loadings of the MVs on the second-order factor are computed by multiplying the factor loading of each MV on the corresponding first-order factor by the factor loading of this first-order factor on the second-order factor.
  - For example, the standardized loading of the Comprehension subtest score on *g* is computed as $.762 \times .818 = .623$.
- The loadings of the MVs on a specific factor can be computed by multiplying the factor loading of each MV on the corresponding first-order factor by the standard deviation of the corresponding specific factor.
  - For example, the standardized loading of the Comprehension subtest score on *Gc* is $.762 \times .576 = .439$ (the [standardized] variance of *Gc* is .332 and $\sqrt{.332} = .576$).

- Model Fit
  - This model fits the data very similarly to the four-factor model, although since the higher-order model is more parsimonious (it estimates 24 parameters instead of the 26 parameters the four-factor model estimates), it is probably a better model for this data.

Correlations among First-Order Factors. Actual are in Upper Triangle and Implied are in Lower Triangle

	Gf	Gc	Gsm	Gs
Gf	1.00	0.78	0.82	0.71
Gc	0.76	1.00	0.77	0.56
Gsm	0.86	0.76	1.00	0.71
Gs	0.68	0.60	0.68	1.00

Residual Correlations of First-Order Factors

	Gf	Gc	Gsm	Gs
Gf				
Gc	0.02			
Gsm	-0.04	0.01		
Gs	0.03	-0.04	0.03	

- The residuals range from -.04 to .03, which are likely not of much concern.

Variances of First Order Factors

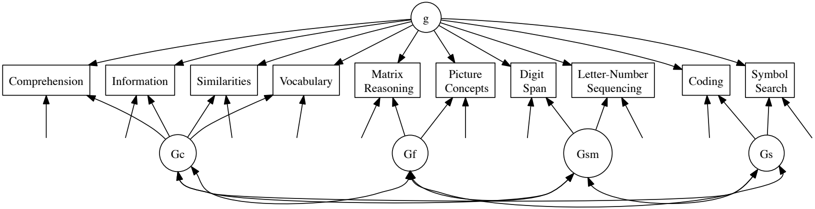
	Observed Variance	Residual Variance	R^2
Gc	4.81	1.60	0.67
Gf	4.00	0.53	0.87
Gsm	3.87	0.55	0.86
Gs	3.16	1.46	0.54

- g explains between 54 and 87% of the first order factors' variances.

Talk Outline

Higher Order Factor Models

- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
- Examples
  - First-Order Factor Model
  - Higher-Order Factor Model
  - Hierarchical Factor Model
  - EFA Models



Hierarchical Model of the WISC-IV subtests

Higher Order Factor Models

Examples: Hierarchical Factor Model

- The specification of the hierarchical model in `lavaan` is

```
1 ## Hierarchical model
2 WISC.hierarchical.model<-'
3 gc =~ Comprehension + Information +  Similarities + Vocabulary
4 gf =~ Matrix.Reasoning + Picture.Concepts
5 gsm =~  Digit.Span + Letter.Number
6 gs =~ Coding + Symbol.Search
7 g=~ Comprehension + Information + Matrix.Reasoning + Picture.Concepts + Similarities +
      Vocabulary +  Digit.Span + Letter.Number + Coding + Symbol.Search
8 g~~0*gc + 0*gf + 0*gsm + 0*gs
9 '
```

- The default in `lavaan` is for all exogenous variables to be correlated with each other, so line 8 indicates that *g* needs to be uncorrelated with *Gc*, *Gf*, *Gsm* and *Gs*.

```
WISC.hierarchical.fit<-cfa(model=WISC.hierarchical.model, sample.cov=WiscIV.cov, sample.
    nobs=550, std.lv=TRUE)
summary(WISC.hierarchical.fit, fit.measure=TRUE, standardized=TRUE)

lavaan (0.5-9) converged normally after 103 iterations

  Number of observations                           550

  Estimator                                         ML
  Minimum Function Chi-square                   13.485
  Degrees of freedom                                19
  P-value                                        0.813

Chi-square test baseline model:

  Minimum Function Chi-square                 2552.014
  Degrees of freedom                                45
  P-value                                        0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    1.000
  Tucker-Lewis Index (TLI)                       1.005

Loglikelihood and Information Criteria:

  Loglikelihood user model (H0)             -12545.215
  Loglikelihood unrestricted model (H1)     -12538.472
```

Higher Order Factor Models (cont.)

Examples: Hierarchical Factor Model

```

  Number of free parameters                         36
  Akaike (AIC)                               25162.429
  Bayesian (BIC)                             25317.586
  Sample-size adjusted Bayesian (BIC)        25203.307

Root Mean Square Error of Approximation:

  RMSEA                                          0.000
  90 Percent Confidence Interval          0.000  0.024
  P-value RMSEA <= 0.05                          1.000

Standardized Root Mean Square Residual:

  SRMR                                           0.011

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  gc =~
    Comprehension     2.360    0.162   14.599    0.000    2.360    0.820
    Information       2.092    0.461    4.541    0.000    2.092    0.696
    Similarities      2.176    0.409    5.314    0.000    2.176    0.719
    Vocabulary        2.501    0.305    8.193    0.000    2.501    0.829
  gf =~
```

Higher Order Factor Models (cont.)

Examples: Hierarchical Factor Model

```
    Matrix.Resnng      1.510    0.395    3.826    0.000    1.510    0.523
    Pictur.Cncpts      1.612    0.208    7.743    0.000    1.612    0.541
  gsm =~
    Digit.Span         1.931    0.186   10.387    0.000    1.931    0.649
    Letter.Number      2.103    0.257    8.176    0.000    2.103    0.704
  gs =~
    Coding             1.862    0.164   11.360    0.000    1.862    0.630
    Symbol.Search      2.251    0.278    8.101    0.000    2.251    0.722
  g =~
    Comprehension      0.271    0.703    0.386    0.700    0.271    0.094
    Information        1.547    0.622    2.486    0.013    1.547    0.514
    Matrix.Resnng      1.446    0.375    3.850    0.000    1.446    0.501
    Pictur.Cncpts      0.635    0.414    1.533    0.125    0.635    0.213
    Similarities       1.356    0.652    2.080    0.038    1.356    0.448
    Vocabulary         0.963    0.749    1.286    0.198    0.963    0.319
    Digit.Span         0.547    0.488    1.122    0.262    0.547    0.184
    Letter.Number      0.868    0.525    1.655    0.098    0.868    0.291
    Coding             0.331    0.370    0.894    0.371    0.331    0.112
    Symbol.Search      0.982    0.419    2.345    0.019    0.982    0.315

Covariances:
  gc ~~
    g                  0.000                               0.000    0.000
  gf ~~
    g                  0.000                               0.000    0.000
  gsm ~~
    g                  0.000                               0.000    0.000
  gs ~~
    g                  0.000                               0.000    0.000
```

Higher Order Factor Models (cont.)

Examples: Hierarchical Factor Model

```
  gc ~~
    gf                0.691    0.116    5.975    0.000    0.691    0.691
    gsm               0.728    0.065   11.235    0.000    0.728    0.728
    gs                0.495    0.092    5.374    0.000    0.495    0.495
  gf ~~
    gsm               0.799    0.082    9.749    0.000    0.799    0.799
    gs                0.657    0.084    7.819    0.000    0.657    0.657
  gsm ~~
    gs                0.681    0.063   10.730    0.000    0.681    0.681

Variances:
    Comprehension     2.637    0.458                      2.637    0.318
    Information       2.273    0.245                      2.273    0.251
    Similarities      2.592    0.216                      2.592    0.283
    Vocabulary        1.922    0.213                      1.922    0.211
    Matrix.Resnng     3.966    0.454                      3.966    0.476
    Pictur.Cncpts     5.862    0.515                      5.862    0.661
    Digit.Span        4.836    0.409                      4.836    0.546
    Letter.Number     3.748    0.382                      3.748    0.420
    Coding            5.168    0.534                      5.168    0.591
    Symbol.Search     3.686    0.608                      3.686    0.379
    gc                1.000                               1.000    1.000
    gf                1.000                               1.000    1.000
    gsm               1.000                               1.000    1.000
    gs                1.000                               1.000    1.000
    g                 1.000                               1.000    1.000
```

Higher Order Factor Models (cont.)

Examples: Hierarchical Factor Model

- All of the fit indices indicate that this model fits the data better any of the other three. Moreover, the χ^2 value also indicates that this model is a fairly good representation of the data (Barrett, 2007).

Talk Outline

Higher Order Factor Models

- Second-Order Factor Models
- Hierarchical Factor Models
- Exploratory Factor Models
- Examples
  - First-Order Factor Model
  - Higher-Order Factor Model
  - Hierarchical Factor Model
  - EFA Models

- For the Schmid-Leiman transformation, use the `schmid()` function in the `psych` package.

```
> schmid(WiscIV.cor,nfactors=4)
Schmid-Leiman analysis


Schmid Leiman Factor loadings greater than  0.2
        g   F1*   F2*   F3*   F4*   h2   u2   p2
1   0.59 0.52             -0.20 0.66 0.34 0.54
2   0.65 0.53                   0.73 0.27 0.57
3   0.58                   0.37 0.49 0.51 0.67
4   0.49                        0.27 0.73 0.88
5   0.65 0.53                   0.71 0.29 0.58
6   0.67 0.58                   0.80 0.20 0.57
7   0.66                        0.44 0.56 0.98
8   0.76                        0.59 0.41 0.99
9   0.43            0.77        0.78 0.22 0.24
10  0.57            0.32        0.45 0.55 0.73
```

```
 The orthogonal loadings were
Standardized loadings based upon correlation matrix
     F1   F2   F3   F4   h2   u2
1  0.73 0.28 0.16 0.05 0.65 0.35
2  0.70 0.22 0.17 0.42 0.74 0.26
3  0.28 0.29 0.17 0.55 0.50 0.50
4  0.24 0.31 0.18 0.30 0.27 0.73
5  0.72 0.26 0.08 0.36 0.72 0.28
6  0.80 0.29 0.15 0.22 0.80 0.20
7  0.29 0.54 0.19 0.17 0.44 0.56
8  0.32 0.62 0.18 0.26 0.59 0.41
9  0.11 0.16 0.85 0.10 0.78 0.22
10 0.20 0.34 0.44 0.32 0.45 0.55

                 F1   F2   F3   F4
SS loadings    2.57 1.28 1.14 0.95
Proportion Var 0.26 0.13 0.11 0.10
Cumulative Var 0.26 0.38 0.50 0.59
```

- Exploratory bi-factor analysis (EBFA) is a rotation option in the `psych` package's `fa()` function as well as is the `GPArotation` package.
  - (The `fa()` function actually uses the algorithm in the `GPArotation` package.)
- EBFA is deigned to extract a second-order factor as well as first-order factors.
  - For the WISC data, five factors should be extracted instead of four: one for *g* and the other four for *Gc*, *Gs*, *Gf* and *Gsm* factors.

```
> fa(WiscIV.cor, nfactors=5, n.obs=550, fm="pa", rotate="bifactor", max.iter = 500)
Factor Analysis using method =  pa
Call: fa(r = WiscIV.cor, nfactors = 5, n.obs = 550, rotate = "bifactor",
    max.iter = 500, fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
    PA1   PA2   PA3   PA5   PA4   h2     u2
1   0.67 -0.01 -0.02  0.47  0.02 0.67 0.3285
2   0.83 -0.06 -0.08  0.14 -0.14 0.74 0.2633
3   0.65  0.07  0.03 -0.19 -0.03 0.46 0.5424
4   0.51  0.86  0.01 -0.01  0.00 0.99 0.0064
5   0.81 -0.04 -0.18  0.18 -0.08 0.73 0.2699
6   0.81 -0.05 -0.10  0.34 -0.05 0.79 0.2063
7   0.59  0.01  0.05  0.00  0.53 0.64 0.3639
8   0.66  0.05  0.09 -0.02  0.21 0.50 0.5037
9   0.41  0.06  0.54  0.02  0.04 0.46 0.5387
10  0.59 -0.01  0.47 -0.10  0.01 0.58 0.4233

                       PA1  PA2  PA3  PA5  PA4
SS loadings           4.44 0.75 0.57 0.44 0.36
Proportion Var        0.44 0.08 0.06 0.04 0.04
Cumulative Var        0.44 0.52 0.58 0.62 0.66
Proportion Explained  0.68 0.11 0.09 0.07 0.05
Cumulative Proportion 0.68 0.79 0.88 0.95 1.00
```

Psychometrics

Table of Contents

Psychometrics

- Introduction
- Classical Test Theory
  - Classical Test Theory Hierarchy Indicators
  - Classical Test Theory vs. Confirmatory Factor Analysis
- Measuring Reliability
  - Coefficient Alpha
- Information
- Single Indicator Models
- Example
  - Reliability

Table of Contents (cont.)

Single Indicator Model

Talk Outline

Psychometrics

- Introduction
- Classical Test Theory
- Measuring Reliability
- Information
- Single Indicator Models
- Example

Talk Outline

Psychometrics

- Introduction
- Classical Test Theory
- Measuring Reliability
- Information
- Single Indicator Models
- Example

- Usually science is interested in the relationship between constructs.
- In the behavioral and social sciences, often our measurements of these constructs are not perfect, as they contain an unknown amount of error.
- There are two types of error:
  - Random
  - Systematic

- *Systematic error* typically reflect specific effects due to an individual or situation.
  - e.g.,if you are making copies of a math test and, unbeknownst to you, the copy machine blurred the problems on the right column on every even page so that the 1s looked like 7s, then the fact that every student who received a blurred copy of the test would be off on his/her calculations, whereas the students who received the un-blurred tests did not, would be an example of systemic measurement error.
- *Random error* can be defined as anything that is not systemic error, which prevents the observed score from equaling the true score (to be defined later).
  - For example, if Smitty normally does very well on biology exams. The night before one exam, however, his roommate unexpectedly brought a new dog home and it kept Smitty up all night. His abnormally test score the next day is likely going to be influenced by this random event.

- Because systematic error can be specified, it can often be removed either by design (or, perhaps, as part of the data analysis).
- But what about the effect of random error?
  - Random errors will affect the strength of the correlation between two (or more) variables
  - Charles Spearman (1904) first recognized the influence of error on observed correlations.

> Now, suppose that we wish to ascertain the correspondence between a series of values, p, and another series, q. By practical observation we evidently do not obtain the true objective values, p and q, but only approximations which we will call p' and q'. Obviously, p' is less closely connected with q', than is p with q, for the first pair only correspond at all by the intermediation of the second pair; *the real correspondence between p and q, shortly,* r_{pq}*, has been "attenuated" into* $r_{p'q'}$ (Spearman, 1904, p. 90, emphasis added)

- To understand how this error influences relationships, we need to first delve into classical test theory.

Talk Outline

Psychometrics

- Introduction
- Classical Test Theory
- Measuring Reliability
- Information
- Single Indicator Models
- Example

- Classical test theory (CTT; Allen & Yen, 1979; Lord & Novick, 1968) is concerned with the whole test (i.e., (weighted) sum of the items, average response).
- More specifically, it is interested in the reliability of this whole test (observed) score.
- To get at reliability, CTT posits that an observed score on a test, X, is made up two independent latent components:
  - True score, ξ, and
  - Error, E.

- The CTT components are related in the following manner

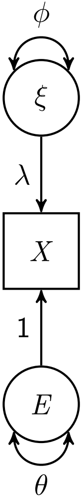
Classical Test Theory "Model"

$$X = a + \lambda\xi + E \tag{6}$$

- Equation (6) is often simplified by setting $a = 0$ and $\lambda = 1$, yielding

Alternative Classical Test Theory "Model"

$$X = T + E \tag{1a}$$



Path Diagram of Classical Test Theory Model

- According to Spearman (1904), (random) errors are "accidental deviations" that are different for every individual, occur without bias, occur in "every direction according to the known laws of probability," and can be thought of as "augmenting and diminishing" observed values.
- Over many observations, these errors tend to "more and more perfectly counterbalance one another." True scores, on the other hand, are the expected value (i.e., average over the entire distribution or many, many, many trials) of the observed score.

- The goal of CTT-based analysis is to reduce error variance as much as possible.
- This is usually done by
  - standardizing the testing conditions (i.e., get rid of systematic errors)
  - aggregating over as many items as possible, which will cause the random errors to cancel out
- From CTT perspective, items are exchangeable, thus their properties are not taken into account.

- The "randomness" of the random error results in

$$\mathrm{E}[X] = \xi \tag{7}$$

  and

$$\rho_{E,\xi} = 0 \tag{8}$$

- The implications from these relationships are
  - The long term average of X is ξ, i.e., E[X]$= \xi$.
  - Since $X = \xi + E$ and E[X]$= \xi$, then E[E]$= 0$.

- From Equations (6)-(8) (or Figure 15), the variance of X can be decomposed into

$$\text{VAR[X]} = \phi + \theta + 2\sigma_{\xi,E} \tag{9}$$

where,
ϕ is the true score variance, and
θ is the error variance.

- From the results of Equation (8), $2\sigma_{\xi,E} = 0$, so (9) becomes

Observed Score Variance

$$\text{VAR[X]} = \phi + \theta \tag{10}$$

- Using the results from Equation (10), we can now define score *reliability*.

Classical Test Theory Reliability

$$\rho_{XX'} = \frac{\phi}{\text{VAR[X]}} = \frac{\phi}{\phi + \theta} \tag{11}$$

- Reliability is the amount of variance for a variable that is due to variance in the "True Score."

- Alternatively, Equation (11) can be written as

Classical Test Theory Reliability (Alternative)

$$\rho_{XX'} = 1 - \frac{\theta}{\phi + \theta}, \tag{12}$$

- This says reliability is 1 - the proportion of the observed score variance due to error variance.

- How to obtain this true score variance, or, alternatively, the error variance?
- To do this, we need to discuss the hierarchy of CTT indicators (Allen & Yen, 1979; Lord & Novick, 1968).

Talk Outline

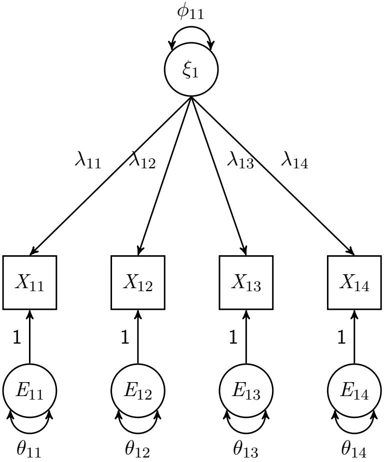
Psychometrics

- Introduction
- Classical Test Theory
  - Classical Test Theory Hierarchy Indicators
  - Classical Test Theory vs. Confirmatory Factor Analysis
- Measuring Reliability
- Information
- Single Indicator Models
- Example

- *Congeneric* indicators are ones that measure the same latent variable.
  - There are no restrictions on the factor loadings or error variances except that the error variances are independent.
  - The equation is

$$X_i = \lambda_i \xi_1 + E_i \tag{13}$$

(although sometimes a constant [intercept], a_i is added.)



Congeneric Indicators

- From Wright's rules, we can derive the reliability of X (i.e., the sum of $X_1, X_2, \ldots X_k$) for *congeneric indicators*.
- We do this by calculating the proportion of variance due to ξ and the proportion due to E.

Reliability for Congeneric Indicators

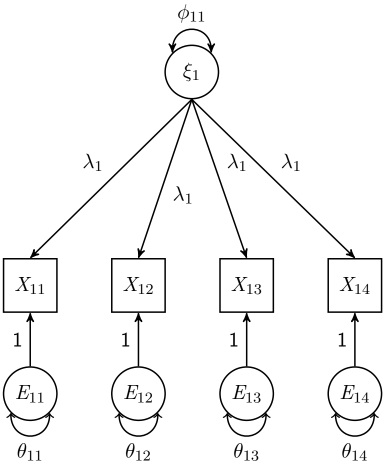
$$\rho_{XX'} = \frac{\left(\sum_{i=1}^{k} \lambda_i\right)^2 \phi_{11}}{\left(\sum_{i=1}^{k} \lambda_i\right)^2 \phi_{11} + \sum_{i=1}^{k} \theta_{ii} + 2 \sum_{1 \le i < j \le k} \theta_{ij}} \tag{14}$$

- This measure of reliability is also called ω (McDonald, 1999).

- A more restrictive CTT model is that of τ-*equivalent indicators*
  - This specifies a congeneric model but makes the indicators for a given construct have equal factor loadings.
  - In this model, the indictors have equivalent relationships with the underlying construct they measure.
  - Thus, a change in the latent variable of *m* units results in the same amount of change on each indicator.
  - The errors can differ for the indicators, however.
  - The equation is

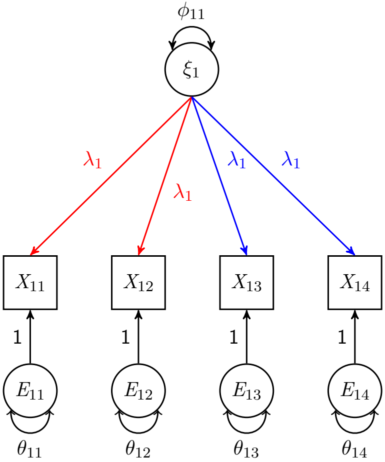
CTT "Model" for τ-equivalent Indicators

$$X_i = \lambda \xi_1 + E_{1i} \tag{15}$$



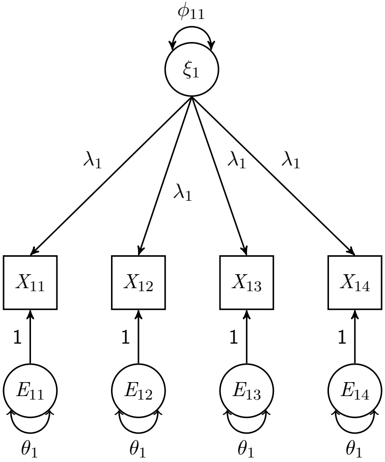
τ-Equivalent Indicators

- Because τ-*equivalent indicators* requires equal loadings
  - All indicator covariances are the same: $\lambda_i \phi \lambda_j = \lambda^2 \phi$
  - Indicator variances can differ: $\lambda_i \phi \lambda_i + \theta_i = \lambda^2 \phi + \theta$
- Sometimes this is called a compound symmetry heterogeneous model



τ-Equivalent Indicators

- *Parallel indicators* adds to the τ-equivalent model the restriction that the error variances are the same.
- If indicators are parallel, it lends support to the notion that the indicators are interchangeable (at least psychometrically) and justifies the practice of summing the indicators to get a manifest version of the latent variable.



Parallel Indicators

- Because *parallel indicators* requires equal loadings and error variances
  - All indicator covariances are the same: $\lambda_i \phi \lambda_j = \lambda^2 \phi$
  - All indicator variances are the same: $\lambda_i \phi \lambda_i + \theta = \lambda^2 \phi + \theta$
- Sometimes this is called a compound symmetry model

- If the indicators are parallel, then for given respondent, his/her expected scores (i.e., true scores) on the indicators are the same.
- Thus, the true score variance, ϕ, is the same for all the indicators, as is the error variance, θ.

Correlation Between Two Parallel Indicators

$$\rho_{P_1 P_2} = \frac{\sigma_{P_1 P_2}}{\sqrt{\sigma^2_{P_1}\sigma^2_{P_2}}} = \frac{\sigma^2_{\xi}}{\sigma^2_{P}} = \frac{\phi}{\phi + \theta} = \rho_{XX'} \tag{16}$$

where
P_1 and P_2 are parallel indictors of the same construct.

- That is, the correlation between two parallel indicators is a measure of reliability.

- Spearman's solution to the problem of estimating the true relationship between two variables, p and q, given observed scores p' and q' was to introduce additional *parallel* variables.
- From these parallel measures, he estimated the reliability of each set of measures $(r_{p'p'}, r_{q'q'})$
- He then used the reliability estimate to find

Spearman's Correction of the Correlation for Attenuation

$$r_{pq} = \frac{r_{p'q'}}{\sqrt{r_{p'p'} r_{q'q'}}} \tag{17}$$

Talk Outline

Psychometrics

- Introduction
- Classical Test Theory
  - Classical Test Theory Hierarchy Indicators
  - Classical Test Theory vs. Confirmatory Factor Analysis
- Measuring Reliability
- Information
- Single Indicator Models
- Example

- Assumptions
  - Most of CTT analysis is based on the assumption of parallel or τ-equivalent indicators.
  - CFA can *test* whether each indicator relates to the factor, as well if the relate differently.
  - That is, CFA can tell where some items are "better" than others
- Comparability
  - CTT does not separate item properties from observed/true score properties
  - CTT assumes the sum of the items estimates the true score
  - In CTT, item properties are sample dependent
  - In CFA, latent trait are estimated separately from item responses
  - CFA separates person traits from items properties
  - Thus, item properties are not dependent on a specific sample

Talk Outline

Psychometrics

- Introduction
- Classical Test Theory
- Measuring Reliability
- Information
- Single Indicator Models
- Example

Talk Outline

Psychometrics

- Introduction
- Classical Test Theory
- Measuring Reliability
  - Coefficient Alpha
- Information
- Single Indicator Models
- Example

- Guttman (1945) and Cronbach (1951) came up with a way to estimate reliability from one administration of a test, instead of multiple administration of parallel forms.
- It is frequently referred to as α.
- α has many definitions
  - It is the mean of all possible split-half correlations
  - It is the expected correlation with a hypothetical alternative form of the same length
  - It is the lower-bound estimate of reliability assuming that that all items are τ-equivalent

Coefficient α

$$\alpha = \frac{k}{k-1} \times \frac{\sigma_X^2 - \sum_{i=1}^{k} \sigma_{\sigma_{X_i}}^2}{\sigma_X^2} \tag{18}$$

where

$X = \sum_{i=1}^{k} X_i$,

$\sum_{i=1}^{k} \sigma_{\sigma_{X_i}}^2$ is the sum of the indicator variances, and

σ_X^2 is the variance of $X = \sum_{i=1}^{k} X_i$, which is the sum of all the indicator variances and $2\times$ indicator covariances.

- The idea behind α is that if the indicators are related to each other, the variance of their total, X, should be larger than the sum of the indicator variances.
- It assumes that the indicators are τ-equivalent (i.e., each indictor contributes equally to the construct).
- It assumes local independence (i.e., no residual covariance)
- It is **not** an index of model fit
- It is **not** a test of the indicators dimensionality
  - It does **not** index the extent to which indicators measure the same construct.
  - Could have set of indicators that form two constructs that have the α values as a set of indicators that measure one construct [see example in Schmitt (1996)].

- Through factor analysis, can test the assumptions of τ-equivalence and parallel indicators
  - Model 1: Indicator loadings can vary.
  - Model 2: Indicator loadings are constrained to be equal.
  - Model 3: Indicator loadings are constrained to be equal and error variances constrained to be equal.
- If τ-equivalent assumptions do not hold, do not use α, as it will not measure reliability accurately.
- Can use alternatives, such as ω, instead.
- ω assumes unidimensionality, but not τ-equivalence.
- $\omega = \alpha$ when the indicators are τ-equivalent.

Talk Outline

Psychometrics

- Introduction
- Classical Test Theory
- Measuring Reliability
- Information
- Single Indicator Models
- Example

- Information is a measure of precision of the estimate of the latent variable.
- Information tells us the proportion of indicator variance that is "true" relative to the amount that is due to "error".
- Unstandardized loadings, alone, are not enough, as their relative contribution depends on size of error variance.
- In (linear) CFA, the standardized loadings will give you the same rank order of the items as far as their information goes.

Indicator Information for CFA

$$I_i(\xi) = \frac{\lambda_i^2}{\theta_i} \tag{19}$$

- $I_i(\xi)$ does not change depending of the value of ξ.
- Thus, items with larger $I_i(\xi)$ values are alwaya better than those with low $I_i(\xi)$ values.

- One can also obtain the amount of information for an entire test, $TI(\xi)$, by summing up the $I_i(\xi)$ across all *i* indicators.

Test Information

$$TI(\xi) = \sum_{i=1}^{k} I_i(\xi) \tag{20}$$

- We can obtain an estimate of reliability from $TI(\xi)$ using the CTT definition of reliability.

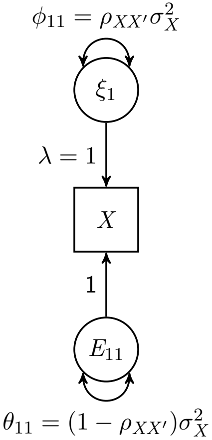
$$\begin{aligned}
\rho_{XX'} &= \frac{\phi}{\phi + \theta} \\
&= \frac{\phi/\phi}{\phi/\phi + \theta/\phi} \\
&= \frac{1}{1 + \theta/\phi} \\
&= \frac{1}{1 + \frac{1}{TI(\xi)}} \\
&= \frac{TI(\xi)}{TI(\xi) + \frac{TI(\xi)}{TI(\xi)}} \\
&= \frac{TI(\xi)}{TI(\xi) + 1}
\end{aligned} \tag{21}$$

Talk Outline

Psychometrics

- Introduction
- Classical Test Theory
- Measuring Reliability
- Information
- Single Indicator Models
- Example

- Single indicator models are CFAâĂŘlike models where a "factor" is measured by a single indicator.



Single Indicator Model

- Identification constraining of
  - factor variance: $\theta_{11} = (\rho_{XX'})\sigma^2_X$ (Reliable portion of X),
  - factor loading: $\lambda = 1$, and
  - unique variance: $\theta_{11} = (1 - \rho_{XX'})\sigma^2_X$ (Unreliable portion of X)
- Assumptions
  - The indicator is unidimensional (only one factor)
  - The reliability of the indicator is known (usually use a previously reported reliability coefficient)

Talk Outline

Psychometrics

- Introduction
- Classical Test Theory
- Measuring Reliability
- Information
- Single Indicator Models
- Example

Talk Outline

Psychometrics

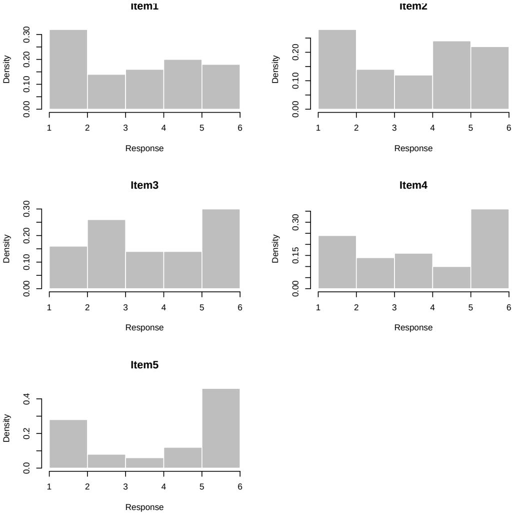
- Introduction
- Classical Test Theory
- Measuring Reliability
- Information
- Single Indicator Models
- Example
  - Reliability
  - Single Indicator Model

- Generate item data using *ltm* package(Rizopoulos, 2006)

```
> # 50 response patterns under a GPCM model
> # with 5 items, with 6 categories each
> thetas <- lapply(1:5, function(u) c(seq(-1, 1*(1/u), len = 5)))
> loadings<-c(1.2, 1.2, 1.2, 1.2, 3)
> for(i in 1:5){
+ thetas[[i]] <-c(thetas[[i]], loadings[i])
+ }
> set.seed(45456)
> items.data<-data.frame(rmvordlogis(50, thetas))
> names(items.data)<-paste("Item", seq(1,5,1), sep="")
>
> head(items.data)
  Item1 Item2 Item3 Item4 Item5
1     5     4     5     6     6
2     6     6     6     6     6
3     4     3     5     3     2
4     5     5     6     4     6
5     6     6     6     6     6
6     1     2     3     2     2
```

- Generate item data using *ltm* package(Rizopoulos, 2006)

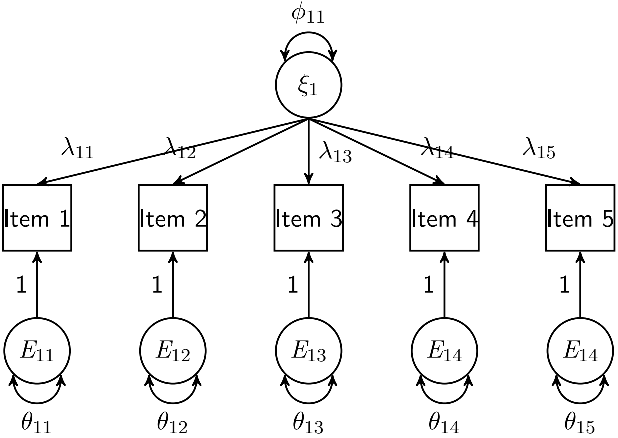
```
#Plot of Item Responses
> par(mfrow=c(3, 2))
> colnames <- dimnames(items.data)[[2]]
> for (i in 1:5) {
+     hist(items.data[,i], xlim=c(1, 6), main=colnames[i], probability=TRUE, col="gray",
      border="white", xlab="Response")
+ }
```



Histogram of Item Responses

- Examine Dimensionality

```
> fa.parallel(items.data)
Parallel analysis suggests that the number of factors =  1  and the number of components
      =  1

> VSS(items.data, plot=FALSE)
The Velicer MAP criterion achieves a minimum of NA  with  1  factors

Velicer MAP
[1] 0.08 0.18 0.40 1.00    NA
```



Congeneric Indicators

- Coefficient α

```
> ##alpha
> items.cov<-cov(items.data)
> dim(items.cov)[1]
[1] 5
>
> (dim(items.cov)[1]/(dim(items.cov)[1]-1))* ((sum(items.cov)- sum(diag(items.cov)))/sum(
    items.cov))
[1] 0.8948386
```

- Coefficient α

```
library(psych)
> alpha(items.data)

Reliability analysis
Call: alpha(x = items.data)

 raw_alpha std.alpha G6(smc) average_r mean  sd
      0.89      0.89    0.88      0.63    4 1.5

 Reliability if an item is dropped:
      raw_alpha std.alpha G6(smc) average_r
Item1      0.88      0.88    0.87      0.66
Item2      0.88      0.88    0.85      0.65
Item3      0.88      0.88    0.85      0.64
Item4      0.87      0.87    0.85      0.63
Item5      0.84      0.85    0.81      0.58

 Item statistics
       n    r r.cor r.drop mean  sd
Item1 50 0.80  0.72   0.69  3.6 1.7
Item2 50 0.81  0.76   0.71  3.8 1.8
Item3 50 0.83  0.77   0.73  4.1 1.6
Item4 50 0.84  0.79   0.74  4.1 1.8
Item5 50 0.91  0.91   0.86  4.3 1.9
```

- Test Congeneric Model

```
congeneric.model<-'
LV=~ NA*Item1 + l1*Item1 + l2*Item2 + l3*Item3 + l4*Item4 + l5*Item5
LV~~1*LV

Item1~~e1*Item1
Item2~~e2*Item2
Item3~~e3*Item3
Item4~~e4*Item4
Item5~~e5*Item5

#Relaibility
omega := ((l1+l2+l3+l4+l5)^2) / ((l1+l2+l3+l4+l5)^2 + e1+e2+e3+e4+e5)

#Information
I1:= l1^2/e1
I2:= l2^2/e2
I3:= l3^2/e3
I4:= l4^2/e4
I5:= l5^2/e5
TestInfo := I1+I2+I3+I4+I5
'
```

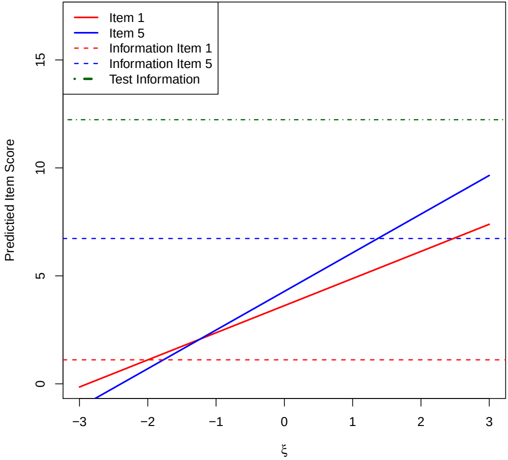
```
> congeneric.fit<-cfa(congeneric.model, data=items.data, meanstructure=TRUE)
> summary(congeneric.fit, standardized=TRUE)
lavaan (0.5-9) converged normally after  17 iterations

  Number of observations                            50

  Estimator                                         ML
  Minimum Function Chi-square                    3.916
  Degrees of freedom                                 5
  P-value                                        0.562

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  LV =~
    Item1    (l1)     1.256    0.217    5.788    0.000    1.256    0.725
    Item2    (l2)     1.379    0.216    6.378    0.000    1.379    0.777
    Item3    (l3)     1.205    0.196    6.150    0.000    1.205    0.758
    Item4    (l4)     1.418    0.223    6.363    0.000    1.418    0.776
    Item5    (l5)     1.790    0.211    8.465    0.000    1.790    0.933

Intercepts:
    Item1             3.620    0.245   14.789    0.000    3.620    2.092
    Item2             3.820    0.251   15.225    0.000    3.820    2.153
```

Psychometrics (cont.)

Example: Reliability

```
    Item3               4.100    0.225   18.227    0.000    4.100    2.578
    Item4               4.060    0.258   15.717    0.000    4.060    2.223
    Item5               4.280    0.271   15.773    0.000    4.280    2.231
    LV                  0.000                               0.000    0.000

Variances:
    LV                  1.000                               1.000    1.000
    Item1    (e1)       1.419    0.315                      1.419    0.474
    Item2    (e2)       1.246    0.289                      1.246    0.396
    Item3    (e3)       1.077    0.245                      1.077    0.426
    Item4    (e4)       1.327    0.308                      1.327    0.398
    Item5    (e5)       0.476    0.222                      0.476    0.129

Defined parameters:
    omega               0.900    0.022   40.124    0.000    0.900    0.900
    I1                  1.111    0.478    2.325    0.020    1.111    1.111
    I2                  1.527    0.630    2.423    0.015    1.527    1.527
    I3                  1.349    0.564    2.391    0.017    1.349    1.349
    I4                  1.514    0.626    2.421    0.015    1.514    1.514
    I5                  6.729    3.847    1.749    0.080    6.729    6.729
    TestInfo           12.231    4.195    2.916    0.004   12.231   12.231
```

```
> fitMeasures(congeneric.fit)
          chisq                df            pvalue    baseline.chisq
          3.916             5.000             0.562           149.554
    baseline.df  baseline.pvalue               cfi               tli
         10.000             0.000             1.000             1.016
           logl unrestricted.logl              npar               aic
       -423.922          -421.964            15.000           877.844
            bic            ntotal              bic2             rmsea
        906.524            50.000           859.441             0.000
 rmsea.ci.lower    rmsea.ci.upper      rmsea.pvalue              srmr
          0.000             0.173             0.628             0.024
    srmr_nomean
          0.028
```



Predicted Item Responses and Information

- Test τ-equivalent model

```
tau.model<-'
LV=~ NA*Item1 + l1*Item1 + l1*Item2 + l1*Item3 + l1*Item4 + l1*Item5
LV~~1*LV

Item1~~e1*Item1
Item2~~e2*Item2
Item3~~e3*Item3
Item4~~e4*Item4
Item5~~e5*Item5

#Relaibility
omega := ((l1+l1+l1+l1+l1)^2) / ((l1+l1+l1+l1+l1)^2 + e1+e2+e3+e4+e5)

#Information
I1:= l1^2/e1
I2:= l1^2/e2
I3:= l1^2/e3
I4:= l1^2/e4
I5:= l1^2/e5
TestInfo := I1+I2+I3+I4+I5
'
```

```
> tau.fit<-cfa(tau.model, data=items.data, meanstructure=TRUE)
> summary(tau.fit, standardized=TRUE)
lavaan (0.5-9) converged normally after  12 iterations

  Number of observations                            50

  Estimator                                         ML
  Minimum Function Chi-square                   13.179
  Degrees of freedom                                 9
  P-value                                        0.155

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  LV =~
    Item1     (l1)    1.424    0.158    8.992    0.000    1.424    0.773
    Item2     (l1)    1.424    0.158    8.992    0.000    1.424    0.780
    Item3     (l1)    1.424    0.158    8.992    0.000    1.424    0.822
    Item4     (l1)    1.424    0.158    8.992    0.000    1.424    0.789
    Item5     (l1)    1.424    0.158    8.992    0.000    1.424    0.834

Intercepts:
    Item1             3.620    0.261   13.886    0.000    3.620    1.964
    Item2             3.820    0.258   14.800    0.000    3.820    2.093
```

Psychometrics (cont.)

Example: Reliability

```
    Item3               4.100    0.245   16.735    0.000    4.100    2.367
    Item4               4.060    0.255   15.899    0.000    4.060    2.248
    Item5               4.280    0.242   17.716    0.000    4.280    2.505
    LV                  0.000                               0.000    0.000

Variances:
    LV                  1.000                               1.000    1.000
    Item1     (e1)      1.370    0.322                      1.370    0.403
    Item2     (e2)      1.303    0.309                      1.303    0.391
    Item3     (e3)      0.973    0.247                      0.973    0.324
    Item4     (e4)      1.233    0.296                      1.233    0.378
    Item5     (e5)      0.891    0.232                      0.891    0.305

Defined parameters:
    omega               0.898    0.023   39.370    0.000    0.898    0.898
    I1                  1.480    0.483    3.065    0.002    1.480    1.480
    I2                  1.556    0.511    3.048    0.002    1.556    1.556
    I3                  2.083    0.712    2.928    0.003    2.083    2.083
    I4                  1.645    0.543    3.029    0.002    1.645    1.645
    I5                  2.277    0.790    2.881    0.004    2.277    2.277
    TestInfo            9.042    2.259    4.002    0.000    9.042    9.042
```

```
> fitMeasures(tau.fit)
            chisq                df            pvalue    baseline.chisq
           13.179             9.000             0.155           149.554
      baseline.df baseline.pvalue               cfi               tli
           10.000             0.000             0.970             0.967
             logl unrestricted.logl              npar               aic
         -428.553          -421.964            11.000           879.107
              bic            ntotal              bic2             rmsea
          900.139            50.000           865.612             0.096
   rmsea.ci.lower    rmsea.ci.upper      rmsea.pvalue              srmr
            0.000             0.200             0.229             0.107
      srmr_nomean
            0.123
```

- Test Parallel model

```
parallel.model<-'
LV=~ NA*Item1 + l1*Item1 + l1*Item2 + l1*Item3 + l1*Item4 + l1*Item5
LV~~1*LV

Item1~~e1*Item1
Item2~~e1*Item2
Item3~~e1*Item3
Item4~~e1*Item4
Item5~~e1*Item5

#Relaibility
omega := ((l1+l1+l1+l1+l1)^2) / ((l1+l1+l1+l1+l1)^2 + e1+e1+e1+e1+e1)

#Information
I1:= l1^2/e1
I2:= l1^2/e1
I3:= l1^2/e1
I4:= l1^2/e1
I5:= l1^2/e1
TestInfo := I1+I2+I3+I4+I5
'
```

```
> parallel.fit<-cfa(parallel.model, data=items.data, meanstructure=TRUE)
> summary(parallel.fit, standardized=TRUE)
lavaan (0.5-9) converged normally after   8 iterations

  Number of observations                            50

  Estimator                                         ML
  Minimum Function Chi-square                   15.587
  Degrees of freedom                                13
  P-value                                        0.272

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  LV =~
    Item1    (l1)     1.406    0.157    8.936    0.000    1.406    0.794
    Item2    (l1)     1.406    0.157    8.936    0.000    1.406    0.794
    Item3    (l1)     1.406    0.157    8.936    0.000    1.406    0.794
    Item4    (l1)     1.406    0.157    8.936    0.000    1.406    0.794
    Item5    (l1)     1.406    0.157    8.936    0.000    1.406    0.794

Intercepts:
    Item1             3.620    0.251   14.449    0.000    3.620    2.043
    Item2             3.820    0.251   15.248    0.000    3.820    2.156
```

Psychometrics (cont.)

Example: Reliability

```
    Item3               4.100    0.251   16.365    0.000    4.100    2.314
    Item4               4.060    0.251   16.206    0.000    4.060    2.292
    Item5               4.280    0.251   17.084    0.000    4.280    2.416
    LV                  0.000                               0.000    0.000

Variances:
    LV                  1.000                               1.000    1.000
    Item1     (e1)      1.162    0.116                      1.162    0.370
    Item2     (e1)      1.162    0.116                      1.162    0.370
    Item3     (e1)      1.162    0.116                      1.162    0.370
    Item4     (e1)      1.162    0.116                      1.162    0.370
    Item5     (e1)      1.162    0.116                      1.162    0.370

Defined parameters:
    omega               0.895    0.024   38.054    0.000    0.895    0.895
    I1                  1.702    0.425    4.002    0.000    1.702    1.702
    I2                  1.702    0.425    4.002    0.000    1.702    1.702
    I3                  1.702    0.425    4.002    0.000    1.702    1.702
    I4                  1.702    0.425    4.002    0.000    1.702    1.702
    I5                  1.702    0.425    4.002    0.000    1.702    1.702
    TestInfo            8.509    2.126    4.002    0.000    8.509    8.509
```

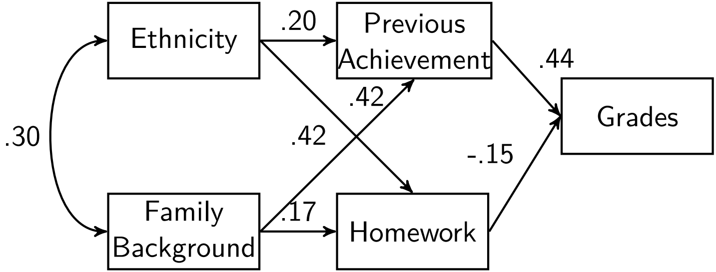
```
> fitMeasures(parallel.fit)
            chisq                df            pvalue    baseline.chisq
           15.587            13.000             0.272           149.554
      baseline.df baseline.pvalue                 cfi               tli
           10.000             0.000             0.981             0.986
             logl unrestricted.logl              npar               aic
         -429.757          -421.964             7.000           873.515
              bic            ntotal              bic2             rmsea
          886.899            50.000           864.927             0.063
   rmsea.ci.lower    rmsea.ci.upper      rmsea.pvalue              srmr
            0.000             0.161             0.385             0.102
      srmr_nomean
            0.117
```

Talk Outline

Psychometrics

- Introduction
- Classical Test Theory
- Measuring Reliability
- Information
- Single Indicator Models
- Example
  - Reliability
  - Single Indicator Model

- Path model taken from Keith (2006, chapter 13)



Path Model

Covariance Matrix for Path Model

		1	2	3	4	5
1	ETHNICITY	0.18	0.11	1.20	0.03	0.08
2	FAMBACK	0.11	0.69	3.54	0.18	0.34
3	PREACH	1.20	3.54	79.17	2.07	6.44
4	HOMEWORK	0.03	0.18	2.07	0.65	0.34
5	GRADES	0.08	0.34	6.44	0.34	2.19

```
#Single indicator Models
> #Homework data from Keith book (figure 13.4)
> HW.cor.matrix<-matrix(c
    (1, 0.3041, 0.3228, 0.0832, 0.1315, 0.3041, 1, 0.4793, 0.2632, 0.2751,
0.3228, 0.4793, 1, 0.2884, 0.489, 0.0832, 0.2632, 0.2884, 1, 0.2813, 0.1315,
0.2751, 0.489, 0.2813, 1), 5,5)
> HW.sd.vector<-c(.4186, .8311, 8.8978, .8063, 1.479)
> HW.cov.matrix <-cor2cov(HW.cor.matrix, HW.sd.vector)
> dimnames(HW.cov.matrix) <- list(c("ETHNICITY", "FAMBACK", "PREACH", "HOMEWORK", "GRADES
    "), c("ETHNICITY", "FAMBACK", "PREACH", "HOMEWORK", "GRADES"))
> round(HW.cov.matrix, 3)
          ETHNICITY FAMBACK PREACH HOMEWORK GRADES
ETHNICITY     0.175   0.106  1.202    0.028  0.081
FAMBACK       0.106   0.691  3.544    0.176  0.338
PREACH        1.202   3.544 79.171    2.069  6.435
HOMEWORK      0.028   0.176  2.069    0.650  0.335
GRADES        0.081   0.338  6.435    0.335  2.187
```

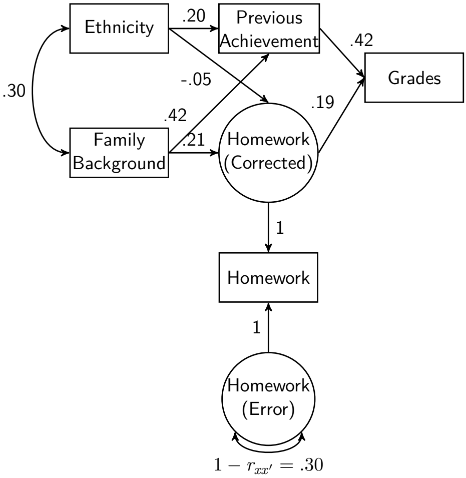
```
> original.model<-'
+ # regressions
+ GRADES ~ PREACH + HOMEWORK
+ PREACH ~ ETHNICITY + FAMBACK
+ HOMEWORK ~ PREACH + ETHNICITY + FAMBACK
+ '
> original.fit<-sem(original.model, sample.cov=HW.cov.matrix, sample.nobs=1000)
> summary(original.fit, standardized=TRUE)
```

```

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Regressions:
  GRADES ~
    PREACH            0.074    0.005   15.665    0.000    0.074    0.445
    HOMEWORK          0.281    0.052    5.387    0.000    0.281    0.153
  PREACH ~
    ETHNICITY         4.147    0.605    6.852    0.000    4.147    0.195
    FAMBACK           4.496    0.305   14.750    0.000    4.496    0.420
  HOMEWORK ~
    PREACH            0.020    0.003    6.298    0.000    0.020    0.220
    ETHNICITY        -0.076    0.062   -1.225    0.220   -0.076   -0.039
    FAMBACK           0.165    0.034    4.902    0.000    0.165    0.170

Variances:
    GRADES            1.616    0.072                      1.616    0.739
    PREACH           58.190    2.602                     58.190    0.736
    HOMEWORK          0.581    0.026                      0.581    0.895
```

- What if $r_{XX'} = .70$ for the Homework variable?

```
> #Account for unreliability of HW variable
> reliable.model<-'
+ #measurement model
+ HMWK =~ HOMEWORK
+
+ #constrain error variance of HMWK to be .30*VAR(HOMEWORK) = .30*.8063^2
+ HOMEWORK ~~ (.30*.650)*HOMEWORK
+
+ # regressions
+ GRADES ~ PREACH + HMWK
+ PREACH ~ ETHNICITY + FAMBACK
+ HMWK ~ PREACH + ETHNICITY + FAMBACK
+ '
```

```
> reliable.fit<-sem(reliable.model, sample.cov=HW.cov.matrix, sample.nobs=1000)
> summary(reliable.fit, standardized=TRUE)

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  HMWK =~
    HOMEWORK          1.000                               0.674    0.836

Regressions:
  GRADES ~
    PREACH            0.070    0.005   14.138    0.000    0.070    0.423
    HMWK              0.422    0.078    5.427    0.000    0.284    0.192
  PREACH ~
    ETHNICITY         4.147    0.605    6.852    0.000    4.147    0.195
    FAMBACK           4.496    0.305   14.750    0.000    4.496    0.420
  HMWK ~
    PREACH            0.020    0.003    6.306    0.000    0.030    0.263
    ETHNICITY        -0.081    0.062   -1.319    0.187   -0.121   -0.050
    FAMBACK           0.167    0.033    4.979    0.000    0.247    0.206

Variances:
    HOMEWORK          0.195                               0.195    0.300
    GRADES            1.591    0.073                      1.591    0.728
    PREACH           58.190    2.602                     58.190    0.736
    HMWK              0.386    0.026                      0.849    0.849
```



Path Model with Corrected Homework Variable

Categorical Outcomes

Table of Contents

Parameterizations of Models with Categorical Outcomes

- IRT Approach
- Underlying Variable Approach
  - Marginal Parameterization
  - Conditional Parameterization
  - Scaling the Factor

Relationship Between Factor Analysis and Item Response Models

Data Analysis

Talk Outline

Parameterizations of Models with Categorical Outcomes

- IRT Approach
- Underlying Variable Approach

Parameterizations

- There are multiple ways to parameterize factor models with categorical outcomes (Kamata & Bauer, 2008).
- Education measurement often takes the logistic IRT perspective (Hambleton & Swaminathan, 1985).
- Psychology/statistics often takes the underlying variables approach (e.g., Bartholomew, Knott, & Moustaki, 2011).

Talk Outline

Parameterizations of Models with Categorical Outcomes

- IRT Approach
- Underlying Variable Approach

- We previously discussed the logistic IRT approach

Two Parameter Logistic (2PL) Model

$$p(x_{ij} = 1|\theta_i, a_j, b_j) = \frac{1}{\exp(a_j[b_j - \theta_i]) + 1}$$

where
x_{ij} is the *i*th person's response to item *j*,
θ_i is the *i*th person's level on the latent trait, θ,
b_j is the *j*th item's location, and
a_j is the *j*th item's discrimination

- The two-parameter model can be re- parameterized as

$$f(\alpha_i \theta + \beta_i)$$

where
α_i is the *i*th item's slope
β_i is the *i*th item's intercept, and
$f()$ is the logistic or normal cumulative distribution.

- Of note, *b* from the original IRT model can be obtained via $b = \frac{-\beta}{\alpha}$

Cumulative Normal Distribution

$$p(Z) = \int_{-\infty}^{Z} \frac{1}{\sqrt{2}\pi} exp(-.5t^2)\, dt$$

Cumulative Logistic Distribution

$$p(Z) = \frac{1}{\exp(-Z) + 1}$$

Talk Outline

Parameterizations of Models with Categorical Outcomes

IRT Approach

Underlying Variable Approach

- Underlying variable approach assumes the categorical outcomes are realizations of continuous underlying response variables that are incompletely observed.
- For each categorical outcome, x_i, there is an incompletely observed continuous variable x_i^* and $x_i^* \sim N(\mu_i, \sigma_i^2)$.

$$x_i^* = \upsilon_i + \lambda_i \theta + \epsilon_i$$

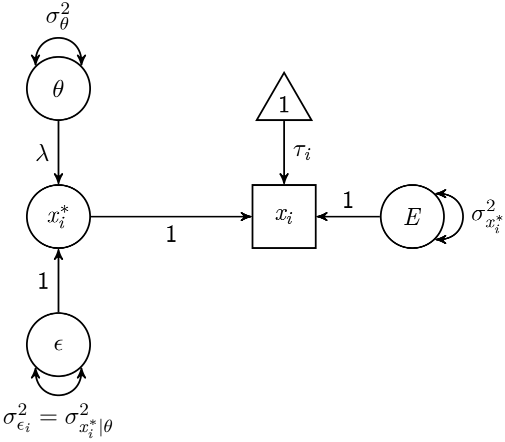
where
υ_i is the intercept for item *i*
λ_i is the factor loading for item *i*
θ is the latent factor level for person *j* (subscript not shown), and
ϵ_i is the residual for item *i*, $\epsilon_i \sim N$

- x_i and x_i^* are related as follows

$$x_i = \begin{cases} 1 & \text{if } x_i^* \geq \tau_i \\ 0 & \text{if } x_i^* < \tau_i \end{cases}$$

where
τ_i is the threshold.

- Since the only information known about x_i^* is its relationship to x_i and its distributional form, nothing is lost if μ_i and σ_i^2 are constrained to certain values.



Path Diagram of Underlying Variable Approach

Parameterizations

Underlying Variable Approach

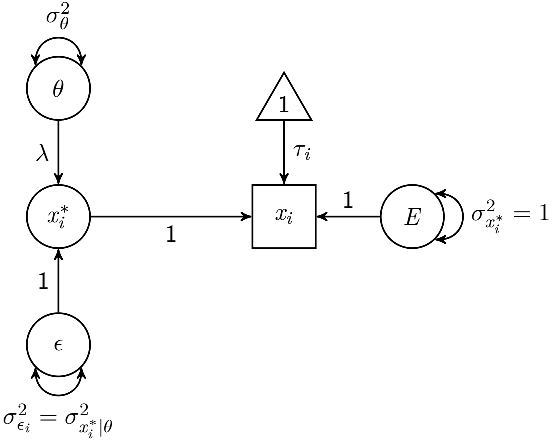
- The model is underidentified, so the scale and location of either $\sigma^2_{x_i^*}$ or $\sigma^2_{\epsilon_i}$ have to be constrained.
- The two typical parameterizations of the underlying variable approach, then take one of two forms
  - constraining $\sigma^2_{x_i^*}$
  - constraining $\sigma^2_{\epsilon_i}$
- For both models, often the intercept υ_i is set to zero and τ_i is estimated.
  - Nothing says that τ_i could not be set to 0 and estimate υ_i, though.

Talk Outline

Parameterizations of Models with Categorical Outcomes

- IRT Approach
- Underlying Variable Approach
  - Marginal Parameterization
  - Conditional Parameterization
  - Scaling the Factor

- In Mplus, it is called the *Delta* parameterization (default)
- Currently, the only way `lavaan` handles categorical data
- In this approach, $\sigma^2_{x_i^*}$ is constrained to be 1.0 for all items.
- It gets its name from the fact that the marginal distribution of x_i^* is standardized.
- $\sigma^2_{x_i^*} = \sigma^2_{\epsilon_i} + \lambda_i^2 \sigma^2_\theta$, so $\sigma^2_{\epsilon_i} = 1 - \lambda_i^2 \sigma^2_\theta$.
- This is the polychoric/tetrachoric method of estimating a correlation (Joreskog, 1994).
- Common method used with binary factor models.



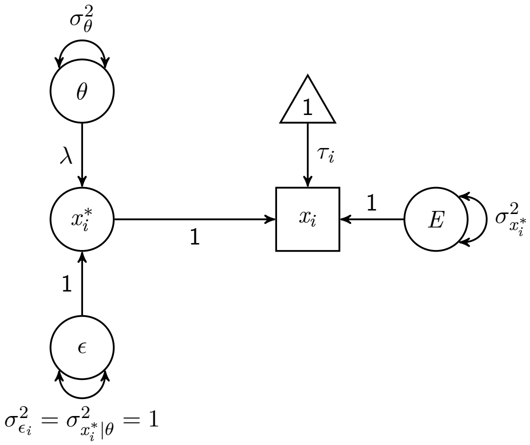
Path Diagram of Underlying Variable Approach

Talk Outline

Parameterizations of Models with Categorical Outcomes

- IRT Approach
- Underlying Variable Approach
  - Marginal Parameterization
  - Conditional Parameterization
  - Scaling the Factor

- In Mplus, it is the *Theta* parameterization
- In this approach, $\sigma^2_{\epsilon_i}$ is constrained to be 1.0 for all items.
- It gets its name from the fact that $\sigma^2_{\epsilon_i}$, the conditional distribution of x^*_i (i.e, $\sigma^2_{\epsilon_i} = \sigma^2_{x^*_i|\theta}$), is standardized.
- $\sigma^2_{x^*_i} = \lambda^2_i \sigma^2_\theta + \sigma^2_{\epsilon_i} = \lambda^2_i \sigma^2_\theta + 1$
- Similar to probit regression model.



Path Diagram of Underlying Variable Approach

Talk Outline

Parameterizations of Models with Categorical Outcomes

- IRT Approach
- Underlying Variable Approach
  - Marginal Parameterization
  - Conditional Parameterization
  - Scaling the Factor

- As with linear factor models, the factor, θ, must be scaled.
- Key Indicator
  - Allow mean and variance of θ to be estimated.
  - (Linear): set one intercept to 0 and one loading to 1.
  - (Categorical): set one threshold, τ_i, to 0 and one loading, λ_i to 1.
- Standardized θ
  - All item parameters are estimated.
  - (Linear): set latent variable mean to 0 and variance to 1.
  - (Categorical): set latent variable mean, $\mu_\theta = 0$, and variance $\sigma^2_\theta = 1$.
  - This is the approach taken by most IRT software.

Talk Outline

Relationship Between Factor Analysis and Item Response Models

FA and IRT

- Using the regression form of the two-parameter IRT model

$$f(\alpha_i\theta + \beta_i)$$

- Takane and de Leeuw (1987) showed that when $\mu_\theta = 0$ and $\sigma^2_\theta = 1$:
  - $\alpha_i = \frac{\lambda_i}{\sqrt{\sigma^2_{\epsilon_i}}}$
  - $\beta_i = \frac{-\tau_i}{\sqrt{\sigma^2_{\epsilon_i}}}$
  - where ϵ_i is the residual for the *i*th item

FA and IRT

- In the *conditional* underlying variable model $\sigma^2_{\epsilon_i} = 1$, so $\sqrt{\sigma^2_{\epsilon_i}} = 1$.
  - This means that for the conditional model with standardized θ, $\alpha_i = \lambda_i$ and $\beta_i = -\tau$.
- In the *marginal* underlying variable model, $\sigma^2_{\epsilon_i} = 1 - \lambda_i^2 \sigma^2_\theta$.
  - This means that for the marginal model with standardized θ, $\alpha_i = \frac{\lambda_i}{\sqrt{1-\lambda_i^2\sigma^2_\theta}}$ and $\beta_i = \frac{-\tau}{\sqrt{1-\lambda_i^2\sigma^2_\theta}}$.
- Similar conversions can be derived for the *Key Indicator* models.

FA and IRT

	Key Indicator	Standardized θ
Marginal	$\alpha_i = \frac{\lambda_i \sqrt{\sigma^2_\theta}}{\sqrt{1-\lambda^2_i \sigma^2_\theta}}$	$\alpha_i = \frac{\lambda^2_i}{\sqrt{1-\lambda^2_i}}$
	$\beta = \frac{-[\tau_i - \lambda_i \mu_\theta]}{\sqrt{1-\lambda^2_i \sigma^2_\theta}}$	$\beta_i = \frac{-\tau_i}{\sqrt{1-\lambda^2_i}}$
Conditional	$\alpha_i = \lambda_i \sqrt{\sigma^2_\theta}$	$\alpha_i = \lambda_i$
	$\beta_i = -[\tau_i - \lambda_i \mu_\theta]$	$\beta_i = -\tau_i$

Conversion Formulae, taken from Kamata and Bauer (2008, p. 144)

Talk Outline

Data Analysis

Data Analysis

- Data is $N = 1000$ respondents on $n = 6$ items on the LAST6 (Bock & Aitkin, 1981)
- In `psych` package: `lsat6`.

Data Analysis

- Descriptive statistics

```
> descript(lsat6)

Descriptive statistics for the 'lsat6' data-set

Sample:
 5 items and 1000 sample units; 0 missing values

Proportions for each level of response:
                logit
Q1  0.076 0.924 2.4980
Q2  0.291 0.709 0.8905
Q3  0.447 0.553 0.2128
Q4  0.237 0.763 1.1692
Q5  0.130 0.870 1.9010
```

Data Analysis

- There are multiple IRT packages in `R`: http://cran.cc.uoc.gr/web/views/Psychometrics.html
- We'll use the `ltm` package (Rizopoulos, 2006).
  - Uses Logistic distribution
  - By default, it estimates β and α instead of *b* and *a*.

Data Analysis

```
> library(ltm)
> library(psych) # For the lsat6 data
> lsat.IRT<-ltm(lsat6~z1, IRT.param=FALSE, control = list(GHk = 100, iter.em = 20))
> coef(lsat.IRT)
   (Intercept)        z1
Q1   2.7727010 0.8250292
Q2   0.9900525 0.7233071
Q3   0.2496680 0.8900640
Q4   1.2847498 0.6889666
Q5   2.0533611 0.6571570
```

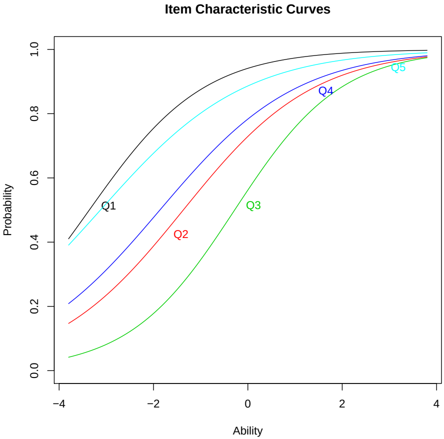
- `(Intercept)`$= \alpha$ and `z1`$= \beta$
- `GHk`: Quadrature points
- `iter.em`: EM iterations

Data Analysis

```
#2 PL under "regular IRT" model
> ltm(lsat6~z1, IRT.param=TRUE, control = list(GHk = 100, iter.em = 20))

Call:
ltm(formula = lsat6 ~ z1, IRT.param = TRUE, control = list(GHk = 100,
    iter.em = 20))

Coefficients:
    Dffclt  Dscrmn
Q1  -3.361   0.825
Q2  -1.369   0.723
Q3  -0.281   0.890
Q4  -1.865   0.689
Q5  -3.125   0.657
```

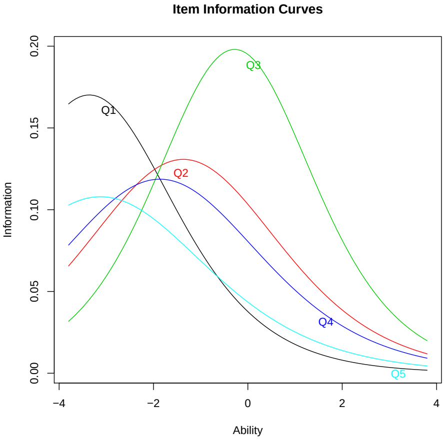
Data Analysis

```
#Item Characteristic Curves
> plot(lsat.IRT)
```

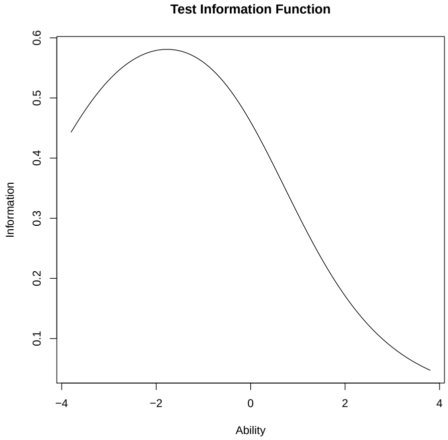


Data Analysis

```
#Item Information Curves
> plot(lsat.IRT, type = "IIC")
```



```
#Test Information Function
> plot(lsat.IRT, type = "IIC", items=0)
```



- For a 2 Parameter Normal Ogive Model, use the `irt.fa()` function in the `psych` package.

```
> ls <- irt.fa(lsat6, plot=FALSE, correct=TRUE)
> ls$irt
$difficulty
$difficulty[[1]]
        Q1         Q2         Q3         Q4         Q5
-1.5496525 -0.6024765 -0.1523948 -0.7705779 -1.1887859


$discrimination
         MR1
Q1 0.4126107
Q2 0.4448568
Q3 0.5550697
Q4 0.3978793
Q5 0.3374248
```

- Using the regression form of the two-parameter Normal Ogive model

```
> lsat.NO <- irt.fa(lsat6, plot=FALSE, correct=TRUE)
> lsat.NO$fa
Factor Analysis using method =  minres
Call: fa(r = r, nfactors = nfactors, n.obs = n.obs)
Standardized loadings (pattern matrix) based upon correlation matrix
    MR1   h2   u2
Q1  0.38 0.15 0.85
Q2  0.41 0.17 0.83
Q3  0.49 0.24 0.76
Q4  0.37 0.14 0.86
Q5  0.32 0.10 0.90

> lsat.NO$tau
        Q1         Q2         Q3         Q4         Q5
-1.4325027 -0.5504657 -0.1332445 -0.7159860 -1.1263911
```

- `lavaan` will estimate λ and τ from the *marginal* underlying variable approach, which can be converted to IRT parameters.

```
> twoP.model<-'
+ Theta =~ l1*Q1 + l2*Q2 + l3*Q3 + l4*Q4 + l5*Q5
+ Q1 | th1*t1
+ Q2 | th2*t1
+ Q3 | th3*t1
+ Q4 | th4*t1
+ Q5 | th5*t1
+
+ #Convert regression to IRT
+ alpha1 := (l1)/sqrt(1-l1^2)
+ alpha2 := (l2)/sqrt(1-l2^2)
+ alpha3 := (l3)/sqrt(1-l3^2)
+ alpha4 := (l4)/sqrt(1-l4^2)
+ alpha5 := (l5)/sqrt(1-l5^2)
+ beta1 := (-th1)/sqrt(1-l1^2)
+ beta2 := (-th2)/sqrt(1-l2^2)
+ beta3 := (-th3)/sqrt(1-l3^2)
+ beta4 := (-th4)/sqrt(1-l4^2)
+ beta5 := (-th5)/sqrt(1-l5^2)
+ '
```

Data Analysis

```
> twoP.fit<-cfa(twoP.model, data=data.frame(lsat6),  std.lv=TRUE, ordered=c("Q1","Q2","Q3
    ","Q4", "Q5"))
> summary(twoP.fit, standardized=TRUE)
lavaan (0.5-9) converged normally after  27 iterations

  Number of observations                          1000

  Estimator                                       DWLS      Robust
  Minimum Function Chi-square                    4.051       4.744
  Degrees of freedom                                 5           5
  P-value                                        0.542       0.448
  Scaling correction factor                                  0.867
  Shift parameter                                            0.070
    for simple second-order correction (Mplus variant)

Parameter estimates:

  Information                                 Expected
  Standard Errors                           Robust.sem

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  Theta =~
    Q1        (l1)    0.389    0.112    3.486    0.000    0.389    0.389
    Q2        (l2)    0.397    0.083    4.801    0.000    0.397    0.397
    Q3        (l3)    0.471    0.088    5.347    0.000    0.471    0.471
    Q4        (l4)    0.377    0.083    4.536    0.000    0.377    0.377
    Q5        (l5)    0.342    0.093    3.690    0.000    0.342    0.342
```

Data Analysis (cont.)

```

Intercepts:
    Theta             0.000                               0.000    0.000

Thresholds:
    Q1|t1    (th1)   -1.433    0.059  -24.431    0.000   -1.433   -1.433
    Q2|t1    (th2)   -0.550    0.042  -13.133    0.000   -0.550   -0.550
    Q3|t1    (th3)   -0.133    0.040   -3.349    0.001   -0.133   -0.133
    Q4|t1    (th4)   -0.716    0.044  -16.430    0.000   -0.716   -0.716
    Q5|t1    (th5)   -1.126    0.050  -22.395    0.000   -1.126   -1.126

Variances:
    Theta             1.000                               1.000    1.000

Defined parameters:
    alpha1            0.423    0.143    2.957    0.003    0.423    0.423
    alpha2            0.433    0.107    4.044    0.000    0.433    0.433
    alpha3            0.534    0.128    4.159    0.000    0.534    0.534
    alpha4            0.407    0.105    3.892    0.000    0.407    0.407
    alpha5            0.364    0.112    3.258    0.001    0.364    0.364
    beta1            -1.555    0.100  -15.586    0.000   -1.555   -1.555
    beta2            -0.600    0.051  -11.809    0.000   -0.600   -0.600
    beta3            -0.151    0.046   -3.297    0.001   -0.151   -0.151
    beta4            -0.773    0.054  -14.232    0.000   -0.773   -0.773
    beta5            -1.199    0.067  -17.798    0.000   -1.199   -1.199
```

	`psych::irt.fa()`				`lavaan`, standardized θ			
	IRT		Regression		IRT		Regression	
	α	β	λ	τ	α	β	λ	τ
Q1	0.41	-1.55	0.38	-1.43	0.42	-1.55	0.39	-1.43
Q2	0.45	-0.60	0.41	-0.55	0.43	-0.60	0.4	-0.55
Q3	0.56	-0.15	0.49	-0.13	0.53	-0.15	0.47	-0.13
Q4	0.40	-0.77	0.37	-0.72	0.41	-0.77	0.38	-0.72
Q5	0.34	-1.19	0.32	-1.13	0.36	-1.20	0.34	-1.13

Power, Nonnormality, and Missing Data

Table of Contents

Nonnomral Variables

- Scaled χ^2
- Bootstrapping
  - Confidence Interval Review
  - Example
- Example of CFA with Nonnormal Data

Power Review

- Example

Power Through a Monte Carlo Study

- Example (Continued)

Power in Structural Equation Modeling

Table of Contents (cont.)

- Power to Detect an Added Path
    - Example
- Overall Power to Reject a Model
    - Example
- Monte Carlo

Missing Data

- Motivation
- Types of Missing Data
    - Missing Completely At Random
    - Missing At Random
    - Missing Not at Random

Table of Contents (cont.)

- Traditional Data Handling with Missing Values
  - Listwise Deletion
  - Pairwise Deletion
  - Mean Imputation
- Modern Data Handling with Missing Values
  - Full Information Maximum Likelihood
  - Multiple Imputation
  - Auxiliary Variables
- Example
- Missing Data in `R`

Talk Outline

Nonnomral Variables

- Scaled χ^2
- Bootstrapping
- Example of CFA with Nonnormal Data

Nonnomral Variables

- A major assumption of SEM is that the manifest variables are multivariate normal.
- There a multiple ways that variables can be nonnormal:
  - Categorical
  - Skewness and/or kurtosis, univariate or multivariate
  - Outliers

Nonnomral Variables

- When have non-normal variables, there are multiple ways to deal with the data (Enders, 2001; West, Finch, & Curran, 1995)
  - If categorical and have fewer than 4 response alternatives, use IRT/categorical FA methods or parceling
  - Use a scaled χ^2 and robust standard errors
  - Use bootstrapped estimated of the standard errors

Talk Outline

Nonnomral Variables

Scaled χ^2

Bootstrapping

Example of CFA with Nonnormal Data

- Researchers have developed a set of corrected normal-theory test statistics that adjust the goodness-of-fit χ^2 for bias due to (multivariate) nonnormality.
- Correcting the regular χ^2 value for nonnormality requires the estimation of a scaling correction factor (c), which reflects the amount of average multivariate kurtosis.
- One divides the goodness-of-fit χ^2 value for the model by c to obtain the scaled χ^2.

- Several corrections have been proposed for the χ^2 model test, the most often used are the Satorra and Bentler (1994) and the Yuan and Bentler (1998) corrections.
- In addition to the robust χ^2, robust standard errors (using sandwich estimators; White, 1982), using the observed residual variances to correct the asymptotic standard errors.
- The robust χ^2 tests and standard errors are generally more accurate than the asymptotic tests when data are non-normal (Curran, West, & Finch, 1996).

The scaling factor for testing the difference between the baseline and nested model, c_d is

$$c_d = \frac{d_0\,c_0 - d_1\,c_1}{d_0 - d_1}$$

where
d_i are the degrees of freedom for model i, and
c_0 is the scaling factor (i.e., ratio of χ^2 values for regular and robust estimators) for model i.

The scaled difference test statistic, T_d^* is

$$T_d^* = \frac{T_0 - T_1}{c_d}$$

where
T_i is the unscaled χ^2 value for model i.

- In `lavaan` (and Mplus), the Satorra-Benter correction is called MLM and the Yuan-Bentler correction is called MLR.
- MLM uses "classic" robust standard errors, while MLR uses Huber-White (sandwich) robust estimators.

Talk Outline

Nonnomral Variables

- Scaled χ^2
- Bootstrapping
- Example of CFA with Nonnormal Data

- The idea behind bootstrapping is to mimic the sampling distribution of the statistic(s) of interest by *resampling with replacement* many, many times (Efron & Tibshirani, 1994).
- They are typically used when either
  - the statistic(s) of interest do not have a easy to compute distribution.
  - the assumptions for the statistic(s) of interest are not met.
- They can be used for many things, but a common use in SEM is to develop confidence intervals.

Talk Outline

Nonnomral Variables

- Scaled χ^2
- Bootstrapping
  - Confidence Interval Review
  - Example
- Example of CFA with Nonnormal Data

- Confidence intervals (CI) concern a statistic
  - e.g., mean, variance
- Range from $> 0\%$ to $< 100\%$

- Interpretation of a CI:

  If we took *a lot* of samples from the same population, and construct $X\%$ CIs each time, approximately $X\%$ of them will contain the value of the parameter.

- **Not** the probability that a parameter lies between the upper and lower point.
  - The parameter is fixed (i.e., does not have a distribution of possible values), but the confidence interval is random (as it depends on the random sample).
  - The probability that the parameter is actually inside the given interval is either 0 or 1 (the unknown parameter is not-random, so is either there or not).
- **Not** "how confident" you are about a *statistic*
  - Confidence is in the method
- It is "... one interval generated by a procedure that will give correct intervals 95% of the time" (Antelman, 1997, p. 375)
- For an alternative approach, see (Edwards, Lindman, & Savage, 1963)

- CI can be used to do hypothesis testing
  - Set H_0 and α
  - Gather data
  - Calculate the $(1 - [\alpha/2])$100% CI
  - If the the $(1 - [\alpha/2])$100% CI does not contain null value, reject H_0, otherwise fail to reject.

Talk Outline

Nonnomral Variables

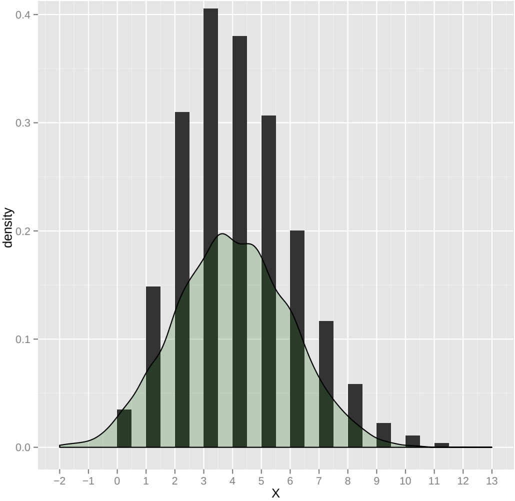
- Scaled χ^2
- Bootstrapping
  - Confidence Interval Review
  - Example
- Example of CFA with Nonnormal Data

- Let's work through an example.
- Say $X \sim P(4)$, that is, X comes from a Poisson distribution with $\lambda = 4$. [6]
- Its probability distribution function, for k cases, is $p(k) = \frac{e^{-\lambda}\lambda^k}{k!}$, so with $\lambda = 4$ we have $p(k) = \frac{e^{-4}4^k}{k!}$.
- Thus, the probability of observing, say, 5 cases is .16.

[6] Poison distributions are frequently used with count data (Atkins, Baldwin, Zheng, Gallop, & Neighbors, in press).

Nonnomral Variables

Bootstrapping: Example



Poisson Distribution with $\lambda = 4$

Nonnomral Variables

Bootstrapping: Example

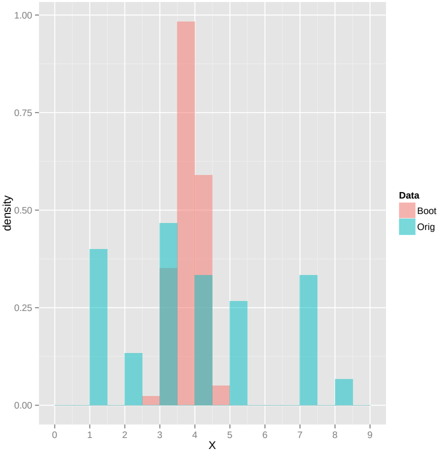
- Let's draw $n = 30$ random cases from a poisson distribution with $\lambda = 4$, $X \sim p(\lambda = 4)$.

```
> set.seed(45678)
> X<-rpois(30,4)
> X
 [1] 7 3 2 3 5 7 3 2 5 1 1 4 3 1 5 7 1 4 3 4 4 3 1 5 1 8 3 7 4 7
```

- The sample mean and SE are $\bar{x} = \frac{\sum_{i=1}^{30} X_i}{30} = 3.8$ and $\frac{sd}{\sqrt{30}} = 0.39$.

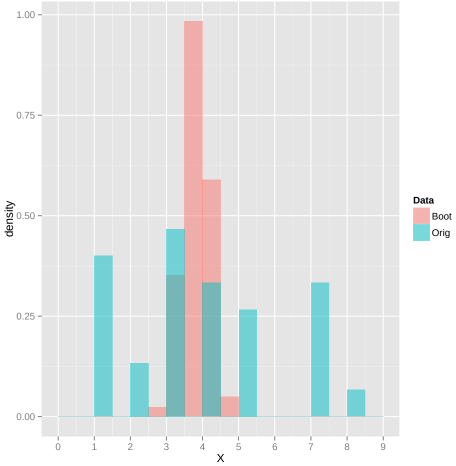
```
> mean(X)
[1] 3.8
> sd(X)/sqrt(length(X))
[1] 0.3906964
```

- Since the dataset size, *n*, is not large and is not "normal", there is likely some suspicions about the accuracy of $\bar{x}$ and its confidence interval, which is based on the normality assumption.
- Now, lets collect, at random and with replacement, $m = 1000$ samples of size $n = 30$ from the original dataset.
- These are called *bootstrap samples*, X^*.
- For each X^* we can calculate the its mean, $\bar{x}^*$



Poisson Distribution with $\lambda = 4$ and $m = 1000$ Bootstrapped Means from the Original Data

- Notice that while $X \sim p(\lambda = 4)$, $\bar{x}^*$ looks like it came from a Normal distribution.
- Usually, the bootstrapped distribution of a statistic will mimic the sampling distribution of the statistic.



- The mean and standard deviation of the *m* bootstrapped means are

$$\bar{x}^*_m = \frac{1}{m}\sum_{i=1}^{m}\bar{x}^*_i$$

$$s_{\bar{x}^*} = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}\left(\bar{x}^* - \bar{x}^*_m\right)^2}$$

- For the data used for the previous figure, $\bar{x}^*_m = 3.81$ and $s_{\bar{x}^*} = 0.37$.
- The *bias* of the bootstrapped statistic is

$$\bar{x}^*_m - \bar{x} = 3.81 - 3.80 = 0.01$$

- For the *mean*, the bias tends to be quote small.

- This bootstrap bias is an approximation of the bias between $\bar{x}$ and μ.
- Likewise $s_{\bar{x}^*}$ is an approximation of the SE
  - i.e., SE $= 0.39$ and $s_{\bar{x}^*} = 0.37$.
- Again, for the mean, the difference between SE and $s_{\bar{x}^*}$ tends to be quite small.
- We could now create a $(1-\alpha)\%$ CI for the mean

$$(1-\alpha)\% CI = \bar{x}_m^* \pm t_{df=n-1, 1-\alpha/2} s_{\bar{x}^*}$$

- We could also calculate a bootstrapped T, via $T^* = \frac{\bar{x}_m^* - \bar{x}}{s_{\bar{x}^*}/\sqrt{n}}$, and then etting m bootstrapped estimates of it.

- An alternative way of estimating a $(1-\alpha)\%$ CI is to use the *m* values of X^*
- For this method, we take the values at the $\alpha/2$ and $1-\alpha/2$ quantiles of X^* as the estimates of the lower and upper bound, respectively, of the confidence interval.

- There are multiple ways to do bootstrapping in R.

```
> ## Bootstrap Method 1
> set.seed(45678)
> X<-rpois(30,4)
> m<-1000 # nnumber of iterations
> xstar<- numeric(1000)
> for (i in 1:m) xstar[i] <- mean(sample(X,replace=T))
> ### mean of xstar
> mean(xstar)
[1] 3.804
> ### sd of xstar
> sd(xstar)
[1] 0.3909094
> ### CI --percentile method
> alpha<-.05
> quantile(xstar,alpha/2) # lower limit
    2.5%
3.066667
> quantile(xstar,1-alpha/2) #upper limit
   97.5%
4.633333
> ### CI --bootstrapped standard errors  method
> mean(xstar) - qt(1-alpha/2, 29)*sd(xstar) # lower limit
[1] 3.004501
> mean(xstar) + qt(1-alpha/2, 29)*sd(xstar) # upper limit
[1] 4.603499
```

```
> ## Bootstrap Method 2
> mean.boot<-function(data,d){return(mean(data[d]))} #Have to write a function that
    contains the statistics and has an index
> X.boot<-boot(data=X, mean.boot, R=1000)
> #The 1000 Bootstrapped Means
> X.star<-X.boot$t
> mean(X.star)
[1] 3.824167
> apply(X.star, 2, sd)
[1] 0.3757103
>
> boot.ci(X.boot, conf = 0.95)
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 1000 bootstrap replicates

CALL :
boot.ci(boot.out = X.boot, conf = 0.95)

Intervals :
Level      Normal              Basic
95%   ( 3.039,  4.512 )   ( 3.034,  4.500 )

Level     Percentile            BCa
95%   ( 3.100,  4.566 )   ( 3.033,  4.500 )
Calculations and Intervals on Original Scale
```

Talk Outline

Nonnomral Variables

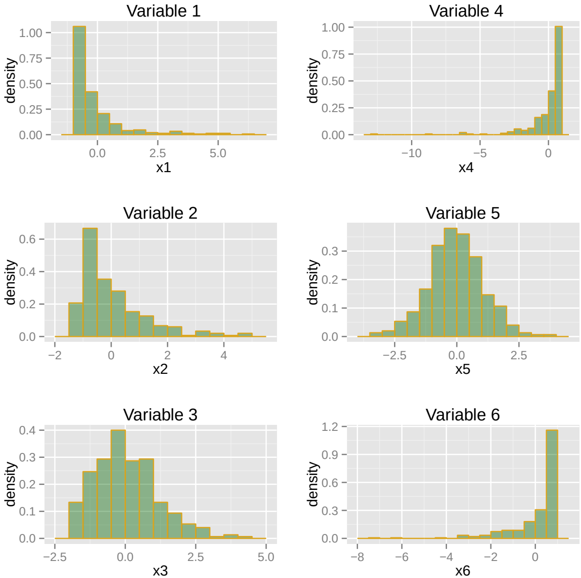
- Scaled χ^2
- Bootstrapping
- Example of CFA with Nonnormal Data

- Have `lavaan` simulate data with skewness and kurtosis "issues".

```
# specify population model
population.model <- '
f1 =~ .65*x1 + 0.8*x2 + .7*x3
f2 =~ .87*x4 + 0.5*x5 + .9*x6
f1~~.5*f2
f1~~1*f1
f2~~1*f2
'
set.seed(34566)
sample.data <- simulateData(population.model, sample.nobs=300L, skewness=c
    (3,2.1,1,-2.5,0,-3))
```

Nonnomral Variables

Example of CFA with Nonnormal Data



Nonnomral Variables

Example of CFA with Nonnormal Data

- "Regular" Estimation

```
> # fit model
> sample.model <- '
+ f1 =~ x1 + x2 + x3
+ f2 =~ x4 + x5 + x6
+ '
> fit <- cfa(sample.model, data=sample.data)
> summary(fit, standardized=TRUE)
lavaan (0.5-9) converged normally after  47 iterations

  Number of observations                           300

  Estimator                                         ML
  Minimum Function Chi-square                   12.552
  Degrees of freedom                                 8
  P-value                                        0.128

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  f1 =~
```

Nonnomral Variables (cont.)

Example of CFA with Nonnormal Data

```
    x1                 1.000                               0.455    0.425
    x2                 1.421    0.347    4.095    0.000    0.647    0.545
    x3                 1.613    0.408    3.948    0.000    0.735    0.603
  f2 =~
    x4                 1.000                               1.282    0.784
    x5                 0.373    0.069    5.409    0.000    0.478    0.408
    x6                 0.719    0.117    6.121    0.000    0.922    0.726

Covariances:
  f1 ~~
    f2                 0.146    0.060    2.429    0.015    0.250    0.250

Variances:
    x1                 0.943    0.095                      0.943    0.820
    x2                 0.992    0.134                      0.992    0.703
    x3                 0.943    0.156                      0.943    0.636
    x4                 1.028    0.265                      1.028    0.385
    x5                 1.148    0.101                      1.148    0.834
    x6                 0.763    0.144                      0.763    0.473
    f1                 0.207    0.077                      1.000    1.000
    f2                 1.643    0.322                      1.000    1.000
```

Nonnomral Variables

Example of CFA with Nonnormal Data

- Bootstrapped standard error and confidence intervals

```
> #Bootstrapped
> boot.fit<-sem(model=sample.model, data=sample.data, se="boot", bootstrap=1000)
> parameterEstimates(boot.fit, ci=TRUE, boot.ci.type="perc")
   lhs op rhs   est    se      z pvalue ci.lower ci.upper
1   f1 =~  x1 1.000 0.000     NA     NA    1.000    1.000
2   f1 =~  x2 1.421 0.459  3.097  0.002    0.841    2.643
3   f1 =~  x3 1.613 1.092  1.477  0.140    0.856    4.932
4   f2 =~  x4 1.000 0.000     NA     NA    1.000    1.000
5   f2 =~  x5 0.373 0.090  4.138  0.000    0.190    0.550
6   f2 =~  x6 0.719 0.193  3.720  0.000    0.390    1.131
7   x1 ~~  x1 0.943 0.257  3.678  0.000    0.521    1.517
8   x2 ~~  x2 0.992 0.227  4.377  0.000    0.519    1.439
9   x3 ~~  x3 0.943 0.246  3.832  0.000    0.313    1.264
10  x4 ~~  x4 1.028 0.422  2.438  0.015   -0.030    1.621
11  x5 ~~  x5 1.148 0.114 10.044  0.000    0.930    1.364
12  x6 ~~  x6 0.763 0.200  3.817  0.000    0.310    1.114
13  f1 ~~  f1 0.207 0.120  1.724  0.085    0.038    0.496
14  f2 ~~  f2 1.643 0.550  2.986  0.003    0.793    2.841
15  f1 ~~  f2 0.146 0.061  2.395  0.017    0.015    0.262
> parameterEstimates(boot.fit, ci=TRUE, boot.ci.type="bca.simple")
   lhs op rhs   est    se      z pvalue ci.lower ci.upper
1   f1 =~  x1 1.000 0.000     NA     NA    1.000    1.000
2   f1 =~  x2 1.421 0.459  3.097  0.002    0.826    2.615
3   f1 =~  x3 1.613 1.092  1.477  0.140    0.827    4.409
```

Nonnomral Variables (cont.)

Example of CFA with Nonnormal Data

```
4    f2 =~  x4 1.000 0.000     NA     NA    1.000    1.000
5    f2 =~  x5 0.373 0.090  4.138  0.000    0.195    0.559
6    f2 =~  x6 0.719 0.193  3.720  0.000    0.394    1.138
7    x1 ~~  x1 0.943 0.257  3.678  0.000    0.552    1.615
8    x2 ~~  x2 0.992 0.227  4.377  0.000    0.544    1.447
9    x3 ~~  x3 0.943 0.246  3.832  0.000    0.423    1.305
10   x4 ~~  x4 1.028 0.422  2.438  0.015    0.228    1.709
11   x5 ~~  x5 1.148 0.114 10.044  0.000    0.933    1.377
12   x6 ~~  x6 0.763 0.200  3.817  0.000    0.351    1.124
13   f1 ~~  f1 0.207 0.120  1.724  0.085    0.046    0.523
14   f2 ~~  f2 1.643 0.550  2.986  0.003    0.815    2.929
15   f1 ~~  f2 0.146 0.061  2.395  0.017    0.052    0.317
```

Nonnomral Variables

Example of CFA with Nonnormal Data

- Robust estimation, Satorra-Bentler/MLM

```
> #Robust Estimation
> robust.fit<-sem(model=sample.model, data=sample.data, estimator="MLM")
> summary(robust.fit, standardized=TRUE)
lavaan (0.5-9) converged normally after  47 iterations

  Number of observations                           300

  Estimator                                         ML       Robust
  Minimum Function Chi-square                   12.552       11.759
  Degrees of freedom                                 8            8
  P-value                                        0.128        0.162
  Scaling correction factor                                   1.068
    for the Satorra-Bentler correction

Parameter estimates:

  Information                                 Expected
  Standard Errors                           Robust.sem

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  f1 =~
    x1                1.000                               0.455    0.425
    x2                1.421    0.401    3.547    0.000    0.647    0.545
```

Nonnomral Variables (cont.)

Example of CFA with Nonnormal Data

```
    x3                 1.613    0.500    3.229    0.001    0.735    0.603
  f2 =~
    x4                 1.000                               1.282    0.784
    x5                 0.373    0.080    4.669    0.000    0.478    0.408
    x6                 0.719    0.162    4.435    0.000    0.922    0.726

Covariances:
  f1 ~~
    f2                 0.146    0.059    2.459    0.014    0.250    0.250

Intercepts:
    x1                -0.088    0.062   -1.425    0.154   -0.088   -0.082
    x2                -0.062    0.069   -0.897    0.370   -0.062   -0.052
    x3                -0.009    0.070   -0.124    0.901   -0.009   -0.007
    x4                -0.176    0.095   -1.867    0.062   -0.176   -0.108
    x5                -0.102    0.068   -1.504    0.133   -0.102   -0.087
    x6                -0.042    0.073   -0.566    0.571   -0.042   -0.033
    f1                 0.000                               0.000    0.000
    f2                 0.000                               0.000    0.000

Variances:
    x1                 0.943    0.253                      0.943    0.820
    x2                 0.992    0.204                      0.992    0.703
    x3                 0.943    0.173                      0.943    0.636
    x4                 1.028    0.323                      1.028    0.385
    x5                 1.148    0.110                      1.148    0.834
    x6                 0.763    0.193                      0.763    0.473
    f1                 0.207    0.104                      1.000    1.000
    f2                 1.643    0.510                      1.000    1.000
```

Nonnomral Variables

Example of CFA with Nonnormal Data

- Robust estimation, Yuan-Bentler/MLR

```
> #Robust Estimation
> robust.fit<-sem(model=sample.model, data=sample.data, estimator="MLR")
> summary(robust.fit, standardized=TRUE)
lavaan (0.5-9) converged normally after  47 iterations

  Number of observations                           300

  Estimator                                         ML      Robust
  Minimum Function Chi-square                   12.552      13.328
  Degrees of freedom                                 8           8
  P-value                                        0.128       0.101
  Scaling correction factor                                  0.942
    for the Yuan-Bentler correction

Parameter estimates:

  Information                                 Observed
  Standard Errors                         Robust.huber.white

                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  f1 =~
    x1                1.000                               0.455    0.425
    x2                1.421    0.378    3.761    0.000    0.647    0.545
```

Nonnomral Variables (cont.)

Example of CFA with Nonnormal Data

```
    x3                 1.613    0.643    2.509    0.012    0.735    0.603
  f2 =~
    x4                 1.000                               1.282    0.784
    x5                 0.373    0.079    4.713    0.000    0.478    0.408
    x6                 0.719    0.156    4.612    0.000    0.922    0.726

Covariances:
  f1 ~~
    f2                 0.146    0.058    2.503    0.012    0.250    0.250

Intercepts:
    x1                -0.088    0.062   -1.427    0.154   -0.088   -0.082
    x2                -0.062    0.069   -0.898    0.369   -0.062   -0.052
    x3                -0.009    0.070   -0.124    0.901   -0.009   -0.007
    x4                -0.176    0.094   -1.870    0.061   -0.176   -0.108
    x5                -0.102    0.068   -1.507    0.132   -0.102   -0.087
    x6                -0.042    0.073   -0.567    0.570   -0.042   -0.033
    f1                 0.000                               0.000    0.000
    f2                 0.000                               0.000    0.000

Variances:
    x1                 0.943    0.255                      0.943    0.820
    x2                 0.992    0.225                      0.992    0.703
    x3                 0.943    0.221                      0.943    0.636
    x4                 1.028    0.317                      1.028    0.385
    x5                 1.148    0.110                      1.148    0.834
    x6                 0.763    0.187                      0.763    0.473
    f1                 0.207    0.117                      1.000    1.000
    f2                 1.643    0.499                      1.000    1.000
```

Talk Outline

Power Review

Example

Power Review

- Null Hypothesis Significance Testing
  - Neyman-Pearson method of testing competing hypotheses
- Null hypothesis (H_0)
  - The (antithesis) of the hypothesis we are interested in analyzing
- Alternative hypothesis (H_a, H_1)
  - A contrary hypothesis to H_0
  - Usually that a parameter is $\neq$ to some specific value
- Sampling distribution
  - Probability distribution of a statistic

Power Review

- Type 1 error (α)
  - *p*(rejecting H_0 based on data [statistic]— H_0 is true)
- What happens when α is large?
  - Frequently reject H_0, when it is true
  - Say there is an effect, when there isnâĂŹt one
  - Large *false* $+$ rate

Power Review

- Type 2 error (β)
  - *p*(accepting H_0 based on data [statistic]— H_0 is false)
- Power $= 1 - \beta$
- What happens when β is large ?
  - Frequently accept H_0, when it is false
  - Say there is not an effect, when there is one
  - Large *false* - rate

Power Review

- Power, α (probability of a type 1 error), n, and effect size are all related to each other.
  - If you know three of the values, the fourth is known if you know how to extract that value.

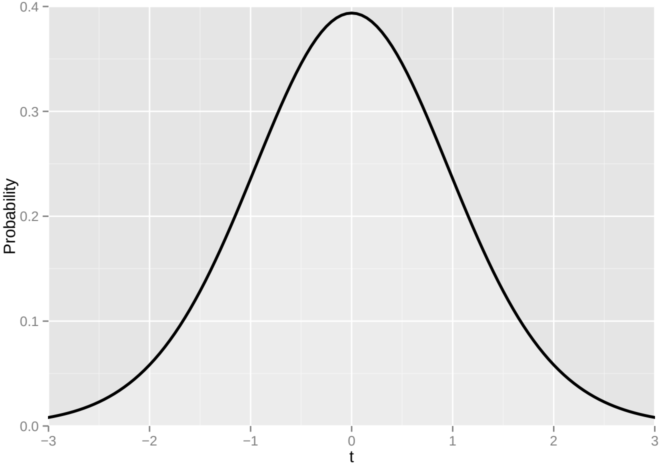
Talk Outline

Power Review

Example

- ACT variable
  - $n = 20$
  - mean=26
  - SD=4.1
- Hypothesize that this year's entering class' average ACT scores are "significantly" larger than last year's entering class's average ACT scores, which was 24.
- H_0:
  - This years entering class' average ACT score is ≤ 24
- H_a:
  - This years entering class' average ACT score is > 24

- If H_0 is true, then $\bar{X} \sim T_{df=19}$



Power Review

Example

- How many times do we want to reject H_0, when it is true?
  - That is, say that the mean ACT score is higher, when it is not.
  - 10% ?
  - $\alpha = .10$

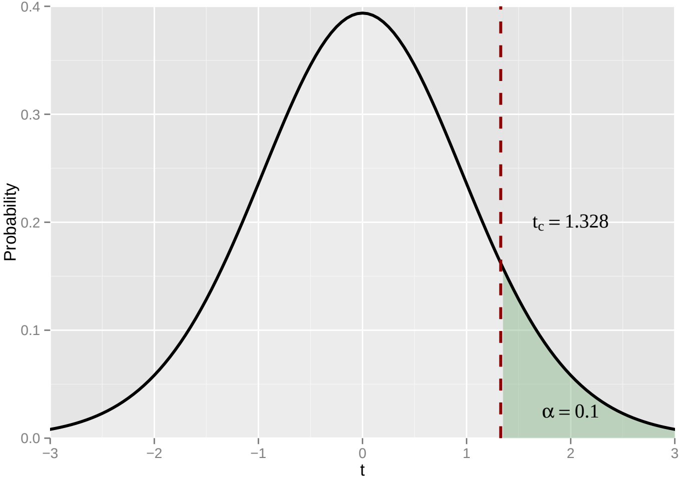
Power Review

Example

- How many times do we want to reject H_0, when it is true?
  - That is, say that the mean ACT score is higher, when it is not.
  - 10% ?
  - $\alpha = .10$
- Same condition as null hypothesis

- If H_0 is true, then $\bar{X} \sim T_{19df}$
- If H_0 is true, will reject it (wrongly) 10% of the time
- $\alpha = .10$

- If H_0 is true, then $\bar{X} \sim T_{df=19}$



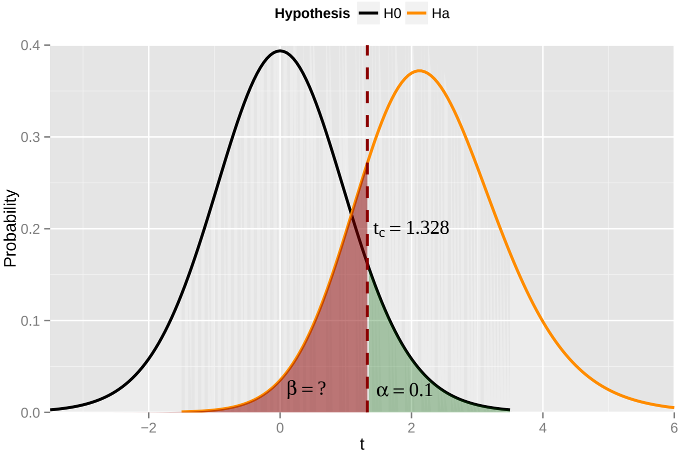
- For an effect size, we can use Cohen's (1988) *d*

$$d = \frac{\mu - \mu_0}{\sigma} = \frac{26 - 24}{4.1} = 0.49$$

- *d* can be transformed into a noncentrality parameter, Δ, for a *t*-distribution.
  - Δ can be conceptualized as an index of the magnitude of difference between H_0 and H_a.
- For the single sample scenario:

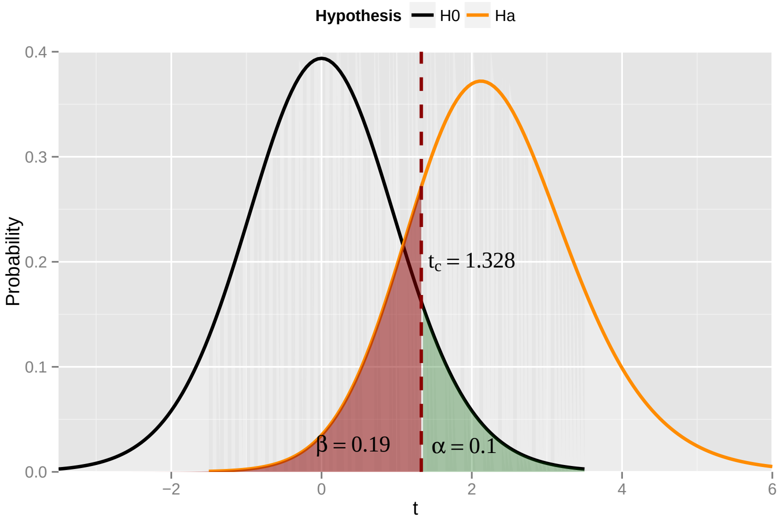
$$\Delta = d\sqrt{n} = 0.49\sqrt{20} \approx 2.20$$

- If H_a is true, then $\bar{X} \sim T_{df=19,\Delta=2.20}$



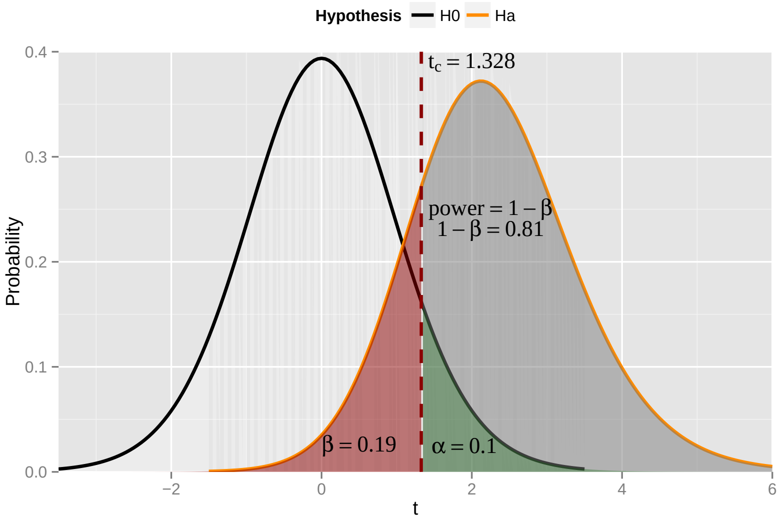
- How do we get β?
  - $\beta = p(\text{accepting } H_0 | H_0 \text{ is false})$
  - $\beta = p(\text{accepting } H_0 | H_a \text{ is true})$
  - $p(t < 1.32 | H_a \text{ is true})$

```
> # Beta-- p(type II error)
> pt(1.32,19,ncp=2.2,lower.tail = TRUE)
[1] 0.19006
```


Hypothesis
H0
Ha
0.4
0.3
0.2
0.1
0.0
Probability
−2
0
2
4
6
t
tc = 1.328
β = 0.19
α = 0.1

Power Review

Example



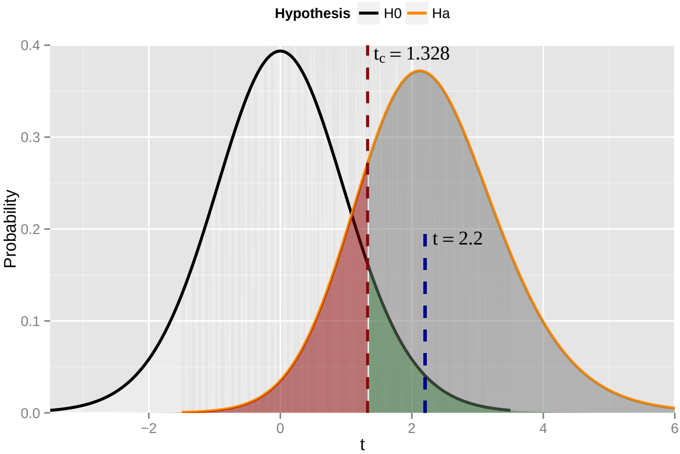
Power Review

Example

- To test and plot the current mean (i.e., ACT $= 26$) on the distribution graph, it need to be converted to the t-metric

$$t = \frac{\bar{X} - \mu}{\sigma_{\bar{X}}} = \frac{26 - 24}{4.1/\sqrt{20}} = 2.2$$

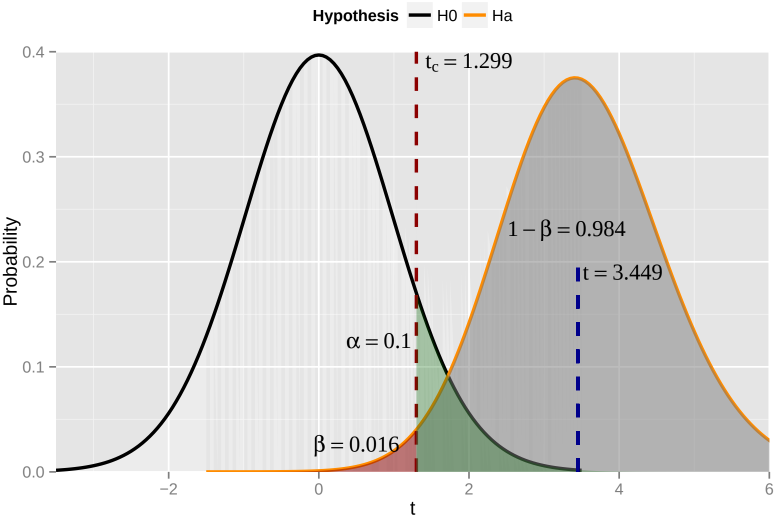
- Because $t > t_c$, reject H_0



- What would happen if $n = 50$?
  - Under H_0, $\bar{X} \sim T_{49\,df}$
  - $t = \frac{26-24}{4.1/\sqrt{50}} = 3.5$

Power Review

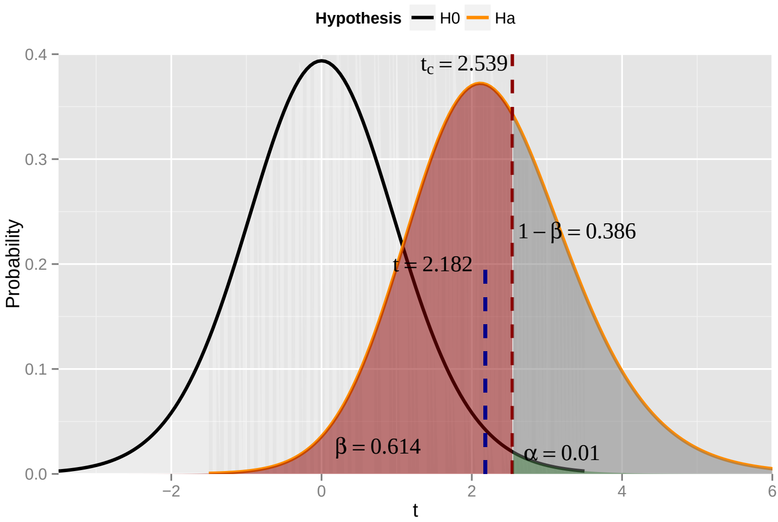
Example



- What would happen if $n = 50$?
  - Under H_0, $\bar{X} \sim T_{49\,df}$
  - $t = \frac{26-24}{4.1/\sqrt{50}} = 3.5$
- Holding everything else constant
  - as $n \uparrow$, power $\uparrow$

Power Review
Example

- What would happen if $\alpha = .01$
  - $t_c \neq 1.32$
  - $t_c = 2.5$



Hypothesis
H0
Ha
0.4
0.3
0.2
0.1
0.0
Probability
$t_c = 2.539$
$1 - \beta = 0.386$
$t = 2.182$
$\beta = 0.614$
$\alpha = 0.01$
−2
0
2
4
6
t

Power Review

Example

- What would happen if $\alpha = .01$
  - $t_c = 2.5$
- Holding everything else constant
  - as $\alpha \downarrow$, power $\downarrow$

Talk Outline

Power Through a Monte Carlo Study

- Example (Continued)

Monte Carlo Power

- An alternative to the traditional power analysis is a Monte Carlo (MC) study Muthen and Muthen (2002).
- In MC studies,
  - data are simulated from a population with hypothesized parameter values.
  - a large number of samples from the population are drawn.
  - a model is estimated for each sample.
  - parameter values and standard errors are averaged over the samples.
  - the following criteria are examined: parameter estimate bias, standard error bias, and coverage.

Monte Carlo Power

Parameter Estimate Bias

$$\theta_{\text{bias}} = \frac{\hat{\theta} - \theta}{\theta}$$

where
θ is the hypothesized parameter value, and
$\hat{\theta}$ is the average parameter value from the *m* simulations.

Standard Error Bias

$$\sigma_{\text{bias}} = \frac{\hat{\sigma}_\theta - \sigma_\theta}{\sigma_\theta}$$

where
σ_θ is the SD of the parameter estimate over the *m* replications, and
$\hat{\sigma}_\theta$ is the average of the estimated standard errors for the parameter estimate over the *m* replications.

Monte Carlo Power

- *Coverage* is the percent of the m replications that the (1-α)% confidence interval contains θ.
- *Power* is proportion of the m replications for which the null hypothesis is rejected for the parameter at the α level.

Monte Carlo Power

- Muthen and Muthen (2002) suggest the following criteria to determine sample size:
  1. Parameter and standard error biases do not exceed 10% for *any* parameter in the model.
  2. Standard error bias for the parameter for which power is being assessed does not exceed 5%.
  3. coverage remains between 0.91 and 0.98.
- Once these three conditions are satisfied, they suggest selecting a sample size to keep power close to 0.80.

Talk Outline

Power Through a Monte Carlo Study

Example (Continued)

Monte Carlo Power
Example (Continued)

- H_0:
  - This years entering class' average ACT score is ≤ 24
- Let's make it a more stringent hypothesis
  - H_0: This years entering class' average ACT score is $\neq 24$
  - This means that the test is 2-tailed (i.e., $\alpha = .05/2$ in a given tail of the sampling distribution).

Monte Carlo Power

Example (Continued)

- Let walk through the MC process with $m = 2$
  1. Simulate the data

```
> n <- 20 #sample size
> mu<- 26 #hypothesized mean
> sigma <- 4.1 #hypothesized SD
> set.seed(34567)
> x1<-rnorm(n,mu,sigma)
> set.seed(56981)
> x2<-rnorm(n,mu,sigma)
> x1
 [1] 27.77249 17.76224 25.80838 21.68914 27.34067 23.21717 23.16883 29.30207
 [9] 23.77690 20.16772 20.49027 15.25286 31.00157 23.61168 22.08402 28.01689
[17] 24.88833 30.23753 27.67120 29.43361
> x2
 [1] 25.00732 30.55491 23.12045 22.13123 23.58853 21.90874 24.70908 24.78498
 [9] 18.32644 29.24183 31.61460 21.92098 18.03927 23.48655 27.13335 32.16530
[17] 22.80072 26.12504 27.20693 32.44824
```

- Let walk through the MC process with $m = 2$
  2. Calculate the average and SD of the average values.

```
> mean(c(mean(x1), mean(x2))) #mean of the means
[1] 24.9752
> sd(c(mean(x1), mean(x2))) #SD of the means
[1] 0.4815713
```

  3. Calculate the average standard error

```
> stderr <- function(x) sqrt(var(x)/length(x)) #function for standard error of mean
> mean(c(stderr(x1), stderr(x2))) #mean of the standard errors
[1] 0.9532781
```

Monte Carlo Power

Example (Continued)

- Let walk through the MC process with $m = 2$
  4. Calculate the coverage

```
> alpha<-.1 #type 1 error rate
> #Simulation 1
> mean1<-mean(x1)
> se1<-stderr(x1) #standard error
> error1 <- qt(1-alpha/2,df=n-1)*se1  #CI length
> left1 <- mean1 - error1 #left side of CI
> right1 <- mean1 + error1 #right side of CI
> cov1<-ifelse(mu <= right1 && mu >= left1, 1, 0)  #coverage
> cov1
[1] 1
> #Simulation 2
> mean2<-mean(x2)
> se2<-stderr(x2) #standard error
> error2 <- qt(1-alpha/2,df=n-1)*se2  #CI length
> left2 <- mean2 - error2 #left side of CI
> right2 <- mean2 + error2 #right side of CI
> cov2<-ifelse(mu <= right2 && mu >= left2, 1, 0)  #coverage
> cov2
[1] 1
>
> mean(cov1,cov2) #average (1-alpha)% coverage
[1] 1
```

Monte Carlo Power

Example (Continued)

- Let walk through the MC process with $m = 2$
  5. Calculate the power

```
> sig1<-ifelse(t.test(x1, mu=24, alternative = "two.sided")$p.value < alpha/2, 1, 0)
> sig2<-ifelse(t.test(x2, mu=24, alternative = "two.sided")$p.value < alpha/2, 1, 0)
> mean(sig1, sig2) #power
[1] 0
```

Monte Carlo Power

Example (Continued)

- Instead of doing it piecemeal, we want to do everything "at once"
- Write a function that will calculate everything, and output the information of interest

```
sim.oneSampleMean<-function(mu=NULL, sigma=NULL, n=NULL, alpha=.05, m=100){
nparm<-1 #Number of parameters estimating
simulations<-matrix(NA, m, 5) #Container for simulated data's statistics

for(i in 1:m){
x.data <-rnorm(n, mu, sigma)
x.mean<-mean(x.data)
x.var<-var(x.data)
x.se<-stderr(x.data)
error <- qt(1-alpha/2,df=n-1)*x.se
left <- x.mean - error
right <- x.mean + error
simulations[i,1]<-i #simulation number
simulations[i,2]<-x.mean # theta_hat
simulations[i,3]<-x.se # Standard error
simulations[i,4]<-ifelse(mu <= right && mu >= left, 1, 0) #Coverage
simulations[i,5]<-ifelse(t.test(x.data, mu=24, alternative = "two.sided")$p.value <=
     alpha/2, 1 , 0) #power
}

```

Monte Carlo Power (cont.)

Example (Continued)

```
results<-matrix(NA, nparm, 8)
colnames(results)<-c("Starting", "Average", "SD", "SE.Average", "Coverage", "Power", "PE.
      bias", "SE.bias")
results[1,1]<-mu
results[1,2]<-mean(simulations[,2])
results[1,3]<-sd(simulations[,2])
results[1,4]<-mean(simulations[,3])
results[1,5]<-mean(simulations[,4])
results[1,6]<-mean(simulations[,5])
results[1,7]<-(mu - mean(simulations[,2]))/mu
results[1,8]<-(mean(simulations[,3]) - sd(simulations[,2]))/sd(simulations[,2])
results<-round(results, 3)

results
}
```

Monte Carlo Power

Example (Continued)

```
> #Change number of replications
> #m=10
> sim.oneSampleMean(mu=26, sigma=4.1, n=20, alpha=.05, m=10)
     Starting  Average   SD SE.Average Coverage Power PE.bias SE.bias
[1,]       26  25.839 0.88      0.933        1   0.3   0.006   0.061
> #m=100
> sim.oneSampleMean(mu=26, sigma=4.1, n=20, alpha=.05, m=100)
     Starting  Average    SD SE.Average Coverage Power PE.bias SE.bias
[1,]       26  25.769 0.911      0.922     0.97  0.33   0.009   0.012
> #m=1000
> sim.oneSampleMean(mu=26, sigma=4.1, n=20, alpha=.05, m=1000)
     Starting  Average    SD SE.Average Coverage Power PE.bias SE.bias
[1,]       26  26.002 0.916      0.895    0.952 0.408       0  -0.023
> #m=10000
> sim.oneSampleMean(mu=26, sigma=4.1, n=20, alpha=.05, m=10000)
     Starting  Average    SD SE.Average Coverage Power PE.bias SE.bias
[1,]       26  26.023 0.926      0.904    0.949 0.431  -0.001  -0.023
```

Monte Carlo Power

Example (Continued)

```
> #Change sample sizes
> #n=10
> sim.oneSampleMean(mu=26, sigma=4.1, n=10, alpha=.05, m=10000)
     Starting Average    SD SE.Average Coverage Power PE.bias SE.bias
[1,]       26  26.018 1.302      1.266    0.953 0.185  -0.001  -0.027
> #n=20
> sim.oneSampleMean(mu=26, sigma=4.1, n=20, alpha=.05, m=10000)
     Starting Average    SD SE.Average Coverage Power PE.bias SE.bias
[1,]       26  25.994 0.909      0.906    0.952 0.416       0  -0.003
> #n=30
> sim.oneSampleMean(mu=26, sigma=4.1, n=30, alpha=.05, m=10000)
     Starting Average    SD SE.Average Coverage Power PE.bias SE.bias
[1,]       26  26.003 0.749      0.743    0.949 0.618       0  -0.008
> #n=50
> sim.oneSampleMean(mu=26, sigma=4.1, n=50, alpha=.05, m=10000)
     Starting Average    SD SE.Average Coverage Power PE.bias SE.bias
[1,]       26  25.989 0.578      0.576     0.95 0.865       0  -0.004
> #n=75
> sim.oneSampleMean(mu=26, sigma=4.1, n=75, alpha=.05, m=10000)
     Starting Average    SD SE.Average Coverage Power PE.bias SE.bias
[1,]       26   25.99 0.476      0.472    0.945 0.968       0   -0.01
```

Talk Outline

Power in Structural Equation Modeling

- Power to Detect an Added Path
- Overall Power to Reject a Model
- Monte Carlo

Talk Outline

Power in Structural Equation Modeling

- Power to Detect an Added Path
- Overall Power to Reject a Model
- Monte Carlo

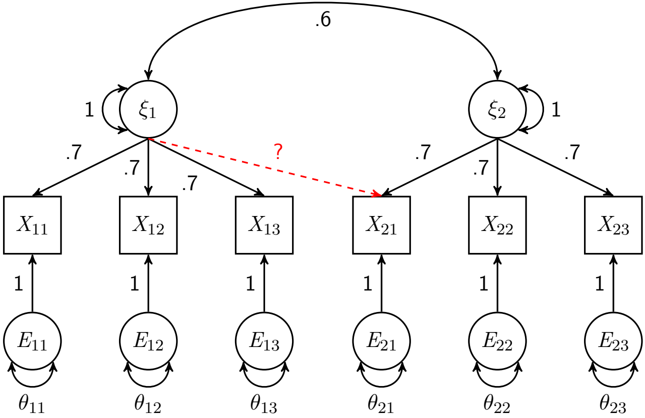
- Following Loehlin (2004) and Satorra and Saris (1985), the power to detect an added path to a model is a 3-step procedure.
- In this situation, the effect size is the NCP of χ^2, Δ.
  - That is the resulting χ^2 given by fitting two CFA models (with and without the parameter of interest).

- 3-step procedure
  1. Obtain fitted covariance matrix under H_a, $\mathbf{\Sigma}_{H_a}$, that the added path coefficient > 0.
  2. Using $\mathbf{\Sigma}_{H_a}$, fit the original model, i.e., without the added path, and obtain the χ^2, which is an approximation of Δ
  3. Obtain the probability of getting a value as or more extreme than α under a χ^2 distribution with NCP $= \Delta$

Talk Outline

Power in Structural Equation Modeling

- Power to Detect an Added Path
  - Example
- Overall Power to Reject a Model
- Monte Carlo



Path Model for Power Analysis, taken from Loehlin (2004, p. 71)

- To have lavaan give the implied covariance (correlation) matrix, use the do.fit=FALSE argument in the cfa() (or sem()) function, which tells lavaan to use the starting values as the parameter estimates.
- Then obtain the implied covariance matrix from this model using the fitted() function.
- The fitted() function returns both the fitted covariance matrix as well as the fitted means, so add the $cov suffix just returns the covariance matrix.

Power in SEM

Power to Detect an Added Path

```
> #power for adding single path
> Fig2.10.model<-'
+ G=~  .7*A + .7*B + .7*C + .3*D
+ H=~  .7*D + .7*E + .7*F
+ G~~.6*H
+
+ G~~1*G
+ H~~1*H
+ '
> fig2.10.fit<-cfa(Fig2.10.model, do.fit=FALSE)
> fig2.10.cov<-fitted(fig2.10.fit)$cov
> fig2.10.cov
  A     B     C     D     E     F
A 1.490
B 0.490 1.490
C 0.490 0.490 1.490
D 0.504 0.504 0.504 1.832
E 0.294 0.294 0.294 0.616 1.490
F 0.294 0.294 0.294 0.616 0.490 1.490
```

- Notice that the variance values in `fig2.10.cov` are not one.
- This is because we did not set the residual variance values in the model, so `lavaan` fitted them with the default of 1.
- For this data and mode, the amount of variance in, say X_{11}, is $.7 \times .7 = .49$ and the residual variance is 1, thus the implied "correlation" is 1.49.
- We can alter the residual variances in `Fig2.10.model` (they would be 1-R^2 for each residual variance), but this will become quite complex quickly for models where the R^2 is made of complex paths.
- Another alternative is to set the variances to 1 manually using the `diag()` function.
  - If you input a matrix into the `diag()` function, it will return the principal diagonal of the matrix.
  - Then, we just need to reassign those values to 1, which we do by repeating 1 six times using the `rep()` function.

```
> diag(fig2.10.cov)<-rep(1,6) #Puts ones on the diagona
> fig2.10.cov
  A     B     C     D     E     F
A 1.000
B 0.490 1.000
C 0.490 0.490 1.000
D 0.504 0.504 0.504 1.000
E 0.294 0.294 0.294 0.616 1.000
F 0.294 0.294 0.294 0.616 0.490 1.000
```

- Now use the implied covariance matrix (i.e., `fig2.10.cov`) as input for the "original" factor model with an $n = 500$

```
> Fig2.10.original.model<-'
+ G=~  A + B + C
+ H=~  D + E + F
+ '
>
> fig2.10.original.fit<-cfa(Fig2.10.original.model, sample.cov=fig2.10.cov, sample.nobs
    =500)
```

- The NCP (Δ) is just the χ^2 value of fitting this original model to the data generated from the model with the extra path.

```
> NCP<-fitMeasures(fig2.10.original.fit, fit.measure="chisq") #gives chi-square
> NCP
chisq
14.96
```

- Instead of consulting a table for power estimates, we can calculate power directly.
- We already have a measure of effect size (albeit an unusually scaled one) and have specified n, so all that is left is pick an α value.
- Since our ES measure has an unusual scale, though, we need to put α on a comparable scale (i.e, transform it to a critical value), which we can do by using the quantile χ^2 function in R, i.e, qchisq().

```
> #Transform alpha to chi-square metric
> cv<-qchisq(.95,df=1)#Gives the critical value
> cv
[1] 3.841459
```

- The .95 in the qchisq() is $1 - \alpha$, so if you want a more stringent or liberal α value, α', calculate $1 - \alpha'$ and replace the .95 with the newly calculated value.

- Now we have all the information we need (cv=3.84, df=1, and $\Delta = 14.96$) to calculate power for this single-path. Specifically, power in this case is the probability of getting a critical value (CV) of 3.84 given $CV \sim \chi^2_{df=1,\Delta=14.96}$

```
> pchisq(cv, df=1,ncp=NCP, lower.tail=FALSE) #The power to detect one path
[1] 0.9717973
```

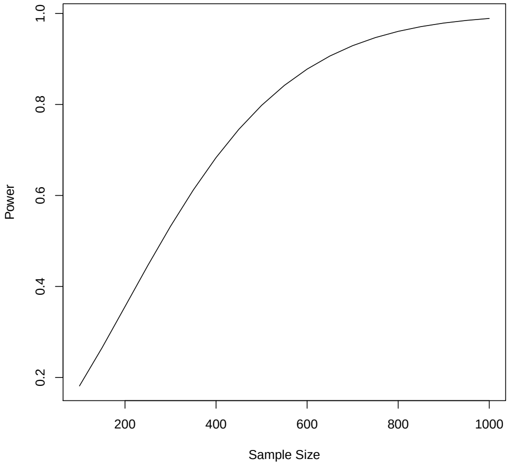
- We use the `lower.tail=FALSE` argument here, which is equivalent to specifying `1-pchisq(..., lower.tail=TRUE)`

- To get the power for the model for a generic extra path, we follow the same procedure, only now we with the $df = 8$.

```
> cv<-qchisq(.95,df=8)#Gives the critical value
> pchisq(cv, df=8, ncp=NCP, lower.tail=FALSE) #The power for overall model
[1] 0.798017
```

- Instead of getting a single sample size needed for a given power level, it is usually more useful to get a power curve, that is the power for a range of sample sizes.
- We make such a curve using a `for()` loop in `R`

```
#power curve
n.start<-100 #n value to start power curve
n.stop<-1000 #n value to end power curve
incriment<-50 #How fine tuned you want the curve, larger values are less fine tuned
df<-8 #degrees of freedom
alpha<-.05
sample.sizes<-seq(n.start,n.stop,incriment) #makes a vector of sample sizes, given the n.
    start, n.stop and increment
values<-matrix(NA, ncol=2, nrow=length(sample.sizes)) #make an empty matrix

#For loop to generate power at given n values
for (i in 1:length(sample.sizes)){
model.fit<-cfa(Fig2.10.original.model, sample.cov=fig2.10.cov, sample.nobs=sample.sizes[i
    ])
values[i,1]<-sample.sizes[i]
NCP<-fitMeasures(model.fit, fit.measure="chisq") #chi-square
cv<-qchisq(1-alpha,df=df)#critical value
values[i,2]<-pchisq(cv, df=df,ncp=NCP, lower.tail=FALSE) #The power for overall model
}

#make the power curve
plot(values[,1], values[,2], main="Power Curve", xlab="Sample Size", ylab="Power", type="
    l")
```



Power Curve

Talk Outline

Power in Structural Equation Modeling

- Power to Detect an Added Path
- Overall Power to Reject a Model
- Monte Carlo

- This method asks:
  - If the model fits the data well in the population (RMSEA $\leq .05$), then is the sample sufficient to be able to reject the hypothesis that the the model fits bad (RMSEA $\geq .10$)?

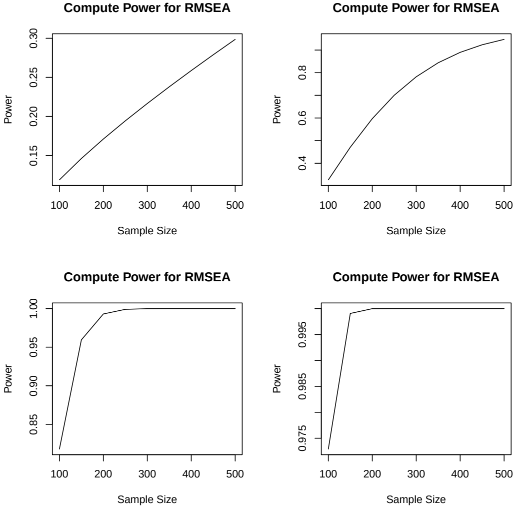
Talk Outline

Power in Structural Equation Modeling

- Power to Detect an Added Path
- Overall Power to Reject a Model
  - Example
- Monte Carlo

- The semTools packages has a function to plot power curves for RMSEA, as well as determine sample size.

```
> library(semTools)

> par(mfrow=c(2,2))
> plotRMSEApower(.05, .1, df=1, nlow=100, nhigh=500, steps=50, alpha=.05)
> plotRMSEApower(.05, .1, df=10, nlow=100, nhigh=500, steps=50, alpha=.05)
> plotRMSEApower(.05, .1, df=50, nlow=100, nhigh=500, steps=50, alpha=.05)
> plotRMSEApower(.05, .1, df=100, nlow=100, nhigh=500, steps=50, alpha=.05)
```



Compute Power for RMSEA
Power
0.15
0.20
0.25
0.30
100
200
300
400
500
Sample Size
Compute Power for RMSEA
Power
0.4
0.6
0.8
100
200
300
400
500
Sample Size
Compute Power for RMSEA
Power
0.85
0.90
0.95
1.00
100
200
300
400
500
Sample Size
Compute Power for RMSEA
Power
0.975
0.985
0.995
100
200
300
400
500
Sample Size

Power in SEM

Overall Power to Reject a Model: Example

```
> findRMSEAsamplesize(rmsea0=.05, rmseaA=.1, df=1, power=.80, alpha=.05)
[1] 2475
> findRMSEAsamplesize(rmsea0=.05, rmseaA=.1, df=8, power=.80, alpha=.05)
[1] 376
```

Talk Outline

Power in Structural Equation Modeling

- Power to Detect an Added Path
- Overall Power to Reject a Model
- Monte Carlo

- Using a Monte Carlo study for SEMs is the same as we specified for the *t*-test.
- The `simsem` (Pornprasertmanit, Miller, & Schoemann, 2012) package is set up to do this.

```
> Fig2.10.model<-'
+ G=~  .7*A + .7*B + .7*C + .3*D
+ H=~  .7*D + .7*E + .7*F
+ G~~.6*H
+
+ G~~1*G
+ H~~1*H
+ '
>
> Fig2.10.fit<-cfa(Fig2.10.model, do.fit=FALSE)
>
> Fig2.10.datamodel<-model.lavaan(Fig2.10.fit, std=TRUE) #Build the data generation
    template and analysis template
>
> Fig2.10.sim.n100<-sim(100, Fig2.10.datamodel,n=100, multicore=TRUE)
> Fig2.10.sim.n500<-sim(100, Fig2.10.datamodel,n=500, multicore=TRUE)
```

```
> summary(Fig2.10.sim.n100)
RESULT OBJECT
Model Type
[1] "CFA"
========= Fit Indices Cutoffs ============
          Alpha
Fit Indices      0.1     0.05     0.01    0.001     Mean     SD
      Chi     13.502   16.058   19.473   23.131    7.609  4.426
      AIC   1673.167 1682.506 1695.852 1710.482 1638.609 31.880
      BIC   1725.270 1734.610 1747.955 1762.586 1690.713 31.880
      RMSEA    0.096    0.114    0.133    0.152    0.035  0.041
      CFI      0.926    0.888    0.852    0.814    0.976  0.038
      TLI      0.842    0.761    0.682    0.601    0.989  0.116
      SRMR     0.052    0.057    0.064    0.068    0.037  0.012
========= Parameter Estimates and Standard Errors ============
       Labels Estimate.Average Estimate.SD Average.SE Power..Not.equal.0.
1.G=~A                    0.553       0.145      0.116               0.958
1.G=~B                    0.566       0.128      0.117               1.000
1.G=~C                    0.566       0.131      0.116               1.000
1.G=~D                    0.234       0.345      0.412               0.326
1.H=~D                    0.487       0.327      0.417               0.463
1.H=~E                    0.584       0.132      0.130               1.000
1.H=~F                    0.575       0.137      0.129               1.000
1.A~~A (smc1)              0.656       0.158      0.125               1.000
1.B~~B (smc2)              0.668       0.127      0.128               1.000
1.C~~C (smc3)              0.642       0.116      0.125               0.989
1.D~~D (smc4)              0.513       0.137      0.183               0.874
1.E~~E (smc5)              0.651       0.152      0.146               0.968
```

```
1.F~~F (smc6)          0.636      0.128     0.143            0.958
1.H~~G                 0.627      0.172     0.171            0.947
1.A~1                  0.006      0.126     0.099            0.158
1.B~1                 -0.007      0.104     0.100            0.053
1.C~1                  0.004      0.117     0.099            0.095
1.D~1                 -0.009      0.100     0.099            0.032
1.E~1                  0.005      0.090     0.100            0.032
1.F~1                 -0.009      0.104     0.099            0.095
       Std.Est Std.Est.SD Average.Param Average.Bias Coverage
1.G=~A   0.558      0.137         0.573       -0.020    0.895
1.G=~B   0.564      0.108         0.573       -0.007    0.926
1.G=~C   0.570      0.110         0.573       -0.007    0.947
1.G=~D   0.229      0.332         0.222        0.013    0.916
1.H=~D   0.496      0.324         0.517       -0.030    0.947
1.H=~E   0.581      0.118         0.573        0.010    0.958
1.H=~F   0.577      0.114         0.573        0.001    0.968
1.A~~A    0.670      0.143         0.671       -0.015    0.863
1.B~~B    0.671      0.122         0.671       -0.003    0.916
1.C~~C    0.663      0.125         0.671       -0.029    0.937
1.D~~D    0.522      0.135         0.546       -0.033    0.989
1.E~~E    0.648      0.136         0.671       -0.020    0.937
1.F~~F    0.654      0.132         0.671       -0.036    0.947
1.H~~G    0.627      0.172         0.600        0.027    0.947
1.A~1    0.005      0.129         0.000        0.006    0.842
1.B~1   -0.007      0.104         0.000       -0.007    0.947
1.C~1    0.003      0.118         0.000        0.004    0.905
1.D~1   -0.009      0.103         0.000       -0.009    0.968
1.E~1    0.005      0.090         0.000        0.005    0.968
1.F~1   -0.010      0.104         0.000       -0.009    0.905
```

```
========= Correlation between Fit Indices ============
         Chi    AIC    BIC  RMSEA    CFI    TLI   SRMR
Chi    1.000  0.011  0.011  0.958 -0.890 -0.958  0.910
AIC    0.011  1.000  1.000  0.021  0.117  0.081 -0.036
BIC    0.011  1.000  1.000  0.021  0.117  0.081 -0.036
RMSEA  0.958  0.021  0.021  1.000 -0.901 -0.924  0.848
CFI   -0.890  0.117  0.117 -0.901  1.000  0.919 -0.806
TLI   -0.958  0.081  0.081 -0.924  0.919  1.000 -0.903
SRMR   0.910 -0.036 -0.036  0.848 -0.806 -0.903  1.000
================== Replications =====================
Number of replications = 100
Number of converged replications = 95
Number of nonconverged replications:
   1. Nonconvergent Results = 1
   2. Nonconvergent results from multiple imputation = 0
   3. At least one SE were negative or NA = 0
   4. At least one variance estimates were negative = 4
   5. At least one correlation estimates were greater than 1 or less than -1 = 0
```

Power in SEM

Monte Carlo

```
> summary(Fig2.10.sim.n500)
RESULT OBJECT
Model Type
[1] "CFA"
========= Fit Indices Cutoffs ============
          Alpha
Fit Indices      0.1     0.05     0.01    0.001     Mean     SD
      Chi     12.582   14.294   18.413   18.482    7.306  3.749
      AIC   8192.703 8216.559 8254.730 8265.257 8098.079 71.945
      BIC   8276.995 8300.851 8339.022 8349.549 8182.371 71.945
      RMSEA    0.040    0.046    0.057    0.057    0.014  0.017
      CFI      0.988    0.983    0.976    0.974    0.996  0.006
      TLI      0.975    0.964    0.948    0.945    0.999  0.019
      SRMR     0.023    0.024    0.028    0.029    0.016  0.004
========= Parameter Estimates and Standard Errors ============
       Labels Estimate.Average Estimate.SD Average.SE Power..Not.equal.0.
1.G=~A                    0.578       0.055      0.053                1.00
1.G=~B                    0.582       0.050      0.053                1.00
1.G=~C                    0.566       0.059      0.053                1.00
1.G=~D                    0.211       0.104      0.095                0.65
1.H=~D                    0.526       0.104      0.097                1.00
1.H=~E                    0.580       0.059      0.056                1.00
1.H=~F                    0.574       0.055      0.056                1.00
1.A~~A (smc1)              0.657       0.059      0.057                1.00
1.B~~B (smc2)              0.661       0.058      0.058                1.00
1.C~~C (smc3)              0.666       0.065      0.057                1.00
1.D~~D (smc4)              0.535       0.059      0.059                1.00
1.E~~E (smc5)              0.662       0.057      0.062                1.00
```

Power in SEM (cont.)

Monte Carlo

```
1.F~~F (smc6)            0.666       0.062      0.061                  1.00
1.H~~G                   0.595       0.076      0.073                  1.00
1.A~1                    0.002       0.052      0.045                  0.08
1.B~1                   -0.001       0.042      0.045                  0.04
1.C~1                   -0.002       0.047      0.044                  0.04
1.D~1                    0.002       0.045      0.045                  0.03
1.E~1                    0.004       0.045      0.045                  0.07
1.F~1                   -0.003       0.050      0.045                  0.07
       Std.Est Std.Est.SD Average.Param Average.Bias Coverage
1.G=~A   0.580      0.049         0.573        0.005     0.93
1.G=~B   0.582      0.043         0.573        0.009     0.95
1.G=~C   0.568      0.052         0.573       -0.008     0.93
1.G=~D   0.211      0.104         0.222       -0.010     0.93
1.H=~D   0.527      0.104         0.517        0.008     0.95
1.H=~E   0.579      0.049         0.573        0.007     0.95
1.H=~F   0.574      0.048         0.573        0.000     0.97
1.A~~A    0.661      0.057         0.671       -0.014     0.94
1.B~~B    0.660      0.049         0.671       -0.011     0.93
1.C~~C    0.674      0.060         0.671       -0.005     0.91
1.D~~D    0.537      0.056         0.546       -0.011     0.96
1.E~~E    0.662      0.057         0.671       -0.010     0.96
1.F~~F    0.668      0.055         0.671       -0.005     0.94
1.H~~G    0.595      0.076         0.600       -0.005     0.96
1.A~1    0.002      0.053         0.000        0.002     0.92
1.B~1   -0.001      0.042         0.000       -0.001     0.96
1.C~1   -0.002      0.047         0.000       -0.002     0.96
1.D~1    0.002      0.045         0.000        0.002     0.97
1.E~1    0.004      0.045         0.000        0.004     0.93
1.F~1   -0.003      0.050         0.000       -0.003     0.93
```

```
========= Correlation between Fit Indices ============
          Chi    AIC    BIC  RMSEA    CFI    TLI   SRMR
Chi     1.000 -0.168 -0.168  0.952 -0.920 -0.994  0.958
AIC    -0.168  1.000  1.000 -0.160  0.174  0.175 -0.151
BIC    -0.168  1.000  1.000 -0.160  0.174  0.175 -0.151
RMSEA   0.952 -0.160 -0.160  1.000 -0.941 -0.943  0.883
CFI    -0.920  0.174  0.174 -0.941  1.000  0.916 -0.839
TLI    -0.994  0.175  0.175 -0.943  0.916  1.000 -0.956
SRMR    0.958 -0.151 -0.151  0.883 -0.839 -0.956  1.000
================== Replications =====================
Number of replications = 100
Number of converged replications = 100
Number of nonconverged replications:
   1. Nonconvergent Results = 0
   2. Nonconvergent results from multiple imputation = 0
   3. At least one SE were negative or NA = 0
   4. At least one variance estimates were negative = 0
   5. At least one correlation estimates were greater than 1 or less than -1 = 0
```

Talk Outline

Missing Data

Motivation

Types of Missing Data

Traditional Data Handling with Missing Values

Modern Data Handling with Missing Values

Example

Missing Data in R

Talk Outline

Missing Data

Motivation

Types of Missing Data

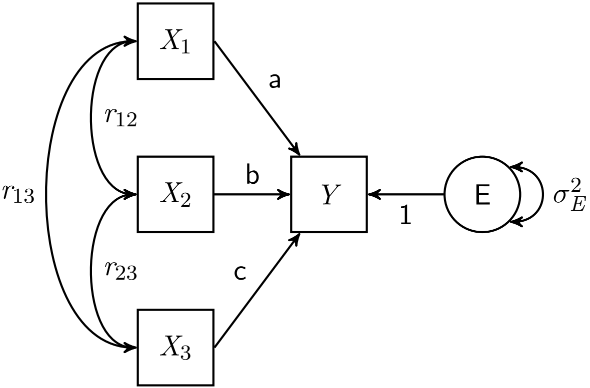
Traditional Data Handling with Missing Values

Modern Data Handling with Missing Values

Example

Missing Data in R

- Outcome (Y): math achievement
- Predictor: household wealth (X_1)
- Covariates:
  - Child cognitive ability (X_2)
  - (Average) parental education (X_3)
- You collect data on n students
  - 100% complete on Y
  - g% are missing on X_1, X_2, and X_3



Model for Motivational Example of Missing Data

Talk Outline

Missing Data

- Motivation
- Types of Missing Data
- Traditional Data Handling with Missing Values
- Modern Data Handling with Missing Values
- Example
- Missing Data in R

- R. J. A. Little and Rubin (2002) posit three different types of missing data:
  1. Missing Completely at Random (MCAR)
  2. Missing at Random (MAR)
  3. Missing Not At Random (MNAR)

Talk Outline

Missing Data

- Motivation
- Types of Missing Data
  - Missing Completely At Random
  - Missing At Random
  - Missing Not at Random
- Traditional Data Handling with Missing Values
- Modern Data Handling with Missing Values
- Example
- Missing Data in R

- **Definition.** The missing values on a given variable are unrelated to the underlying variable, as well as any other variable in the data.
- In our example, the reason why students do not have scores on any of the predictors is random, i.e., is completely unrelated Y, any of the predictors, or any other variable.
- For example, the data coder made random input errors; the respondent's pencil broke during an item and he/she forgot to go back and complete it.
- Key: no systematic reason why data are missing.
- Alternative Framework: students with completely observed data represent a *random subsample* of the complete data set.

- MCAR can be assessed.
- Good: *t*-tests.
  - Make two groups: those with data on the variable and those with missing data on the variable.
  - Compare the means between the two groups for every other variable in the data.
- Better: Little's (1988) χ^2
  - See `BaylorEdPsych` (Beaujean, 2012) for a rough implementation in `R`

Talk Outline

Missing Data

Motivation

Types of Missing Data

Missing Completely At Random

Missing At Random

Missing Not at Random

Traditional Data Handling with Missing Values

Modern Data Handling with Missing Values

Example

Missing Data in R

- Definition. The missing values on a given variable are unrelated to the underlying variable, but are related other variables in the data.
- In our example, the reason why students do not have scores on Xi is completely unrelated to X_i, but could be related to *Y* or X_j $(j = 1, 2, 3; j \neq i)$
- Concretely: Students with higher math achievement scores are found to not have cognitive ability information more often than other students. However, within a math achievement group, there is no relationship between missingness and cognitive ability.

- Cannot test for data being missing at random.

Talk Outline

Missing Data

- Motivation
- Types of Missing Data
  - Missing Completely At Random
  - Missing At Random
  - Missing Not at Random
- Traditional Data Handling with Missing Values
- Modern Data Handling with Missing Values
- Example
- Missing Data in R

- Definition. The missing values on a given variable are related to the underlying variable.
- In our example, the reason why students do not have scores on X_i related to X_i, $(i = 1, 2, 3)$.
- Concretely: Students whose parents have lower education levels, tend to report their (average) parental education less often.
- Synonymous with *non-ignorable* missing data.

Talk Outline

Missing Data

- Motivation
- Types of Missing Data
- Traditional Data Handling with Missing Values
- Modern Data Handling with Missing Values
- Example
- Missing Data in R

- "Traditional" techniques for handling missing data generally require MCAR
- "Modern" techniques for handling missing data generally only require MAR
- To understand why, we need to understand estimation bias.

- A statistic , $\hat{\theta}$, used to estimate a parameter, θ, is *unbiased* if and only if the expected value of the statistic is the parameter, i.e. $\mathsf{E}[\hat{\theta}] = \theta$.
- For example, the mean is an unbiased statistic

$$\mathsf{E}[\bar{X}] = \mathsf{E}\left[\frac{\sum_{i=1}^{n} X_i}{n}\right] = \frac{n\mu}{n} = \mu$$

- Variance is not an unbiased statistic

$$
\mathsf{E}[S^2] = \mathsf{E}\left[\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{n}\right] = \frac{1}{n}\left[\sum_{i=1}^{n}\mathsf{E}[(X_i - \mu)]^2 - n\mathsf{E}[(\bar{X} - \mu)^2]\right]
$$

$$
= \frac{1}{n}\left[n\sigma^2 - n\frac{\sigma^2}{n}\right] = \sigma^2(1 - \frac{1}{n})
$$

report. . . unanticipated events in data collection. These include missing data, attrition, and nonresponse. Discuss analytic techniques devised to ameliorate these problems. . . . The use of techniques to ensure that the reported results are not produced by anomalies in the data . . . should be a standard component of all analyses . . . Special issues arise in modeling when we have missing data. *The two popular methods for dealing with missing data that are found in basic statistics packages, listwise and pairwise deletion of missing values, are among the worst methods available for practical applications.* (Wilkinson & American Psychological Association Science Directorate Task Force on Statistical Inference, 1999, p.598, emphasis added)

Talk Outline

- Missing Data
  - Motivation
  - Types of Missing Data
  - Traditional Data Handling with Missing Values
    - Listwise Deletion
    - Pairwise Deletion
    - Mean Imputation
  - Modern Data Handling with Missing Values
  - Example
  - Missing Data in R

- Definition. Deletes all cases that have a missing value on any of the variables under examination in the model
- Provides unbiased estimates only if data MCAR
- However, as the n $\downarrow$, the standard error $(\sigma_\theta) \uparrow$ and statistical power $(1-\beta) \downarrow$

Talk Outline

Missing Data

- Motivation
- Types of Missing Data
- Traditional Data Handling with Missing Values
  - Listwise Deletion
  - Pairwise Deletion
  - Mean Imputation
- Modern Data Handling with Missing Values
- Example
- Missing Data in R

Missing Data

Traditional Data Handling with Missing Values : Pairwise Deletion

- Definition. Deletes cases on an analysis-by-analysis basis, where each statistic is calculated by using the cases with complete data from the variables needed for the statistic.
- *Within* a study, different subsets of cases are used for each analysis (are these comparable ?)
- For a covariance matrix, it is likely to be singular/non-positive definite (i.e., may not be invertible).
- Provides unbiased estimates only if data MCAR, but still leaves question of what the sample size is for the study.

Talk Outline

Missing Data

- Motivation
- Types of Missing Data
- Traditional Data Handling with Missing Values
  - Listwise Deletion
  - Pairwise Deletion
  - Mean Imputation
- Modern Data Handling with Missing Values
- Example
- Missing Data in R

- Definition. (Usually) the (arithmetic) mean for each variable is calculated using the available data, and is subsequently used to replace the missing response on that variable
- Many problems with using this technique
- $\downarrow \sigma^2_X \Rightarrow \downarrow \sigma_{XY}$
- Estimates are biased for all statistics (except the mean) under all missing data mechanisms.

Talk Outline

Missing Data

- Motivation
- Types of Missing Data
- Traditional Data Handling with Missing Values
- Modern Data Handling with Missing Values
- Example
- Missing Data in R

- Two common modern techniques
  - Full information maximum likelihood
  - Multiple imputation

Talk Outline

Missing Data

- Motivation
- Types of Missing Data
- Traditional Data Handling with Missing Values
- Modern Data Handling with Missing Values
  - Full Information Maximum Likelihood
  - Multiple Imputation
  - Auxiliary Variables
- Example
- Missing Data in R

- Full Information Maximum Likelihood (FIML)
- Maximum Likelihood (ML) is a general procedure to obtain both an estimator for a statistic, as well as estimates once you have data.
- Estimates can be found analytically for very simple models, but for more complex one it uses an iteration procedure until it comes across the "most likely" value for the statistic, given the data.
- Can be used both with and without missing data.
- But, does require distributional assumptions about the model under investigation.

- Typical covariance structure models use the sample's covariance (and mean) statistics as input into the estimator.
- The goal is to minimize a fit function value
- For FIML, though, each individual, *i*, contributes what data they have to the fit function.
- Consequently, we are interested in maximizing a (log) likelihood function $f(\cdot)$ that is comprised of the sum of likelihoods from each respondent.

- For a normally distributed variable

$$f_i(\mathbf{X}|\mu_i, \Sigma_i) = C_i - \frac{1}{2}\ln|\sigma_i|\frac{1}{2}MD_i$$

where
$\mathbf{X}_i$ is the "complete data" data matrix for the *i*th person
C_i is a "constant" to keep the function on the probability metric
Σ_i is the estimated (co)variance matrix using only variables on which *i*th person has complete data, and
MD_i is *i*th person's Mahalanobis distance matrix (a function of X_i, μ_i , and Σ_i), again, using only variables on which *i*th person has complete data.

- With FIML, usually observations do not have to be deleted from the analysis.
- There is no fixing of the data before estimation begins.
- Participants with partial data can contribute to the estimation of all the parameters (assuming the variables are related to each other) because FIML uses the "complete data" from the respondents with missing data as well as the relationship between all the variables
- Assumes missing data are MCAR or MAR.

- Assumes data are multivariate normal (although some programs can incorporate robust statistics for data that depart from this assumption)
- Have to have an *a priori* model for the data analysis

Talk Outline

Missing Data

- Motivation
- Types of Missing Data
- Traditional Data Handling with Missing Values
- Modern Data Handling with Missing Values
  - Full Information Maximum Likelihood
  - Multiple Imputation
  - Auxiliary Variables
- Example
- Missing Data in R

- Unlike mean (or regression) imputation, *multiple imputation* (MI) creates multiple ($m > 5$) data sets which contain (different) plausible estimates of the missing values.
- The data analysis is then computed on all imputed data sets.
- The parameter estimates from each analysis are then pooled to produce a final estimate.
- Typically a 3-step process
  - Impute
  - Analyze
  - Pool

- Imputing the data is the most complex step, and differs by computer program.
- Gist: Non-missing data used to make covariance matrix
- Covariance matrix used to make *augmented* regression equations to predict missing values
- The augmentation is the addition of random error to the regression equations
- Then the new covariance matrix is used to make new augmented regression equations to predict missing values.
- Catch: In between the creation of new imputed data sets, many other are created and discarded to alleviate autocorrelation

- For the data analysis, the m imputed (and complete) data sets are used to estimate p parameters of interest m times.

- Once the $m \times p$ estimates are calculated, then pool the $m \times p$ parameter estimates and their $m \times p$ standard errors.

- Requires multivariate normality
- Precludes nominal or ordinal variables (have to use augmented procedures to deal with these data types)
- Does not requires an a priori analysis model
- Because the imputation and analysis phase are independent, MI procedures can be used with (almost) any kind of model

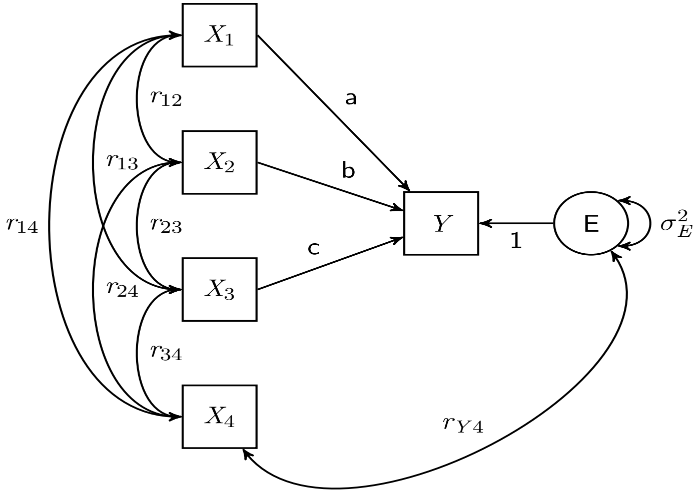
Talk Outline

Missing Data

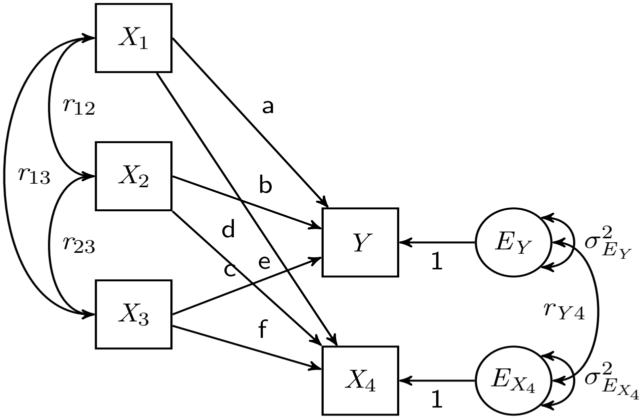
- Motivation
- Types of Missing Data
- Traditional Data Handling with Missing Values
- Modern Data Handling with Missing Values
  - Full Information Maximum Likelihood
  - Multiple Imputation
  - Auxiliary Variables
- Example
- Missing Data in R

- An *auxiliary variable* (AV) is a variable that you are not interested in, per se, but is included in the model because it is either a potential cause or correlate of missingness, or a correlate of the variable that is missing.
- Can be used with both FIML and MI.
- In our motivational example, say the number of hours parents are home (X_4) is related to missing data on household wealth (X_1), child cognitive ability (X_2) and (average) parental education (X_3). However, the number of hours parents are home are not of interest to the study, per se. For MAR to hold, though, you have to take X_4 into account.

- Graham (2003) suggests:
  - AVs should be correlated with observed (not latent) exogenous in the model.
  - AVs should be correlated with the residual terms from observed (not latent) endogenous variables.
  - AVs should be correlated with each other.



Model for Motivational Example of Missing Data with Auxiliary Variable
"Saturated Correlates" Model



Model for Motivational Example of Missing Data with Auxiliary Variable
"Extra DV" Model

Talk Outline

Missing Data

- Motivation
- Types of Missing Data
- Traditional Data Handling with Missing Values
- Modern Data Handling with Missing Values
- **Example**
- Missing Data in R

Missing Data
Example

- Average math achievement scores (*Y*) Household wealth: (X_1) Child cognitive ability (X_2) (Average) parental education (X_3).
- Number of hours parents are home (X_4) is an auxiliary variable.
- Collect data on 100 students and have complete data on Y, but 19% are missing on $X_i (i = 1, 2, 3)$.

- Simulate data

$$\begin{bmatrix} Y_{p=1} \\ - \\ X_{p=3|1} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \hline 0 \end{bmatrix}, \begin{bmatrix} 1.00 & 0.20 & 0.30 & 0.40 & 0.50 \\ 0.20 & 1.00 & 0.35 & 0.60 & 0.70 \\ 0.30 & 0.35 & 1.00 & 0.45 & 0.65 \\ 0.40 & 0.60 & 0.45 & 1.00 & 0.55 \\ \hline 0.50 & 0.70 & 0.65 & 0.55 & 1.00 \end{bmatrix} \right)$$

- MCAR: Randomly deleted $\approx$ 19% of values on variables $X_1 - X_3$. Little's (1988) $\chi^2_{df=15} = 13.937$, $p = 0.530$.
- MAR: For X_i $(i = 1, 2, 3)$, deleted largest m_i values of λ_i, where $\lambda_i = \sum_{j=1}^{4} X_j - X_i$, constrained such that $\frac{\sum_{i=1}^{3} m_i}{n_{X_1} + n_{X_2} + n_{X_3}} \approx .19$
- MNAR: For X_i $(i = 1, 2, 3)$, deleted largest m_i values
- m_1: 21
- m_2: 18
- m_3: 17

Pairwise complete n

	Math	Wealth	Child IQ	Parent Ed
Math	100	79	82	83
Wealth	79	79	61	63
Child IQ	82	61	82	65
Parent Ed	83	63	65	83

Covariance Matrix and Means for Full Data Set

	Math	Wealth	Child IQ	Parent Ed	Parent Home
Math	0.883	0.270	0.327	0.500	0.402
Wealth	0.270	1.299	0.356	0.849	0.871
Child IQ	0.327	0.356	1.078	0.569	0.602
Parent Ed	0.500	0.849	0.569	1.323	0.699
Parent Home	0.402	0.871	0.602	0.699	0.961
Mean	-0.044	0.099	-0.002	0.068	0.061

- One way to examine how close the covariance matrices with missing data are to the matrix without missing data is the root mean square residual (RMR)

$$RMR = \sqrt{\frac{\sum_{i=1}^{n}(r_i - r_i^*)^2}{p}}$$

where
r is the original covariance,
r^* is the covariance from the missing data, and
p is the number of correlations.

- Smaller values are better.

Covariance and Means for MCAR data, Listwise Deletion. RMR: .031

	Math	Wealth	Child IQ	Parent Ed	Parent Home
Math	0.92	0.28	0.08	0.47	0.31
Wealth	0.28	1.27	0.25	0.99	0.96
Child IQ	0.08	0.25	0.81	0.34	0.46
Parent Ed	0.47	0.99	0.34	1.43	0.73
Parent Home	0.31	0.96	0.46	0.73	1.07
Means	0.05	0.27	-0.02	0	0.2

Comparisons

	MCAR		MAR		MNAR	
Analysis	n	RMR	n	RMR	n	RMR
Listwise	45	0.031	44	0.401	65	0.401
Pairwise	61	0.002	61	0.075	61	0.158
Mean Imputation	100	0.034	100	0.077	100	0.239
FIML	100	0.003	100	0.002	100	0.110
MI	100	0.004	100	0.002	100	0.115

MCAR Results: Point Estimates

Data Set	a	Δ a	b1	Δ b1	b2	Δ b2	b3	Δ b3	R^2	*n*
Full	-0.06	–	-0.07	–	0.13	–	0.36	–	0.23	100
Listwise	0.07	0.13	-0.08	-0.02	-0.05	-0.18	0.40	0.04	0.18	45
Pair	-0.03	0.03	-0.11	-0.04	0.11	-0.03	0.40	0.04	0.25	61
Mean Imputation	-0.05	0.02	-0.02	0.05	0.15	0.01	0.34	-0.02	0.21	100
FIML	-0.03	0.03	-0.08	-0.01	0.10	-0.03	0.36	0.00	0.22	100
FIML/Aux	-0.05	0.01	-0.06	0.01	0.11	-0.02	0.34	-0.02	0.21	100
MI (m=5)	-0.03	0.03	-0.05	0.02	0.11	-0.02	0.32	-0.04	0.21	100

MCAR Results: Standard Errors

Data Set	σ_a	$\Delta\sigma_a$	σ_{b_1}	$\Delta\sigma_{b_1}$	σ_{b_2}	$\Delta\sigma_{b_2}$	σ_{b_3}	$\Delta\sigma_{b_3}$	*n*
Full	0.08	–	0.10	–	0.09	–	0.10	–	100
Listwise	0.14	0.06	0.18	0.08	0.16	0.07	0.17	0.07	45
Pair	0.11	0.02	0.12	0.03	0.11	0.02	0.13	0.02	61
Mean Imputation	0.09	0.00	0.10	0.00	0.10	0.00	0.10	-0.01	100
FIML	0.09	0.00	0.12	0.02	0.10	0.01	0.12	0.02	100
FIML/aux	0.08	0.00	0.11	0.01	0.10	0.00	0.12	0.01	100
MI (m=5)	0.09	0.00	0.14	0.04	0.11	0.02	0.14	0.04	100

MAR Results: Point Estimates

Data Set	a	Δ a	b1	Δ b1	b2	Δ b2	b3	Δ b3	R^2	*n*
Full	-0.06	–	-0.07	–	0.13	–	0.36	–	0.23	100
Listwise	0.11	0.17	-0.15	-0.08	0.03	-0.11	0.33	-0.03	0.14	44
Pair	0.07	0.14	-0.35	-0.28	-0.15	-0.28	0.62	0.26	0.30	61
Mean Imputation	-0.01	0.05	-0.12	-0.05	0.11	-0.02	0.36	0.00	0.23	100
FIML	0.00	0.07	-0.12	-0.06	0.09	-0.05	0.36	0.00	0.24	100
FIML/aux	-0.02	0.05	-0.14	-0.07	0.08	-0.06	0.38	0.02	0.25	100
MI (m=5)	-0.04	0.02	-0.09	-0.02	0.15	0.02	0.36	0.00	0.25	100

MAR Results: Standard Errors

Data Set	σ_a	$\Delta\sigma_a$	σ_{b_1}	$\Delta\sigma_{b_1}$	σ_{b_2}	$\Delta\sigma_{b_2}$	σ_{b_3}	$\Delta\sigma_{b_3}$	*n*
Full	0.08	–	0.10	–	0.09	–	0.10	–	100
Listwise	0.23	0.15	0.15	0.06	0.13	0.04	0.14	0.04	44
Pair	0.12	0.03	0.17	0.07	0.16	0.07	0.18	0.08	61
Mean Imputation	0.09	0.01	0.11	0.01	0.10	0.01	0.10	0.00	100
FIML	0.10	0.01	0.13	0.03	0.12	0.03	0.14	0.03	100
FIML/aux	0.09	0.01	0.13	0.03	0.11	0.02	0.13	0.03	100
MI (m=5)	0.09	0.00	0.12	0.02	0.10	0.01	0.12	0.01	100

MAR Results: Point Estimates

Data Set	a	Δ a	b1	Δ b1	b2	Δ b2	b3	Δ b3	R^2	*n*
Full	-0.06	–	-0.07	–	0.13	–	0.36	–	0.23	100
Listwise	-0.05	0.01	-0.06	0.01	0.15	0.01	0.43	0.07	0.18	65
Pair	0.09	0.16	0.01	0.08	0.08	-0.06	0.38	0.02	0.16	61
Mean Imputation	0.10	0.16	0.03	0.09	0.09	-0.04	0.37	0.01	0.13	100
FIML	0.06	0.13	-0.02	0.04	0.07	-0.06	0.40	0.04	0.16	100
FIML/aux	0.13	0.19	0.03	0.10	0.10	-0.04	0.39	0.03	0.18	100
MI (m=5)	0.06	0.12	0.03	0.10	0.11	-0.02	0.37	0.00	0.18	100

MAR Results: Standard Errors

Data Set	σ_a	$\Delta\sigma_a$	σ_{b_1}	$\Delta\sigma_{b_1}$	σ_{b_2}	$\Delta\sigma_{b_2}$	σ_{b_3}	$\Delta\sigma_{b_3}$	*n*
Full	0.08	–	0.10	–	0.09	–	0.10	–	100
Listwise	0.14	0.06	0.15	0.05	0.16	0.06	0.15	0.04	65
Pair	0.13	0.04	0.15	0.06	0.16	0.07	0.15	0.05	61
Mean Imputation	0.11	0.02	0.13	0.04	0.14	0.05	0.13	0.02	100
FIML	0.10	0.02	0.14	0.04	0.14	0.04	0.13	0.03	100
FIML/aux	0.09	0.01	0.13	0.03	0.13	0.04	0.13	0.03	100
MI (m=5)	0.10	0.01	0.12	0.03	0.14	0.05	0.15	0.04	100

Talk Outline

Missing Data

- Motivation
- Types of Missing Data
- Traditional Data Handling with Missing Values
- Modern Data Handling with Missing Values
- Example
- Missing Data in R

```
# Generate Data
library(MASS)

true.data<-matrix(c
      (1, .20, .30, .40, .50,.20,1,.35,.60,.70,.30,.35,1,.45,.65,.40,.60,.45,1,.55,.50,
.70,.65,.55,1), 5,5)

set.seed(5456)
data.start<-mvrnorm(100, c(0,0,0,0,0),true.data)
colnames(data.start)<-c("MATH", "WEALTH", "CHILD_IQ", "PARENT_ED", "PARENT_HOME")
data.full<-data.frame(data.start)

# MCAR
MCAR.data<-data.start
var.mis<-3 #number of varibales you want missing data on
sampling1<-rep(NA,nrow(data.full)*var.mis*.25)
sampling2<-rep(NA,nrow(data.full)*var.mis*.25)
for( i in 1:nrow(data.full)*var.mis*.25){
set.seed(i)
sampling1[i]<-sample(2:4,1)#just have var 2 &3 have missing data
set.seed(i)
sampling2[i]<-sample(100,nrow(data.full)*var.mis*.25)  }
for (j in 1:nrow(data.full)*var.mis*.25){
MCAR.data[sampling2[j],sampling1[j]]<-NA
}

MCAR.data<-data.frame(MCAR.data)
```

- Analysis for full data set

```
> librar(lavaan)
> full.model<-'
+ MATH~ a*1+ b1*WEALTH + b2*CHILD_IQ+ b3*PARENT_ED
+ '
>
> full.fit<-sem(full.model, data=data.full)
> summary(full.fit)
lavaan (0.5-9) converged normally after   1 iterations

  Number of observations                           100

  Estimator                                         ML
  Minimum Function Chi-square                    0.000
  Degrees of freedom                                 0
  P-value                                        1.000

Parameter estimates:

  Information                                 Expected
  Standard Errors                             Standard

                   Estimate  Std.err  Z-value  P(>|z|)
Regressions:
  MATH ~
```

```
    WEALTH   (b1)    -0.065    0.095   -0.688    0.492
    CHILD_IQ (b2)     0.134    0.090    1.485    0.138
    PARENT_E (b3)     0.362    0.102    3.560    0.000

Intercepts:
    MATH      (a)    -0.062    0.082   -0.755    0.450

Variances:
    MATH              0.669    0.095
```

- Listwise deletion

```
#Listwise Deletion
MCAR.Listwise.fit<-sem(full.model, data=MCAR.data)
summary(MCAR.Listwise.fit, rsquare=TRUE)
```

- Pairwise deletion

```
pairwise.MCAR.cov<-cov(MCAR.data, use="pairwise.complete.obs")
pairwise.MCAR.mean<-c(mean(MCAR.data$MATH,na.rm=TRUE),mean(MCAR.data$WEALTH, na.rm=TRUE),
    mean(MCAR.data$CHILD_IQ,na.rm=TRUE),mean(MCAR.data$PARENT_ED, na.rm=TRUE),mean(MCAR.
    data$PARENT_HOME, na.rm=TRUE))
names(pairwise.MCAR.mean)<-colnames(pairwise.MCAR.cov)

MCAR.Pairwise.fit<-sem(full.model, sample.cov=pairwise.MCAR.cov, sample.nobs=61, sample.
    mean=pairwise.MCAR.mean)
summary(MCAR.Pairwise.fit, rsquare=TRUE)
```

- Mean imputation

```
> library(Hmisc)
> #Create mean-imputed data set
> MCAR.MeanI.data<-MCAR.data
> MCAR.MeanI.data$WEALTH<-impute(MCAR.MeanI.data$WEALTH, fun=mean)
> MCAR.MeanI.data$CHILD_IQ<-impute(MCAR.MeanI.data$CHILD_IQ, fun=mean)
> MCAR.MeanI.data$PARENT_ED<-impute(MCAR.MeanI.data$PARENT_ED, fun=mean)

> MCAR.meanImputation.fit<-sem(full.model, data=MCAR.MeanI.data)
> summary(MCAR.meanImputation.fit, rsquare=TRUE)
```

- FIML

```
MCAR.FIML.fit<-sem(full.model, data=MCAR.data, missing="fiml")
summary(MCAR.FIML.fit, rsquare=TRUE)
```

- FIML with Auxiliary variables

```
> library(semTools)
> #FIML wuth auxuliary variable--second DV
> MCAR.FIMLAux.fit<-auxiliary(MCAR.FIML.fit, aux="PARENT_HOME", data=MCAR.data)
> summary(MCAR.FIMLAux.fit, rsquare=TRUE)

> #FIML with Auxiliary variable--saturated correlations
> MCAR.FIML.Aux.model<-'
> MATH~  b1*WEALTH + b2*CHILD_IQ+ b3*PARENT_ED + 0*PARENT_HOME
> MATH + WEALTH + CHILD_IQ + PARENT_ED ~~ PARENT_HOME
> '
> MCAR.FIML.Aux.fit<-sem(model=MCAR.FIML.Aux.model, data=MCAR.data, missing="fiml", fixed
    .x=FALSE)
> summary(MCAR.FIML.Aux.fit, rsquare=TRUE)
```

- Multiple imputation

```
library(Amelia)
library(semTools)
MCAR.sim <- amelia(MCAR.data,m=5)
MCAR.MI.fit <- runMI(full.model, data=MCAR.sim$imputations, fun="sem")
summary(MCAR.MI.fit)
```

References

Allen, M. J., & Yen, W. M. (1979). *Introduction to measurement theory*. Long Grove, IL: Waveland Press.

Antelman, G. (1997). *Elementary Bayesian statistics*. Cheltenham, UK: Edward Elgar.

Atkins, D. C., Baldwin, S., Zheng, C., Gallop, R. J., & Neighbors, C. (in press). A tutorial on count regression and zero-altered count models for longitudinal addictions data.

Barrett, P. (2007). Structural equation modelling: Adjudging model fit. *Personality and Individual Differences*, *42*(5), 815-824. doi: 10.1016/j.paid.2006.09.018

Bartholomew, D., Knott, M., & Moustaki, I. (2011). *Latent variable models and factor analysis: : A unified approach* (3rd ed.). New York: John Wiley Sons.

References (cont.)

Beaujean, A. A. (2012). BaylorEdPsych: R package for Baylor University Educational Psychology quantitative courses (Version 0.5) [Computer software]. Waco, TX: Baylor University.

Beaujean, A. A., Freeman, M. J., Youngstrom, E., & Carlson, G. (2012). The structure of cognitive abilities in youths with manic symptoms: A factorial invariance study. *Assessment*, *19*(4), 462-471. doi: 10.1177/1073191111399037

Bock, R., & Aitkin, M. (1981). Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika*, *46*(4), 443-459. doi: 10.1007/BF02293801

Carroll, J. B. (1993). *Human cognitive abilities: A survey of factor-analytic studies*. New York, NY: Cambridge University Press.

Chen, F. F., West, S. G., & Sousa, K. H. (2006). A comparison of bifactor and second-order models of quality of life. *Multivariate Behavioral Research*, *41*(2), 189-225. doi: 10.1207/s15327906mbr4102_5

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Mahwah, NJ: Erlbaum.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, *16*, 297-334. doi: 10.1007/bf02310555

Curran, P., West, S., & Finch, J. (1996). The robustness of test statistics to nonnormality and specification error in confirmatory factor analysis. *Psychological Methods*, *1*(1), 16-29. doi: 10.1037/1082-989X.1.1.16

Edwards, W., Lindman, H., & Savage, L. J. (1963). Bayesian statistical inference for psychological research. *Psychological Review*, *70*(3), 193-242. doi: 10.1037/h0044139

Efron, B., & Tibshirani, R. J. (1994). *An introduction to the bootstrap*. London: Chapman Hall.

References (cont.)

Enders, C. K. (2001). The impact of nonnormality on full information maximum-likelihood estimation for structural equation models with missing data. *Psychological Methods*, *6*(4), 352-370. doi: 10.1037/1082-989X.6.4.352

Gorsuch, R. L. (1983). *Factor analysis*. Hillsdale, NJ: Lawrence Erlbaum.

Graham, J. W. (2003). Adding missing-data-relevant variables to FIML-based structural equation models. *Structural Equation Modeling: A Multidisciplinary Journal*, *10*(1), 80-100. doi: 10.1207/S15328007SEM1001_4

Guttman, L. (1945). A basis for analyzing test-retest reliability. *Psychometrika*, *10*(4), 255-282. doi: 10.1007/BF02288892

Hambleton, R. K., & Swaminathan, H. (1985). *Item response theory: Principles and applications*. Boston, MA: Kluwer.

Jennrich, R., & Bentler, P. (2011). Exploratory bi-factor analysis. *Psychometrika*, *76*(4), 537-549. doi: 10.1007/s11336-011-9218-4

References (cont.)

Joreskog, K. G. (1994). On the estimation of polychoric correlations and their asymptotic covariance matrix. *Psychometrika*, *59*(3), 381-389. doi: 10.1007/bf02296131

Kamata, A., & Bauer, D. J. (2008). A note on the relation between factor analytic and item response theory models. *Structural Equation Modeling: A Multidisciplinary Journal*, *15*(1), 136 - 153. doi: 10.1080/10705510701758406

Keith, T. Z. (2006). *Multiple regression and beyond*. Boston: Pearson.

Kenny, D. A., Kashy, D., & Bolger, N. (1988). Data analysis in social psychology. In D. Gilbert, S. Fiske, & G. Lindzey (Eds.), *Handbook of social psychology* (4th ed., p. 233-265). New York: McGraw-Hill.

Little, R. J. A. (1988). A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, *83*, 1198-1202.

Little, R. J. A., & Rubin, D. B. (2002). *Statistical analysis with missing data* (2nd ed.). Hoboken, NJ: John Wiley.

References (cont.)

Little, T. D., Cunningham, W. A., Shahar, G., & Widaman, K. F. (2002). To parcel or not to parcel: Exploring the question, weighing the merits. *Structural Equation Modeling*, *9*(2), 151-173. doi: 10.1207/s15328007sem0902_1

Loehlin, J. C. (2004). *Latent variable models: An introduction to factor, path, and structural equation analysis* (4th ed.). Mahwah, NJ: Erlbaum.

Lord, F. M., & Novick, M. R. (1968). *Statistical theories of mental tests*. Reading, MA: Addison-Wesley.

McDonald, R. P. (1999). *Test theory: A unified treatment*. Mahwah, NJ: Erlbaum.

McIver, J. P., Carmines, E. G., & Zeller, R. A. (1980). Multiple indicators (Appendix). In R. A. Zeller & E. G. Carmines (Eds.), *Measurement in the social sciences* (p. 162-185). Cambridge, UK: Cambridge University Press.

References (cont.)

Muthen, L. K., & Muthen, B. O. (2002). How to use a Monte Carlo study to decide on sample size and determine power. *Structural Equation Modeling*, *9*(4), 599-620. doi: 10.1207/S15328007SEM0904_8

Neale, M. C., & Maes, H. H. M. (1992). *Methodology for genetic studies of twins and families*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Patton, S., Benedict, H., & Beaujean, A. A. (submitted). Parental bonds, attachment anxiety, media internalization, and body dissatisfaction: A mediation model. *Developmental Psychology*.

Pornprasertmanit, S., Miller, P., & Schoemann, A. (2012). *simsem: SIMulated Structural Equation Modeling.*

Preacher, K. J., & Kelley, K. (2011). Effect size measures for mediation models: Quantitative strategies for communicating indirect effects. *Psychological Methods*, *16*(2), 93-115. doi: 10.1037/a0022658

References (cont.)

Rindskopf, D., & Rose, T. (1988). Some theory and applications of confirmatory second-order factor analysis. *Multivariate Behavioral Research*, *23*(1), 51-67. doi: 10.1207/s15327906mbr2301_3

Rizopoulos, D. (2006). ltm: An R package for latent variable modeling and item response theory analyses. *Journal of Statistical Software*, *17*(5), 1-25.

Satorra, A., & Bentler, P. M. (1994). Corrections to test statistics and standard errors in covariance structure analysis. In A. von Eye & C. C. Clogg (Eds.), *Latent variables analysis: Applications for developmental research* (p. 399âĂŞ419). Thousands Oaks, CA: Sage.

Satorra, A., & Saris, W. (1985). Power of the likelihood ratio test in covariance structure analysis. *Psychometrika*, *50*(1), 83-90. doi: 10.1007/bf02294150

Schmid, J., & Leiman, J. (1957). The development of hierarchical factor solutions. *Psychometrika*, *22*(1), 53-61. doi: 10.1007/bf02289209

References (cont.)

Schmitt, N. (1996). Uses and abuses of coefficient alpha. *Psychological Assessment*, *8*(4), 350-353. doi: 10.1037/1040-3590.8.4.350

Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, *15*(1), 72-101. doi: 10.2307/1412159

Takane, Y., & de Leeuw, J. (1987). On the relationship between item response theory and factor analysis of discretized variables. *Psychometrika*, *52*(3), 393-408. doi: 10.1007/BF02294363

West, S. G., Finch, J. F., & Curran, P. J. (1995). Structural equation models with non-normal variables: Problems and remedies. In R. A. Hoyle (Ed.), *Structural Equation Modeling: Concepts, Issues and Applications* (p. 56-75). Newbury Park, CA: Sage.

White, H. (1982). Maximum likelihood estimation of misspecified models. *Econometrica*, *50*(1), 1-25.

References (cont.)

Wilkinson, L., & American Psychological Association Science Directorate Task Force on Statistical Inference. (1999). Statistical methods in psychology journals: Guidelines and explanations. *American Psychologist*, *54*(8), 594-604. doi: 10.1037/0003-066X.54.8.594

Yuan, K.-H., & Bentler, P. M. (1998). Robust mean and covariance structure analysis. *British Journal of Mathematical and Statistical Psychology*, *51*(1), 63-88. doi: 10.1111/j.2044-8317.1998.tb00667.x

Yung, Y.-F., Thissen, D., & McLeod, L. D. (1999). On the relationship between the higher-order factor model and the hierarchical factor model. *Psychometrika*, *64*(2), 113-128. doi: 10.1007/bf02294531