Confirmatory Factor Models
(CFA: Confirmatory Factor Analysis)

- Topics:
  - **Comparison of EFA and CFA**
  - CFA model parameters
  - Two parts of CFA model identification
  - CFA model estimation
  - CFA model fit evaluation

EFA vs. CFA: What gets analyzed

- **EFA: Correlation matrix (of items = indicators)**
  - Only correlations among observed item responses are used
  - Only a standardized solution is provided, so the original means and variances of the observed item responses are irrelevant

- **CFA: Covariance matrix (of items = indicators)**
  - Variances and covariances of observed item responses are analyzed
  - Item response means historically have been ignored (but not by us!)
  - Output includes unstandardized AND standardized solutions
    - **Unstandardized** solution predicts the original **item covariance matrix** (regression solution retains original absolute information)
    - **Standardized** (STDYX) solution predicts the **item correlation matrix** (easier to interpret relative sizes of relationships as correlations)

EFA vs. CFA: Interpretation

- **EFA: Rotation**
  - All items load on all factors (traits), no matter what!
  - Goal is to pick a rotation that gives closest approximation to "simple structure" (clearly-defined factors, fewest cross-loadings)
  - No way of distinguishing latent variables due to "content" (traits being measured) from "method" (correlation induced by common approach)

- **CFA: Your job in the first place!**
  - CFA must be theory-driven: any structure is a testable hypothesis
  - You specify number of latent variables and their inter-correlations
  - You specify which items load on which latent variables (yes/no)
  - You specify any additional relationships for method/other covariance
  - You just need a clue; you don't have to be right (misfit is informative)

EFA vs. CFA: Judging model fit

- **EFA: Eye-balls and Opinion**
  - #Factors? Scree-ish plots, interpretability…
  - Which rotation? Whichever makes most sense… (to you)
  - Which items load on each factor? Arbitrary cut-off of .3-.4ish

- **CFA: Inferential tests via Maximum Likelihood (ML or MLR)**
  - Global model fit test (and local model fit)
  - Standard errors (and significance) of item loadings
  - Standard errors of error variances (and covariances)
  - Ability to test appropriateness of model constraints or model additions via tests for change in model fit

EFA vs. CFA: Factor scores

- **EFA: Don't ever use factor scores from an EFA**
  - Factor scores are indeterminate (especially due to rotation)
  - Inconsistency in how factor models are applied to data
    - Factor model based on common variance only (factor is predictor)
    - Summing items? That's using total variance (component is outcome)

- **CFA: Factor scores *can* be used, but only if necessary**
  - Best option: Test relations among latent factors directly through SEM
    - Factors can either be predictors ("exogenous" variables) or outcomes ("endogenous" variables) or both at once as needed (e.g., as mediators)
    - Relations between factors will be disattenuated for measurement error
  - Factor scores are less indeterminate in CFA, and could be used
    - In reality, though, factor scores are not known single values because they are modeled as random effects, not fixed effects per person
    - Next-best option: Use "plausible values" or other two-stage approaches that acknowledge uncertainty in factor score estimates (stay tuned)

Confirmatory Factor Models
(CFA: Confirmatory Factor Analysis)

- Topics:
  - Comparison of EFA and CFA
  - **CFA model parameters**
  - Two parts of CFA model identification
  - CFA model estimation
  - CFA model fit evaluation

Confirmatory Factor Analysis (CFA)

- **The CFA unit of analysis is the ITEM (as in any LTMM):**

$$y_{is} = \mu_i + \lambda_i F_s + e_{is}$$ → **both items AND subjects matter**

  - Observed response for item *i* and subject *s*
    = intercept of item *i* (μ)
    + subject *s*'s latent factor (F), weighted by item-specific loading λ
    + error (e) of item *i* and subject *s*

- **What does this look like? Linear regression (without an observed X)!**
  - $y_s = \beta_0 + \beta_1 X_s + e_s$ → written for each item → $y_{is} = \beta_{0i} + \beta_{1i} X_s + e_{is}$
  - β_{0i} Intercept = μ_i = expected outcome when F=0
  - β_{1i} Slope of Factor = λ_i = expected change in y for one-unit change in F
  - e_{is} Error (Residual) = e_{is} = how far off predicted y is from real y

Revisiting Vocabulary: Item Psychometric Properties

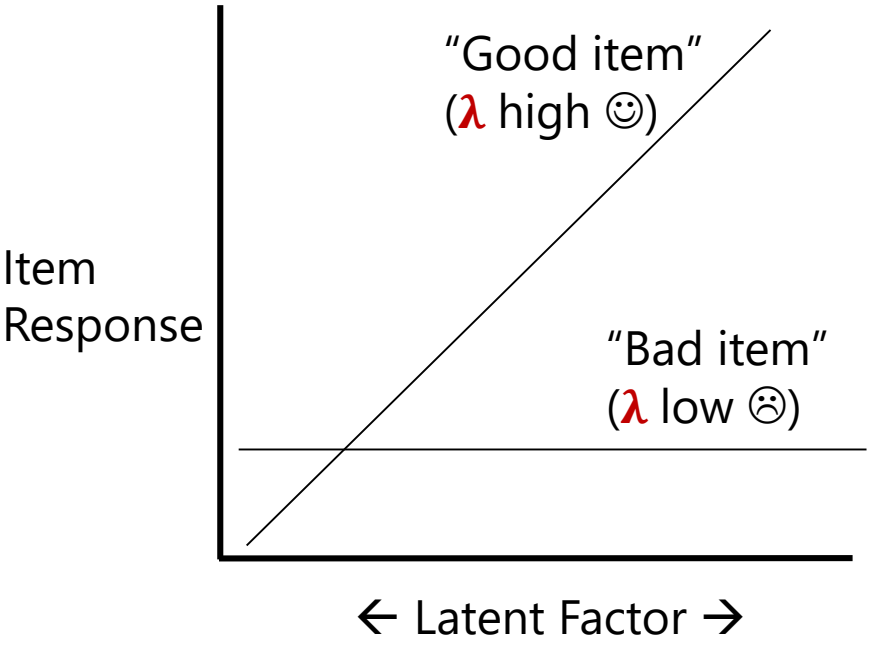
- **Item Discrimination**: How related each item is to the latent trait
  - In CTT, discrimination is given by the item-total (or item-remainder) correlation
    - The total score is the best estimate of the latent trait in CTT
  - In **CFA**, discrimination is given by the **factor loading/slope (λ_i)**
    - We now have a factor that directly represents the covariance among items
    - Stronger standardized factor loadings indicate better, more discriminating items

- **Item Difficulty/Severity**: Location of item on the latent trait metric
  - In CTT, difficulty is given by the item mean
  - In **CFA**, difficulty is given by the **item intercept (μ_i)** – which is still backwards
  - In contrast to other latent trait models (IRT), difficulty (intercepts) are often ignored in CFA… here's why…

Why Item Intercepts Are Often Ignored…

A **"good" item** has a large slope (i.e., factor loading) in predicting the item response from the factor. Because this is a **linear slope**, the item is assumed to be equally discriminating (**equally good**) across the entire latent trait.

Similarly, a **"bad" item** has a flatter linear slope that is **equally bad** across the entire range of the latent trait (where slope=0 means unrelated to trait).



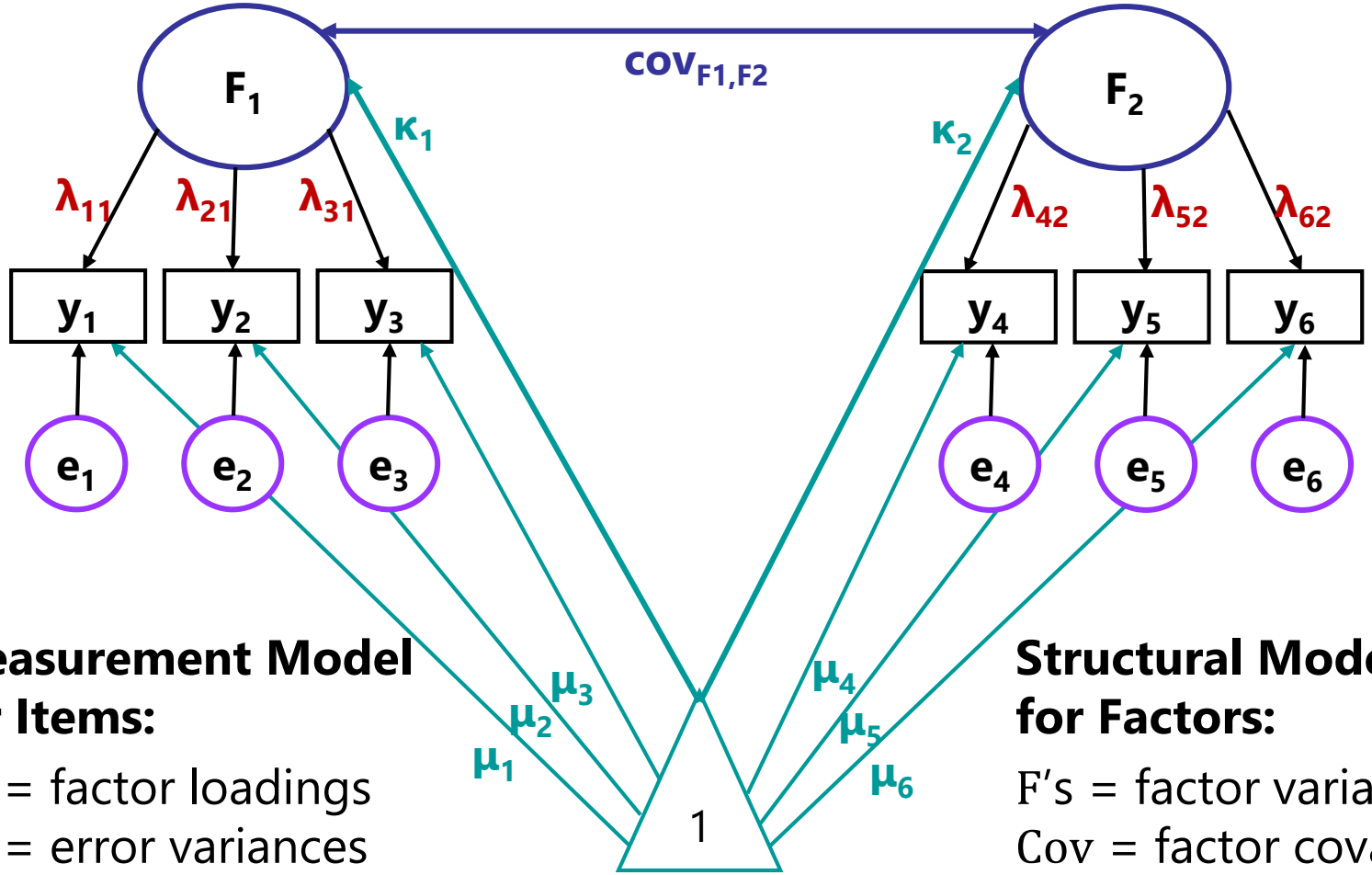
Here item intercepts are irrelevant in evaluating how "good" an item is, so they are not really needed in CFA.

But we will estimate them, because item intercepts are critical when:

- Testing factor mean differences in any latent factor model
- Items need to have a nonlinear slope in predicting the item response from the factor (IRT)

Example Diagram of Two-Factor CFA Model

But some parameters will have to be fixed to known values for the model to be identified.



Measurement Model for Items:

λ's = factor loadings
e's = error variances
μ's = intercepts

Structural Model for Factors:

F's = factor variances
Cov = factor covariances
K's = factor means

2 Types of CFA Parameter Solutions

- Unstandardized → predicts scale-sensitive original item response:
  - **Regression Model:** $y_{is} = \mu_i + \lambda_i F_s + e_{is}$
  - *Useful when comparing solutions across groups or time (when absolute values matter)*
  - Together, the model parameters predict the item means and ***item covariance matrix***
  - Note the solution asymmetry: item parameters μ_i and λ_i will be given in the item metric, but e_{is} will be given as the error variance across persons for that item (squared metric)
  - $Var(y_i) = [\lambda_i^2 * Var(F)] + Var(e_i)$

Careful! There is also STDY and STD

- Standardized → Solution transformed to $Var(y_i) = Var(F) = 1$ via **STDYX**:
  - *Useful when comparing items within a solution (relative values on same scale)*
  - Together, the standardized model parameters predict the **item correlation matrix**
  - Standardized intercept = μ_i / $SD(y_{is})$ → not typically reported
  - Standardized factor loading = $[\lambda_i * SD(F_s)]$ / $SD(y_{is})$ **= item correlation with factor**
  - Standardized error variance = $1 - standardized\ \lambda_i^2$ = "variance due to *not* factor"
  - R^2 for item = $standardized\ \lambda_i^2$ = "variance due to the factor"

Example Two-Factor Model Equations

- Measurement model per numbered item for subject *s*:
  - $y_{1s} = \mu_1 + \lambda_{11}F_{1s} + 0F_{2s} + e_{1s}$
  - $y_{2s} = \mu_2 + \lambda_{21}F_{1s} + 0F_{2s} + e_{2s}$
  - $y_{3s} = \mu_3 + \lambda_{31}F_{1s} + 0F_{2s} + e_{3s}$
  - $y_{4s} = \mu_4 + 0F_{1s} + \lambda_{42}F_{2s} + e_{4s}$
  - $y_{5s} = \mu_5 + 0F_{1s} + \lambda_{52}F_{2s} + e_{5s}$
  - $y_{6s} = \mu_6 + 0F_{1s} + \lambda_{62}F_{2s} + e_{6s}$

Here is the general matrix equation for these 6 item-specific equations:

$$\boldsymbol{Y} = \boldsymbol{\mu} + \boldsymbol{\lambda F} + \boldsymbol{e}$$

where $\boldsymbol{Y}$, $\boldsymbol{\mu}$, and $\boldsymbol{e}$ = 6x1 matrices (because each item gets one value); $\boldsymbol{\lambda}$ = 6x2 matrix, and $\boldsymbol{F}$ =2x1 matrix (because there are two factors)

You decide **how many factors** and if each item has an estimated loading on each factor or not.

Unstandardized loadings ($\boldsymbol{\lambda}$) are the **linear slopes** predicting the item response (y) from the factor (F). **Thus, the model assumes a linear relationship between the factor and the item response.**

Standardized loadings are the slopes in a **correlation** metric (and Standardized Loading^2 = $\boldsymbol{R^2}$).

Intercepts ($\boldsymbol{\mu}$) are the expected item responses ($\hat{\boldsymbol{y}}$) when all factors = 0.

The Role of the CFA Model Parameters

- Data going in to be predicted by the CFA model parameters = item covariance matrix (variances, covariances) and item means

- The CFA **item intercepts** ($\boldsymbol{\mu_i}$) predict the **item means**
  - Item means are unconditional; item intercepts are conditional on $\boldsymbol{F_s = 0}$
  - When each item gets its own intercept (the usual case), the item means will be perfectly predicted (**so no room for mis-fit or mis-prediction**)

- The CFA **item error variances** ($\boldsymbol{Var[e_i]}$) predict the **item variances**
  - Item variances are unconditional; item error variances are conditional (leftover variance after accounting for the contribution of the factor)
  - When each item gets its own error variance (usual case), the item variances will be perfectly predicted (**so no room for misfit or mis-prediction**)

- The CFA **item factor loadings** ($\boldsymbol{\lambda_i}$) predict the **item covariances**
  - Given 3+ items, there will be more covariances among items to predict than item factor loadings to predict them, **thus creating room for misfit**

CFA Model Predictions: (F_1 BY y_1-y_3, F_2 BY y_4-y_6)

Items from same factor (room for misfit or mis-prediction):

- Unstandardized solution: Covariance(y_1, y_3) = $\boldsymbol{\lambda_{11} * Var(F_1) * \lambda_{31}}$
- Standardized solution: Correlation(y_1, y_3) = $\boldsymbol{\lambda_{11} * (1) * \lambda_{31}}$ **← std loadings**
- ONLY reason for correlation is their common factor (local independence, LI)

Items from different factors (room for misfit or mis-prediction):

- Unstandardized: Covariance(y_1, y_6) = $\boldsymbol{\lambda_{11} * Cov(F_1, F_2) * \lambda_{62}}$
- Standardized: Correlation(y_1, y_6) = $\boldsymbol{\lambda_{11} * Cor(F_1, F_2) * \lambda_{62}}$ **← std loadings**
- ONLY reason for correlation is the correlation between factors (again, LI)

Variances are additive (and will be reproduced correctly):

- $\boldsymbol{Var(y_1) = (\lambda_{11}{}^2) * Var(F_1) + Var(e_i)}$ → **note imbalance of λ^2 and e_i**

Model-Predicted Item Covariance Matrix

- Matrix equation: $\boldsymbol{\Sigma = \Lambda \Phi \Lambda^{T} + \Psi}$

$\boldsymbol{\Sigma}$= model-predicted item covariance matrix is created from:

$\boldsymbol{\Lambda}$ = item factor loadings

$\boldsymbol{\Phi}$= factor variances and covariances

$\boldsymbol{\Lambda^{T}}$= item factor loadings transposed ($\rightarrow \lambda^2$)

$\boldsymbol{\Psi}$ = item error variances (residual variances)

$$\begin{pmatrix} \sigma^2_{y1} & \sigma_{y1,y2} & \sigma_{y1,y3} & \sigma_{y1,y4} & \sigma_{y1,y5} & \sigma_{y1,y6} \\ \sigma_{y2,y1} & \sigma^2_{y2} & \sigma_{y2,y3} & \sigma_{y2,y4} & \sigma_{y2,y5} & \sigma_{y2,y6} \\ \sigma_{y3,y1} & \sigma_{y3,y2} & \sigma^2_{y3} & \sigma_{y3,y4} & \sigma_{y3,y5} & \sigma_{y3,y6} \\ \sigma_{y4,y1} & \sigma_{y4,y2} & \sigma_{y4,y3} & \sigma^2_{y4} & \sigma_{y4,y5} & \sigma_{y4,y6} \\ \sigma_{y5,y1} & \sigma_{y5,y2} & \sigma_{y5,y3} & \sigma_{y5,y4} & \sigma^2_{y5} & \sigma_{y5,y6} \\ \sigma_{y6,y1} & \sigma_{y6,y2} & \sigma_{y6,y3} & \sigma_{y6,y4} & \sigma_{y6,y5} & \sigma^2_{y6} \end{pmatrix}$$

$$= \begin{pmatrix} \lambda_{11} & 0 \\ \lambda_{21} & 0 \\ \lambda_{31} & 0 \\ 0 & \lambda_{42} \\ 0 & \lambda_{52} \\ 0 & \lambda_{62} \end{pmatrix} \begin{pmatrix} \sigma^2_{F1} & \sigma_{F1,F2} \\ \sigma_{F2,F1} & \sigma^2_{F2} \end{pmatrix} \begin{pmatrix} \lambda_{11} & \lambda_{21} & \lambda_{31} & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_{42} & \lambda_{52} & \lambda_{62} \end{pmatrix}$$

$$+ \begin{pmatrix} \sigma^2_{e1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{e2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{e3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{e4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{e5} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{e6} \end{pmatrix}$$

Model-Predicted Item Covariance Matrix

- $\Sigma = \Lambda\Phi\Lambda^{T} + \Psi$ → **Predicted Covariance Matrix**

The **loadings** control how related items from the same factor are predicted to be.

Items within Factor 1

$$\begin{pmatrix}
\lambda_{11}^2\sigma_{F1}^2 + \sigma_{e1}^2 & \lambda_{11}\sigma_{F1}^2\lambda_{21} & \lambda_{11}\sigma_{F1}^2\lambda_{31} & \lambda_{11}\sigma_{F2,F1}\lambda_{42} & \lambda_{11}\sigma_{F2,F1}\lambda_{52} & \lambda_{11}\sigma_{F2,F1}\lambda_{62} \\
\lambda_{21}\sigma_{F1}^2\lambda_{11} & \lambda_{21}^2\sigma_{F1}^2 + \sigma_{e2}^2 & \lambda_{21}\sigma_{F1}^2\lambda_{31} & \lambda_{21}\sigma_{F2,F1}\lambda_{42} & \lambda_{21}\sigma_{F2,F1}\lambda_{52} & \lambda_{21}\sigma_{F2,F1}\lambda_{62} \\
\lambda_{31}\sigma_{F1}^2\lambda_{11} & \lambda_{31}\sigma_{F1}^2\lambda_{21} & \lambda_{31}^2\sigma_{F1}^2 + \sigma_{e3}^2 & \lambda_{31}\sigma_{F2,F1}\lambda_{42} & \lambda_{31}\sigma_{F2,F1}\lambda_{52} & \lambda_{31}\sigma_{F2,F1}\lambda_{62} \\
\lambda_{42}\sigma_{F2,F1}\lambda_{11} & \lambda_{42}\sigma_{F2,F1}\lambda_{21} & \lambda_{42}\sigma_{F2,F1}\lambda_{31} & \lambda_{42}^2\sigma_{F2}^2 + \sigma_{e4}^2 & \lambda_{42}\sigma_{F2}^2\lambda_{52} & \lambda_{42}\sigma_{F2}^2\lambda_{62} \\
\lambda_{52}\sigma_{F2,F1}\lambda_{11} & \lambda_{52}\sigma_{F2,F1}\lambda_{21} & \lambda_{52}\sigma_{F2,F1}\lambda_{31} & \lambda_{52}\sigma_{F2}^2\lambda_{42} & \lambda_{52}^2\sigma_{F2}^2 + \sigma_{e5}^2 & \lambda_{52}\sigma_{F2}^2\lambda_{62} \\
\lambda_{62}\sigma_{F2,F1}\lambda_{11} & \lambda_{62}\sigma_{F2,F1}\lambda_{21} & \lambda_{62}\boxed{\sigma_{F2,F1}}\lambda_{31} & \lambda_{62}\sigma_{F2}^2\lambda_{42} & \lambda_{62}\sigma_{F2}^2\lambda_{52} & \lambda_{62}^2\sigma_{F2}^2 + \sigma_{e6}^2
\end{pmatrix}$$

Items within Factor 2

The only reason why items from different factors should be related is the **covariance** between the two factors.

The **loadings** also control how much of the item response is due to factor versus error.

Confirmatory Factor Models
(CFA: Confirmatory Factor Analysis)

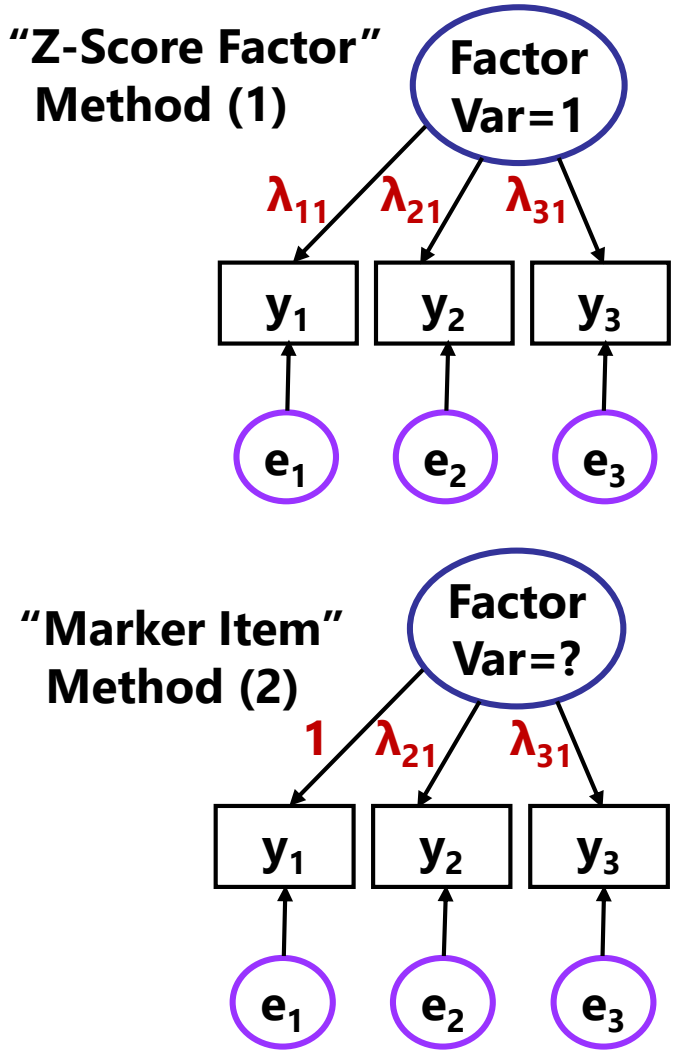
- Topics:
  - Comparison of EFA and CFA
  - CFA model parameters
  - **Two parts of CFA model identification**
  - CFA model estimation
  - CFA model fit evaluation

Two Parts of Model Identification

- **Part 1: Create a scale for each latent factor**
  - Each latent factor needs a mean and a variance
  - Necessary but not sufficient for estimating the CFA model

- **Part 2: Ensure the CFA model is estimable**
  - Data going in versus estimated parameters going out:
    - Item means → item intercepts (usually 1:1 ratio)
    - Item variances → item residual variances (usually 1:1 ratio)
    - Item covariances → item factor loadings (must have ratio ≥ 1)
  - In practice, this means the number of estimated loadings may not exceed the number of observed item covariances

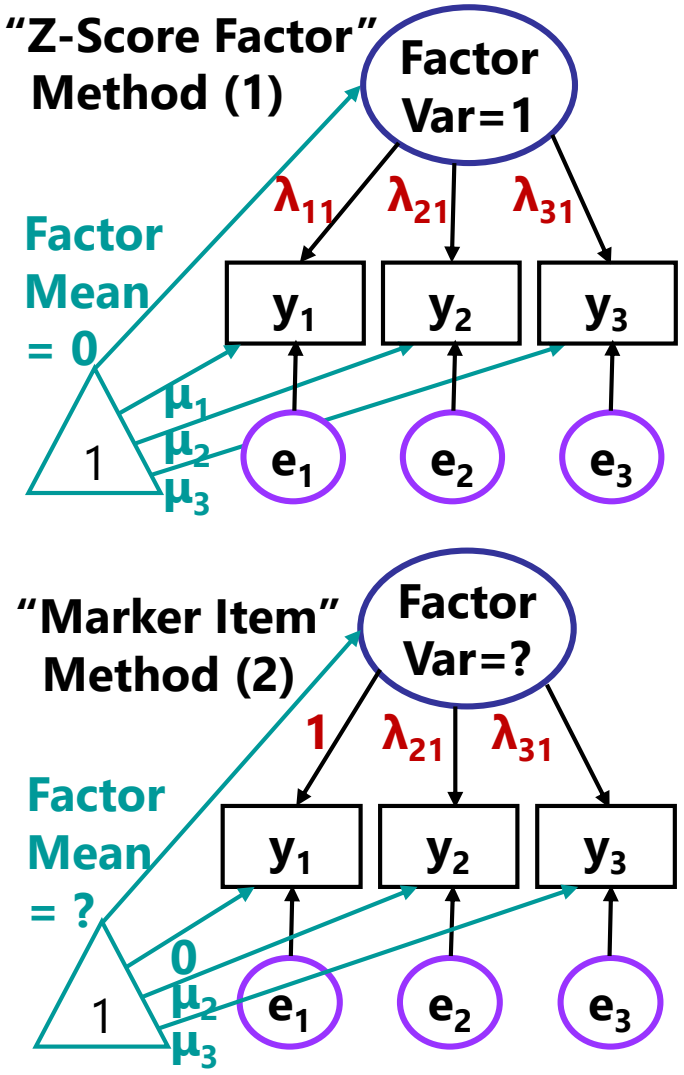
CFA Model Identification Part 1:
Create a Scale for the Latent Factor Variance



- The factor doesn't exist, so **it needs a scale** (it needs a mean and variance):
- There are two **equivalent** options to create a scale for the factor **VARIANCE**:

  - **(1) Fix factor variance to 1: "z-score"**
    - Factor is interpreted as standard z-scores
    - **Can't** be used in models with higher-order factors (coming later in this course)
  - **(2) Fix a "marker item" loading to 1**
    - Factor variance is then estimated the "reliable" part of the marker item variance
    - Std. loading = 0.9, item variance = 16? Factor variance = $(0.9^2)*16 = 12.96$
    - Can cause the model to blow up if marker item has no correlation with the factor at all

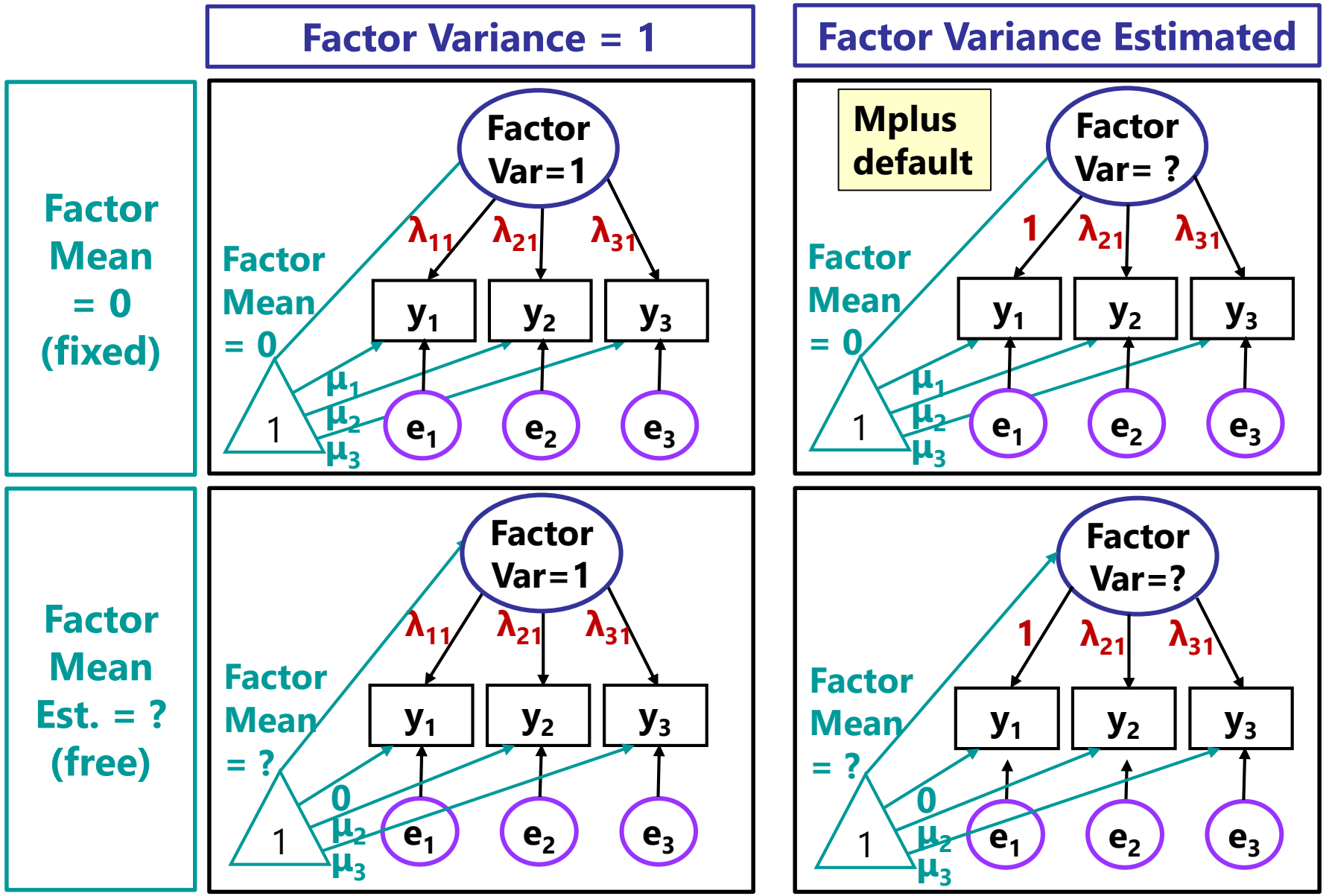
CFA Model Identification Part 1:
Create a Scale for the Latent Factor Mean



- The factor doesn't exist, so **it needs a scale** (it needs a mean and variance):
- There are two **equivalent** options to create a scale for the factor **MEAN**:

  - **(1) Fix factor mean to 0: "z-score"**
    - Factor is interpreted as standard z-scores
    - **Can** be used in models with higher-order factors (coming later in the course)
    - Item intercepts = item means
  - **(2) Fix a "marker item" intercept to 0**
    - Factor mean = mean of marker item
    - Item intercepts = expected item responses when factor = 0 (→ marker = 0)

Possible Factor Means and Variances



Part 2 of CFA Model Identification

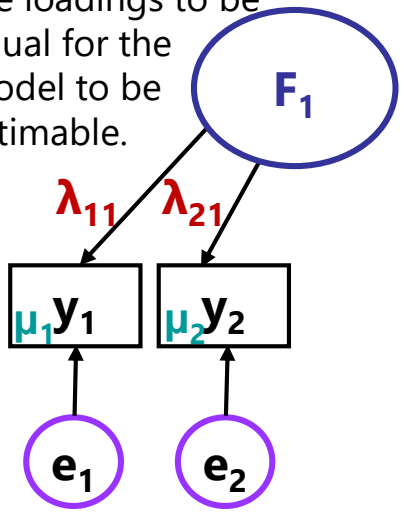
- Make sure the model is estimable; then try to reproduce observed item covariance matrix using as few estimated parameters as possible
  - (Robust) Maximum likelihood used to estimate model parameters
    - **Measurement Model**: Item factor loadings, item intercepts, item error variances
    - **Structural Model**: Factor variances, factor covariances, factor means
  - Global model fit is evaluated as difference between model-predicted and data-observed covariance matrix (but only covariances usually contribute to misfit)

- How many possible parameters can you estimate: what is total DF?
  - **Total DF =** $\frac{v(v+1)}{2} + v$ where v **is the # items** (NOT people, like usual)
    - Total DF = number of item means, variances, and covariances
    - e.g., if $v = 4$ items, then $\text{DF} = \frac{4(4+1)}{2} + 4 = 14$
  - **Model DF** = data input − model output
  - **Model DF** = # possible parameters − # estimated parameters

Under-Identified Factor: 2 Items

- Model is **under-identified** if there are more unknown parameters then item variances, covariances, and means with which to estimate them
  - Model **cannot be estimated** because there are an infinite number of different parameter estimates that would result in the same and perfect fit
  - Example: x + y = 7 ??

You'd have to constrain the loadings to be equal for the model to be estimable.



In other words, the assumptions required to calculate two-score reliability in CTT are the result of model under-identification.

Total possible **DF** = unique pieces of data = **5**

0 factor variances		**1 factor variance**
0 factor means		**1 factor mean**
2 item loadings	**OR**	**1 item loading**
2 item intercepts		**1 item intercept**
2 error variances		**2 error variances**

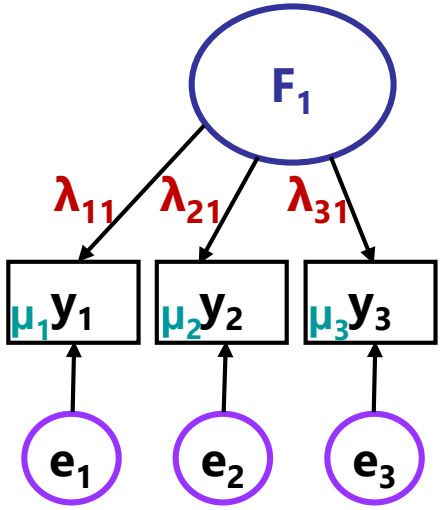
DF = 5 – 6 = – 1

If $cor(y_1, y_2) = .64$, then:

$\lambda_{11} = .800, \lambda_{21} = .800$??
$\lambda_{11} = .900, \lambda_{21} = .711$??
$\lambda_{11} = .750, \lambda_{21} = .853$??

Just-Identified Factor: 3 Items

- Model is **just-identified** if there are as many unknown parameters as item variances, covariances, and means with which to estimate them
  - The model is estimable, so the parameter estimates have a unique solution
  - But those parameters will **perfectly reproduce** the data-observed covariance matrix, so **model fit is not testable**—it's just a re-arrangement of the data
  - Example: Solve x + y = 7, 3x – y = 1



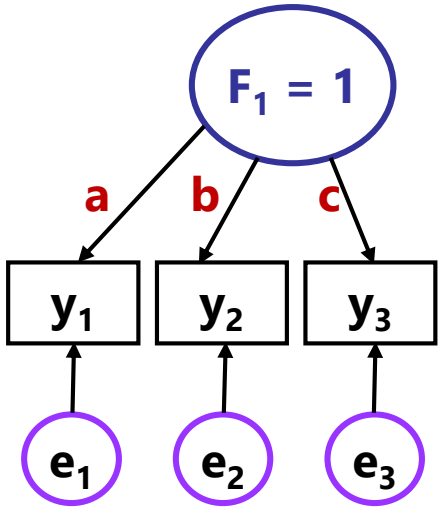
Total possible **DF** = unique pieces of data = **9**

0 factor variances		**1 factor variance**
0 factor means		**1 factor mean**
3 item loadings	**OR**	**2 item loadings**
3 item intercepts		**2 item intercepts**
3 error variances		**3 error variances**

DF = 9 – 9 = 0

Not really a model—more like a description

Example: Solving a Just-Identified Model

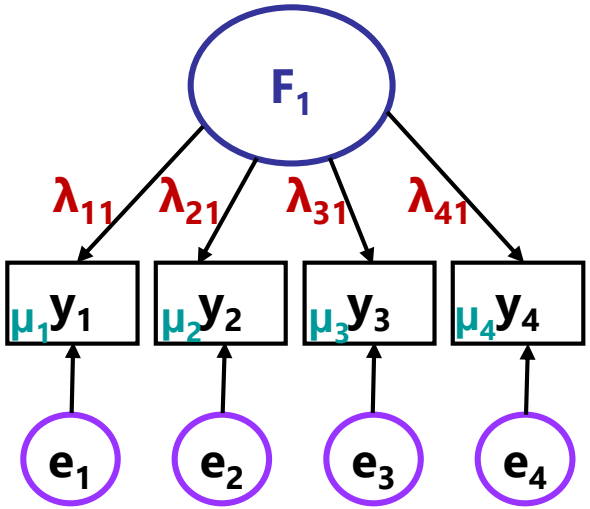


	y_1	y_2	y_3
y_1	1.00		
y_2	.595	1.00	
y_3	.448	.544	1.00

- Step 1: $ab = .595$
  $ac = .448$
  $bc = .544$
- Step 2: $b = .595/a$
  $c = .488/a$
  $(.595/a)(.448/a) = .544$
- Step 3: $.26656/a^2 = .544$
  $a = .70$
- Step 4: $.70b = .595 \quad b = .85$
  $.70c = .448 \quad c = .64$
- Step 5: $Var(e_1) = 1 - a^2 = .51$

Over-Identified Factor: 4+ Items

- Model is **over-identified** if there are fewer unknown parameters than item variances, covariances, and means with which to estimate them
  - The model is estimable, so the parameter estimates have a unique solution
  - But now the parameters will NOT perfectly reproduce the observed matrix → **if DF > 0, we can test model fit!**



Total possible **DF** = unique pieces of data = **14**

0 factor variances		**1 factor variance**
0 factor means		**1 factor mean**
4 item loadings	**OR**	**3 item loadings**
4 item intercepts		**3 item intercepts**
4 error variances		**4 error variances**

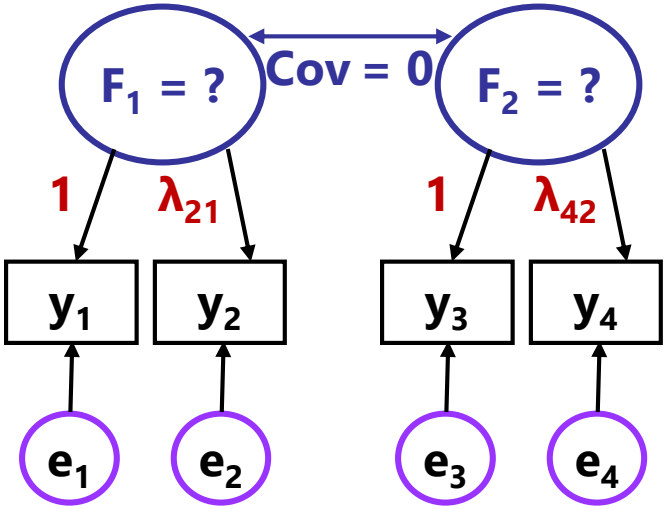
DF = 14 – 12 = 2

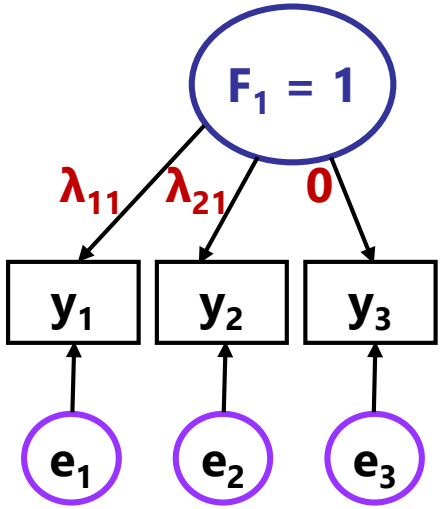
Model fit: Did we do a "good enough" job reproducing the item covariance matrix with 2 fewer parameters than it was possible to use?

Oops: Empirical Under-Identification

- Did your model blow up (errors instead of output)? Double-check:
  - Part 1: Make sure each factor has a scale: a mean and a variance
  - Part 2: Make sure you aren't estimating more parameters than you have DF

- Sometimes you can set up your model correctly and it will STILL blow up because of **empirical under-identification**
  - It's not you; **it's your data**—here are two examples of when these models should have been identified, but weren't because of an unexpected 0 relationship





That Other Kind of Measurement Model…

Remember the difference between principal components and factor analysis in terms of 'types' of items?

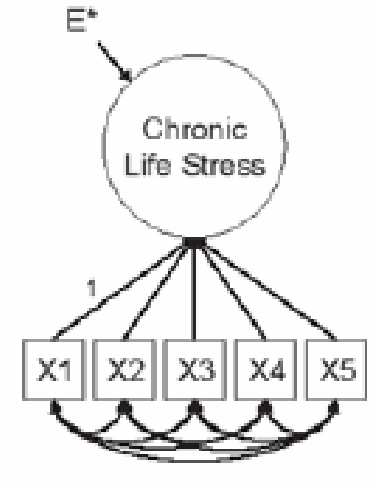
Factor Model:

- Composed of "Reflective" or "Effects" items
- Factor is thought to **cause** observed item responses
- Items should be correlated
- **Is identified** with 3+ items (fit is testable with 4+ items)

Component Model:

- Composed of "Formative" or "Emergent" or "Cause" items
- Component is **result** of observed item responses
- Items may not be correlated
- **Will not be identified** no matter how many items *without additional variables in the model*

Formative (Component) Models
(see Brown 2015 ch. 8 pp. 322-331)

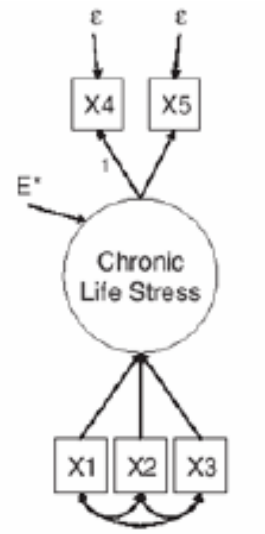


Model A Parameters:

4 factor loadings/regression paths
1 factor disturbance (variance left over)
10 item correlations
5 item variances
5 item means
$DF = 20 - 25 = -5$

Not identified

Formative measurement models are not identified without including other outcomes or predictors of the formative latent factor



Model C Parameters:

4 factor loadings/regression paths
1 factor disturbance (variance left over)
3 item correlations
5 item variances/error variances
5 item means/intercepts
$DF = 20 - 18 = 2$

Identified

Model C has both formative and reflective indicators—the latter might also be "outcomes"

Intermediate Summary: CFA

- CFA is a **linear model** in which continuous observed item responses are predicted from latent factors (traits) and error
  - Goal is to reproduce observed **item covariance matrix** using parameters (item intercepts, loadings, and error variances; factor variances/covariances)
  - Factor model makes specific testable mathematical predictions about how item responses should relate to each other: **loadings predict covariances**
  - Need at least 3 items per latent factor for the model to be identified; need **at least 4 items per latent factor for model fit to be testable**

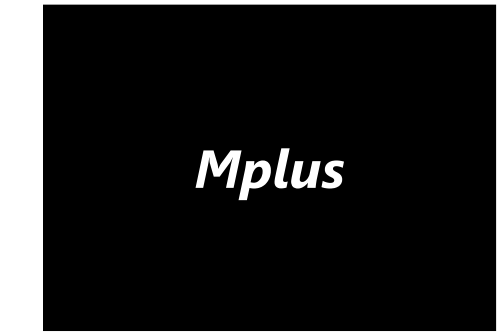
- CFA framework offers significant advantages over CTT by offering the potential for comparability across samples, groups, and time
  - CTT: No separation of observed item responses from true score
    - Sum across items = true score; item properties belong to that sample only
  - CFA: Latent factor is estimated *separately* from item responses
    - Separates interpretation of person trait levels from specific items given
    - Separates interpretation of item properties from specific persons in sample

Confirmatory Factor Models
(CFA: Confirmatory Factor Analysis)

- Topics:
  - Comparison of EFA and CFA
  - CFA model parameters
  - Two parts of CFA model identification
  - **CFA model estimation**
  - CFA model fit evaluation

Where the Answers Come From: The Big Picture of ML Estimation

ESTIMATOR = Robust Maximum Likelihood;



... answers ...

Any questions?

What all do we have to estimate?

- For example, a model with <u>two correlated factors for $v = 6$ items:</u>
  - F_1 measured by items 1,2,3; F_2 measured by items 4,5,6
  - If we fix both factors to have mean=0 and variance=1, then we need:
    6 intercepts (μ_i) + 6 factor loadings (λ_i) + 6 error variances ($\sigma^2_{e_i}$)
    + 1 factor covariance [$Cov(F_1, F_2)$]= 19 total parameters

- **Item parameters** are **FIXED effects** → inference about specific item
  - It's ok if missing data leads to different numbers of total items across persons

- What about the all the individual person **factor scores**?
  - The individual factor scores are NOT part of the model—in other words, factor scores are modeled as **RANDOM effects** assumed to be multivariate normal
  - So we need the **factor means, variances, and covariances** as sufficient statistics, but **we don't need** the factor scores for the **individual respondents**

The End Goals of Maximum Likelihood (ML) Estimation

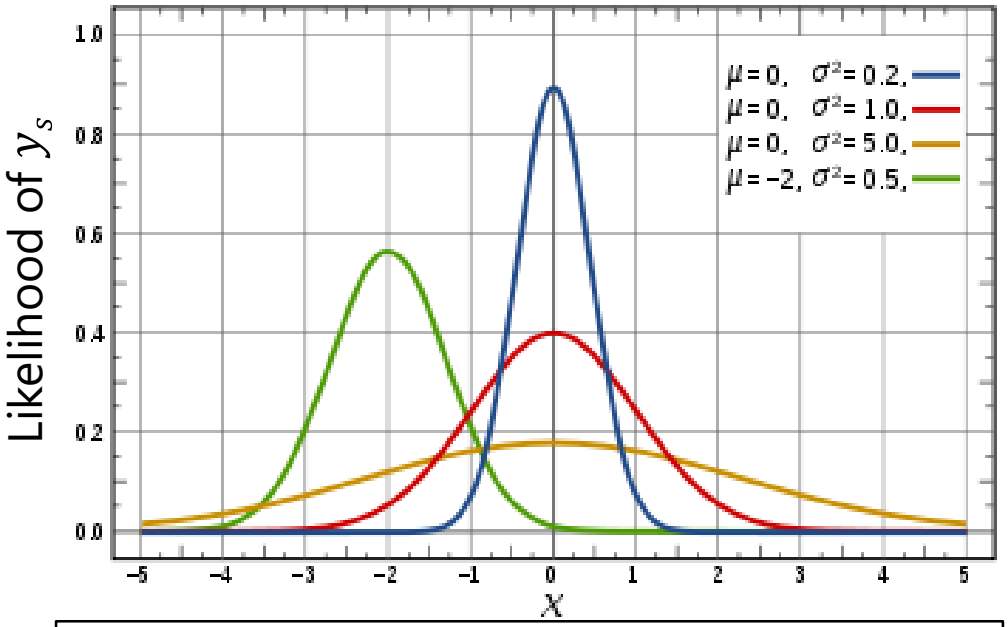
1. Obtain "most likely" values for each unknown parameter in our model (intercepts, loadings, error variances, factor means, factor variances, factor covariances) → the answers → **the estimates**

2. Obtain some kind of index as to how likely each parameter value actually is (i.e., "really likely" or pretty much just a guess?)
   → **the standard error (SE) of the estimates (smaller is better)**

3. Obtain some kind of index as to how well the model we've specified actually describes the data → **the model fit indices**

How does all this happen? The magic of multivariate normal…
(but let's start with univariate normal first)

Univariate Normal Distribution



Univariate Normal PDF:

$$f(y_s) = \frac{1}{\sqrt{2\pi\sigma_e^2}} * \exp\left[-\frac{1}{2} * \frac{(y_s - y_s)^2}{\sigma_e^2} \right]$$

Sum over persons for log of $f(y_i)$= Model Log-Likelihood → Model Fit

- This PDF tells us how **likely** (i.e., **tall**) any value of $\boldsymbol{y_s}$ is given two things:
  - Conditional mean $\boldsymbol{\hat{y}_s}$
  - Residual variance $\boldsymbol{\sigma_e^2}$
- We can see this work using the NORMDIST function in excel!
  - Easiest for **empty** model:

$$\boldsymbol{y_s = \beta_0 + e_s}$$

- We can check our math via software using ML!

Multivariate Normal for $\mathbf{Y}_s$: all $v = 6$ item responses from person s

Univariate Normal PDF: $f(y_s) = \left(2\pi\sigma_e^2\right)^{-1/2} * \exp\left[-\frac{1}{2} * \left(y_s - y_s\right)\left(\sigma_e^2\right)^{-1}\left(y_s - y_s\right)\right]$

Multivariate Normal PDF: $f(\mathbf{Y}_s) = (2\pi)^{-v_s/2} * |\boldsymbol{\Sigma}|^{-1/2} * \exp\left[-\frac{1}{2} * \left(\mathbf{Y}_s - \boldsymbol{\mu}\right)^T \left(\boldsymbol{\Sigma}_s\right)^{-1}\left(\mathbf{Y}_s - \boldsymbol{\mu}\right)\right]$

- In our CFA model, the only fixed effects that predict the 6 item responses in $\boldsymbol{Y}_s$ are the item intercepts (now $v = 6$ of them in the vector $\boldsymbol{\mu}$)
- CFA model also gives us the **predicted** variance and covariance matrix across the items ($\boldsymbol{\Sigma}$), assumed the same across persons:
  - In matrices: $\boldsymbol{\Lambda}$ = loadings, $\boldsymbol{\Phi}$ = factor variances and covariances, $\boldsymbol{\Psi}$ = item error variances
  - Variance of Item i: $\boldsymbol{Var(y_i)} = \boldsymbol{\lambda_i^2} * \boldsymbol{Var(F)} + \boldsymbol{Var(e_i)}$
  - Covariance of items on same factor: $\boldsymbol{Cov(y_1, y_2)} = \boldsymbol{\lambda_{11}} * \boldsymbol{Var(F_1)} * \boldsymbol{\lambda_{21}}$
  - Covariance of items on different factors: $\boldsymbol{Cov(y_1, y_6)} = \boldsymbol{\lambda_{11}} * \boldsymbol{Cov(F_1, F_2)} * \boldsymbol{\lambda_{62}}$

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda\Phi\Lambda}^T + \boldsymbol{\Psi}$$

- Uses $|\boldsymbol{\Sigma}|$ = determinant of $\boldsymbol{\Sigma}$ = summary of *non-redundant* info
- $\boldsymbol{\Sigma}_s^{-1}$ → matrix inverse → like dividing (so can't be 0 or negative)

Now Try Some Possible Answers...
(e.g., for those 19 parameters in this example)

- Plug predictions into **log-likelihood** function, sum over persons:

$$\text{Model } (H_0) \text{ Likelihood: } L = \prod_{s=1}^{N} \left\{ (2\pi)^{-v_s/2} * |\boldsymbol{\Sigma}|^{-1/2} * \exp\left[-\frac{1}{2} (\mathbf{Y}_s - \boldsymbol{\mu})^T (\boldsymbol{\Sigma}_s)^{-1} (\mathbf{Y}_i - \boldsymbol{\mu}) \right] \right\}$$

$$\text{Model } (H_0) \text{ Log Likelihood: } LL = \sum_{s=1}^{N} \left\{ \left[-\frac{v_s}{2} \log(2\pi) \right] + \left[-\frac{1}{2} \log|\boldsymbol{\Sigma}| \right] + \left[-\frac{1}{2} (\mathbf{Y}_s - \boldsymbol{\mu})^T (\boldsymbol{\Sigma}_s)^{-1} (\mathbf{Y}_i - \boldsymbol{\mu}) \right] \right\}$$

- Try one set of possible parameter values, compute LL (total **height**)
- Try another possible set, compute revised LL....
  - Different algorithms are used to decide which values to try given that each parameter has its own likelihood distribution → like an uncharted mountain
  - Calculus helps the program scale this multidimensional mountain
    - At the top, all first partial derivatives (linear slopes at that point) ≈ 0
    - *Positive* first partial derivative? Too *low*, try again. *Negative*? Too *high*.
    - Matrix of partial first derivatives = "score function" = "gradient"

End Goals 1 and 2: Model Estimates and SEs

- Process terminates (the model "**converges**") when the next set of tried parameter values don't improve the LL very much…
  - e.g., Mplus default convergence criteria for this H_0 Model LL = .00005 (other values are used for different estimation problems—see manual)
  - Those are the values for our model parameters that, relative to the other possible values tried, are "most likely" → **Model (H_0) LL** and **estimates**

- But we also need to know how trustworthy those estimates are…
  - **Precision** is indexed by the steepness of the multidimensional mountain, where steepness → more negative partial second derivatives
  - Matrix of partial second derivatives = "Hessian matrix"
  - Hessian matrix $*$ -1 = "information matrix" $\text{Each parameter SE} = \frac{1}{\sqrt{\text{information}}}$
  - So steeper function = more information = more precision = **smaller SE**

End Goal #3: How well do the model predictions match the data?

- Use your model LL_{H_0} from predicting $\mathbf{\Sigma}$ → so how good is it?
- Get the best possible LL_{H_1} if we used the real data ($\mathbf{S}$) instead:

$$\text{Saturated Model } (H_1) \text{ Log Likelihood: } LL = \sum_{s=1}^{N}\left\{\left[-\frac{v_s}{2}\log(2\pi)\right]+\left[-\frac{1}{2}\log|\mathbf{S}|\right]+\left[-\frac{1}{2}v_s\right]\right\}$$

- Compute the **ML fitting function** that indexes how far off the model predictions are from the real data → χ^2:

$$\text{ML Fitting Function: } F_{ML} = \frac{LL_{H_1 data}}{N} - \frac{LL_{H_0 model}}{N} \quad \text{where} \quad \chi^2 = 2 * N * F_{ML}$$

- Combining and re-arranging the terms in LL for H_0 and H_1 yields this common (complete data) expression for the ML fitting function:

$$F_{ML} = \frac{1}{2}\sum_{s=1}^{N}\left\{\underbrace{\log|\mathbf{\Sigma}| - \log|\mathbf{S}|}_{\text{how far off}} + \underbrace{\text{trace}\left[(\mathbf{\Sigma})^{-1}\mathbf{S}\right] - v_s}_{\text{correction for \#items}}\right\} / N$$

If the model fits perfectly, both parts should be 0.

What about item non-normality?

- The use of this multivariate normal ML function assumes:
  - Persons and items are conditionally independent
  - Item responses can be missing at random (MAR; ignorable)
  - Factor scores ($\boldsymbol{F_s}$) have a multivariate normal distribution
  - Item residuals ($\boldsymbol{e_{is}}$) have a multivariate normal distribution
  - So in this case, the original item responses should have a multivariate normal distribution, too (given prediction by normal $\boldsymbol{F_s}$+ normal $\boldsymbol{e_{is}}$)

- Impact of non-normality of item responses:
  - Linear model predicting item response from factor may not work well
    - if $\boldsymbol{y_{is}}$ is not really continuous, the slope needs to shut off at its boundaries
  - SEs and χ^2-based model fit statistics will be incorrect
  - Three fixes: **1. Robust ML** (or 2. transform the data, or 3. use a different kind of factor model → IRT/IFA… stay tuned)

Robust ML for Non-Normality: MLR

- **MLR in Mplus**: ≈ Yuan-Bentler T_2 (permits MCAR or MAR missing data)
  - Still **a linear model** between the item responses and latent factor, so the parameter estimates will be the same as in regular ML

- Adjusts **fit statistics** using an estimated **scaling factor** → for kurtosis:
  - Scaling factor = 1.000 = perfectly multivariate normal → same as regular ML!
  - Scaling factor > 1.000 = leptokurtosis (too-fat tails; fixes too-big χ^2)
  - Scaling factor < 1.000 = platykurtosis (too-thin tails; fixes too-small χ^2)

- **SEs** computed with Huber-White 'sandwich' estimator → uses an information matrix from the variance of the partial first derivatives to correct the information matrix from the partial second derivatives
  - Leptokurtosis (too-fat tails) → increases information; fixes too small SEs
  - Platykurtosis (too-thin tails) → lowers information; fixes too big SEs

- Because MLR simplifies to ML if the item responses actually are multivariate normally distributed, **we will use MLR as our default estimator for CFA**

SEM Estimation in STATA v. 16

- Although SEMs can be estimated using STATA, it appears there are (currently) fewer combinations allowed than in Mplus:
  - SEM option method( ) allows estimator choices:
    - ML = regular limited-information ML (no missing data allowed)
    - MLMV = full-information ML (MCAR or MAR missing data)
    - ADF = asymptotic distribution free (requires huge N for stable estimation)
  - SEM option vce( ) allows robust standard error choices:
    - Robust = Huber-White 'sandwich' version (as given by MLR in Mplus)
      - Can be used with MLMV for missing data to adjust parameter SEs
      - No scaling correction factor given to compute fit statistics (none given)
    - Sbentler = Satorra–Bentler version (MLM in Mplus)
      - Can only be used with ML estimation method (so no missing data allowed)

- Because there appears to be no combination that allows missing data + robust estimation of fit statistics and SEs, I'm not going to provide STATA SEM example code…

Confirmatory Factor Models
(CFA: Confirmatory Factor Analysis)

- Topics:
  - Comparison of EFA and CFA
  - CFA model parameters
  - Two parts of CFA model identification
  - CFA model estimation
  - **CFA model fit evaluation**

The Big Picture of Model Fit

- Aspects of the observed data to be predicted *(assuming a z-score metric for the factors for simplicity)*:
- CFA model equation: $\boldsymbol{y_{is} = \mu_i + \lambda_i F_s + e_{is}}$
  - **Mean** per item: **Captured** by intercept $\boldsymbol{\mu_i}$ per item
    - Not a source of misfit (unless constraints are applied on the intercepts)
  - **Variance** per item: **Captured** by weighted factor + error
    - $\boldsymbol{Var(y_i) = \lambda_i^2 * Var(F) + Var(e_i)}$ → output given as $\boldsymbol{\lambda_i}$ and $\boldsymbol{Var(e_i)}$
    - Factor and error variances are additive → not a source of misfit (whatever $\boldsymbol{F_s}$ doesn't get, $\boldsymbol{e_{is}}$ picks up to get back to total $\boldsymbol{y_i}$ variance)
  - **Covariance** among items: **Predicted** via factor loadings $\boldsymbol{\lambda_i}$
    - Loadings (multiplied) predict what observed covariance should be… but they may not be right → **THE PRIMARY SOURCE OF MISFIT**

Baselines for Assessing Fit in CFA (Item means all saturated in both)

Independence (Null) Model

$$\begin{pmatrix} \sigma^2_{y1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{y2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{y3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{y4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{y5} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{y6} \end{pmatrix}$$

Saturated (Unstructured; H_1) Model

$$\begin{pmatrix} \sigma^2_{y1} & \sigma_{y1,y2} & \sigma_{y1,y3} & \sigma_{y1,y4} & \sigma_{y1,y5} & \sigma_{y1,y6} \\ \sigma_{y2,y1} & \sigma^2_{y2} & \sigma_{y2,y3} & \sigma_{y2,y4} & \sigma_{y2,y5} & \sigma_{y2,y6} \\ \sigma_{y3,y1} & \sigma_{y3,y2} & \sigma^2_{y3} & \sigma_{y3,y4} & \sigma_{y3,y5} & \sigma_{y3,y6} \\ \sigma_{y4,y1} & \sigma_{y4,y2} & \sigma_{y4,y3} & \sigma^2_{y4} & \sigma_{y4,y5} & \sigma_{y4,y6} \\ \sigma_{y5,y1} & \sigma_{y5,y2} & \sigma_{y5,y3} & \sigma_{y5,y4} & \sigma^2_{y5} & \sigma_{y5,y6} \\ \sigma_{y6,y1} & \sigma_{y6,y2} & \sigma_{y6,y3} & \sigma_{y6,y4} & \sigma_{y6,y5} & \sigma^2_{y6} \end{pmatrix}$$

Parsimony ← → **Good fit**

All item means and variances estimated separately;
no covariances are estimated

$$y_{is} = \mu_i + \lambda_i F_s + e_{is}$$

All item means and variances estimated separately;
all covariances estimated are separately now, too

Your CFA model "H_0" will go here

Baseline model comparisons are already given in Mplus output…

```
MODEL FIT INFORMATION (Abbreviated)

Number of Free Parameters                       18

Loglikelihood
     H0 Value                          -11536.404
     H0 Scaling Correction Factor           1.4158
           for MLR

     H1 Value                          -11322.435
     H1 Scaling Correction Factor           1.4073
           for MLR

Chi-Square Test of Model Fit
     Value                             307.799*
     Degrees of Freedom                      9
     P-Value                            0.0000
     Scaling Correction Factor              1.3903
           for MLR

Chi-Square Test of Model Fit for the Baseline Model
     Value                            1128.693
     Degrees of Freedom                     15
     P-Value                            0.0000
```

$\boldsymbol{H_1}$ Saturated (Unstructured) Model

$$\begin{pmatrix} \sigma^2_{y1} & \sigma_{y1,y2} & \sigma_{y1,y3} & \sigma_{y1,y4} & \sigma_{y1,y5} & \sigma_{y1,y6} \\ \sigma_{y2,y1} & \sigma^2_{y2} & \sigma_{y2,y3} & \sigma_{y2,y4} & \sigma_{y2,y5} & \sigma_{y2,y6} \\ \sigma_{y3,y1} & \sigma_{y3,y2} & \sigma^2_{y3} & \sigma_{y3,y4} & \sigma_{y3,y5} & \sigma_{y3,y6} \\ \sigma_{y4,y1} & \sigma_{y4,y2} & \sigma_{y4,y3} & \sigma^2_{y4} & \sigma_{y4,y5} & \sigma_{y4,y6} \\ \sigma_{y5,y1} & \sigma_{y5,y2} & \sigma_{y5,y3} & \sigma_{y5,y4} & \sigma^2_{y5} & \sigma_{y5,y6} \\ \sigma_{y6,y1} & \sigma_{y6,y2} & \sigma_{y6,y3} & \sigma_{y6,y4} & \sigma_{y6,y5} & \sigma^2_{y6} \end{pmatrix}$$

"Model fit" χ^2 is from a $-2\Delta LL$ test of your chosen $\boldsymbol{H_0}$ model vs. saturated $\boldsymbol{H_1}$ model

Independence (Null) Model

$$\begin{pmatrix} \sigma^2_{y1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma^2_{y2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{y3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma^2_{y4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma^2_{y5} & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma^2_{y6} \end{pmatrix}$$

"Baseline model" χ^2 is from $-2\Delta LL$ test of null model vs. saturated $\boldsymbol{H_1}$ model (ignore)

4 Steps in Assessing Model Fit

1. **Global model fit**
   - *Does the model "work" overall: Does it reproduce the observed data?*

2. Local model fit
   - *Are there any more specific problems (that cause global misfit)?*

3. Inspection of model parameters
   - *Are the estimates, SEs, and the item responses they predict plausible?*

4. Reliability and information per item
   - *How "good" is my test? How useful is each item?*

Step 1: Indices of Global Model Fit

- Primary fit index: obtained model $\boldsymbol{\chi^2 = 2 * N * F_{ML}}$
  - χ^2 is evaluated based on model DF (# parameters left over)
  - Tests null hypothesis that $\boldsymbol{\Sigma = S}$ (that model = data is perfect), so **significance is bad** (i.e., smaller χ^2, bigger *p*-value is better)
    - Is LRT ($-2\Delta LL$) of your H_0 model versus saturated best H_1 model
    - Btw, don't use "ratio rules" like $\chi^2/\mathrm{DF} > 2$ or $\chi^2/\mathrm{DF} > 3$
  - Just using χ^2 to index model fit is usually insufficient, however:
    - χ^2 depends largely on sample size (is overpowered with large N)
    - Is "unreasonable" null hypothesis (perfect fit, really??)
    - Btw, χ^2 is only possible given balanced data (as typical for CFA)

- Because of these issues, additional fit indices are usually used in conjunction with the χ^2 test (that function like effect sizes)
  - **Absolute** Fit Indices (besides χ^2)—relative to "**saturated**" best model
  - **Comparative** (Incremental) Fit Indices—relative to "**null**" worst model
  - Cite a reference for any cut-offs you use… it's now more complicated!

Step 1: Indices of Global Model Fit

- Absolute Fit: **SRMR**
  - **Standardized Root Mean Square Residual**
  - Get difference of standardized $\mathbf{S} - \mathbf{\Sigma}$ → "residual" (leftover) matrix
  - Sum the squared residuals of the predicted correlation matrix across items, divide by number of matrix elements, then take square root:
  - $SRMR = \sqrt{\frac{2\sum_{i=1}^{I}\sum_{j=1}^{J}\left[\frac{s_{ij}-\sigma_{ij}}{s_{ii}s_{jj}}\right]^2}{I(I-1)}}$
  - Ranges from 0 to 1: **smaller is better**
  - Convention: ".08 or less" → good fit

- Less common variant: **RMR** (**Root Mean Square Residual**)

Step 1: Indices of Global Model Fit

Parsimony-Corrected: **RMSEA**

- **Root Mean Square Error of Approximation**
- Relies on a "non-centrality parameter" (NCP) for T (target H_0)
  - NCP indexes how far off your model is → adjusted χ^2 distribution
  - $NCP_T = \max(\chi^2_T - DF_T, 0)$ → scaled discrepancy $d_T = NCP_T/N$
  - $\text{RMSEA} = \sqrt{\frac{\max(\chi^2_T - DF_T, 0)}{DF_T * N}} = \sqrt{\frac{d}{DF_T}}$ → how far off per DF left

- RMSEA ranges from 0 to 1; **smaller is better**
  - Conventions: < .05 or .06 = "good", .05 to .08 = "adequate"
  - In addition to point estimate, get 90% confidence interval (CI)
  - RMSEA penalizes for model complexity—it's discrepancy in fit per DF left in model (but not sensitive to N, although CI can be)
  - Also get test of "close fit": null hypothesis that $\text{RMSEA} \leq .05$

Step 1: Indices of Global Model Fit

Comparative (Incremental) Fit Indices (bigger is better)

- Fit evaluated relative to "null" (independence) model of 0 covariances
- Relative to that, your model fit should be great!
- Conventions: > .90 = "adequate", > .95 = "good"

- **CFI: Comparative Fit Index (ranges from 0 to 1)**
  - Also based on idea of NCP ($\chi^2_T - \mathrm{DF}_T$)
  - $CFI = \frac{\max(\chi^2_N - DF_N, 0) - \max(\chi^2_T - DF_T, 0)}{\max(\chi^2_N - DF_N, 0)}$

T = target model (H_0)
N = null model (no covariances)

- **TLI: Tucker-Lewis Index** (= **Non-Normed Fit Index**)
  - $TLI = \frac{\frac{\chi^2_N}{DF_N} - \frac{\chi^2_T}{DF_N}}{\frac{\chi^2_N}{DF_N} - 1}$ (so can go negative or > 1)

4 Steps in Model Evaluation

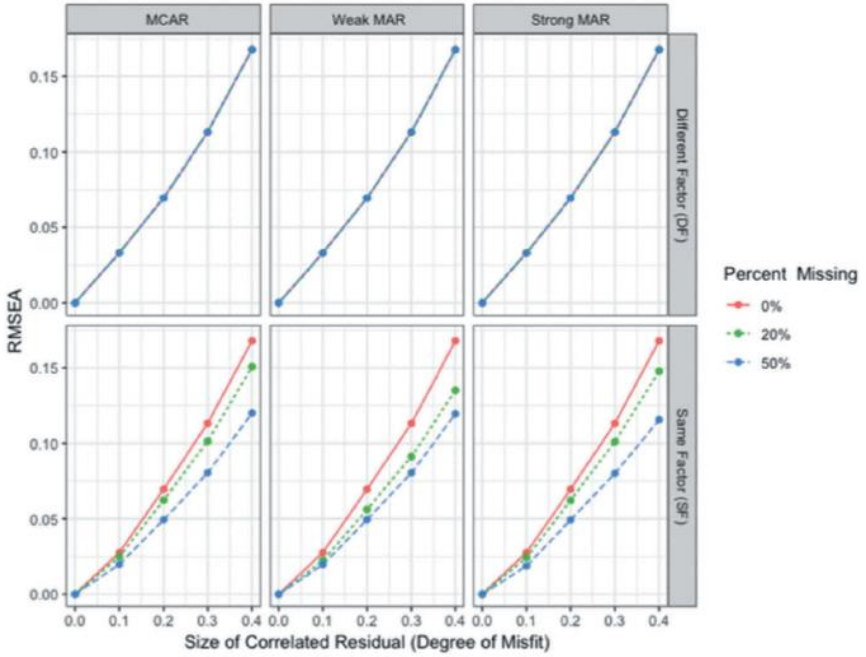
1. **Assess global model fit (summary)**
   - Recall that item intercepts, factor means, and variances are just-identified → *misfit comes from mis-predicted covariances*
   - χ^2 is sensitive to large N, so pick at least one global fit index from each class; hope they agree (e.g., CFI, RMSEA) that fit is "good"

- Conventions of "good" absolute model fit largely stem from simulation studies reported in Hu & Bentler (1999)
  - Been cited 68,000+ times! But no one study can cover everything…
    - Held indicator reliability relatively constant: standardized loadings .70-.80
    - Small-ish model of 15 indicators measuring 3 correlated factors
    - Complete data, generated using perfectly multivariate normal indicators
  - Research now suggests standards for what is "good" model fit will vary significantly as a function of these unaddressed features…
    - Here are examples from recent studies (on your reading list or reference given)

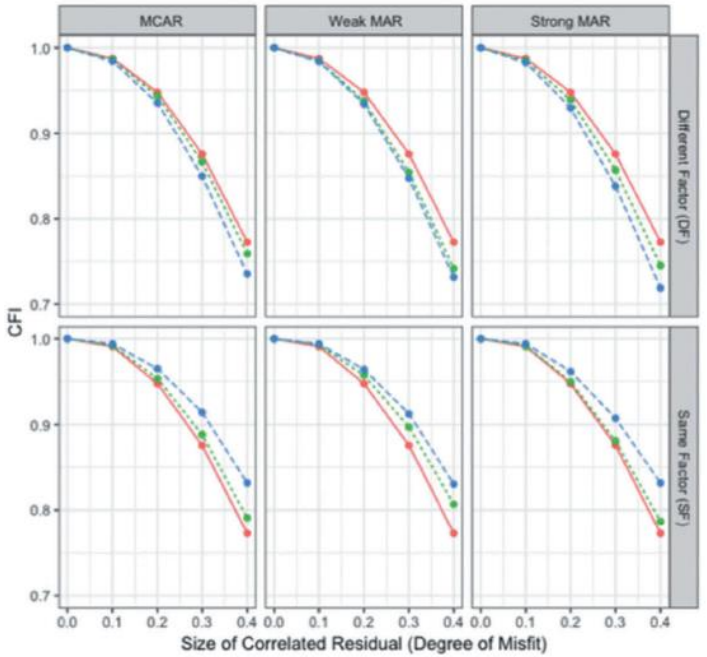
Good Fit is Easier with Missing Data

- Zhang & Savalei (2020): Cases that don't have the indicators with mis-specification will contribute better fit (higher LL)!
  - Figure 4: Fit when a correlated residual (error) of increasing size is ignored

RMSEA gets less worse with more mis-specification when missing the indicators that have ignored correlated residuals



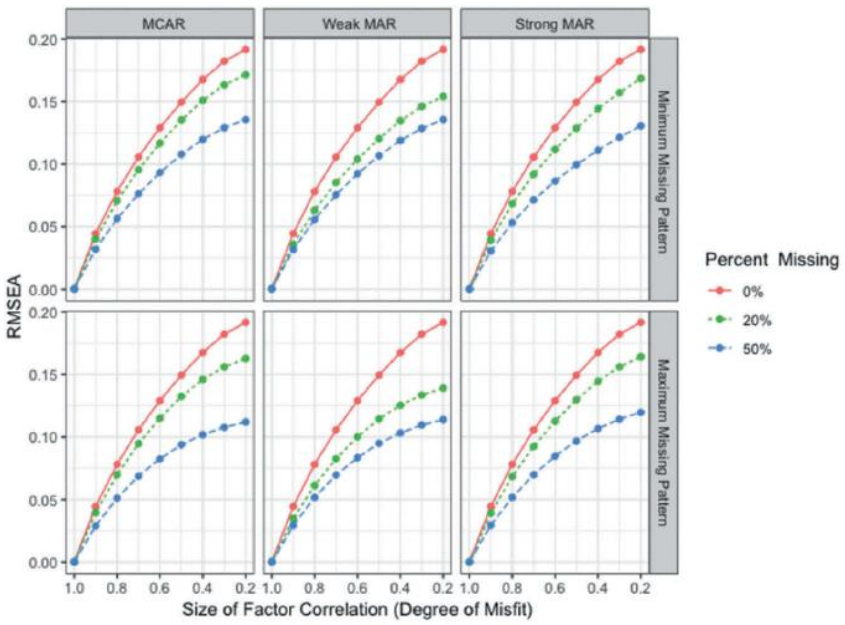
CFI also gets less worse with more mis-specification when missing the indicators that have ignored correlated residuals



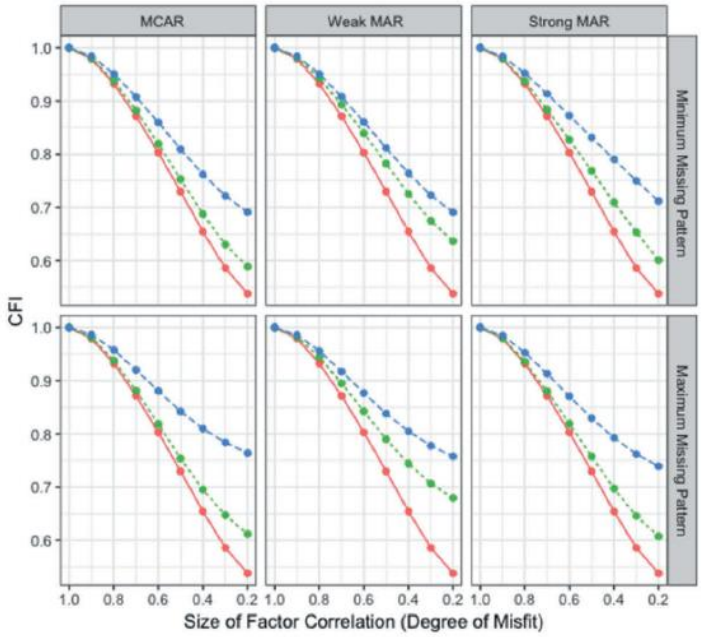
Good Fit is Easier with Missing Data

- Zhang & Savalei (2020): Same problem when misfit is due to structural mis-specification (i.e., not localized to indicator errors)
  - Figure 5: Fit when one factor is specified instead of two correlated factors

RMSEA gets less worse with more mis-specification with greater amounts of missing indicators with more relevance



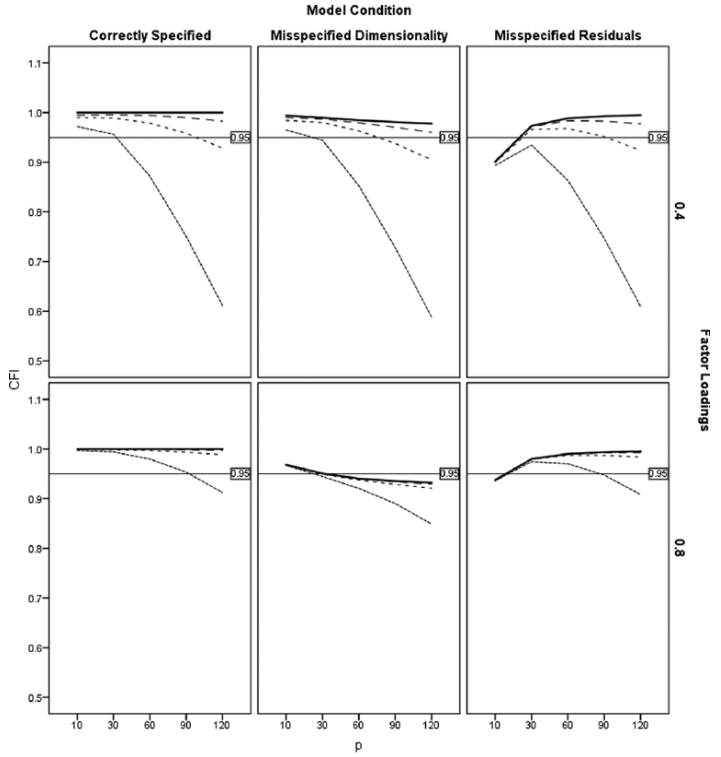
CFI also gets less worse with more mis-specification with greater amounts of missing indicators with more relevance



Good Fit by Number of Indicators…

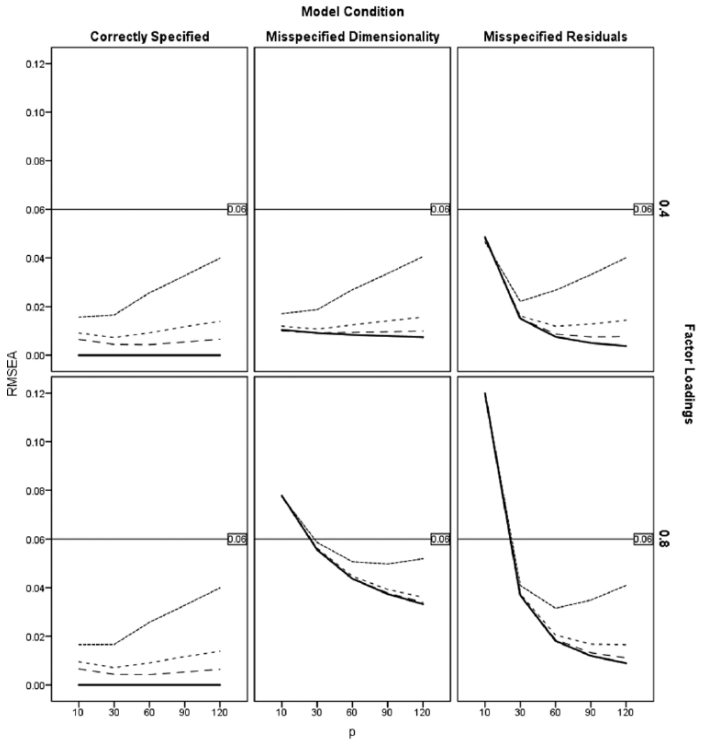
- …It's complicated… see Shi, Lee, & Maydeu-Olivares (2019)
  - Figure 1 and 3: Effects of # indicators for N=200, 500, 1000, and population

CFI gets more worse with more indicators, smaller N, and low reliability (for $\lambda = .40$, CFI is much more variable)



Correct model: RMSEA gets a little worse (still ok) with more indicators and smaller N

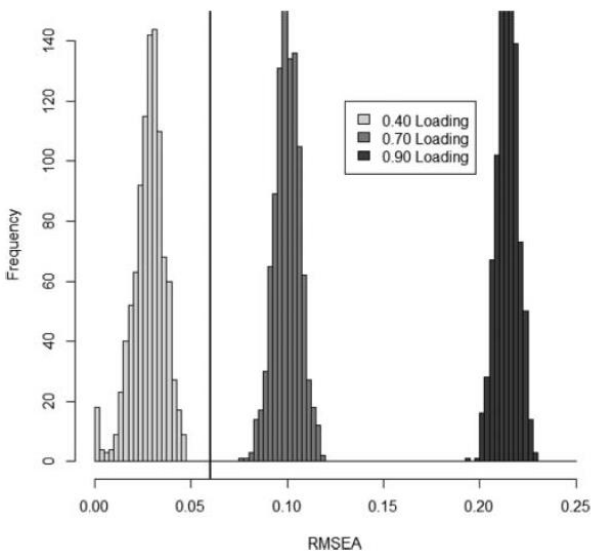
Incorrect models: RMSEA gets better with more indicators (less so with small N)

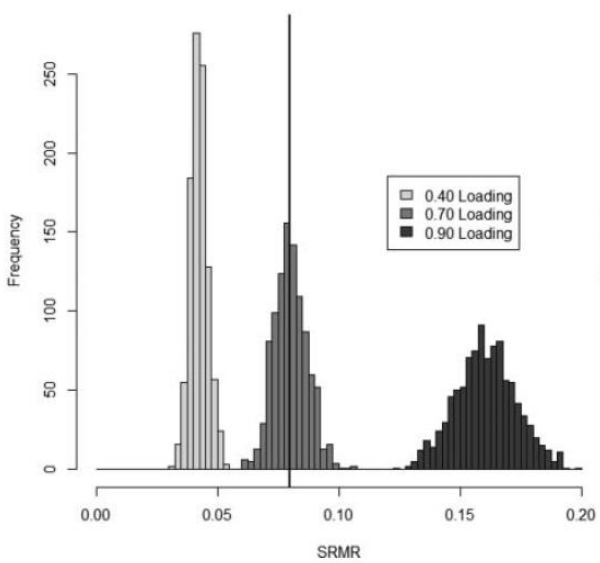


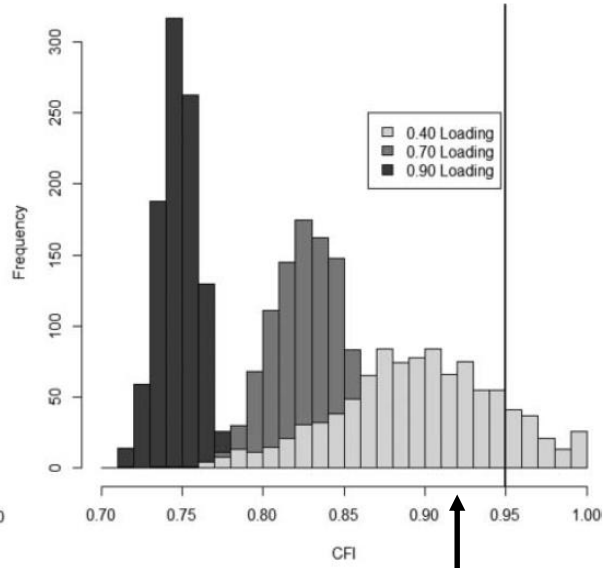
Right: Mis-specified residuals (errors) → misfit limited to only some indicators (so having more properly specified indicators makes fit better on average)

Good Fit* Is Easier With Lower Reliability

- Lower reliability → Lower standardized factor loadings
  - McNeish, An, & Hancock (2018): 15 indicators measuring 3 factors
  - Figures 2 and 3: missing factor covariance → always good fit if $\lambda = .40$!
  - Strong signal (i.e., more reliability) makes it easier to detect when model does not adequately capture that signal



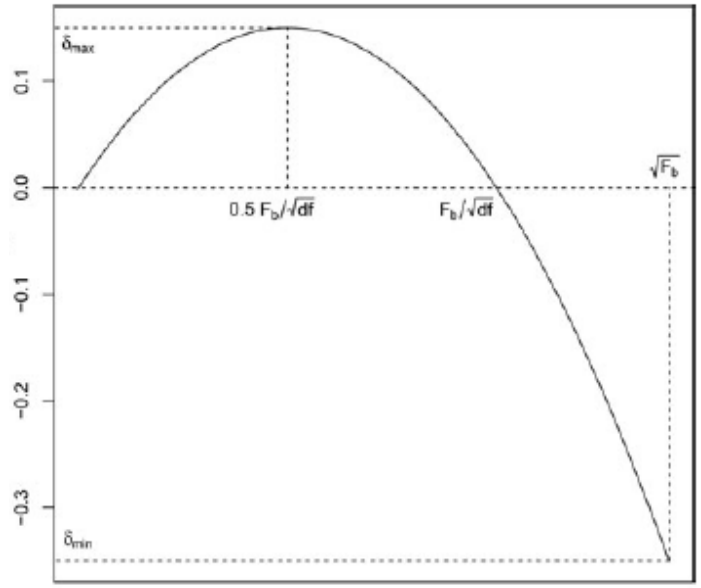




Wide variability in CFI with $\lambda = .40$!

When Fit Indices Disagree

- *Opposite pattern also found for CFI using more incorrect models: CFI was lower (worse fit) with lower reliability
  - From Heene, M., Hilbert, S., Draxler, C., Ziegler, M., & Bühner, M. (2011). Masking misfit in confirmatory factor analysis by increasing unique variances: A cautionary note on the usefulness of cutoff values of fit indices. *Psychological Methods, 16*(3), 319–336. https://doi.org/10.1037/a0024917
- When might RMSEA and CFI disagree? It's a complex function of amount of misfit and DF with which to test it (as well as reliability)



- Figure 1 from: Lai, K. & Green, S. B. (2016). The problem with having two watches: assessment of fit when RMSEA and CFI disagree. *Multivariate Behavioral Research, 51(*2-3), 220-239, DOI: http://dx.doi.org/10.1080/00273171.2015.1134306
- x-axis = amount of misfit in your H_0 model (up to null model, F_b)
- y-axis = model fit discrepancy function; >0 = CFI happier, <0 = RMSEA happier

4 Steps in Model Evaluation

1. **Assess global model fit (summary)**
   - Recall that item intercepts, factor means, and variances are just-identified → *misfit comes from mis-predicted covariances*
   - Be aware that artificially good absolute fit can be created by indicators with low reliability and/or missing data; assessments of global fit can be more variable with smaller N in large models
   - Corrections for non-normality also continually being developed…

- If model fit is not good (yet), you should NOT interpret the model estimates, because they will change as the model changes
  - If model fit is not good, you need to find out WHY → go to step 2
- Even if model fit IS good, it does not mean you are done: still proceed to step 2, assessing local fit
  - This should help protect against erroneous claims of good fit

4 Steps in Model Evaluation: Step 2

2. Identify local misfit: localized model strain

- Global model fit means that the observed and predicted item covariance matrices aren't too far off on the whole… this says nothing about the specific covariances to be predicted
- Should inspect **normalized model residuals** for that → Local fit
  - **RESIDUAL** output option in Mplus or ESTAT RESIDUAL in STATA
  - "Normalized" is residual/SE → **works like a z-score**
  - Relatively large absolute values indicate "localized strain"
  - **Positive** residual → Items are **more** related than you predicted
    - More than just the factor (your model) creating a covariance
  - **Negative** residual → Items are **less** related than you predicted
    - Not as related as your model said they should be
- **Evidence of localized strain tells you where the problems are, but not what to do about them…**

4 Steps in Model Evaluation: Step 2

2. Identify localized model strain, continued…

- Parallel info: **Modification Indices** (aka, voo-doo)
  - LaGrange Multiplier: decrease in model fit χ^2 by adding the listed model parameter (e.g., cross-loading, error covariance)
    - Usually only pay attention if > 3.84 for $\mathrm{DF} = 1$ (for $p < .05$)
    - Get expected parameter estimate for what's to be added, but should only pay attention if its effect size is "meaningful"
    - Also only pay attention if you can INTERPRET AND DEFEND IT
  - Implement these ONE AT A TIME, because one addition to the model can alter the rest of the model substantially

- Keep in mind that voo-doo indices can only try to repair your current model; **they will never suggest a new model**!

Testing Fixes to the Model

- Most common approach for assessing whether adding or subtracting parameters changes model fit is the likelihood ratio test (aka, −2ΔLL "deviance difference" test)
  - Done for you in two cases: comparing saturated H_1 to your H_0 as model χ^2, and comparing saturated H_1 to "null" model
  - Implemented via direct difference in model χ^2 values most often, but this is only appropriate when using regular ML estimation

- Variants of ML for non-normal data (like MLR) require a modified version of this −2ΔLL test (see Mplus website): http://www.statmodel.com/chidiff.shtml
  - Is called "rescaled likelihood ratio test"
  - Includes extra steps to incorporate scaling factors
  - I built you a spreadsheet for this…you're welcome ☺

Testing Fixes to the Model: −2ΔLL

- Comparing nested models via a "**likelihood ratio test**" → **−2ΔLL** (MLR rescaled version)
  - 1. Calculate **−2ΔLL** = $-2*(LL_{fewer} - LL_{more})$
  - 2. Calculate **scaling correction for difference** =

$$\frac{(\#parms_{fewer}*scale_{fewer}) - (\#parms_{more}*scale_{more})}{(\#parms_{fewer} - \#parms_{more})}$$

  - 3. Calculate **rescaled difference** = −2ΔLL / scaling correction
  - 4. Calculate **Δdf** = $\#parms_{more} - \#parms_{fewer}$
  - 5. **Compare rescaled difference to χ^2 with df = Δdf**
    - Add 1 parameter? $LL_{diff} > 3.84$, add 2 parameters: $LL_{diff} > 5.99$…
    - Absolute values of LL are meaningless (is relative fit only)
    - Process generalizes to many other kinds of models

Note: Your LL will always be listed as the **H0** (H1 is for the saturated, perfectly fitting model)

Fewer = simpler model
More = more parameters

Testing Fixes to the Model: −2ΔLL

- If **adding** a parameter, model fit can either get **better** OR stay the same ("not better"):
  - Better = larger LL for H0 and smaller model χ^2
  - e.g., add another factor, add error covariance,

- If **removing** a parameter, model fit can either get **worse** OR stay the same ("not worse")
  - Worse = smaller LL for H0 and larger model χ^2
  - e.g., constrain item loadings equal → test "tau-equivalence"

- When testing parameters that have a boundary (e.g., factor correlation ≠ 1?), this test will be slightly conservative
  - Should use $p < .10$ instead of $p < .05$ (or mixture χ^2 distribution)

Testing Fixes to the Model, cont.

- For comparing **non-nested models** (e.g., should y_1 load on F_2 or F_1 instead?), the -2ΔLL test is not applicable given same DF

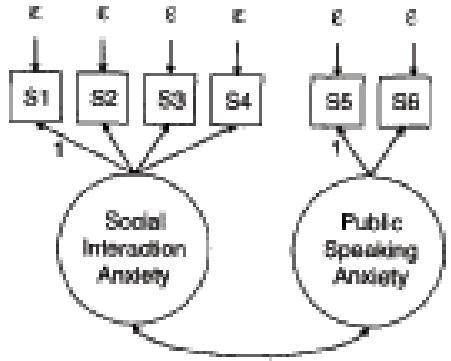
- Use information criteria instead: **AIC** and **BIC**
  - Akaike IC: AIC = -2LL + 2*#parameters
  - Bayesian (Schwartz) IC = -2LL + log(N)*#parameters
  - Are NOT significance tests, just "smaller is better", is "evidence"
  - **Still cannot be used on models with different items (outcomes)**

- For both nested or non-nested model comparisons, differences in other fit indices should be examined, too
  - No real critical values for changes in other fit indices, however
  - They may disagree (especially RMSEA, which likes parsimony)

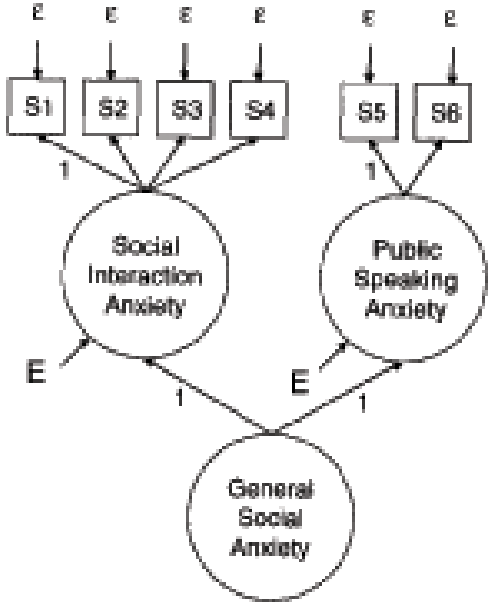
Fixing the Model by Expanding

- A common (and relatively easy to fix) source of misfit is caused by items that are too correlated after accounting for their common factor—some possible solutions:
  - Add **error covariance(s)** (i.e., as suggested by voo-doo indices)
    - Is additive: $Cov(y_1, y_2)$ = cov due to Factor + cov due to error covariance, so the residual covariance basically plugs the hole in the covariance matrix
    - In models that do not allow error covariances (e.g., IFA, stay tuned), you can do the same via a separate uncorrelated "**method factor**" (for positive covariance, fix both loadings = 1; for negative covariance, use 1 and −1)
    - **Either way, this means you have unaccounted for multidimensionality** → Explicit acknowledgement that you have measured your latent factor + something else that those items have in common (e.g., stem, valence, specific content) of unknown origin, so you must be able to defend error covariances
  - Lots of problematic pairings? **Re-consider factor dimensionality**
    - I'd generally recommend against adding cross-loadings, because if the item measures more than one thing, it will complicate the interpretation of factors

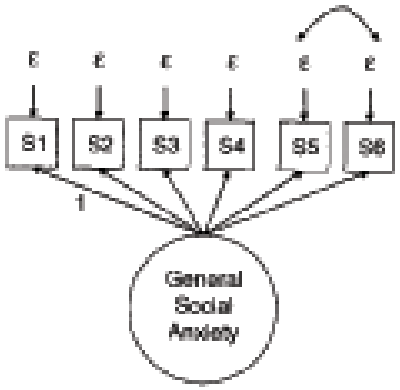
Equivalent Ways of Addressing Multidimensionality… (Brown, 2015 p. 181)



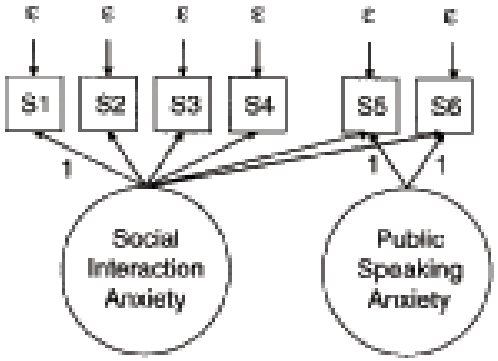
Two-Factor Models



Error Covariance



Factor + Method Factor



Here a general factor of "Social Interaction Anxiety" includes two items about public speaking specifically.

The extra relationship between the two public speaking items can be modeled in different, yet statistically equivalent ways… **error covariances represent another factor** (which is why you should be able to explain and predict them if you include them).

When to Simplify the Model

- Factors correlated > .85ish may suggest a simpler structure
  - Nested model comparison: fix factor variances to 1 so factor covariance becomes factor correlation, then test $r \neq 1$ at $p < .10$ (because r is bounded from −1 to 1)

- When might you consider dropping an item?
  - Non-significant loadings: If the item isn't related, it is NOT measuring the latent trait, and so you pry don't need it
  - Negative loadings: Make sure to reverse-coded as needed ahead of time, otherwise, this indicates a big problem!
  - Problematic leftover positive covariances between two items—such redundancy implies you may not need both (redundancy may indicate a "bloated specific")
  - If one item is responsible for many of the suggested error covariances, perhaps you might remove it (but be cautious, because often fewer items → less reliability)

- **However: models with different items (outcomes) are NOT COMPARABLE AT ALL because their LL values are based on different input data!**
  - No model comparisons of any kind (including −2LL, AIC, and BIC)
  - To do a true comparison, you'd need to leave the item in the model but set its loading = 0 (which is the same as the original test of its loading)

What else can go wrong?

- Error message: "**non-positive definite (NPD)**"
  - Both $\mathbf{S}$ (data) and $\boldsymbol{\Sigma}$ (predicted) matrices must be positive definite
    - Because they get inverted in the LL formula (like matrix division)
  - Non-positive definite means that the determinant is ≈ 0, or that the matrix is singular (has redundant information)
    - Double-check that data are being read in correctly; otherwise you may need to drop items that are too highly correlated
    - **NPD means your model is broken and you can't keep it**

- **Structural under-identification**
  - Does every factor have a mean and variance and at least 3 items?
  - Does the marker item actually load on the factor???
- **Empirical under-identification**
  - More likely with smaller sample sizes, fewer indicators per factor, and items with low communalities (R^2 accounted for by factor)

Open in case of emergency…

- If good model fit seems hopeless, you may need to go back to the drawing board… almost
  - Actual EFA uses weird constraints to identify the model, so don't use it

- Brown (2015) suggests an "E/CFA" approach of estimating an exploratory-like model staying within a CFA framework:
  - Fix each factor variance to 1 and mean to 0 for identification
  - Each factor gets one item that loads ONLY on it (loading fixed to 1)
  - Rest of items can load on all factors
  - Why bother? To get significance tests of factor loadings
  - May suggest a useful alternative structure, which should then ideally be replicated in an independent sample using CFA

Summary: Model Evaluation Steps 1 and 2

1. **Assess global model fit**
   - Recall that item intercepts, factor means, and variances are usually just-identified → *so misfit comes from mis-predicted covariances*
   - χ^2 is sensitive to large sample size, so pick at least one global fit index from each class (e.g., CFI, RMSEA); cutoffs with caveats
2. **Identify localized model strain**
   - Global model fit means that the observed and predicted covariance matrices aren't too far off on the whole… says nothing about the specific matrix elements (reproduction of each covariance)
   - Consider normalized residuals and modification indices to try and "fix" the model (add or remove factors, add or remove residual covariances, etc.)—Has to be theoretically justifiable!!

Good global and local fit? Great, but we're not done yet…

4 Steps in Model Evaluation: Step 3

3. Inspect **parameter effect sizes** and significance
   - A 1-factor model will fit each of these correlation matrices perfectly:

	y1	y2	y3	y4
y1	1			
y2	**.1**	1		
y3	**.1**	**.1**	1	
y4	**.1**	**.1**	**.1**	1

	y1	y2	y3	y4
y1	1			
y2	**.8**	1		
y3	**.8**	**.8**	1	
y4	**.8**	**.8**	**.8**	1

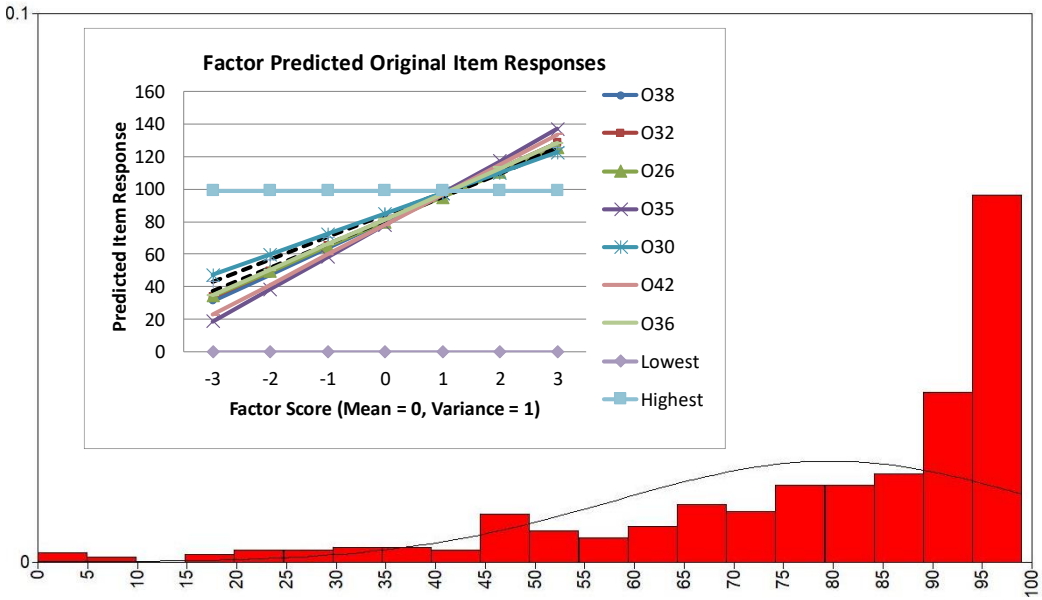
   - **Good model fit does not guarantee a good model**
   - **A good model has meaningful factor loadings**
   - **If your items are not correlated, game over, regardless of fit**

4 Steps in Model Evaluation: Step 3

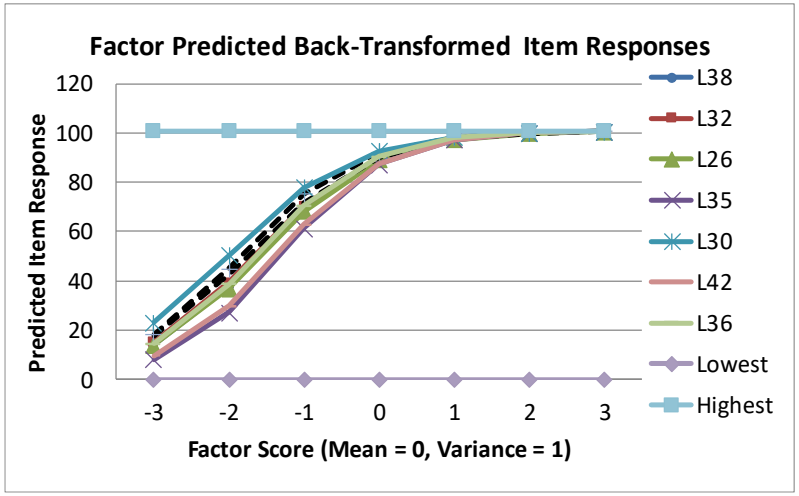
3. Inspect **parameter effect sizes** and significance
   - Model fit does not guarantee meaningful factor loadings
     - Can reproduce lack of covariance quite well and still not have anything useful—e.g., factor loading of 0.2 → 4% shared variance?!?
     - **Effect size (R^2 of item variance from factor) is practical significance**
   - Get SEs and *p*-values for unstandardized estimates (at least report estimate from standardized solution)
     - Marker items won't have significance tests for their unstandardized loadings because they are fixed at 1, but you'll still get standardized factor loadings for them (help to judge relative importance)
   - Make sure all estimates are within bounds AND predicted item responses are plausible at expected latent factor values (±2 SD)
     - No standardized factor loadings > 1 (unless the indicator has cross-loadings, in which case this is actually possible)
     - No negative factor variances or negative error variances

4 Steps in Model Evaluation: Step 3

- CFA is a regression model, so you can plot the responses predicted from the unstandardized item intercepts and slopes (factor loadings) across factor values
- If the predicted responses exceed the possible range within ±2 SD, then **the linear CFA may not be appropriate** (responses are not "normal enough" to use CFA)



- CFI using logit-transformed item responses is a potential solution for bounded/skewed continuous items (creates a logistic curve)
  - $L = min - 1,\ U = max + 1$
  - $Logit = LOG\left(\frac{y_{is} - L}{U - y_{is}}\right)$
  - Predicted $y_{is} = L + (U - L)\frac{\exp(Logit)}{1 + \exp(Logit)}$
- For ordinal responses, choosing an IFA/IRT model is another option (stay tuned)



4 Steps in Model Evaluation: Step 4

4. Calculate item information and model-based reliability

- **Item Information = $(\textbf{unstandardized } \boldsymbol{\lambda})^2 / \mathbf{Var(e)}$**
  → What proportion of item variance is "true" relative to error?
  - Size of unstandardized loadings by themselves is not enough, as their relative contribution depends on size of error variance
  - The standardized loadings will give you the same rank order in terms of item information, which is why information is not often used within CFA (but stay tuned for item and test information in IRT/IFA models)

- **"Omega" Sum Score Reliability =** $\dfrac{\mathbf{Var(F)} * (\boldsymbol{\Sigma\lambda})^2}{[\mathbf{Var(F)} * (\boldsymbol{\Sigma\lambda})^2] + \boldsymbol{\Sigma}\mathbf{Var(e)} + 2\boldsymbol{\Sigma}(\mathbf{e\ cov})}$

  → Factor variance * squared sum of <u>unstandardized</u> factor loadings, over that + summed error variances + 2*summed error covariances
  - Although Omega should be calculated using unstandardized loadings, Omega can differ slightly across methods of model identification
  - **Omega is calculated PER FACTOR because it assumes unidimensionality (which should have been tested already)**

CFA Model Evaluation: Summary

- The primary advantage of working in a CFA framework is obtaining indices of global and local model fit
  - χ^2 and model fit indices indicate how well the model-predicted covariance matrix matches the data-observed covariance matrix…
    - .. But normalized residuals should still be examined for evidence of local misfit (e.g., mis-predicted covariances between certain items)
  - Nested model comparisons via rescaled $-2\Delta LL$ can be conducted in order to compare the fit of augmented or simplified models…
    - … But be careful relying too blindly on modification indices to do so
  - Effect size and significance of model parameters matters, too
    - … How well are your latent factors really defined anyway? Effect size!
    - Watch out for out-of-bound estimates—this means something is wrong
    - Watch for unreasonable predicted responses—this means you shouldn't be using a linear slope CFA model (so you need a nonlinear slope model)

Testing CTT Assumptions in CFA

- **Alpha sum-score reliability** assuming three things:
  - The items measure a single, unidimensional latent factor
  - All factor loadings (discriminations) are equal, or that items are "true-score equivalent" or "**tau-equivalent**"
  - **Local independence** (errors are uncorrelated)

- After assessing unidimensionality of each latent factor, we can then test the assumption of **tau-equivalence** via a $-2\Delta LL$ comparison against a model in which the factor loadings are constrained to be equal
  - If model fit gets worse, the loadings are not equal; items differ in discrimination
  - If so, don't use alpha—use model-based reliability (omega) instead, because omega assumes unidimensionality, but not tau-equivalence

- The assumption of **parallel items** is then testable by constraining item error variances to be equal, too—does model fit get worse?
  - Parallel items is needed to use Spearman-Brown formulas to predict reliability
  - Parallel items will hardly ever hold in real data
  - Note that if tau-equivalence doesn't hold, then neither does parallel items

Conclusion: The Big Picture of CFA

- **The CFA unit of analysis is the ITEM:** $y_{is} = \mu_i + \lambda_i F_s + e_{is}$
  - **Linear** regression relating **continuous** item responses to latent factor predictor
  - Both item AND subject properties matter in predicting item responses
  - Latent factors are estimated as separate entities based on the observed covariances among items—latent factors represent testable assumptions
    - Items are unrelated after controlling for factor(s) → local independence
    - Modeling framework allows exceptions via error covariances and method factors

- **Because item responses are included:**
  - Items are allowed to vary in discrimination (as factor loadings) → thus, exchangeability (tau-equivalence) is a testable hypothesis
  - Because difficulty (item intercepts) do not contribute to the covariance, they don't really matter in CFA (unless you are testing factor mean differences)
  - To make a test better, you need more items
    - **What kind of items? Ones with *greater information* → $\lambda^2/Var(e)$**
  - Measurement error is still assumed constant across the latent trait
    - **People low-medium-high in Factor Score are measured equally well**