Confirmatory Factor Models
(CFA: Confirmatory Factor Analysis)

- Today's topics:
  - **Comparison of EFA and CFA**
  - **CFA model parameters and 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 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
  - Goal is to pick a rotation that gives closest approximation to simple structure (clearly-defined factors, fewest cross-loadings)
  - No way of separating 'content' from 'method' factors

- **CFA: Your job in the first place!**
  - CFA must be theory-driven: structure is a testable hypothesis
  - You specify number of factors and their inter-correlations
  - You specify which items load on which factors (yes/no)
  - You specify any additional relationships for method/other variance

EFA vs. CFA: Judging model fit

- **EFA: Eye-balls and Opinion**
  - #Factors? Scree-ish plots, interpretability…
  - Which rotation? Whichever makes most sense…
  - 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)
  - Significance of item loadings
  - Significance 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 be predictors (exogenous) or outcomes (endogenous) or both at once as needed (e.g., as mediators)
    - Relationships 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
    - Second best option: Use “plausible values” instead (stay tuned)

Confirmatory Factor Analysis (CFA)

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

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

  - Observed response for item *i* and subject *s*

    = intercept of item *i* (μ)

    + subject *s*'s latent trait/factor (F), item-weighted by loading λ

    + error (e) of item *i* and subject *s*

- **What does this look like? Linear regression (without a real X)!**
  - $y_s = \beta_0 + \beta_1 X_s + e_s$ → written for each item → $y_{is} = \beta_{0i} + \beta_{1i} X_s + e_{is}$
  - Intercept β_{0i} = μ_i =
  - Slope of Factor β_{1i} = λ_i =
  - Residual e_{is} = e_{is} =

Revisiting Vocabulary: Item Psychometric Properties

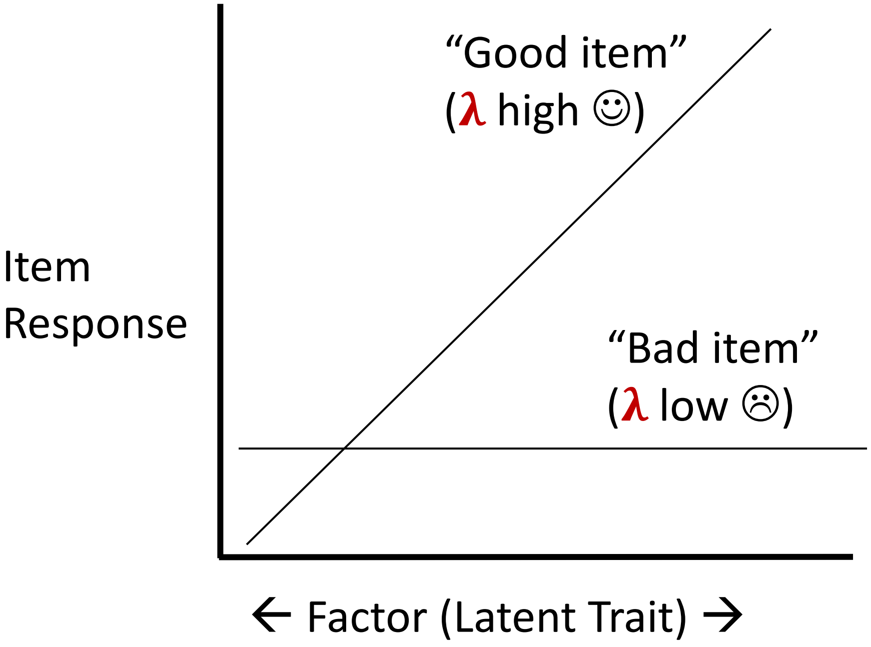
- **Discrimination**: How related each item is to the latent trait
  - In CTT, discrimination is indexed by the item-total correlation
    - The total score is the best estimate of the latent trait in CTT
  - In **CFA**, discrimination is indexed by the **factor loading/slope (λ_i)**
    - We now have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

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

Why Item Intercepts Are Often Ignored…

A **“good” item** has a large slope (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.



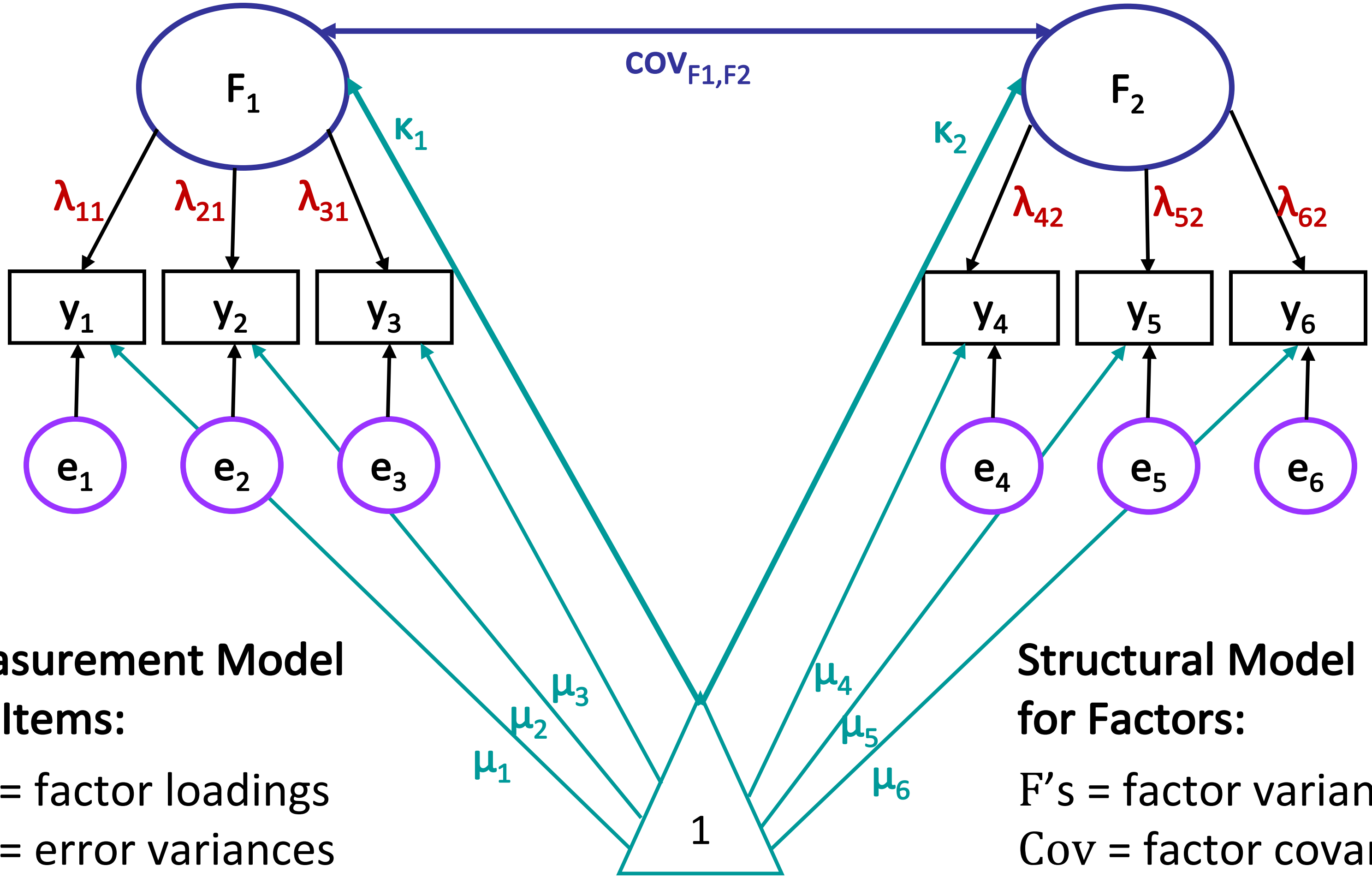
Here item intercepts are irrelevant in evaluating how “good” an item is, so they are not really needed.

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)

CFA Model with Factor Means and Item Intercepts

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 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)
  - $\mathbf{Var}(y_i) = [\lambda_i^2 * \mathbf{Var}(F)] + \mathbf{Var}(e_i)$

Careful! There is also STDY and STD

- Standardized → Solution transformed to $\mathrm{Var}(y_i) = 1, \mathrm{Var}(F) = 1$ 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 / $\mathbf{SD}(y)$ → not typically reported
  - Standardized factor loading = $[\lambda_i * \mathbf{SD}(F)]$ / $\mathbf{SD}(y)$ **= item correlation with factor**
  - Standardized error variance = $\mathbf{1 - standardized}\ \lambda_i^2$ = "variance due to *not* factor"
  - R^2 for item = $\mathbf{standardized}\ \lambda_i^2$ = "variance due to the factor"

CFA Model Equations

- Measurement model per item (numbered) for subject *s*:

  - $y_{1s} = \mu_1 + \lambda_{11}F_{1s} + \mathbf{0}F_{2s} + e_{1s}$
  - $y_{2s} = \mu_2 + \lambda_{21}F_{1s} + \mathbf{0}F_{2s} + e_{2s}$
  - $y_{3s} = \mu_3 + \lambda_{31}F_{1s} + \mathbf{0}F_{2s} + e_{3s}$

  - $y_{4s} = \mu_4 + \mathbf{0}F_{1s} + \lambda_{42}F_{2s} + e_{4s}$
  - $y_{5s} = \mu_5 + \mathbf{0}F_{1s} + \lambda_{52}F_{2s} + e_{5s}$
  - $y_{6s} = \mu_6 + \mathbf{0}F_{1s} + \lambda_{62}F_{2s} + e_{6s}$

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

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

where $\mathbf{Y}$, $\boldsymbol{\mu}$, and $\mathbf{e}$ = 6x1 matrices, $\boldsymbol{\lambda}$ = 6x2 matrix, and $\mathbf{F}$ =2x1 matrix

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 = **R^2**).

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

- The CFA **item error variances** ($\mathbf{Var[e_{is}]}$) 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 mis-fit or mis-prediction)

- The CFA **item factor loadings** (λ_i) predict the **item covariances**
  - Given 3+ items, there will be more covariances among items to predict than item factor loadings to predict them, **creating room for mis-fit**

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

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

- Unstandardized solution: Covariance of y_1, y_3 = $\boldsymbol{\lambda_{11}} * \mathbf{Var(F1)} * \boldsymbol{\lambda_{31}}$
- Standardized solution: Correlation of y_1, y_3= $\boldsymbol{\lambda_{11}} * \mathbf{(1)} * \boldsymbol{\lambda_{31}}$ ← std load
- ONLY reason for correlation is their common factor (local independence, LI)

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

- Unstandardized solution: Covariance of y_1, y_6 = $\boldsymbol{\lambda_{11}} * \mathbf{cov_{F1,\ F2}} * \boldsymbol{\lambda_{62}}$
- Standardized solution: Correlation of y_1, y_6= $\boldsymbol{\lambda_{11}} * \mathbf{cor_{F1,\ F2}} * \boldsymbol{\lambda_{62}}$ ← std load
- ONLY reason for correlation is the correlation between factors (again, LI)

Variances are additive (and will be reproduced correctly):

- **Var(y_1) = (λ_{11}^2)*Var(F_1) + Var(e_i) → but note the imbalance of λ^2 and e**

Model-Predicted Item Covariance Matrix

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

Σ= model-predicted item covariance matrix is created from:

Λ = item factor loadings

Φ= factor variances and covariances

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

Ψ = item error 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

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

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

Items within Factor 1

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

Items within Factor 2

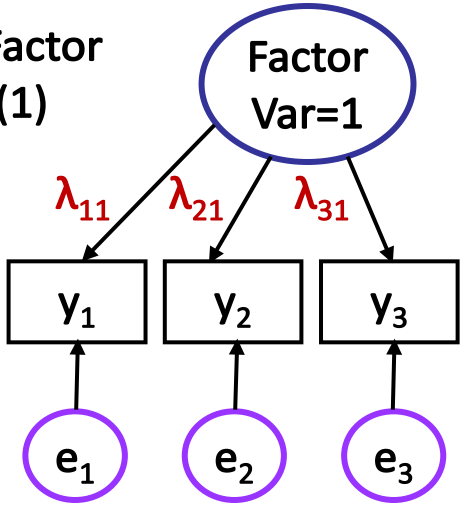
The only reason why items from different factors should be related is the **covariance** between the two factors.

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

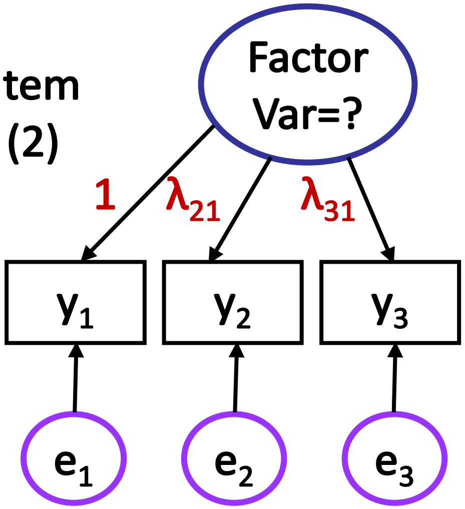
CFA Model Identification:

Create a Scale for the Latent Variable Variance

Z-Score Factor Method (1)



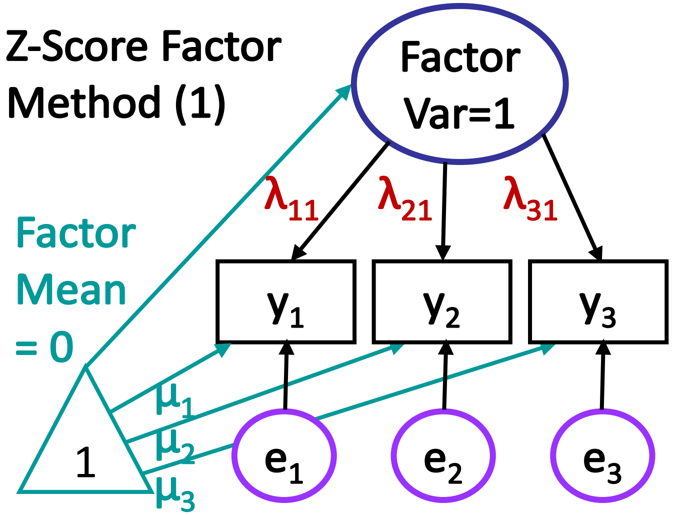
Marker Item Method (2)

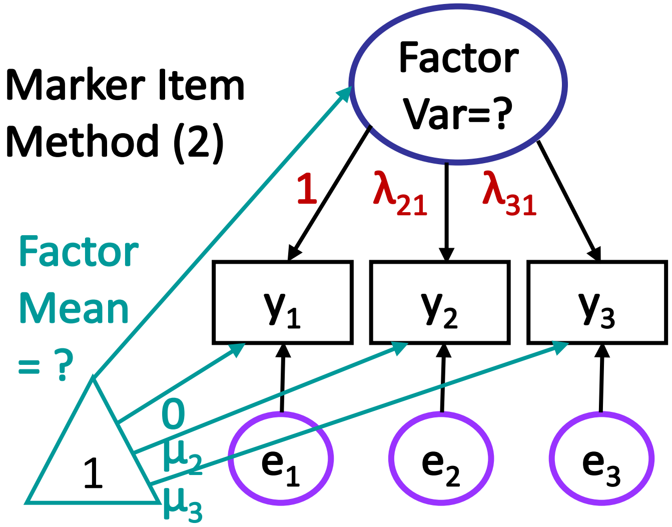


- The factor doesn't exist, so **it needs a scale** (it needs a mean and variance):
- Two **equivalent** options to create a scale for the factor **VARIANCE**:
  - **(1) Fix factor variance to 1: "z-score"**
    - Factor is interpreted as standardized
    - **Can't** be used in models with higher-order factors (coming later in the 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:

Create a Scale for the Latent Variable Mean





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

  - **(1) Fix factor mean to 0: "z-score"**
    - Factor is interpreted as standardized
    - <u>Can</u> 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)

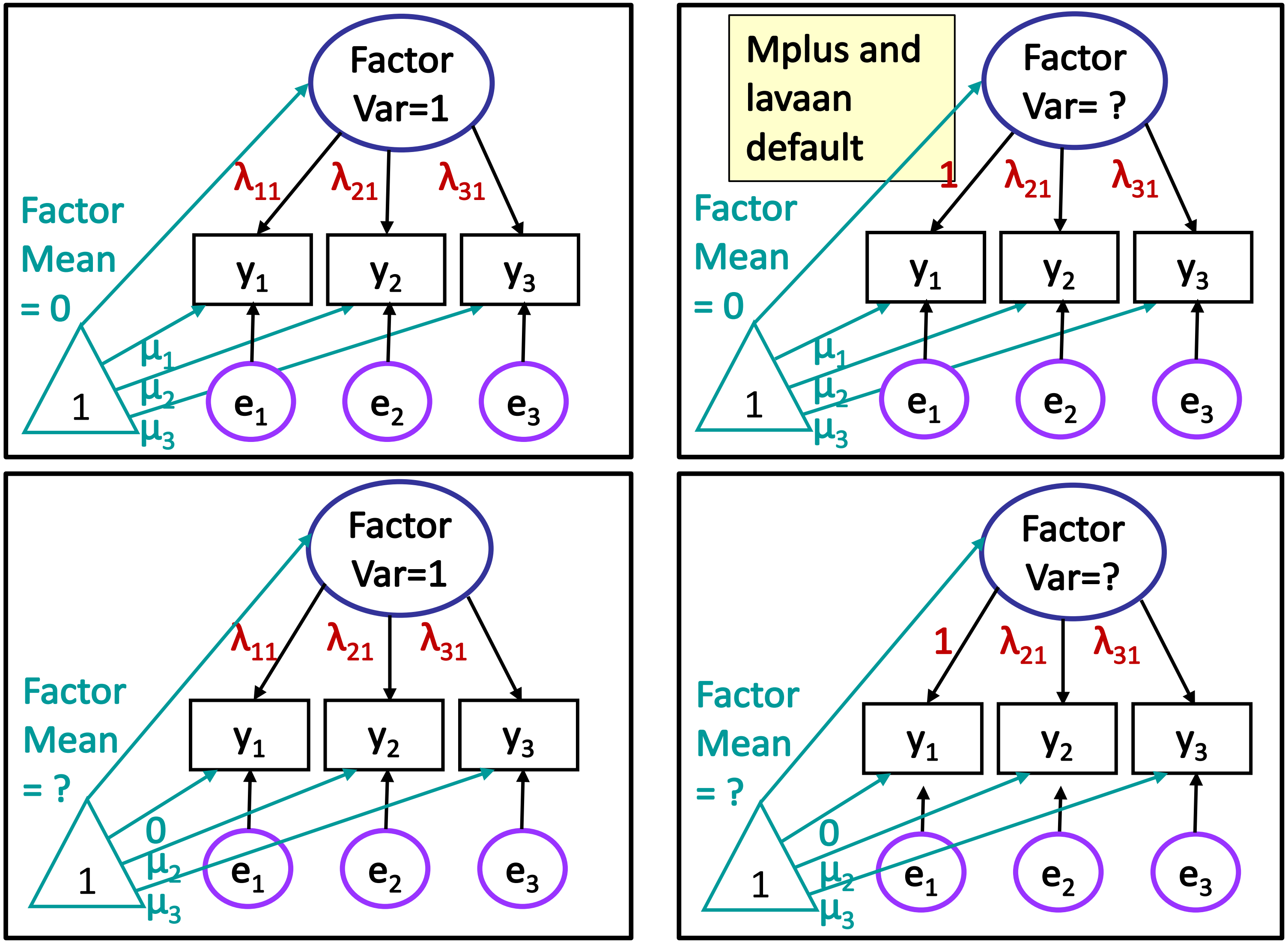
Equivalent CFA Model Identifications

Factor Variance = 1

Factor Variance Estimated

Factor Mean = 0

Factor Mean Est.


Mplus and lavaan default

Factor Model Identification

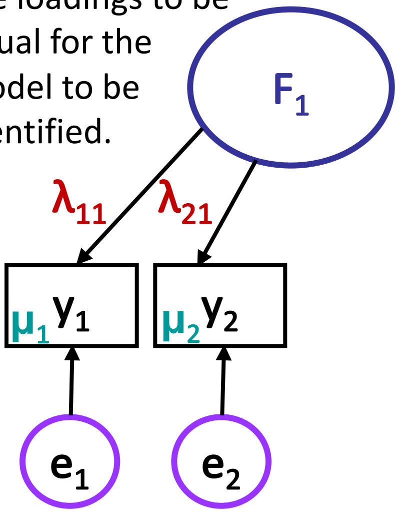
- Goal: *Name that Tune* → Reproduce observed item covariance matrix using as few estimated parameters as possible
  - (Robust) Maximum likelihood used to estimate model parameters
    - **Measurement Model**: Item factor loadings, item intercepts, item error variances
    - **Structural Model**: Factor variances and covariances, factor means
  - Global model fit is evaluated as difference between model-predicted matrix and observed matrix (but only the covariances really contribute to misfit)

- How many possible parameters can you estimate (total DF)?
  - **Total DF depends on # ITEMS →** $\mathbf{v}$ (NOT on # people)
  - Total number of **unique elements** in item covariance matrix
    - Unique item elements = each variance, each covariance, each mean
    - Total unique elements = $(\mathrm{v}(\mathrm{v}+1)\ /\ 2)+\mathrm{v}$ → if 4 items, then ((4*5)/2) + 4 = 14
- Model degrees of freedom (df) = data input − model output
  - Model df = # possible parameters − # estimated parameters

Under-Identified Factor: 2 Items

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

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



In other words, the assumptions required to calculate 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

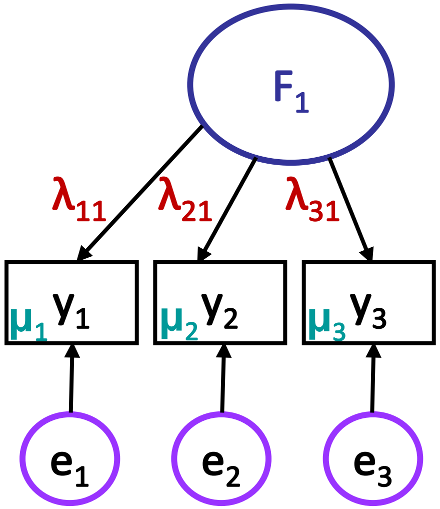
df = 5 – 6 = – 1

If $r_{y1,y2}$ = .64, then:

λ_{11} = .800, λ_{21} = .800 ??
λ_{11} = .900, λ_{21} = .711 ??
λ_{11} = .750, λ_{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
  - Parameter estimates have a unique solution that will perfectly reproduce the observed matrix
  - Example: Solve x + y = 7, 3x – y = 1



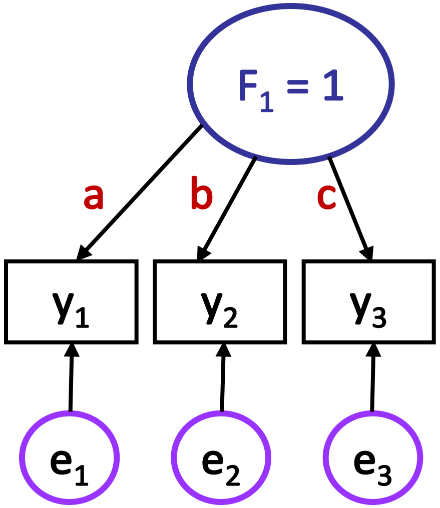
Total possible **df** = unique pieces of data = **9**

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

Solving a Just-Identified Model

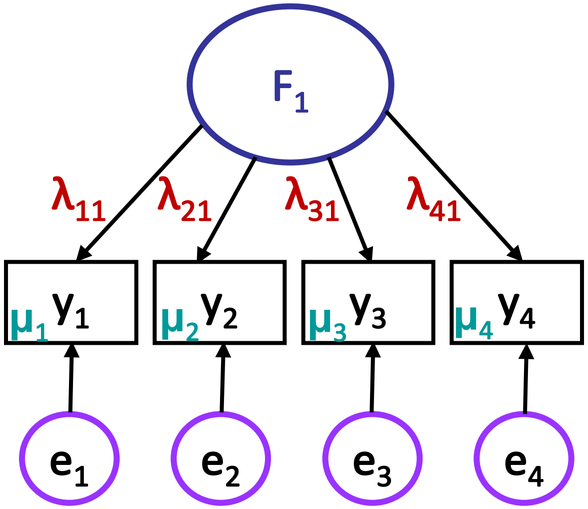


	Y_1	Y_2	Y_3
Y_1	1.00		
Y_2	.595	1.00	
Y_3	.448	.544	1.00

- 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: $\text{Var}(e_1) = 1 - a^2 = .51$

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
  - Parameter estimates have a unique solution that will NOT perfectly reproduce the observed matrix → **if df > 0, NOW we can test model fit**



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

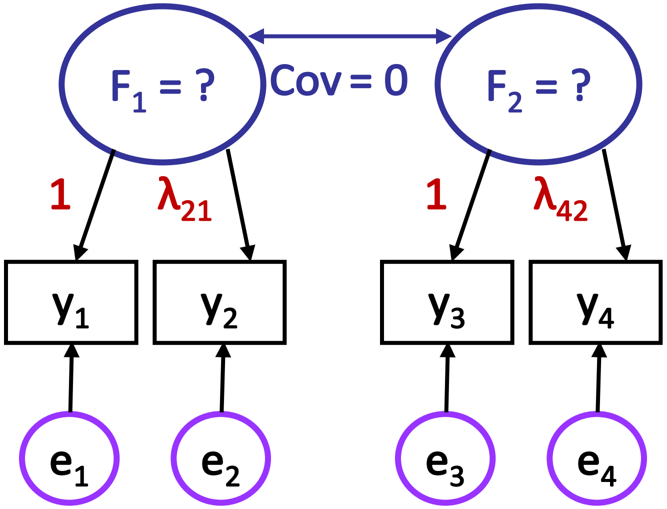
0 factor variances		1 factor variance
0 factor means		1 factor mean
4 item loadings	OR	3 item loadings
4 item intercepts		3 item intercepts
4 error variances		4 error variances

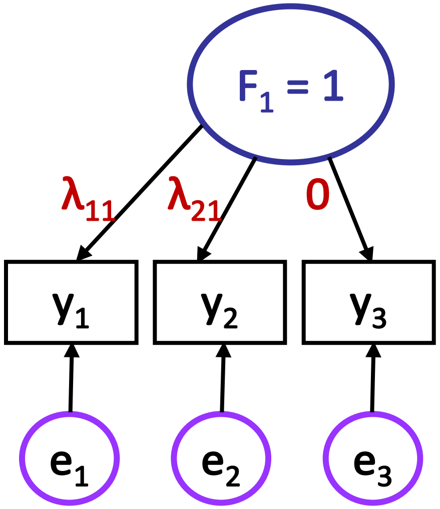
df = 14 – 12 = 2

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)?
  - Make sure each factor is identified (scale of factor mean and variance is set)
- 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?

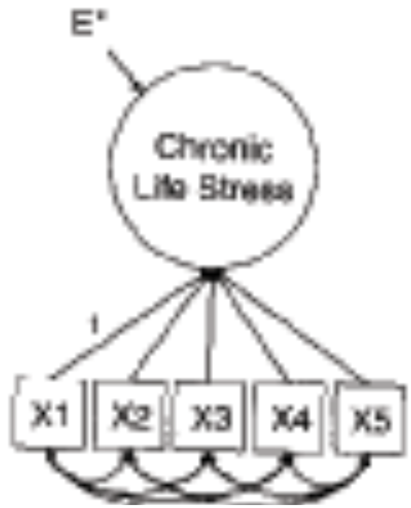
Factor Model:

- Composed of “Reflective” or “Effects” items
- Factor is **cause** of observed item responses
- Items **are** exchangeable and should be correlated
- **Is identified** with 3+ items (fit testable with 4+ items)

Component Model:

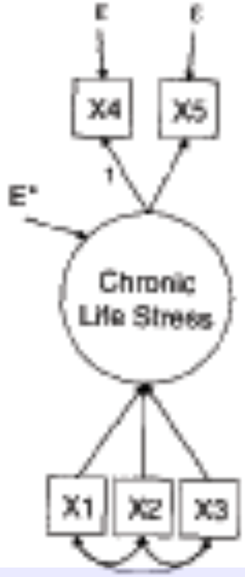
- Composed of “Formative” or “Emergent” or “Cause” items
- Component is **result** of observed item responses
- Items **are not** exchangeable and may not be correlated
- **Will not be identified** no matter how many items *without additional variables in the model*

Formative (Component) Models
(see Brown p. 351-362)



Model 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



Model Parameters:
4 factor loadings/regression paths
1 factor disturbance (variance left over)
3 item correlations
5 item variances/error variances
5 item means

df = 20 – 18 = 2
Identified

Intermediate Summary: CFA

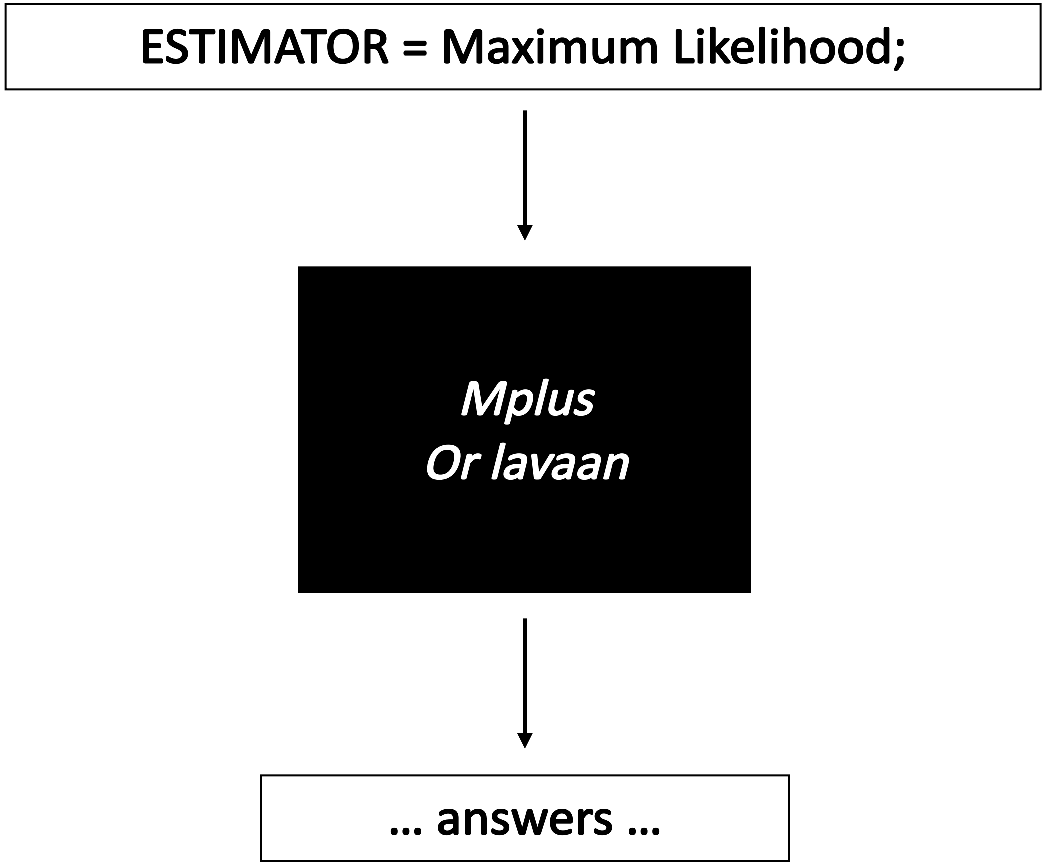
- CFA is a **linear model** in which continuous observed item responses are predicted from latent factors (traits) and error
  - Goal is to reproduce observed **item covariance matrix** using estimated parameters (intercept, loading, and error variance for items, factor variance)
  - Factor model makes specific testable mathematical predictions about how item responses should relate to each other: **loadings predict covariances**
  - Need at least 3 items per factor for the model to be identified; need **at least 4 items 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 trait is estimated separately from item responses
    - Separates interpretation of person traits from specific items given
    - Separates interpretation of item properties from specific persons in sample

Confirmatory Factor Models
(CFA: Confirmatory Factor Analysis)

- Today's topics:
  - Comparison of EFA and CFA
  - CFA model parameters and identification
  - **CFA model estimation**
  - **CFA model fit evaluation**

Where the Answers Come From: The Big Picture of Estimation



ESTIMATOR = Maximum Likelihood;

Mplus
Or lavaan

… answers …

Any questions?

What all do we have to estimate?

- For example, a model with two correlated factors for $v = 6$ items:
  - F1 measured by items 1,2,3; F2 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 (σ_{ei}^2) + 1 factor covariance = 19 total parameters**
- **Item parameters** are **FIXED effects** → inference about specific item
  - Is 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 care about the **factor means, variances, and covariances** as sufficient statistics, but **not** about the factor scores for particular **individuals** per se

The End Goals of Maximum Likelihood (ML) Estimation

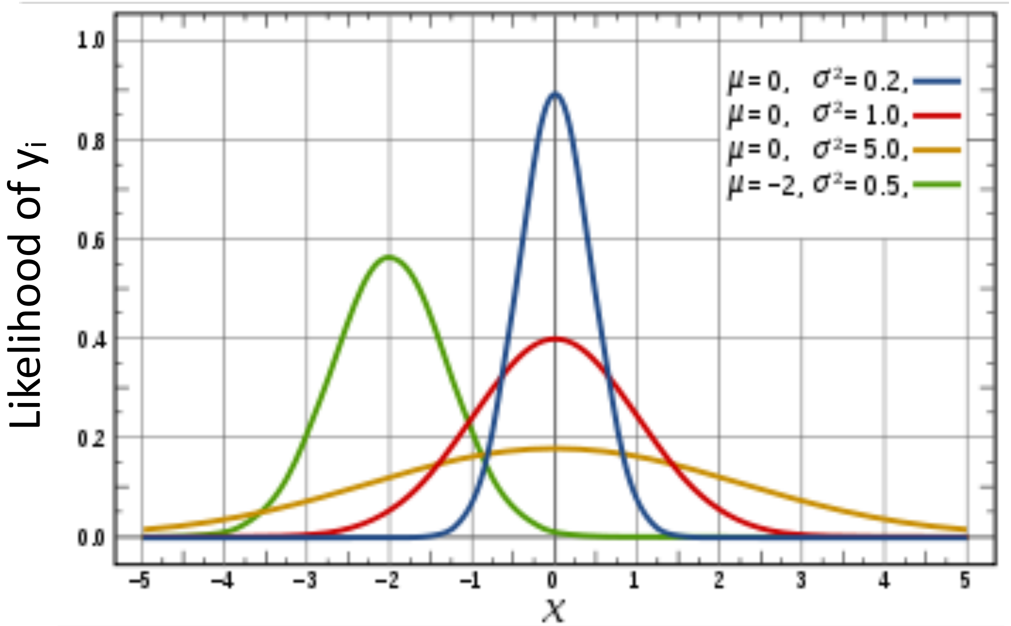
1. Obtain "most likely" values for each unknown parameter in our model (intercepts, loadings, error variances, factor means, factor variances, factor covariances) → the answers → **the estimates**

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**

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:

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

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

- This PDF tells us how **likely** any value of y_i is given two pieces of info:
  - Conditional mean $\hat{y}_i$
  - residual variance σ_e^2

- We can see this work using the NORMDIST function in excel!
  - Easiest for empty model:

$$y_i = \beta_0 + e_i$$

- We can check our math via R, SAS PROC MIXED, or Mplus!

Multivariate Normal for $\mathbf{Y}_s$: all v=6 y_s values from person *s*

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

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

- In our CFA model, the only fixed effects that predict the 6 $\mathbf{Y}_s$ values are the item intercepts (now $v = 6$ of them in the vector $\boldsymbol{\mu}$)
- The 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/covariances, $\boldsymbol{\Psi}$ = error variances

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

  - Variance of Item *i*: $\mathbf{Var(y_i)} = \boldsymbol{\lambda}^2 * \mathbf{Var(F_s)} + \mathbf{Var(e_i)}$
  - Covariance of items on same factor: $\mathbf{Cov(y1, y2)} = \boldsymbol{\lambda}_{\mathbf{11}} * \mathbf{Var(F1)} * \boldsymbol{\lambda}_{\mathbf{21}}$
  - Covariance of items on different factors: $\mathbf{Cov(y1, y6)} = \boldsymbol{\lambda}_{\mathbf{11}} * \mathbf{Cov(F1, F2)} * \boldsymbol{\lambda}_{\mathbf{62}}$
- Uses $|\boldsymbol{\Sigma}|$ = determinant of $\boldsymbol{\Sigma}$ = summary of *non-redundant* info
- $(\boldsymbol{\Sigma})^{-1}$ → 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:

$$\text{Model } \left(\text{H}_0\right) \text{Likelihood: L} = \prod_{s=1}^{N} \left\{ \left(2\pi\right)^{-v_s/2} * \left|\boldsymbol{\Sigma}\right|^{-1/2} * \exp\left[-\frac{1}{2}\left(\mathbf{Y}_s - \boldsymbol{\mu}\right)^T \left(\boldsymbol{\Sigma}_s\right)^{-1} \left(\mathbf{Y}_i - \boldsymbol{\mu}\right) \right] \right\}$$

$$\text{Model } \left(\text{H}_0\right) \text{Log Likelihood: LL} = \sum_{s=1}^{N} \left\{ \left[-\frac{v_s}{2} \log\left(2\pi\right) \right] + \left[-\frac{1}{2} \log\left|\boldsymbol{\Sigma}\right| \right] + \left[-\frac{1}{2}\left(\mathbf{Y}_s - \boldsymbol{\mu}\right)^T \left(\boldsymbol{\Sigma}_s\right)^{-1} \left(\mathbf{Y}_i - \boldsymbol{\mu}\right) \right] \right\}$$

- Try one set of possible parameter values, compute LL
- Try another possible set, compute 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 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 the 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"

$$\boxed{\text{Each parameter SE} = \frac{1}{\sqrt{\text{information}}}}$$

  - So steeper function = more information = more precision = **smaller SE**

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

- Use Model (H_0) LL from predicting **Σ** → so how good is it?
- Get the best possible LL if we used the real data (**S**) instead:

$$\text{Saturated Model } \left(\text{H}_1\right) \text{ Log Likelihood: LL} = \sum_{s=1}^{N} \left\{ \left[-\frac{v_s}{2} \log\left(2\pi\right) \right] + \left[-\frac{1}{2} \log\left|\mathbf{S}\right| \right] + \left[-\frac{1}{2} v_s \right] \right\}$$

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

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

- Combining and re-arranging the terms in LL_{H0} and LL_{H1} yields this more common expression for the ML fitting function:

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

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

What about item non-normality?

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

- Impact of non-normality of item responses:
  - Linear model predicting item response from factor may not work well
    - if y_{is} is not really continuous, the slope needs to shut off at the ends
  - SEs and χ^2-based model fit statistics will be incorrect
  - Three fixes: **Robust ML** (or transform the data, or 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)
  - Still **a linear model** between the items response and latent factor, so the parameter estimates will be the same as in regular ML

- Adjusts **fit statistics** using an estimated **scaling factor** → problematic 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

The Big Picture of Model Fit

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

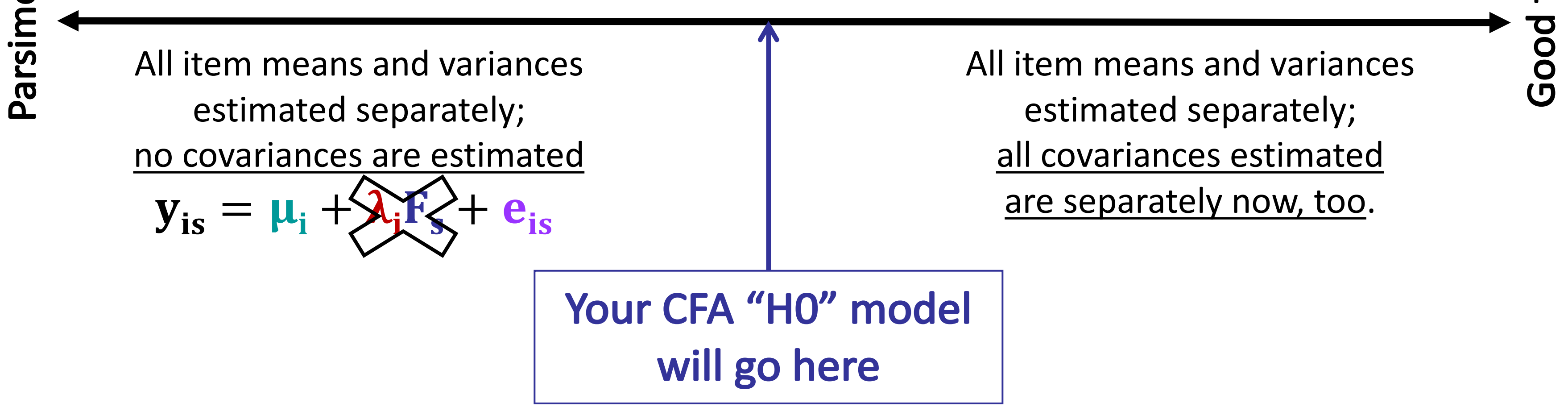
Baselines for Assessing Fit in CFA (Item means all saturated in both)

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; H1) 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; <u>no covariances are estimated</u>

$$\mathbf{y}_{is} = \mu_i + \xcancel{\lambda_i F_s} + e_{is}$$

All item means and variances estimated separately; <u>all covariances estimated are separately now, too</u>.

Your CFA “H0” model will go here

Baseline model results 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
```

H1 Saturated (Unstructured) Model

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

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

"Model fit" χ^2 is from a −2ΔLL test of your chosen **H0** model vs. saturated **H1** model

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

"Baseline model" χ^2 is from −2ΔLL test of null model vs. saturated **H1** model

Independence (Null) Model

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

4 Steps in Assessing Model Fit

1. Global model fit
   - *Does the model 'work' as a whole?*

2. Local model fit
   - *Are there any more specific problems?*

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?*

Indices of Global Model Fit

- Primary fit index: obtained model $\boldsymbol{\chi^2 = 2 * \mathrm{N} * \mathrm{F_{ML}}}$
  - χ^2 is evaluated based on model df (# parameters left over)
  - Tests null hypothesis that $\boldsymbol{\Sigma} = \boldsymbol{S}$ (that model is perfect), so **significance is bad** (i.e., smaller χ^2, bigger *p*-value is better)
  - Just using χ^2 to index model fit is usually insufficient, however:
    - Obtained χ^2 depends largely on sample size (N)
    - Is unreasonable null hypothesis (perfect fit, really??)
    - Only possible given balanced data (as typical in CFA)

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

Indices of Global Model Fit

- Absolute Fit: $\boldsymbol{\chi}^2$
  - Don't use "ratio rules" like $\chi^2/\mathrm{df} > 2$ or $\chi^2/\mathrm{df} > 3$

- Absolute Fit: **SRMR**
  - **Standardized Root Mean Square Residual**
  - Get difference of standardized $\boldsymbol{\Sigma}$ and $\boldsymbol{S}$ → residual matrix
  - Sum the squared residuals of the correlation matrix across items, divide by number of residuals (i.e., matrix elements)
  - Ranges from 0 to 1: smaller is better
  - ".08 or less" → good fit

- See also: **RMR (Root Mean Square Residual)**

Indices of Global Model Fit

Parsimony-Corrected: **RMSEA**

- **Root Mean Square Error of Approximation**
- Relies on a "non-centrality parameter" (NCP)
  - Indexes how far off your model is → χ^2 distribution shoved over
  - NCP → $\mathrm{d} = (\chi^2 - \mathrm{df}) / \mathrm{N}$ Then, $\mathrm{RMSEA} = \sqrt{\mathrm{d/df}}$
- RMSEA ranges from 0 to 1; smaller is better
  - < .05 or .06 = "good", .05 to .08 = "acceptable", .08 to .10 = "mediocre", and >.10 = "unacceptable"
  - In addition to point estimate, get 90% confidence interval
  - RMSEA penalizes for model complexity – it's discrepancy in fit per df left in model (but not sensitive to N, although CI can be)
  - Test of "close fit": null hypothesis that RMSEA ≤ .05

Indices of Global Model Fit

Comparative (Incremental) Fit Indices

- Fit evaluated relative to a ‘null’ or ‘independence’ model (of 0 covariances)
- Relative to that, your model fit should be great!

- **CFI: Comparative Fit Index**
  - Also based on idea of NCP (χ^2 – df)
  - $$\text{CFI} = 1 - \frac{\max[(\chi^2_T - df_T), 0]}{\max[(\chi^2_T - df_T), (\chi^2_N - df_N), 0]}$$

    T = target model
    N = null model
  - From 0 to 1: bigger is better, > .90 = “acceptable”, > .95 = “good”

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

4 Steps in Model Evaluation

1. **Assess global model fit**

   - Recall that item intercepts, factor means, and variances are just-identified → *misfit comes from messed-up covariances*
   - χ^2 is sensitive to large sample size
   - Pick at least one global fit index from each class; hope they agree (e.g., CFI, RMSEA)
   - If model fit is not good, you should NOT be interpreting the model estimates
     - They will change as the model changes
   - If model fit is not good, it's your job to find out WHY
   - If model fit is good, it does not mean you are done, however… proceed to step 2

4 Steps in Model Evaluation

2. **Identify 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
  - Available via **RESIDUAL** output option in Mplus
  - Normalized as 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 creating a covariance
  - **Negative** residual → Items are **less** related than you predicted
    - Not as related as the 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

2. Identify localized model strain, continued…

- Another approach: **Modification Indices** (aka, voo-doo)
  - LaGrange Multiplier: decrease in χ^2 by adding the listed model parameter (e.g., cross-loading, error covariance)
    - Usually only pay attention if > 3.84 for df=1 (for $p < .05$)
    - Get expected parameter estimate for what's to be added – but 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)
  - 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
  - In R you can use the anova() function, which does this automatically
  - In Mplus you can use a spreadsheet Lesa built 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 **difference scaling correction** =

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

  - 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_{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_{H0} and larger model χ^2
  - e.g., constrain items to have same loadings → tau-equivalence
- When testing parameters that have a boundary (e.g., factor correlation ≠ 1?), then this test will be 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

- 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 sets of items**

- 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)

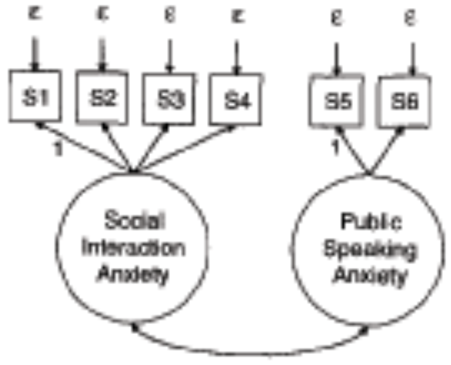
Fixing the Model by Expanding

- A common source of misfit is due to items that remain too correlated after accounting for their common factor

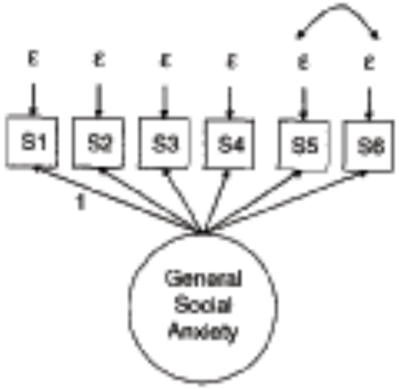
- Solutions for this:
  - Add **error covariance** (i.e., as suggested by voo-doo indices)
    - Is additive: Cov(y_1,y_2) = cov due to Factor + cov due to error covariance, so the error covariance basically plugs the hole in the covariance matrix
    - In models that do not allow error covariances, 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 multi-dimensionality**
      → Explicit acknowledgement that you have measured your factor + something else that those items have in common (e.g., stem, valence, specific content)

  - Lots of problematic pairings? **Re-consider dimensionality altogether**
    - I'd generally recommend against adding cross-loadings, because if the item measures more than one thing, it may not be useful (and will complicate interpretation of factors)

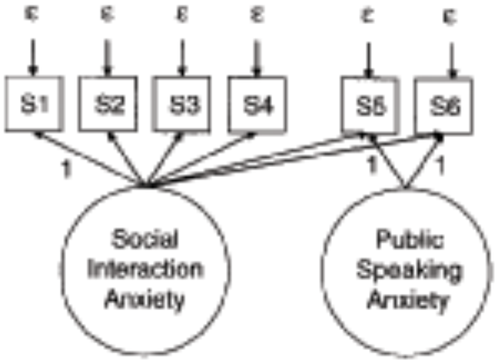
Equivalent Ways of Addressing Multidimensionality…



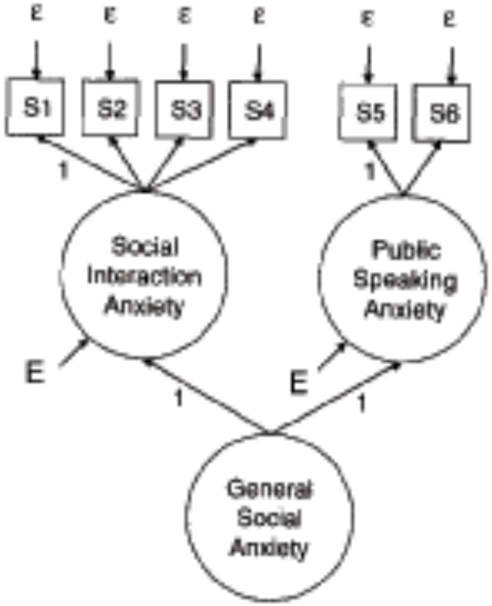
Error Covariance



Factor + Method Factor



Two-Factor Model



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 many different, yet statistically equivalent ways… error covariances really represent another factor (which is why you should be able to explain and predict them if you include them).

When to <u>Simplify</u> the Model

- Factors correlated > .85 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 isn’t measuring the construct, and you most likely 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 (but careful, since fewer items → less reliability)

- However – models with differing # items are NOT COMPARABLE AT ALL because their LL values are based on different data!
  - No model comparisons of any kind (including AIC and BIC)
  - To do a true comparison, you’d need to leave the item in the model but remove its loading (≈ original test of its loading)

What else can go wrong?

- Error message: “**non-positive definite**”
  - 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

- **Structural under-identification**
  - Does every factor have a metric 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 values)

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 suggests an "E/CFA" approach of estimating an exploratory-like model within a CFA framework:
  - Fix each factor variance to 1
  - 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: 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 messed-up covariances*
   - χ^2 is sensitive to large sample size, so pick at least one global fit index from each class (e.g., CFI, RMSEA); hope they agree
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 error 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

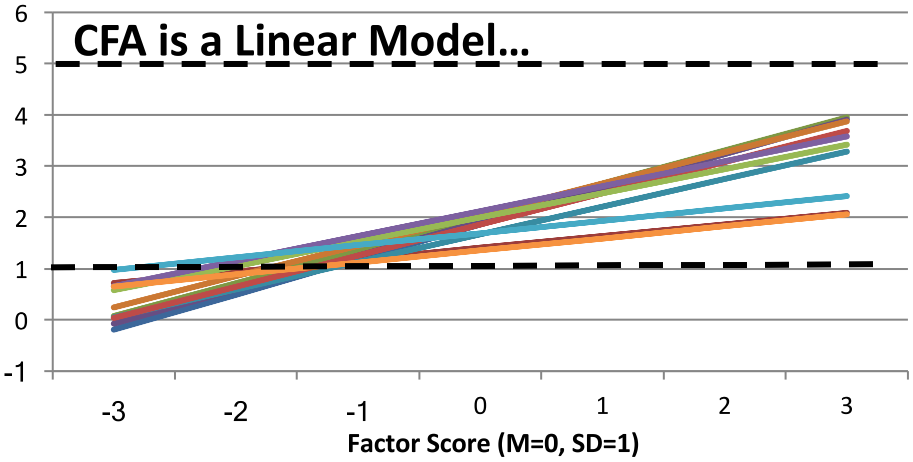
	y1	y2	y3	y4
y1	1			
y2	**.8**	1		
y3	**.8**	**.8**	1	
y4	**.8**	**.8**	**.8**	1

- **Good model fit does not guarantee a good model**
- **A good model has meaningful factor loadings**

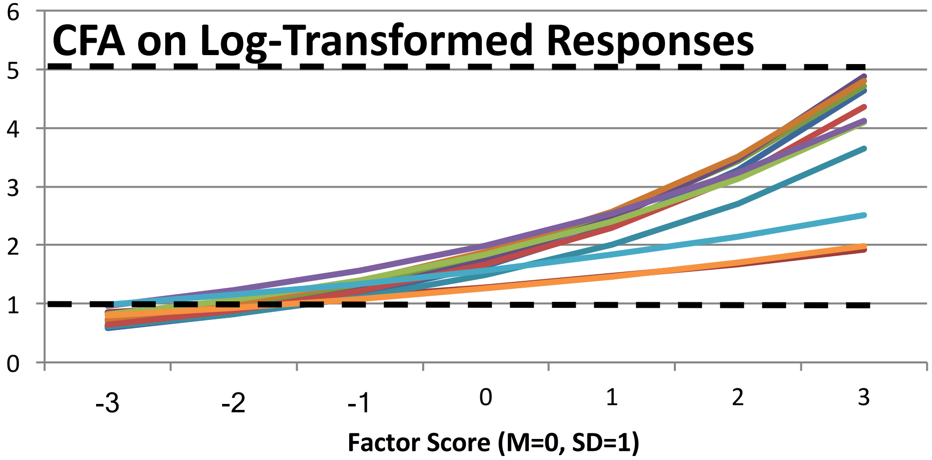
4 Steps in Model Evaluation: Step 3

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 .2 → 4% shared variance?
     - **Effect size (R^2 of item variance from factor) is practical significance**
   - Get SEs and *p*-values for unstandardized (and standardized) estimates (at least report SE from unstandardized)
     - 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
     - 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 levels
- If the predicted responses exceed the possible range, then the linear CFA may not fit (responses are not normal enough for CFA)



- Log-transforming the item response is a potential solution for positively skewed items (creates an exponential curve)
- Choosing an IFA/IRT model for ordinal responses instead is another option (stay tuned)

4 Steps in Model Evaluation: Step 4

4. Calculate item information and model-based reliability

- **Item Information =** $(\mathbf{unstandardized}\ \boldsymbol{\lambda})^2 / \mathbf{Var}(\mathbf{e})$
  → 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)

- **"Omega" Test Reliability =** $(\boldsymbol{\Sigma\lambda})^2\ /\ [(\boldsymbol{\Sigma\lambda})^2 + \boldsymbol{\Sigma}\ \mathbf{Var}(\mathbf{e}) +\ \mathbf{2\Sigma}(\mathbf{e\ cov})]$
  - → Squared sum of ***unstandardized*** 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)**
  - **Omega tells you the reliability for the sum of the items assuming the factor model is the underlying process that fits the data**

Testing CTT Assumptions in CFA

- **Alpha** is reliability assuming two things:
  - All factor loadings (discriminations) are equal, or that the items are "true-score equivalent" or "**tau-equivalent**"
  - **Local independence** (dimensionality now tested within factor models)

- We can test the assumption of **tau-equivalence** via a −2ΔLL comparison against a model in which the 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.
    Omega assumes unidimensionality, but not tau-equivalence
  - Research has shown alpha can be an over-estimate or an under-estimate of reliability depending on particular data characteristics

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

Model Fit: 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 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 improve the fit of the model or to simplify the model…
    - … But careful relying too heavily on modification indices to do so
  - Size and significance of model parameters matters, too
    - … How well are your factors really defined anyway?
    - 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 CFA model (so you need a nonlinear model)

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 responses
  - Factors are estimated as separate entities based on the observed covariances among items – factors represent testable assumptions
    - Items are unrelated after controlling for factors → local independence

- **Because item responses are included:**
  - Items are allowed to vary in discrimination (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 testing factor mean differences)
  - To make a test better, you need more items
    - **What kind of items? Ones with *greater information* → λ^2/Var(e)**
  - Measurement error is still assumed constant across the latent trait
    - **People low-medium-high in Factor Score are measured equally well**