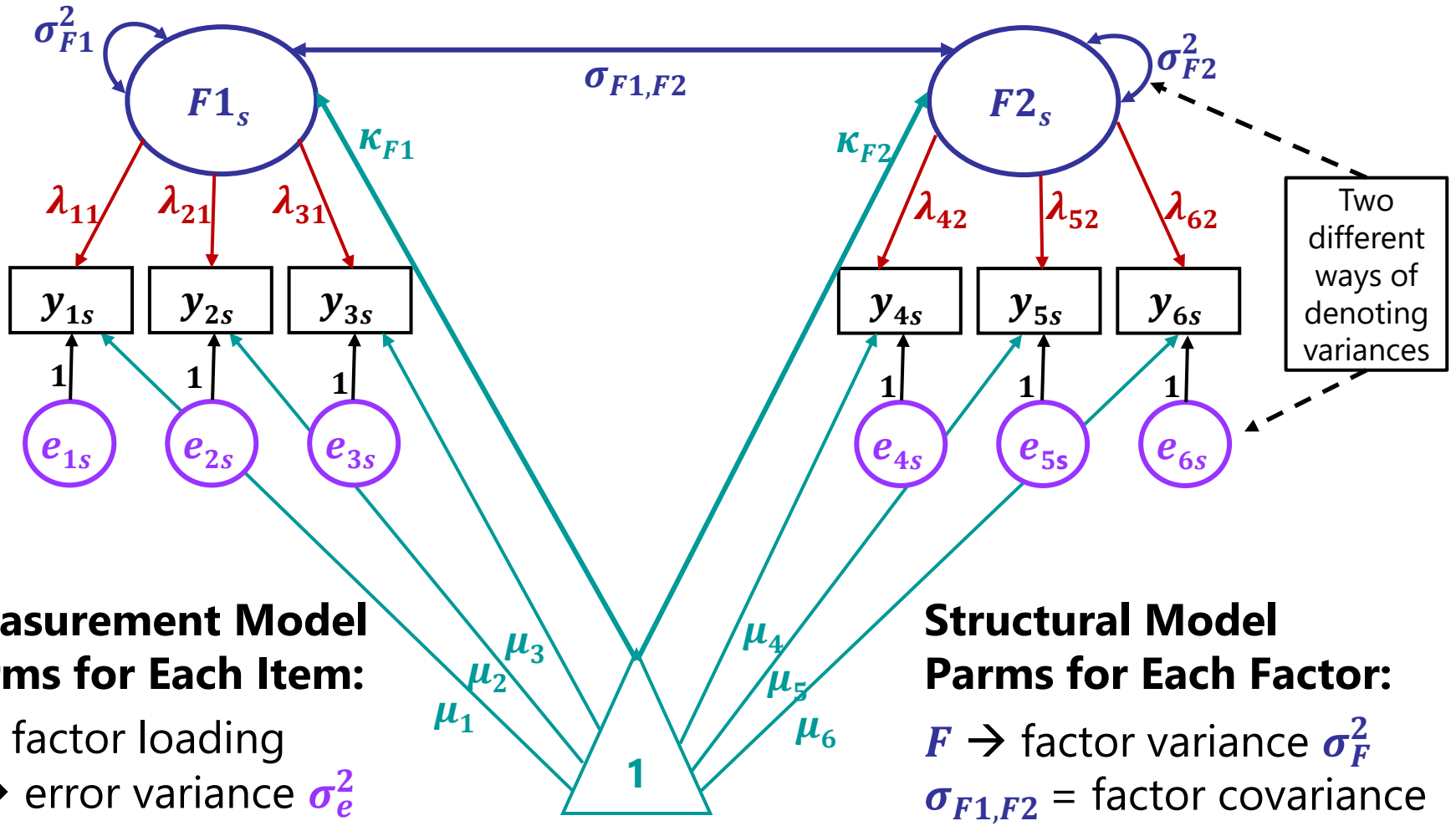
# Confirmatory Factor Analysis (CFA) Part 2

- Topics:
  - CFA model estimation
  - CFA model fit evaluation
    - Check global fit
    - Check local fit
    - Check viability of solution
    - Check effect size

# 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 Parms for Each Item:**

$\lambda$ = factor loading

$e$ → error variance $\sigma^2_e$

$\mu$ = intercept

**Structural Model Parms for Each Factor:**

$F$ → factor variance $\sigma^2_F$

$\sigma_{F1,F2}$ = factor covariance

$\kappa$ = factor mean

# Example "Congeneric" Two-Factor Model

- Measurement model for items ("indicators") 1–6 for subject *s*:
  - $y_{1s} = \mu_1 + \lambda_{11}F1_s + 0F2_s + e_{1s}$
  - $y_{2s} = \mu_2 + \lambda_{21}F1_s + 0F2_s + e_{2s}$
  - $y_{3s} = \mu_3 + \lambda_{31}F1_s + 0F2_s + e_{3s}$
  - $y_{4s} = \mu_4 + 0F1_s + \lambda_{42}F2_s + e_{4s}$
  - $y_{5s} = \mu_5 + 0F1_s + \lambda_{52}F2_s + e_{5s}$
  - $y_{6s} = \mu_6 + 0F1_s + \lambda_{62}F2_s + e_{6s}$

You decide **how many factors** and which items they predict ("congeneric" → diff item parms)

Unstandardized loadings ($\lambda_i$) are **linear slopes** predicting the item response ($y_{is}$) from the factors ($F_s$). **Thus, the model assumes a linear relationship between each factor and item response.**

Intercepts ($\mu_i$) are the expected item responses ($\hat{y}_{is}$) when *all factors predicting that item* = 0.

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

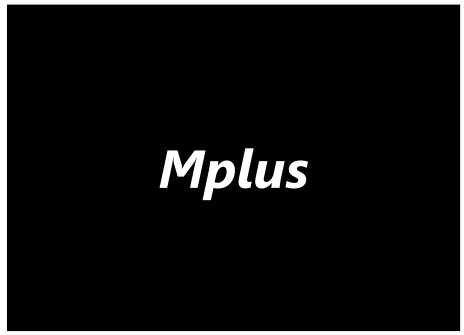
$$\mathbf{Y} = \mathbf{M} + \mathbf{\Lambda U} + \mathbf{E}$$

where $\mathbf{Y}$, $\mathbf{M}$, and $\mathbf{E}$ = $6 \times 1$ matrices (because each item gets one value of each); $\mathbf{\Lambda}$ = $6 \times 2$ matrix and $\mathbf{U}$ = $2 \times 1$ matrix (because two factors)

$\mathbf{M}$ = capital $\mu$, $\mathbf{\Lambda}$ = capital $\lambda$, and $\mathbf{U}$ holds the $F$ values (because $F$ is already capitalized and because random effects = factor scores)

# 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, and 3; $F_2$ measured by items 4, 5, and 6
  - If we fix both factors to have mean=0 and variance=1, then we need:
    **6 intercepts ($\mu_i$) + 6 factor loadings ($\lambda_i$) + 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 (assumed *missing at random* when using ML, which is conditionally random)

- What about the all the individual subject **factor scores**?
  - The subjects' factor scores are NOT part of the model—in other words, **factor scores are RANDOM effects** assuming a multivariate normal distribution
  - So we need the **factor means, variances, and covariances** as sufficient statistics, but **we don't need** the factor scores for the **individual subjects**

# The End Goals of Maximum Likelihood (ML) Estimation

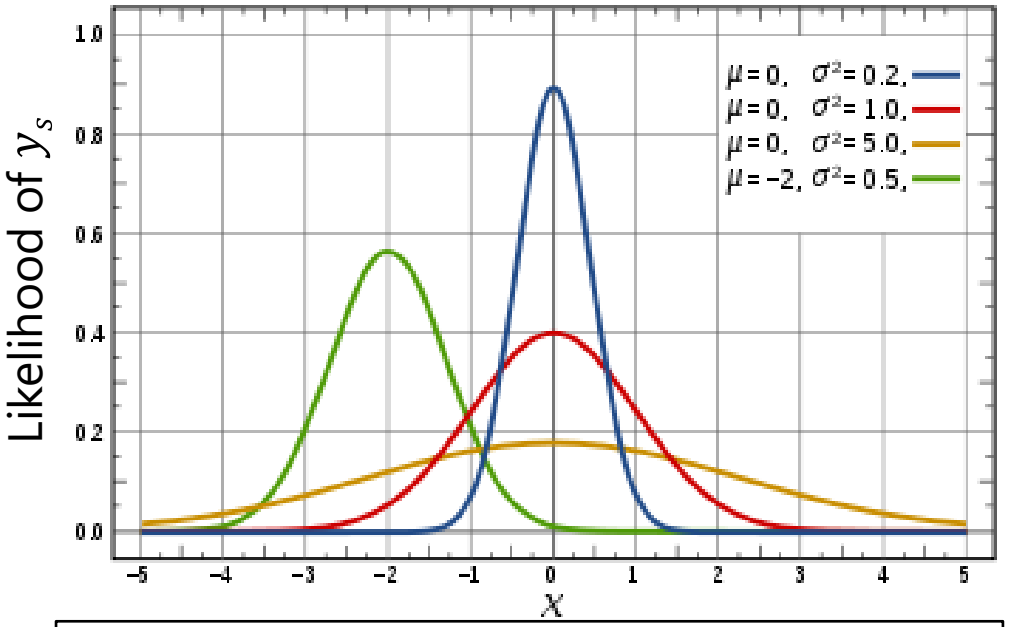
1. Obtain "most likely" values for each unknown parameter in our model (intercepts, loadings, error variances, factor means, factor variances, factor covariances) → the answers → **the estimates**

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:

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

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:

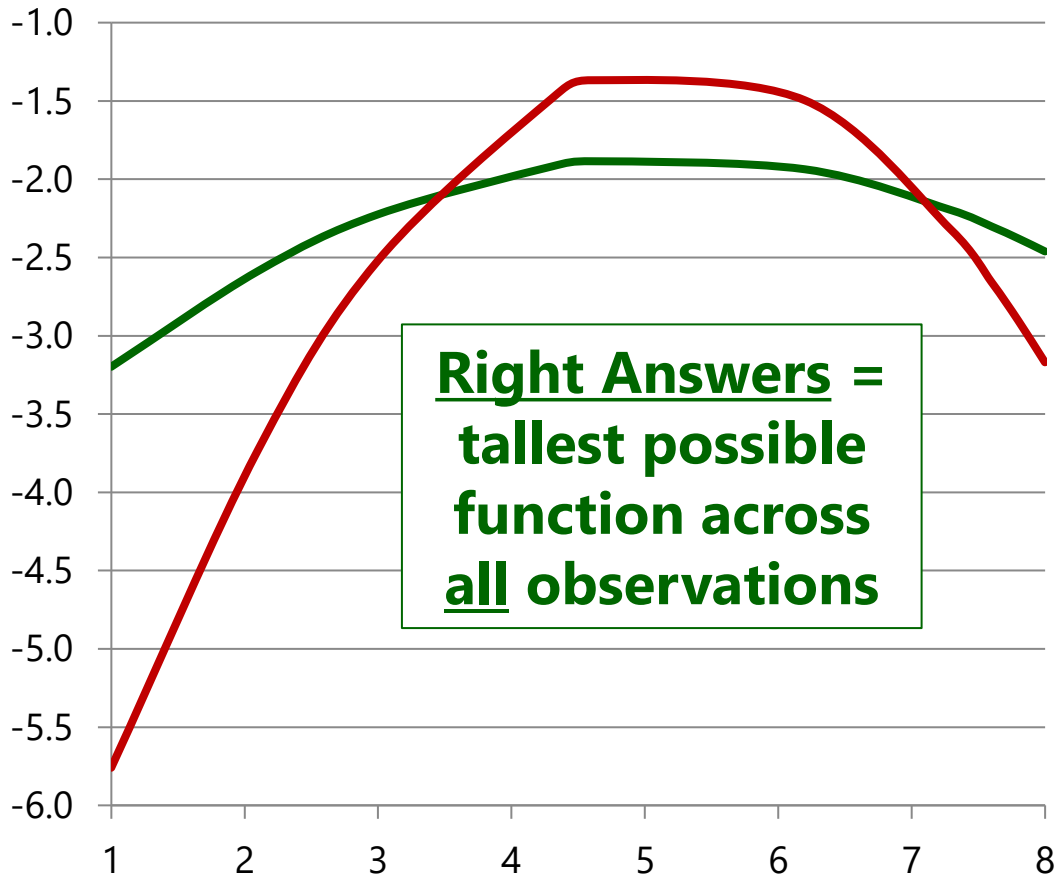
$$\boldsymbol{y_s = \beta_0 + e_s}$$

- We can check our math via software using ML!

# ML via Excel NORMDIST

**Key idea: Normal Distribution formula → data height**

| | | |
|---|---|---|
| **Mean** | **5.19** | **5.24** |
| **Variance** | **6.56** | **2.00** |
| | **Right** | **Wrong** |
| **Outcome** | **Log(Height)** | **Log(Height)** |
| **1.0** | **-3.20** | **-5.76** |
| **2.1** | **-2.59** | **-3.73** |
| **3.0** | **-2.22** | **-2.52** |
| **4.3** | **-1.92** | **-1.49** |
| **4.6** | **-1.89** | **-1.37** |
| **6.2** | **-1.94** | **-1.50** |
| **7.3** | **-2.20** | **-2.33** |
| **7.6** | **-2.30** | **-2.66** |
| **7.8** | **-2.38** | **-2.90** |
| **8.0** | **-2.46** | **-3.17** |
| **SUM = Model LL = taller is better** | | |
| | **-23.09** | **-27.42** |




For a demo in excel, download this file from PSQF 6270: Generalized Linear Models

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

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

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

- 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 $\mathbf{M}$)
- CFA model also makes $\boldsymbol{\Sigma}$, the **recreated** item variance–covariance matrix:
  - 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: $\boldsymbol{Cov(y_1, y_6)} = \boldsymbol{\lambda_{11}} * \boldsymbol{Cov(F_1, F_2)} * \boldsymbol{\lambda_{62}}$

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

- 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 subjects:

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

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

- 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" $\boxed{\text{Each parameter SE} = \frac{1}{\sqrt{\text{information}}}}$
  - So steeper function = more information = more precision = **smaller SE**

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

- Use your model $LL_{H_0}$ from predicting $\boldsymbol{\Sigma}$ → 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 recreations are from the real data moments → $\chi^2$:

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

- 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:

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

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

# What about item non-normality?

- The use of this multivariate normal ML function assumes:
  - Subjects 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 errors ($\boldsymbol{e_{is}}$) 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 $\chi^2$-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 $\chi^2$)
  - Scaling factor < 1.000 = platykurtosis (too-thin tails; fixes too-small $\chi^2$)

- **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 (see Enders 2010 ch. 5)
  - Leptokurtosis (too-fat tails) → increases information; fixes too small SEs
  - Platykurtosis (too-thin tails) → lowers information; fixes too big Ses
  - See Enders (2010 ch. 5) for a readable explanation of how MLR differs from ML

- 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. 18)

- 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:
    - vce(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)
    - vce(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…

# 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 composite model equation: $\boldsymbol{y_{is} = \mu_i + \lambda_i F_s + e_{is}}$
  - **Mean** per item: **Captured** by unique item intercept $\boldsymbol{\mu_i}$ (and factor mean)
    - Not a source of misfit (unless constraints are applied on the intercepts)
  - **Variance** per item: **Captured** by weighted factor + unique item 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 usually a source of misfit (whatever $\boldsymbol{F_s}$ part doesn't get, $\boldsymbol{Var(e_i)}$ picks up to get to total $\boldsymbol{y_i}$ variance)
  - **Covariance** among items: **Attempt to recreate** 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 (MLR here)

```
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
```

# Free parameters = # estimated

$\boldsymbol{H_1}$ Saturated (Unstructured) Model Height from this →

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

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

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

## 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 comparisons are already given in R Lavaan output (ML & MLR)

```
Estimator                                      ML      Robust
Minimum Function Test Statistic           427.937     307.804
Degrees of freedom                              9           9
P-value (Chi-square)                        0.000       0.000
Scaling correction factor                                1.390
  for the Yuan-Bentler correction (Mplus variant)
```

This version of the "minimum function test statistic" is $\chi^2$ from $-2\Delta LL$ test of your chosen $\boldsymbol{H_0}$ model vs. saturated $\boldsymbol{H_1}$ model

```
Model test baseline model:

Minimum Function Test Statistic           1981.034    1128.693
Degrees of freedom                              15          15
P-value                                     0.000       0.000
```

*[other fit indices omitted for now]*

```
Loglikelihood and Information Criteria:
Loglikelihood user model (H0)          -11536.404  -11536.404
Scaling correction factor                                1.416
    for the MLR correction
Loglikelihood unrestricted model (H1)  -11322.435  -11322.435
  Scaling correction factor                              1.407
    for the MLR correction

Number of free parameters                     18          18
  Akaike (AIC)                         23108.808   23108.808
  Bayesian (BIC)                       23198.912   23198.912
  Sample-size adjusted Bayesian (BIC)  23141.739   23141.739
```

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

## Independence (Null) Model

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

$\boldsymbol{H_1}$ Saturated (Unstructured) Model Height from this →

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

# Free parameters = # estimated

# 4 Steps in Assessing Model Fit

1. **Global model fit**
   - *Does the model "work" overall: Does it adequate recreate the observed item means, variances, and covariances?*
   - Recent research has indicated that "standard" cut-offs by which to judge global model fit may not work for all data analyses…

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 predicted item responses 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 $\boldsymbol{\chi^2 = 2 * N * F_{ML}}$
  - $\chi^2$ is evaluated based on model DF (# parameters left over)
  - Tests null hypothesis that $\boldsymbol{\Sigma = S}$ (that model = data → perfect fit), so **significance is bad** (i.e., smaller $\chi^2$, 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 $\chi^2$ to index model fit is usually insufficient, however:
    - $\chi^2$ depends largely on sample size (is overpowered with large $N$)
    - Is "unreasonable" null hypothesis (perfect fit, really???)
    - Btw, $\chi^2$ fit test is only possible given balanced data (as typical for CFA)

- Because of these issues, additional fit indices are usually used along with the $\chi^2$ test (that function like effect sizes for fit)
  - **Absolute** Fit Indices (besides $\chi^2$)—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" (discrepancy) 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 $\chi^2$ 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 Model 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 Model DF left (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 terrible baseline, 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

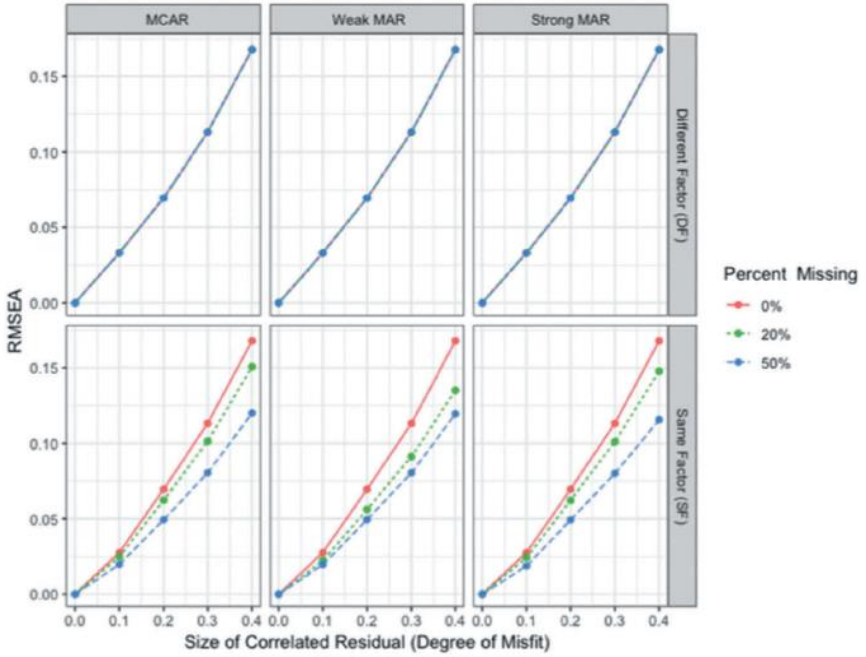
1. **Assess global model fit (summary)**
   - Recall that item intercepts → item means, and item error variances → item variances, so *misfit comes from badly recreated covariances*
   - $\chi^2$ is sensitive to large $N$, so pick at least one global fit index from each class; hope they agree (e.g., CFI, RMSEA) that fit is "good"

- Conventions of "good" absolute model fit largely stem from simulation studies reported in Hu & Bentler (1999)
  - Been cited >100,000 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)

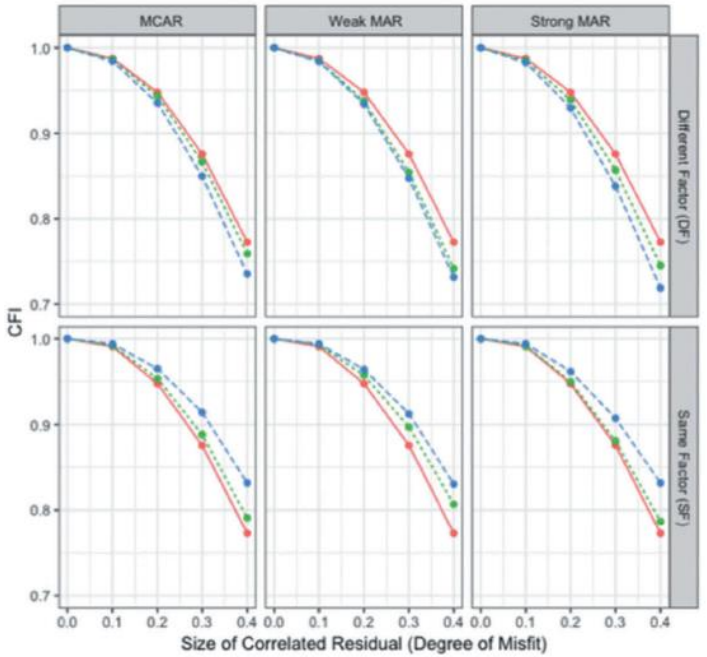
# Good Fit is Easier with Missing Data

- Zhang & Savalei (2020): Cases that don't have responses to 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

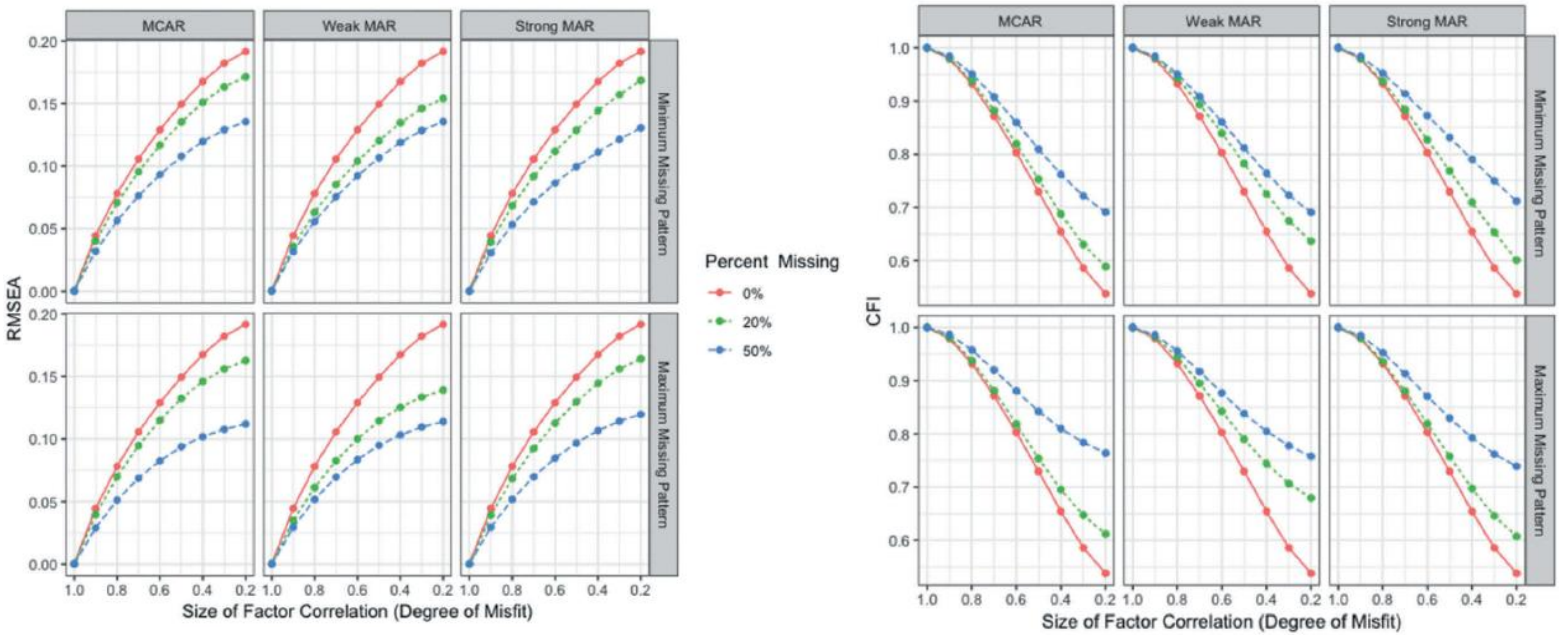


# Good Fit is Easier with Missing Data

- 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

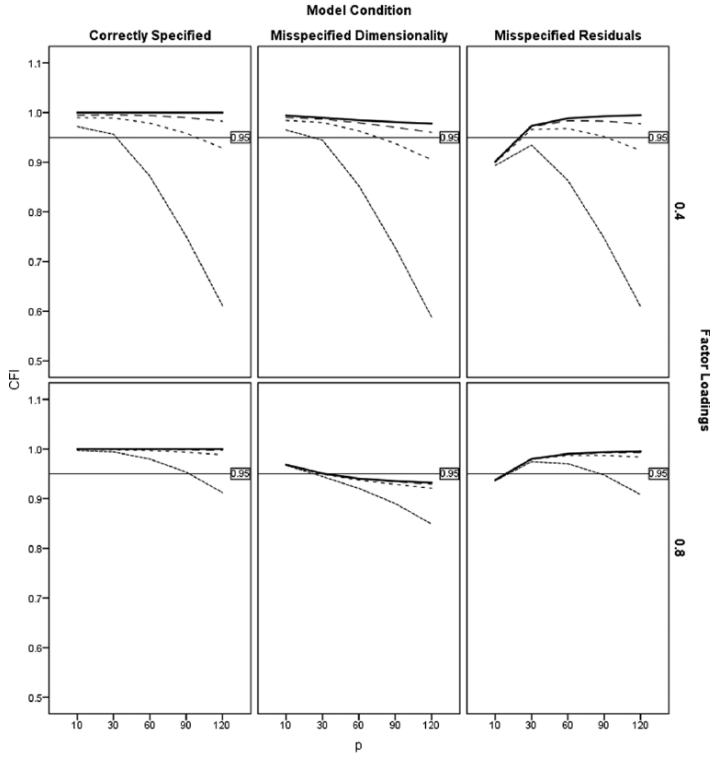
CFI also gets less worse with more mis-specification with greater amounts of missing indicators with more relevance



# Good Fit by Number of Indicators…

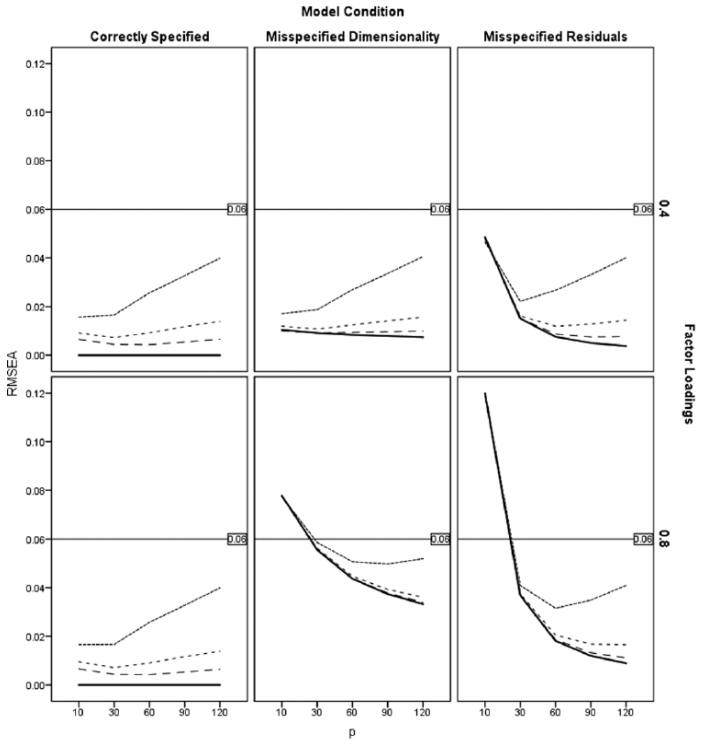
- …It's complicated… see [Shi, Lee, & Maydeu-Olivares (2019)]
  - Figures 1 and 3: Effects of # indicators for $N$=200, 500, 1000, and population

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 $N$

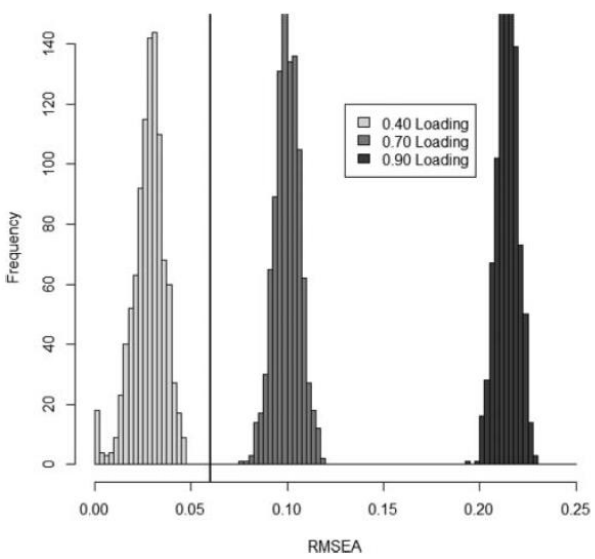
Incorrect models: RMSEA gets better with more indicators (less so with small $N$)

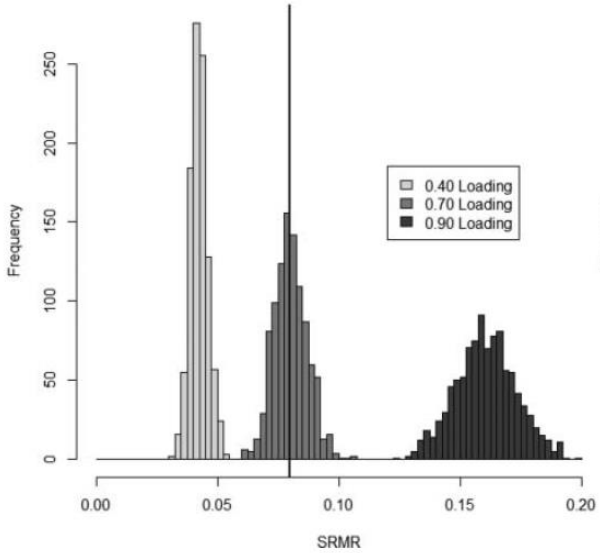


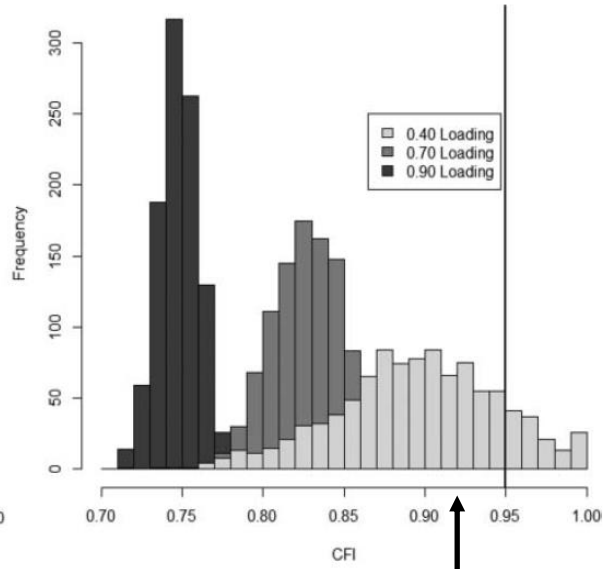

Right: Mis-specified residuals (errors) → misfit limited to only some indicators (so having more properly specified indicators makes fit better on average)

# 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



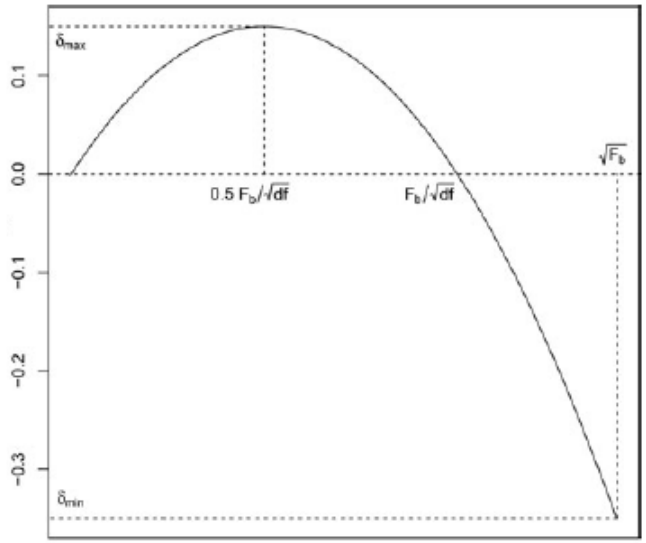




Wide variability in CFI with $\lambda = .40$!

# When Fit Indices Disagree

- *Opposite pattern also found for CFI using more incorrect models: CFI was lower (worse fit) with lower reliability (Heene et al. 2011)
- 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 & Green (2016)
- 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

- In summary: How to determine reasonable cutoffs can be tricky… see West et al. (2023) and this "dynamic" simulation approach

# 4 Steps in Model Evaluation

1. **Assess global model fit (summary)**
   - Recall that *misfit generally comes from badly recreated 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 parameters, because they will change as the model specification 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
  - Good global model fit means that the observed and recreated item means, variances, and covariances aren't too far off on the whole… this doesn't guarantee each specific covariance is recreated well

# 4 Steps in Model Evaluation: Step 2

## 2. Identify local misfit: localized model strain

- What is the discrepancy ("residual") between each model-recreated and data-observed item mean, variance, and covariance?
- Should inspect **normalized model residuals** for that
  - **RESIDUAL** output option in Mplus, residual () in R lavaan, or ESTAT RESIDUAL in STATA
  - "Normalized" is residual/SE → **works like a z-score**
  - Relatively large absolute values indicate "localized strain" (misfit)
  - **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 how to fix them… (positive is easier than negative)**

# 4 Steps in Model Evaluation: Step 2

2. Identify localized model strain, continued…

- Parallel info: **Modification Indices** (*aka*, cheat codes)
  - LaGrange Multiplier: decrease in model fit $\chi^2$ by adding the listed model parameter (e.g., cross-loading, error covariance)
    - Usually only pay attention if $\chi^2 > 3.84$ 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 these "manipulation indices" can only try to repair your current model; **they will never suggest a new factor structure instead** (that's your job)!

# 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 $\chi^2$, and comparing saturated $H_1$ to "null" model
  - Implemented via direct difference in model $\chi^2$ 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" because it has extra steps to incorporate the scaling factors for each model to be compared
  - I built you a spreadsheet for this when using Mplus… you're welcome ☺
  - Can also use anova() in R lavaan (not sure about LRTEST in STATA)

# 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 $\chi^2$ 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 (scale factor of 1 = regular ML)

Note: Your LL will always be listed as the $\boldsymbol{H_0}$ ($H_1$ 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 $H_0$ and smaller model $\chi^2$
  - 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 $H_0$ and larger model $\chi^2$
  - 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 $p < .05$ (i.e., a mixture $\chi^2$ distribution)
  - Same problem as when testing new random effect variances in MLM

# 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 $-2\Delta$LL test is not applicable given same DF

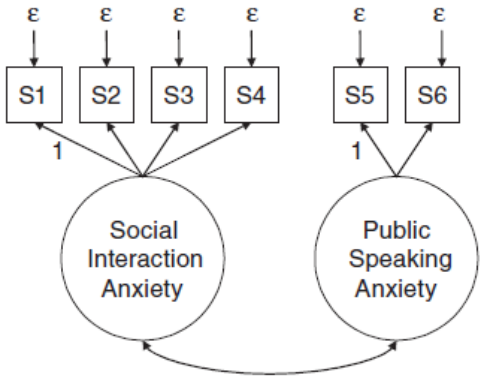
- Use information criteria instead: **AIC** and **BIC**
  - Akaike IC: AIC = $-2$LL + 2*#parameters
  - Bayesian (Schwartz) IC = $-2$LL + log($N$)*#parameters
  - Are NOT significance tests, just "smaller is better", is "evidence"
  - **Still cannot be used on models with different items (outcomes)**

- 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)

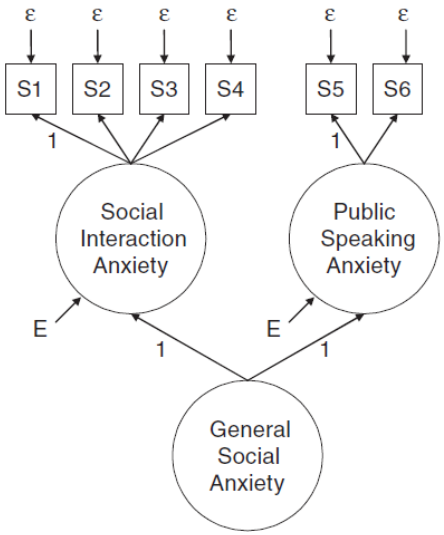
# Fixing the Model by Expanding

- 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 cheat code mod indices)
    - Is then a fourth type of measurement model parameter
    - Is additive: $Cov(y_1, y_2)$ = cov due to Factor + cov due to error covariance, so the error covariance plugs any hole in the recreated 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 interpretation of the factors

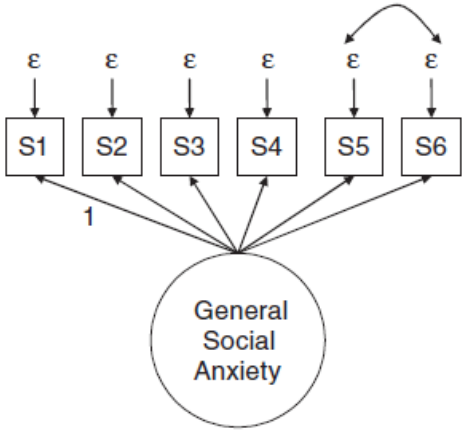
# 4 Equivalent Ways of Addressing Multidimensionality… (Brown, 2015 p. 181)



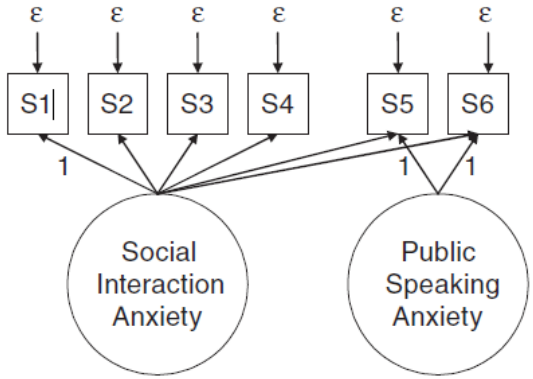

Two-Factor Models




Error Covariance




Factor + Method Factor




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

The extra relation between the two public speaking items can be modeled in different, yet statistically equivalent ways… because **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 $\boldsymbol{\Sigma}$ (model-recreated) 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 relate to its factor???
- **Empirical under-identification**
  - More likely with smaller sample sizes, fewer indicators per factor, and items with low reliabilities ($R^2$ attributable to the factor)

# Open in case of emergency…

- If good model fit seems hopeless, you may choose to go back to the exploratory 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 (or those most likely measured)
  - 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
  - Ideally, any model modifications should be replicated in a new sample!

# 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 badly recreated covariances*
   - $\chi^2$ is sensitive to large sample size, so pick at least one global fit index from each class (e.g., CFI, RMSEA); use cutoffs with caveats
2. **Identify localized model strain**
   - Good global model fit means that the observed and recreated item means, variances, and covariances aren't too far off on the whole… this doesn't guarantee each specific covariance is recreated well
   - Consider normalized residuals and modification indices to try and "fix" your 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 |

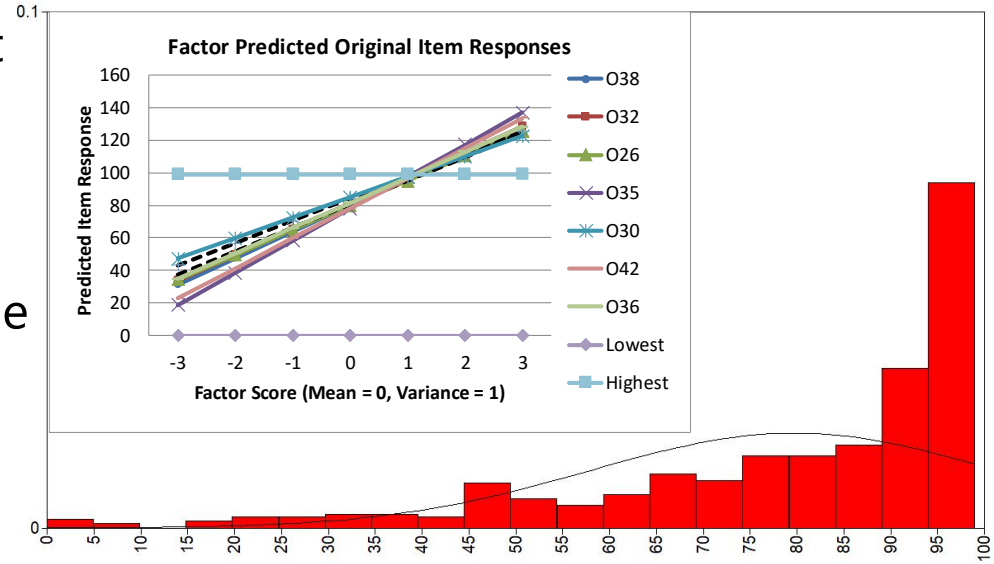
   - **Good model fit does not guarantee a good model**
   - **A good model has meaningful factor loadings**
   - **If your items are not correlated, game over, regardless of fit**

# 4 Steps in Model Evaluation: Step 3

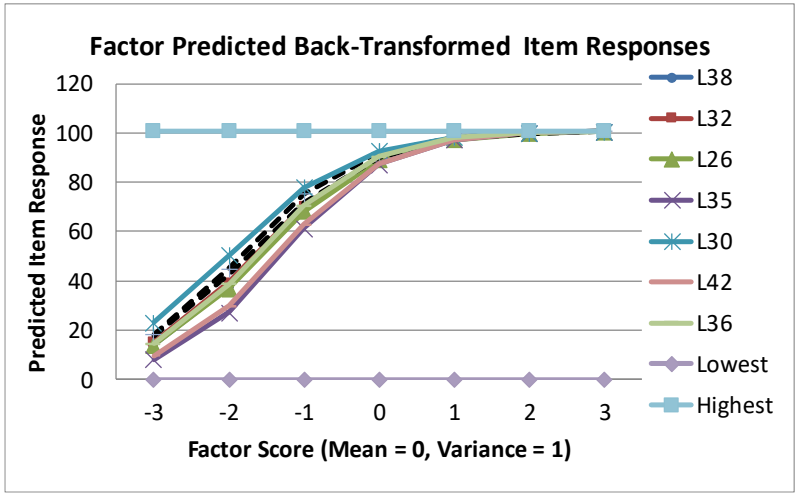
3. Inspect **parameter effect sizes** and significance
   - Model fit does not guarantee meaningful factor loadings
     - Can reproduce lack of covariance quite well and still not have anything useful—e.g., factor loading of 0.2 → 4% shared variance?!?
     - **Effect size ($R^2$ of item variance due to 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 (useful to assess 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 linear model, so you can plot the responses predicted from the unstandardized item intercepts and loadings (slopes) across factor values
- If predicted responses exceed possible range within ±2 SD of factors, then **a linear CFA may not be appropriate**




- CFA 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 $y_{is} = L + (U - L)\frac{\exp(Logit)}{1+\exp(Logit)}$
- Alternatively, CFA using a beta distribution!
- For ordinal responses, an IFA/IRT model is a better option (stay tuned)



# 4 Steps in Model Evaluation: Step 4

4. Calculate item information and model-based reliability

- **Item Information = (unstandardized $\lambda_i$)$^2$ / $\sigma^2_{e_i}$**
  → 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:** $\omega = \frac{\sigma^2_F * (\Sigma\lambda_i)^2}{[\sigma^2_F * (\Sigma\lambda_i)^2] + \Sigma\sigma^2_{e_i} + 2\Sigma(\text{e cov})}$
  → 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)**

# 4 Steps in Model Evaluation: Step 4

- **Omega** is an estimate of **reliability for a sum score** (as is alpha)
  - But why would you want to use a sum score after you just built a better measurement model for the latent factor??? That's the entire point of SEM—to estimate relations directly among latent variables!
- If you want to use a factor score instead of a sum score, then you should report **factor score reliability** instead of omega:
  - Factor score reliability $= \frac{\sigma_F^2}{\sigma_F^2 + \mathbf{SE}_{\mathbf{FS}}^2}$

$\sigma_F^2$ = factor variance from model solution
$\mathbf{SE}_{\mathbf{FS}}^2$ = error variance of EAP factor scores
denominator = total variance = T/(T+E)

---

- Either way, if you must simplify your model, you can use the sum score or a factor score as a single indicator of a latent variable, but still correct for its unreliability as follows (example using Mplus syntax):
  - `Factor BY fscore@1; fscore* (ResVar); Factor*;`
    `[fscore*]; [Factor@0]; ! To mean-center latent factor`
  - `MODEL CONSTRAINT: Resvar =` $(1 - \mathbf{reliability}) * (\mathbf{total\ variance})$`;`
    `! Factor score residual variance = "unreliable" amount of total variance`
  - **Example using R lavaan syntax:**

```
Factor =~ 1*fscore; fscore ~~ ResVar*fscore; Factor ~~ Factor;
fscore ~ 1; Factor ~ 0; # Mean center latent factor
ResVar := (1-reliability)*(total variance)
```

# CFA Model Evaluation: Summary

- The primary advantage of working in a CFA framework is obtaining indices of global and local model fit
  - $\chi^2$ and model fit indices indicate how well the model recreates the item means, variances, and covariances from the original data…
    - … But normalized residuals should still be examined for evidence of local misfit (e.g., badly recreated 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 don't rely blindly on modification indices (cheat codes) 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** assumes 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 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 → **sum scores ≠ factor scores!**
  - 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 treated as separate entities based on the observed covariances among items—latent factors create empirically testable assumptions
    - Items are unrelated after controlling for factor(s) → local independence
    - Extra item relations can be modeled 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, but not just any kind!
    - **What kind of items? Ones with *higher standardized loadings* (more information)**
  - Measurement error is still assumed constant across the latent trait
    - **People low-medium-high in Latent Factor are measured equally well**