The Finale*: Structural Equation Modeling (SEM)

- Topics:
  - The Big Picture of SEM
  - "Predictors" versus "Outcomes"
  - What to do (and what NOT to do) if SEM breaks for you
    - Parceling indicators
    - Using single indicators (sum or factor scores)
    - Multiple plausible values of factor scores

* *For treatment of and examples using path analysis and mediation, see [Lecture 6, Example 5a Part 2, Examples 6a, 6b, and 6c from PSQF 6270]*

Structural Equation Modeling (SEM)

- The term “SEM” gets used to describe many different types of models, but fundamentally, **SEM consists of two distinct parts**:
  - **Measurement model: Mapping of observed indicator outcomes to the latent variable(s) they measure (to create better, “latent” constructs)**
    - “CFA” if indicators are continuous and “normal enough”
    - “IFA” (or “IRT” or “CFA for categorical outcomes”) if indicators are binary or ordinal
    - “IRT” if indicators are nominal (no limited-information version available)
    - “?name?” if indicators require some other link function (e.g., counts)
    - Factors/thetas/traits are (usually) assumed to be multivariate normal
  - **Structural Model: Path analysis using those MVN latent variables**
    - And using **other observed variables** that are not used as part of the measurement model for those latent variables
    - Other **observed variables can be of whatever kind**, so long as the observed outcomes have their distributions modeled properly
      - e.g., a binary predictor variable (i.e., not in the likelihood) does not require a logit, but a **binary outcome variable** does (so then it’s on CATEGORICAL or ORDERED statements)
      - You must **create your own contrasts to include categorical predictors** in Mplus (i.e., there is no “CLASS variable” as in SAS, “factor variable” as in R, or “i. variable” as in Stata)

SEM: Model Identification

- SEM integrates both measurement and path models, so the identification rules for SEM borrow from both
  - Measurement models for each latent variables must be locally identified → each factor has its own scale (mean and variance)
  - The overall model must be identified (mathematically solvable)

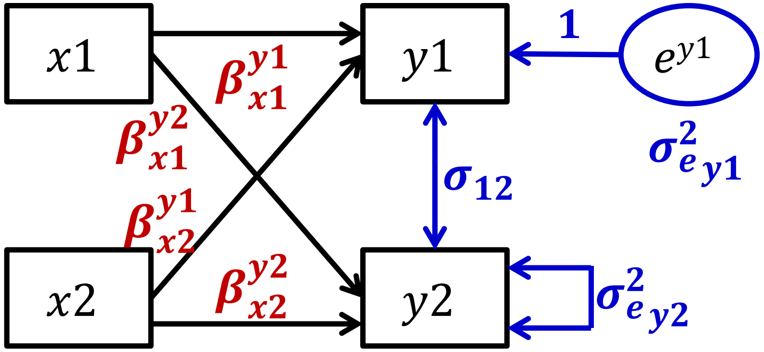
- A necessary (but not sufficient) way of ensuring identification is the "t-rule" (i.e., a counting rule that I never use in SEM)
  - Number of estimated ("free") parameters must be less than the total number of means + variances/covariances of **all** observed variables (v) in the likelihood: Total possible DF = $\frac{v*(v+1)}{2} + v$
  - Practical tip: Don't count, just look at your structural model, and see if it seems logical (e.g., don't have a directed path AND a covariance between two variables), make sure all latent factors are locally identified, and beware of negative factor loadings (because then the factors won't know which way is up)

New (and Confusing) Terminology

- **Predictors** are known as **exogenous** variables (X-ogenous to me)
- **Outcomes** are known as **endogenous** variables (IN-dogenous to me)
- Variables that are both at once are called **endogenous** variables

An example path model:
2 exogenous variables ($x1$ and $x2$)
2 endogenous variables ($y1$ and $y2$)

$$y1_i = \beta_0^{y1} + \beta_{x1}^{y1}(x1_i) + \beta_{x2}^{y1}(x2_i) + e_i^{y1}$$
$$y2_i = \beta_0^{y2} + \beta_{x1}^{y2}(x1_i) + \beta_{x2}^{y2}(x2_i) + e_i^{y2}$$

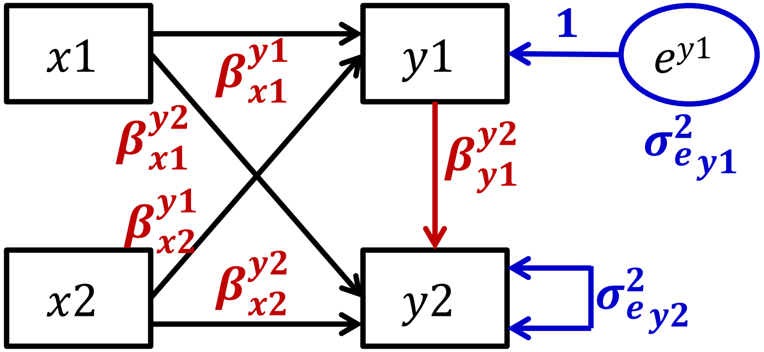


New Mplus code under MODEL:

```
y1 y2 ON x1 x2; y2 WITH y1;
```

A modified example path model:
$y1$ predicts $y2$ (still endogenous)

$$y1_i = \beta_0^{y1} + \beta_{x1}^{y1}(x1_i) + \beta_{x2}^{y1}(x2_i) + e_i^{y1}$$
$$y2_i = \beta_0^{y2} + \beta_{x1}^{y2}(x1_i) + \beta_{x2}^{y2}(x2_i)$$
$$+\beta_{y1}^{y2}(y1_i) + e_i^{y2}$$



New Mplus code under MODEL:

```
y1 y2 ON x1 x2; y2 ON y1;
```

New (and Confusing) Terminology

- What parameters get estimated for exogenous "predictor" and endogenous "outcome" variables differs importantly by program!
  - Only the intercepts, residual variances, and residual covariances of "outcome" variables are estimated as part of the likelihood…
  - But what each program considers an "outcome" depends on estimation!

- By default **in Mplus and in R lavaan** (v 0.6-10+), *truly* exogenous predictor variables cannot have missing data, as in any model
  - Cases with missing predictors are **listwise deleted** (incomplete data then are assumed missing completely at random), no matter which estimator!
  - Because *truly* exogenous predictors are not part of likelihood function
    - Log-likelihood (LL) contains $\hat{y}_i$ for each person and σ_e^2 for each outcome
    - So (conditional) LL can't be calculated without the predictors that create each $\hat{y}_i$
  - But truly exogenous predictors also do not have assumed distributions…
    - Good when you have non-normally-distributed predictors (e.g., ANOVA)!

"Predictors" as Endogenous Outcomes

- **What??? I thought full-information ML allows missing data???**
  - NO: only endogenous outcomes can be incomplete (then assumed missing at random, which means *random only after conditioning on model variables*)
  - Btw, you can add other variables into the likelihood—but not the model—to help (untestable) missing at random assumption using AUXILIARY option
    - Is a "saturated correlates" approach (they just covary with all outcomes)

- **Mplus and R lavaan** each allow a work-around: **Bring exogenous predictors into the likelihood** by listing their means, variances, or covariances as parameters → **predictors then become "outcomes"**
  - Even if nothing predicts the predictor (i.e., it's not *really* a model outcome); you are just estimating an empty model for the predictor as an outcome
  - Incomplete "endogenous predictors" can be included assuming missing at random (MAR), but they also then have distributional assumptions (MNV)
    - Historically Mplus has not let endogenous predictors have other distributions, so you may have to make non-normal predictors an outcome of something else
    - But there may be ways to trick it in doing this that I haven't found yet…

“Predictors” as Endogenous Outcomes

- **SAS CALIS** and **STATA SEM** both default to limited-info ML (uses listwise deletion and assumes MVN for ALL variables), but <u>both can do full-info ML</u>
  - SAS CALIS: full-info via “FIML” (robust “MLMB” does not allow missingness)
    - Can add variables into the likelihood but not the model (as “saturated correlates”) using the AUXILIARY option to help (untestable) missing at random assumption
  - STATA SEM: full-info via “MLMV”; can add “robust” SEs to mimic robust ML
    - No syntax to set up saturated correlates as AUXILIARY variables directly (I think)

- But using **full-info ML FORCES the exogenous predictors into the likelihood**—they are treated as endogenous outcomes whose means, variances, and covariances are estimated as model parameters
  - So incomplete endogenous predictors can then be included assuming <u>missing at random</u> (MAR), but they also then have <u>distributional assumptions</u> (MVN)
  - <u>STATA SEM</u> “xconditional” default computes predictor means, variances, and covariances from the data to save time if complete data (or searches for them with “noxconditional” option, which it invokes on its own when needed)
  - What happens for generalized models in STATA GSEM? Stay tuned…

Reconciling Confusing Vocabulary

- As we've seen, the distinction of "predictor" and "outcome" is no longer as clear-cut as in general(ized) linear models (i.e., univariate regression)
  - Because in path models a variable can be both a predictor and an outcome at the same time! In that case, it's an outcome

- Likewise, the distinction of "exogenous" from "endogenous" (as traditionally used in path models) is not really clear-cut
  - In theory, predictors are exogenous and outcomes are endogenous…
  - …But in practice, that depends on what your software is doing!

- New, more comprehensive rule: **Is a variable in the likelihood**?
  - YES, if its means, variances, or covariances are model parameters
  - YES, if it's only a predictor but you are using full-info ML in SAS CALIS or STATA SEM
  - **IF YES, then I will call it an "outcome":** incomplete cases can then be included (with missing data assumed missing at random), but this flexibility comes at the (potential) cost of assuming a multivariate normal conditional distribution
  - **IF NO, then I will call it a "predictor":** it's not in the likelihood, so cases with incomplete predictors will be dropped, but then no distribution is assumed

SEM: What goes into model fit

- For CFA/IFA only, misfit is almost always due to covariances
  - If each indicator has its own **intercept or thresholds**, then the indicator **means or response category frequencies** will be predicted perfectly
  - If each (conditionally normal) indicator has its own **residual variance**, then the indicator **total variance** will (usually) be predicted perfectly
  - **Factor loadings** are supposed to predict covariances among indicators, so once you have 4+ indicators in a model → **potential for misfit**

- The same is true in SEM, but with a catch, because only some covariances “count” towards model fit in Mplus (and R lavaan)
  - Covariances among “outcomes” (in the likelihood) COUNT
  - Covariances for “predictors” (NOT in the likelihood) with “outcomes” (in the likelihood) COUNT
  - Covariances among “predictors” (NOT in the likelihood) do NOT count

SEM: What to do first?

- **Because SEM is composed of two distinct parts…**
  - Measurement model that maps latent variables onto observed indicators
  - Structural model for relations involving those latent variables

- … **you should build these models <u>sequentially</u>**
  - Start by ensuring each over-identified factor fits adequately
  - **THEN combine** all latent factors and other observed variables in the same model, **estimating all possible relation**s among them (this "saturated" model will be the best-fitting structural model)
    - Helpful to phrase all associations as covariances first **to see bivariate** relations
    - Local misfit will likely only be due to cross-construct mis-predicted covariances (remedy before continuing, creating a new saturated structural model if needed)
  - Then modify the structural model to answer your questions, and see if any simpler model is **NOT worse** than the saturated structural model
    - Can then change to **regression paths to examine unique relations**
    - Will be a nested model only if not all structural relations are directly included

- Because the measurement model will dominate DF for model fit, informative tests of the structural model need to focus THERE only

SEM: What could I do instead?

- A simultaneous estimation of measurement and structural models in SEM is the gold standard, but may not work for you

- SEM is likely to break (i.e., not converge, give crazy SEs) when:
  - Sample sizes are small (few persons relative to # estimated parameters)
  - Many estimated parameters (especially with few persons)
  - Some outcomes are non-normal (link functions are needed)
  - Many latent variables are included (especially with link functions)
  - Latent factors are not locally identified (two indicators is a bad idea)
  - Latent variable interactions are included (which require numeric integration → repeated rectangling of the latent trait distributions)
  - Switching to Bayes estimation *may* fix at least some of this, but if not…

- What to do next? Alternatives range from ok to terrible…
  - Visit this talk and this R code for some newer two-stage approaches

2 Problems with SEM Alternatives (that replace latent circles with observed boxes)

1. A single sum score assumes **unidimensionality** and **parallel items:** equal loadings (discrimination) + equal error variance
   - Factor scores are equivalent to sum scores only under a parallel items model
   - Otherwise, the sum score is inconsistent with the factor model estimated

2. Observed variables are assumed **perfectly reliable** (or said differently, that each person's **trait estimate is known exactly**)
   - If the trait standard error (SE)=0, then we know each person's true value (otherwise, it comes from a distribution with variance given by SE^2)
   - If zero variability of a person's trait estimate is assumed, then the SEs for its relationships with other variables will be downwardly-based (so effects will look more precise and more significant than they should be)
   - If reliability is not perfect, then the estimates of its relationships with other variables will be downwardly-biased (weaker than they should be)

- Let's evaluate **3 strategies** from this view of potential problems…

Option 1: Parceling Indicators

- **Parceling = sum or average only *some* of the indicators**
- For example, for a factor with 12 original indicators:
  - ParcelA = i1+i2+i3+i4, ParcelB = i5+i6+i7+i8, ParcelC = i9+i10+i11+i12
  - `Factor BY ParcelA* ParcelB* ParcelC*; Factor@1; [Factor@0];`
- **Guess what happens to model fit??? It's fake good:**
  - Total possible DF for actual 12 indicators = $\frac{12(12+1)}{2} + 12 = 90$
  - Estimated DF for actual 12 indicators = $12\lambda_i$+$12\mu_i$+$12\sigma^2_{e_i}$ = 36
  - Model DF leftover = 90 − 36 = **54 = lots of room for misfit**
  - Total possible DF for 3 "parcels" = $\frac{3(3+1)}{2} + 3 = 9$
  - Estimated DF for 3 "parcels" = $3\lambda_i$+$3\mu_i$+$3\sigma^2_{e_i}$ = 9
  - Remaining DF leftover = 9 − 9 = **0 = fit is "perfect" (just-identified)**

Option 1: Parceling Indicators

- So contrary to what others may say… **PARCELING IS TOTALLY CHEATING AND YOU SHOULD NOT DO IT**

- That being said, here's how to parcel responsibly if you must:
  - Recognize that **parceling assumes tau-equivalence** (equal loadings) of the indicators within each parcel, so **verify that ahead of time**
  - Otherwise, you will get different model fit and parameter estimates across parceling options → should also report this "parceling allocation variability"; see Sterba & Rights (2023) for more info (and R package)
  - **Be honest** that parceling is an intermediate choice between:
    - Summing completely (one sum score to replace a latent factor)
    - *Summing sort of (parceling only some of the indicators together)*
    - An actual indicator-specific measurement model that reflects *all* the data
  - Recognize that different combinations of indicators to parcels can create very different results (especially for "subscales" of subscales), and **do NOT use parcels to "control for" or HIDE misfit**

Instead, try a simpler measurement model

- One way to save estimated parameters—if can be done without hurting model fit too much—is to **fit constrained measurement models**
- For example, for a factor with 12 original indicators:
  - Total possible DF for actual 12 indicators = $\frac{12(12+1)}{2} + 12 = 90$
  - Used DF for **full one-factor** model = $12\lambda_i + 12\mu_i + 12\sigma^2_{e_i}$ = **36**
  - Used DF for **tau-equivalent** (Rasch) factor model = $1\lambda_i + 12\mu_i + 12\sigma^2_{e_i}$ = **25**
    - **It is more difficult to estimate more loadings than more intercepts or error variances**
  - Used DF for **parallel items** factor model: $1\lambda_i + 12\mu_i + 1\sigma^2_{e_i}$ = **14**
  - Used DF for an **"empty means" parallel items** model: $1\lambda_i + 1\mu_i + 12\sigma^2_{e_i}$ = **3**
  - If not all loadings/residual variances/intercepts can be constrained equal across indicators, perhaps at least some of them can?
    - You can **test the fit of constraints** that **parceling would have just assumed**!
    - SEM allows you to consider and test intermediate possibilities, not just all or nothing with respect to each indicator gets its own parameter(s)
- In IFA/IRT, consider recoding sparse category responses into the next category (fewer thresholds to estimate)

Option 2a: Single-Indicator Models

- If you have determined that a single latent factor fits a set of indicators, one common option is a "single-indicator" sum score replacement

- Assuming perfect reliability (i.e., omega=1) would look like this:
  - `Factor BY sumscore@1; sumscore@0; Factor*;`
  - So sumscore's residual variance = 0 because its variance all goes to "factor variance"

- Better: Correct for **omega reliability** (as estimated from *your* own data, or a plausible upper-bound for reliability based on previous research):
  - Omega: $\boldsymbol{\omega} = \mathbf{Var}(\boldsymbol{F_s}) * (\boldsymbol{\Sigma\lambda_i})^2 \,/\, [\mathbf{Var}(\boldsymbol{F_s}) * (\boldsymbol{\Sigma\lambda_i})^2 + \boldsymbol{\Sigma}\,\mathbf{Var}(\boldsymbol{e_{is}}) + \ \mathbf{2\Sigma}(\boldsymbol{e_{is}}\ \mathbf{cov})]$
  - `Factor BY sumscore@1; sumscore* (ResVar); Factor*;`
  - `MODEL CONSTRAINT: ResVar =` $(\mathbf{1} - \boldsymbol{\omega}) * \mathbf{Var}(\boldsymbol{sumscore_s})$`;`
    - Need to know variance of sumscores (as "total" variance) for inclusion in ResVar formula
  - Sumscore residual variance is then its "error" variance only (rest → "true" factor variance)
  - Note: this is not possible if using IRT/IFA factors (because reliability varies across the trait)

- Either way, the factor can be "mean-centered" by fixing its mean = 0:
  - `[sumscore*]; [Factor@0];` So sumscore intercept holds its mean instead

Option 2b: Single-Indicator Models

- **Can I just treat the factor scores as observed? Not really…**
- Factor score = "theta hat" = random effect = central tendency estimate of a person's *unobserved* latent variable *distribution*
  - EAP estimates in ML → mean; MAP estimates in WLSMV → mode (worse)
  - Variance of each person's latent distribution is given by factor score SE^2
- Because they are from a latent variable, each factor score really has a **distribution of possible values** for each person
  - Factor scores are estimated from a multivariate normal prior distribution, and thus will be **shrunken** (pushed to mean) given low reliability
  - There is likely much uncertainty per person, especially for few indicators
    - Although factor scores (thetas) are routinely used in IRT, it's because they are usually based on *dozens* of items per factor (→ "small enough" SE)

- Btw, you CANNOT create factor scores by using the loadings as such:
  - $\boldsymbol{F_s} = \boldsymbol{\lambda_{11}y_{1s}} + \boldsymbol{\lambda_{21}y_{2s}} + \boldsymbol{\lambda_{21}y_{3s}}$ → Is a COMPONENT model, not a FACTOR model

Option 2b: Single-Indicator Models

- An EAP factor score is an **observed variable** (just like a sum score is), but it is more consistent with factor model structure it came from
- Assuming perfect factor score (fscore) reliability would look like this:
  - `Factor BY fscore@1; fscore@0; Factor*;`
  - So fscore's residual variance = 0 because its variance all goes to "factor variance"

- Better: In CFA, you can use **factor score reliability** estimated from *your* data (proportion of true trait differences relative to total trait variance):
  - Factor score reliability: $\boldsymbol{\rho} = \frac{\boldsymbol{\sigma}_{\mathbf{F}}^{\mathbf{2}}}{\boldsymbol{\sigma}_{\mathbf{F}}^{\mathbf{2}}+\mathbf{SE}_{\mathbf{FS}}^{\mathbf{2}}}$

    $\boldsymbol{\sigma}_{\mathbf{F}}^{\mathbf{2}}$ = factor variance from model solution
    $\mathbf{SE}_{\mathbf{FS}}^{\mathbf{2}}$ = error variance of EAP factor scores
  - `Factor BY fscore@1; fscore* (ResVar); Factor*;`
  - `MODEL CONSTRAINT: Resvar =` $(\mathbf{1}-\boldsymbol{\rho}) * (\boldsymbol{\sigma}_{\mathbf{F}}^{\mathbf{2}}+\mathbf{SE}_{\mathbf{FS}}^{\mathbf{2}})$`;`
    - Need to compute "total" variance (of factor scores + error variance) for inclusion in ResVar formula
  - Fscore residual variance is then its "error" variance only (rest → "true" factor variance)
  - Note this is NOT the same thing as Omega reliability for sum scores, and it's still not possible to do if for IRT/IFA factors (because reliability varies across the trait)

- Either way, the factor can be "mean-centered" by fixing its mean = 0:
  - `[fscore*]; [Factor@0];` So subscale intercept holds its mean instead

Example: Estimating Reliability

```
! Model 4 -- Fully Z-Scored 2-Factor Model with all parameters labeled for reference
    SitP BY Sit2*  Sit4*  Sit6*  (L1-L3);       ! SitP loadings (all free)
    SitN BY Sit1r* Sit3r* Sit5r* (L4-L6);       ! SitN loadings (all free)
    [Sit2*  Sit4*  Sit6*]    (I1-I3);           ! SitP intercepts (all free)
    [Sit1r* Sit3r* Sit5r*]   (I4-I6);           ! SitN intercepts (all free)
    Sit2*   Sit4*  Sit6*     (E1-E3);           ! SitP residual variances (all free)
    Sit1r*  Sit3r* Sit5r*    (E4-E6);           ! SitN residual variances (all free)
    SitP@1 (VarP); SitN@1    (VarN);            ! Factor variances (fixed=1)
    SitP WITH SitN*   (FactCov);                ! Factor covariance (free)
    [SitP@0 SitN@0]   (MeanP MeanN);            ! Factor means (fixed=0)
```

```
MODEL CONSTRAINT:          ! Calculate omega model-based reliability per factor
    NEW(OmegaP OmegaN); ! Using 1 as placeholder for factor variances
    OmegaP = (1*(L1+L2+L3)**2) / ((1*(L1+L2+L3)**2) + (E1+E2+E3));
    OmegaN = (1*(L4+L5+L6)**2) / ((1*(L4+L5+L6)**2) + (E4+E5+E6));
New/Additional Parameters
     OMEGAP                  0.744       0.020      37.956       0.000
     OMEGAN                  0.775       0.014      56.803       0.000
```

Omega Reliability for *Sum* Scores

```
SAMPLE STATISTICS FOR ESTIMATED FACTOR SCORES
     SAMPLE STATISTICS
           Means
              SITP          SITP_SE       SITN          SITN_SE
              ________      ________      ________      ________
 1              0.000         0.472         0.000         0.418
           Covariances
              SITP          SITP_SE       SITN          SITN_SE
              ________      ________      ________      ________
 SITP           0.777
 SITP_SE        0.000         0.000
 SITN           0.533         0.000         0.825
 SITN_SE        0.000         0.000         0.000         0.000
```

***Factor Score* Reliability (proportion of true individual differences)**

SitP: $\rho = \dfrac{1}{1 + .472^2} = .818$

SitN: $\rho = \dfrac{1}{1 + .418^2} = .851$

Option 2: Single-Indicator Models

- **Re-considering** the two potential problems with single-indicator representations of latent factors (sum scores or factor scores):

1. A single sum score assumes **unidimensionality** and **parallel items:** equal loadings (discrimination) + equal error variance
   - **For sum scores:** multidimensionality could be a big problem without any latent trait analyses to support the implied factor structure
     - Even if unidimensionality holds, if parallel items does not fit, the sum scores are not consistent with the model (and may not be available given missing items)
   - **For factor scores:** these reflect the estimated model, but will be shrunken towards the mean (more so for fewer items and greater unreliability)
     - Research suggests that they should be obtained from models that have same covariates as will be used in eventual structural models (see Curran et al., 2018)

2. Assuming perfect reliability of observed variables (or said differently, that that each person's trait estimate is known exactly)
   - **This is a problem unless correcting the single indicator for reliability, but this only possible when using CFA (in which reliability is constant)**
   - **So what to do in IRT/IFA models instead???**

Option 3: Multiple Plausible Values

- **Uncertainty in the factor scores from IFA/IRT models** can be represented explicitly using multiple so-called "**plausible values**" of factor scores
  - Strategy used in some large-scale testing programs (e.g., NAEP)
  - Generate x draws from a person's factor score *distribution*, save those draws to separate datasets, analyze each dataset, then combine results using procedures and rules for multiple imputation of missing data
  - That way the uncertainty of factor scores per person is still represented, along with the factor model parameters that distinguish the indicators
  - Research suggests a minimum of 5 values and a max of ??? (but with diminishing returns after 100 or so)
  - Mplus now provides this using a 4-step process (btw, the amount of analyst effort is the same no matter how many draws you use)
- Could also be implemented given MCMC estimation by using trait values from chain (weighted by 1/#values)

Plausible Values in Mplus, Step by Step

- **Step 1: Estimate factor model** using ML/MLR, save syntax for estimated parameters as start values (use OUTPUT: SVALUES to save typing)
- **Step 2: Feed in estimated parameters** as fixed parameters (replace all ***** with **@**), re-estimate model using ESTIMATOR=BAYES to generate the factor score draws for each person and save to separate data sets
  - Could do BAYES estimation for all of it, but if you have been using ML/MLR, you should use those parameters instead of letting it find new ones
- **Step 3: Merge separate datasets together** to create x complete datasets for analysis (e.g., using my SAS macro as part of Example 9c in this older class)
- **Step 4:** Tell Mplus to estimate your model **using the factor scores as observed variables on each of the x datasets**, and to combine the results (TYPE = IMPUTATION)
  - Will be easier and go faster than analyses of the original latent variables, but still preserves the uncertainty in the factor score estimates per person, along with the factor model from which those factor scores were derived

SEM: My Big Picture

- **SEM is great *when you can do it***
  - Provides a way to make almost any measurement idea an empirical question
  - Measurement models create latent constructs (= random effects) that better represent trait individual differences than can any one outcome
  - Structural models test relations involving those latent constructs
  - Measurement models will dominate global fit tests, so use a saturated structural model as baseline when testing nested structural models
    - Do omitted structural relations make the model fit not worse than saturated?

- **SEM is not a panacea for everything**
  - ML MAY BREAK when your models get too complicated (or realistic)
  - You have named your factors, but it doesn't mean you are right! (Validity)
  - Distributional assumptions matter, but so do linear model assumptions (nonlinear measurement and structural models may be needed)
  - Factor scores are not perfectly determined (and neither are sum scores), so make sure to represent their uncertainty in any SEM alternative