Structural Equation Modeling (SEM)

- Today's topics
  - The Big Picture of SEM
  - What to do (and what NOT to do) when SEM breaks for you
    - Single indicator (ASU) models
    - Parceling indicators
    - Using single factor scores
    - Multiple plausible values of factor scores

What is SEM?

- The term "SEM" gets used to describe many different models, but fundamentally, **SEM consists of two parts**:
  - **Measurement model for each latent variable**
    - "CFA" if indicators are continuous and "normal enough"
    - "IFA" if indicators are binary, ordinal, or nominal
    - "?name?" if indicators require some other link function (e.g., counts)
    - Factors/thetas/traits are assumed to be multivariate normal
  - **Path analysis (regressions) amongst the latent variables**
    - And amongst 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 does not require a logit, but a **binary outcome variable** does (so then it's on the CATEGORICAL statement)
      - THERE IS NO SUCH THING AS A CLASS STATEMENT IN MPLUS (I'm sorry), so you have to create manual contrasts to include categorical predictors

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, variance)
  - The path model must be identified (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 analysis: Total possible df = $\frac{v*(v+1)}{2} + v$
  - Practical tip: don't count, just look at your 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 (then factors won't know which way to go)

SEM: Predictors vs. Outcomes

- New terminology for use in SEM:
  - Predictor variables are called "**exogenous**" (arrows go out of it only)
  - Outcome variables are called "**endogenous**" (arrows go into it)
  - If a variable is *both* a predictor and an outcome, it is "endogenous"

- Some SEM books claim that when using ML, that *all* variables should have a multivariate normal distribution (MVN), but this is NOT true in Mplus for three reasons:
  - You can use ML with link functions and other distributions (e.g., CATEGORICAL tells it to use Bernoulli or Multinomial instead as needed)
  - Exogenous variables are not part of the ML function unless you make them (by referring to their means, variances, or covariances in syntax)
  - Only the *residuals* of endogenous variables are assumed MVN
  - MLR can help with continuous but overly kurtotic endogenous variables

SEM: Predictors vs. Outcomes

- The important distinction is whether each observed variable is **part of the maximum likelihood function or not**
  - Are its means/intercepts, variances/residual variances, or covariances/residual covariances being estimated? Then yes, it is
  - Are *just* its paths predicting endogenous variables being estimated? Then no, it is NOT part of the likelihood
- **Upside** of putting exogenous variables in the likelihood?
  - Predictors can have missing data (assuming missing at random)
- **Downside** of putting exogenous variables in the likelihood?
  - Distributional assumptions then apply, although Mplus gets cranky when exogenous variables are added to CATEGORICAL
    - A silly work-around is to make it a perfect single indicator of a latent factor, that way it becomes an "outcome" officially, but this may cause other problems
  - Covariances amongst "predictors" then contribute to fit…

SEM: What goes into model fit

- Back in CFA/IFA, misfit was almost always due to covariances
  - If each indicator has its own **intercept or thresholds**, then the indicator **means or response frequencies** will be predicted perfectly
  - If each indicator has its own **residual variance**, then the indicator **total variances** will 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
  - Covariances amongst variables in the likelihood COUNT
  - Covariances for "predictors" (NOT in the likelihood) with "outcomes" (in the likelihood) COUNT
  - Covariances amongst "predictors" (NOT in the likelihood) do NOT count

SEM: What to do first?

- **Because SEM is composed of two distinct parts…**
  - Measurement model that identifies latent variables
  - 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
  - When possible, then combine all factors of interest and other observed variables in the same model, estimating all possible relations among them (this "saturated" model is the best-fitting structural model)
  - Then modify the structural model to answer your questions, and see if the simpler model is NOT worse than the saturated structural model

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

SEM: What to do if I can't do it?

- 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 # free parameters)
  - Many free parameters are in the likelihood (especially with few persons)
  - Some outcomes are non-normal (link functions are involved)
  - Many latent variables are included (especially with link functions)
  - Latent factors are not well-identified (2 indicators is not enough)
  - Latent variable interactions are included (which require numeric integration → repeated rectangling of the latent trait distributions)

- What to do then? Alternatives range from ok to terrible…

First try a simpler measurement model

- One way to save estimated parameters—when possible to do so without hurting model fit too much—is to **fit constrained measurement models** (i.e., make the parcels a real structure)
- 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 = 12λ+12μ+12σ_e^2 = **36**
  - Used DF for **tau-equivalent** (Rasch) factor model = 1λ+12μ+12σ_e^2 = **25**
    - **It is more difficult to estimate more loadings than more μ or σ_e^2**
  - Used DF for **parallel items** factor model: 1λ+12μ+1σ_e^2 = **14**
  - Used DF for an **"empty means" parallel items** model: 1λ+1μ+1σ_e^2 = **3**
  - If not all loadings/residual variances/intercepts can be constrained across items, perhaps at least some of them can?
  - Mplus allows you to test intermediate possibilities, not just all or nothing with respect to each indicator gets its own parameter(s)

3 Problems with SEM Alternatives*

1. Assuming **unidimensionality** and **tau-equivalence** (equal discrimination) of indicators within a single sum score
   - If these do not hold, the validity of the factor is questionable
2. Assuming **perfect reliability** of observed variables
   - If reliability is not perfect, then the estimates of its relationships with other variables will be downwardly-biased (weaker than they should be)
3. Assuming each person's **trait estimate is perfectly known**
   - 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)
   - This happens whenever we use only 1 observed trait value per person, **because a trait is essentially a missing value of a predictor variable**

** Thanks to Jonathan Templin for helping me enumerate these problems*

Option 1: Single-Indicator Models

- If you have determined that a single latent factor fits a set of indicators, an option is a "single-indicator" (ASU) factor model

- Assuming perfect reliability (Omega=1) would look like this:
  - **`Factor BY subscale@1; subscale@0; Factor*;`**

- Better: Use **Omega reliability** as estimated from *your* data:
  - "Omega" Reliability: $\boldsymbol{\omega} = (\boldsymbol{\Sigma\lambda})^2 \,/\, [(\boldsymbol{\Sigma\lambda})^2 + \boldsymbol{\Sigma}\,\mathbf{Var}(\mathbf{e}) + \mathbf{2\Sigma}(\mathbf{e\ cov})]$
  - **`Factor BY subscale@1; subscale* (Reliable); Factor*;`**
  - **`MODEL CONSTRAINT: Reliable =`** $(\mathbf{1} - \boldsymbol{\omega}) * \mathbf{Var}(\mathbf{subscale})$**`;`**
  - Subscale residual variance is then the "unreliable variance" only

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

Option 1: Single-Indicator Models

- Problems with using **a sum score** in a single indicator model (or as an observed variable in an analysis more generally):

1. Assuming unidimensionality and tau-equivalence (equal discrimination) of indicators within a single sum score
   - **YEP, this is a definitely a problem.**
2. Assuming perfect reliability of observed variables
   - **This is a problem unless correcting for the omega reliability of the sum score.**
3. Assuming each person's trait estimate is perfectly known
   - **YEP, this is a definitely a problem when using only one number to represent the trait level of each person.**

Option 2: Parceling Indicators

- **Parceling = ASU for only some 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???**
  - Total possible DF for actual 12 indicators = $\frac{12(12+1)}{2}+12=90$
  - Estimated DF for actual 12 indicators = $12\lambda+12\mu+12\sigma_e^2$ = 36
  - Remaining 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" = $3\lambda+3\mu+3\sigma_e^2$ = 9
  - Remaining DF leftover = 9 − 9 = **0 = fit is "perfect" (just-identified)**

Option 2: Parceling Indicators

- 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 **test that ahead of time**
  - If tau-equivalence (a Rasch-type model) holds, then you aren't losing information (or cheating model fit) by combining the item responses
  - **Be honest** that parceling is an intermediate choice between:
    - ASU completely (single-indicator model for a construct)
    - ASU 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 as a way to "control for" or HIDE misfit**

Option 2: Parceling Indicators

- Problems with **using parcels** rather than the original indicators (aside from an invalid assessment of model fit):

1. Assuming unidimensionality and tau-equivalence (equal discrimination) of indicators within a single parcel
   - **YEP, this is a definitely a problem (unless verified ahead of time).**
2. Assuming perfect reliability of observed variables
   - **The parcel is not assumed completely reliable, but the reliability across parcels is likely to be too optimistic (hidden error within).**
3. Assuming each person's trait estimate is perfectly known
   - **This is not a problem if the latent variable is retained in the model, but we are assuming perfectly known parcel-level scores.**

Option 3: Can I just use the factor scores?

- **In a word, NO.**
- Factor score = random effect = mean of a person's *unobserved* latent variable distribution given the observed responses
- Because this is a latent variable, each factor score really has a **distribution of possible values** for each person
  - Factor scores are estimated a multivariate normal prior distribution, and thus will be shrunken (pushed to normal) 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 instead of just a few

- Btw, you CANNOT create factor scores by using the loadings as such:
  - $\mathbf{F} = \boldsymbol{\lambda}_{11}\mathbf{y}_1 + \boldsymbol{\lambda}_{21}\mathbf{y}_2 + \boldsymbol{\lambda}_{21}\mathbf{y}_3 \ldots$ → Is a COMPONENT model, not a FACTOR model

Option 3: Single Factor Scores

- A factor score is an **observed variable** (just like a sum score is)
- Assuming perfect factor score reliability would look like this:
  - **`Factor BY fscore@1; fscore@0; Factor*;`**

- Better: In CFA (but not IRT/IFA in which reliability varies across the trait), you can use **factor score reliability** estimated from *your* data (true trait differences relative to total trait variance):
  - Factor score reliability $\boldsymbol{\rho} = \frac{\boldsymbol{\sigma}_{\mathbf{F}}^{2}}{\boldsymbol{\sigma}_{\mathbf{F}}^{2}+\mathbf{SE}_{\mathbf{F}}^{2}}$

    $\boldsymbol{\sigma}_{\mathbf{F}}^{2}$ =variance of factor scores
    $\mathbf{SE}_{\mathbf{F}}^{2}$=error variance of factor scores
  - **`Factor BY fscore@1; subscale* (Reliable); Factor*;`**
  - **`MODEL CONSTRAINT: Reliable =`** $(\mathbf{1}-\boldsymbol{\rho})*(\boldsymbol{\sigma}_{\mathbf{F}}^{2}+\mathbf{SE}_{\mathbf{F}}^{2})$**`;`**
  - Note this is NOT the same thing as Omega reliability for sum scores

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

Example: Estimating Reliability

```
! Model 4 -- Fully Z-Scored 2-Factor Model
    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*];                     ! SitP intercepts (all free)
    [Sit1r* Sit3r* Sit5r*];                    ! 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; SitN@1;                            ! Factor variances (fixed=1)
    SitP WITH SitN*;                           ! Factor covariance (free)
    [SitP@0 SitN@0];                           ! Factor means (fixed=0)
```

```
MODEL CONSTRAINT:  ! Calculate omega model-based reliability per factor
    NEW(OmegaP OmegaN);
    OmegaP = ((L1+L2+L3)**2) / (((L1+L2+L3)**2) + (E1+E2+E3));
    OmegaN = ((L4+L5+L6)**2) / (((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 <u>true</u> individual differences)

SitP: $\rho = \frac{.777}{.777 + .472^2} = .777$

SitN: $\rho = \frac{.825}{.825 + .418^2} = .825$

Option 3: Single Factor Scores

- Problems with a **single factor score** as an observed variable:

1. Assuming unidimensionality and tau-equivalence (equal discrimination) of indicators within a single sum score
   - **These should be tested first. Unidimensionality should hold, but tau-equivalence doesn't have to (then just let the loadings vary).**

2. Assuming perfect reliability of observed variables
   - **This is not a problem, but factor score unreliability may still create downward bias for relationships with the factor score.**

3. Assuming each person's trait estimate is perfectly known
   - **YEP, this is a definitely a problem when using only one number to represent the trait level of each person.**

Option 4: Multiple Plausible Values

- Using a single factor score instead of a sum score can fix:
  - Assuming (without testing) unidimensionality and tau-equivalence
  - Assuming perfect reliability (can correct using factor score reliability)

- But **uncertainty in the factor scores** is still a problem…
- A potential solution: ***Multiple*** **plausible factor score values**
  - An intermediate option between full SEM and single trait estimates
  - 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
  - Mplus now provides this using a 4-step process

Plausible Values, 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 (see my SAS macro as part of Example 11 to make this easier)
- **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 means to make almost any idea an empirical question
  - Measurement models create latent constructs (= random effects)
  - Structural model test relations among those constructs
  - Do not let your measurement model swamp structural relations tests by looking only at global fit: consider what the baseline model should be
- **SEM is not a panacea for everything**
  - IT WILL BREAK when your models get too complicated (or realistic)
  - You may have named your factors, but it doesn't mean you are right!
  - Distributional assumptions matter, but so do linear model assumptions (nonlinear measurement and structural models may be needed)
  - Factor scores are not real things (and neither are sum scores), so make sure to represent their uncertainty in any SEM alternative