The Finale: Path Analysis and Structural Equation Modeling (SEM)

- Topics:
  - Path analysis (i.e., path modeling):
    - Vocabulary and rules for predictions
    - Assessing model fit and testing mediation
  - The Big Picture of SEM
  - 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

Path Analysis versus SEM

- **Path analysis** (i.e., path models) are multivariate models that include observed variables only, whereas **SEM** also includes latent variables
- The vast, vast majority of textbooks and resources for path analysis and SEM focus on the **multivariate general linear model** case
  - Using an identity link function and conditional multivariate normal distribution (MVN) in which all outcomes have estimated residual variances (and fixed effects that predict their conditional means)
  - Many software packages available: SAS PROC CALIS, **STATA SEM and GSEM**, **Mplus**, Lavaan in R, LISREL, EQS, AMOS (part of SPSS)
    - None use denominator degrees of freedom (give z and χ^2 Wald tests)
  - For path analysis examples with normal residuals using Mplus and STATA, see Part 2 of Examples 4b and 5a; Mediation in Example 6a in this class

- Software for path analysis and SEM with **non-normal outcomes** is harder to find and may have more complexity in estimation
  - **STATA GSEM**, **Mplus**, Lavaan in R (categorical outcomes only with WLSMV)
  - For a mediation path model with binary outcomes, see Example 6b in this class
  - Once you know how to build latent variables (for any kind of indicators), the transition from path analysis to SEM is very straightforward…
    - So let's start with an overview of path models and then extend into SEM…

Path Models: Pictures and Equations

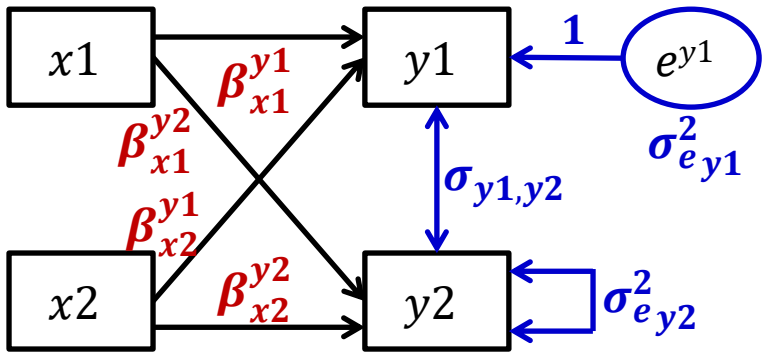
- What are **path models**? "Truly" multivariate models for predicting 2+ outcomes simultaneously for the same unit of analysis
- Models most often expressed as a diagram using these conventions:
  - Boxes = observed variables; circles = latent variables (in SEM) or residuals
  - One-headed arrow = regression (arrow points from predictor to outcome)
  - Two-headed arrow = (residual) covariance; intercepts sometimes via triangle

Diagram translates into these simultaneous regression models (in which superscripts denote the outcome of each parameter):

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

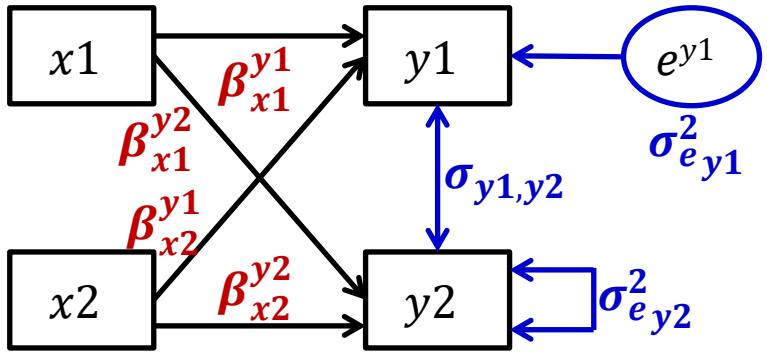
Unstructured R matrix for outcome residual variances and covariance(s): $\begin{bmatrix} \boldsymbol{\sigma^2_{e_{y1}}} & \boldsymbol{\sigma_{12}} \\ \boldsymbol{\sigma_{12}} & \boldsymbol{\sigma^2_{e_{y2}}} \end{bmatrix}$

The idea of residual variable is either expressed using a separate circle (e.g., for $y1$) or a two-headed arrow into itself (e.g., for $y2$).



Multivariate Regression via Path Models

- This example is really just two univariate regression models estimated simultaneously
  - Each $\boldsymbol{\beta_{x1}}$ and $\boldsymbol{\beta_{x2}}$ provide the unique effects of $x1$ and $x2$ for $y1$ and $y2$ outcomes (same as in regression)
  - Can calculate R^2 for each outcome
  - Mplus code under MODEL:
    - `y1 y2 ON x1 x2; y1 WITH y2;`



- So why do both models at once?
  - To test differences in effect size (e.g., does $\boldsymbol{\beta_{x1}^{y1} = \beta_{x1}^{y2}}$?)
  - To test mediation and indirect effects, in which a variable is both a predictor and an outcome in the same analysis (stay tuned)

If these variables came from a dyad of two persons (1 and 2), this could be an example of an "actor–partner model"

- Arrows within same person = "actor effects"
- Arrows across different people = "partner effects"

2 Types of Path Model Solutions

- Unstandardized → predicts scale-sensitive original variables:
  - **Regression Model:** $y1_i = \boldsymbol{\beta_0^{y1}} + \boldsymbol{\beta_{x1}^{y1}}(x1_i) + \boldsymbol{\beta_{x2}^{y1}}(x2_i) + \boldsymbol{e_i^{y1}}$
  - Useful for comparing across groups (whenever absolute values matter)
  - Parameters predict the variables' **means, variances, and covariances**
  - Variance of $\boldsymbol{y1}$ **= [variance explained by predictor fixed effects]** + $\boldsymbol{\sigma^2_{e_{y1}}}$

- Standardized → predicts z-scored versions of variables:
  - Useful when comparing effects within a solution (are then on same scale)
  - Model parameters predict the variables' **correlations**
  - Standardized slope = $[\boldsymbol{\beta_{x1}^{y1}} * \boldsymbol{SD(x1)}] \,/\, \boldsymbol{SD(y1)}$ **= unique correlation**
  - $\boldsymbol{R^2}$ for $\boldsymbol{y1}$ **= 1 − standardized** $\boldsymbol{\sigma^2_{e_{y1}}}$
  - Standardized solutions are usually only reported for path models with conditionally multivariate normal residuals (with estimated variances)

New (and Confusing) Terminology

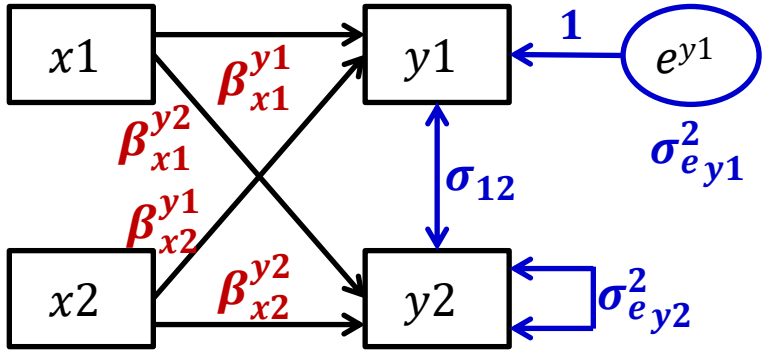
- **Predictors** are known as **exogenous** variables (X-ogenous to me)
- **Outcomes** are known as **endogenous** variables (IN-dogenous to me)
- Variables that are both at once are called **endogenous** variables

Our previous example model:

2 exogenous variables ($x1$ and $x2$)

2 endogenous variables ($y1$ and $y2$)

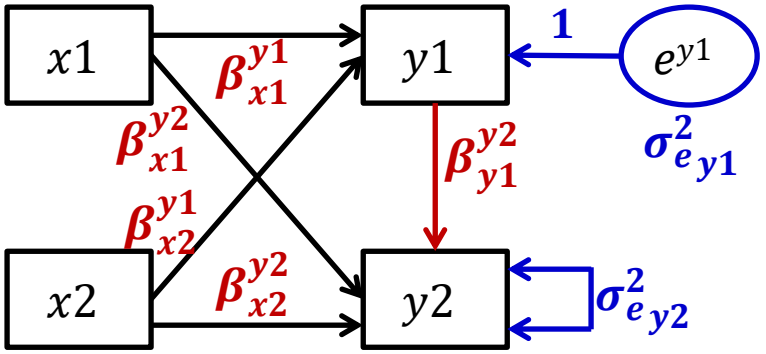
$$y1_i = \boldsymbol{\beta_0^{y1}} + \boldsymbol{\beta_{x1}^{y1}}(x1_i) + \boldsymbol{\beta_{x2}^{y1}}(x2_i) + \boldsymbol{e_i^{y1}}$$
$$y2_i = \boldsymbol{\beta_0^{y2}} + \boldsymbol{\beta_{x1}^{y2}}(x1_i) + \boldsymbol{\beta_{x2}^{y2}}(x2_i) + \boldsymbol{e_i^{y2}}$$



Our modified example model:

$y1$ predicts $y2$ (still endogenous)

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



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**, *truly* exogenous predictor variables cannot have missing data, the same as in any general(ized) linear 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 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** allows a work-around: you ***can* 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* 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 MNV for ALL variables), but both can do full-info ML
  - SAS CALIS: full-info via "FIML" or robust version "MLMB" (I think it's full-info)
    - 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 missing at random (MAR), but they also then have distributional assumptions (MNV)
  - STATA SEM "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 path 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
  - 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 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 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

Model Identification and Model Fit

- "**Model identification**" in path models* refers to estimability and whether the model has spent all possible degrees of freedom (DF)
  - *It also includes the scaling of latent variables in SEM (each latent factor must have a mean and a variance)

- Need to know **Total DF** = possible and **Model DF** = remaining
  - In models in which all variables are in the likelihood as outcomes, **total DF =** $\frac{\boldsymbol{v(v+1)}}{\mathbf{2}} + \boldsymbol{v}$ where $\boldsymbol{v}$ **is the # outcomes** (NOT people, like usual)
    - Total DF = number of outcome means, variances, and covariances
    - e.g., if $v = 4$ outcomes, then $DF = \frac{4(5+1)}{2} + 4 = 14$
    - For truly exogenous predictors, their means, variances, and covariances among them do NOT count towards total DF, but the covariances of those predictors with the outcomes DO count towards total DF (so not an easy-to-use formula)
      - In practice it's still ok to just use v = # outcomes + # predictors (stay tuned)
  - **Model DF** = data input − model output
  - **Model DF** = # possible parameters − # estimated parameters

What Goes In (data used as input)

- Observed mean per outcome
- Observed variance per outcome

What Comes Out (estimated parameters)

- Estimated intercept per outcome (to *perfectly* re-create the observed outcome means)
- Estimated residual variance per outcome (to *perfectly* re-create the observed outcome variances)

- Note of terminology: if the "outcome" is not actually being predicted, then the output labels switch from conditional to unconditional:
  - For a predictor in the likelihood, model estimates its "mean" instead of its "intercept" and its "variance" instead of its "residual variance"
  - For truly exogenous predictors, their means and variances are not potential model parameters, so we can ignore them (as in regular regression models)
- Bottom line: **model misfit does not come from means or variances** (UNLESS constraints on them are used to reduce the number estimated)

What Goes In (data used as input)

- Observed covariance between each pair of outcomes
- Observed covariance of each predictor with each outcome

What Comes Out (estimated parameters)

- Estimated regression path or residual covariance between each pair of outcomes (to predict covariance)
- Estimated regression path or residual covariance of each predictor with each outcome (to predict covariance)

- Note of terminology: if the "outcome" is not actually being predicted, then the output labels switch from conditional to unconditional:
  - For a predictor in the likelihood, model estimates its "covariance" instead of its "residual covariance" with other variables
  - For truly exogenous predictors, the covariances among them are not potential model parameters, so we can ignore them (as in regular regression models)
- **If some sources of direct covariance are omitted**, then observed covariances will not be perfectly reproduced → **room for model misfit**

Labeling Model Identification Scenarios

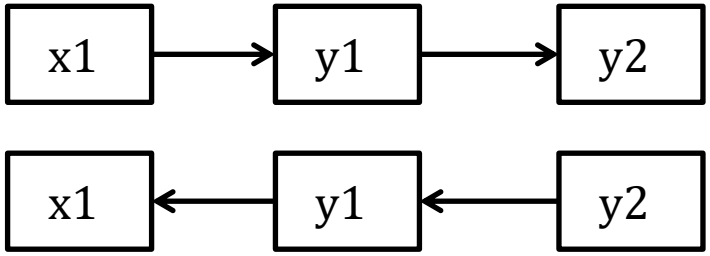
- Two things to know:
  - **Is the model estimable**—can all parameters be found? (no redundancy)
  - **Is the absolute model fit testable**—can we determine if the parameters used adequately re-create the (outcome) means, variances, and covariances?
  - Comes from Model DF = # possible parameters – # estimated parameters

- 3 possible model identification scenarios:
  - **Under-identified**: # possible < # estimated→ negative Model DF
    - Model is not solvable (parameter estimates cannot be found); game over
  - **Just-identified**: # possible = # estimated → 0 Model DF
    - Model is solvable (is most common scenario; perfectly reproduces original data)
    - Absolute model fit will NOT be relevant (which is good for path models)
  - **Over-identified**: # possible > # estimated → positive Model DF
    - Model is still solvable (and is more parsimonious description of original data because some possible direct relationships have been "overlooked")
    - Absolute model fit is then necessary before interpreting model results (generally more of an issue for latent variable measurement models in SEM)

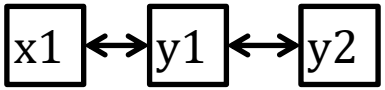
Path Model Identification Examples

(in which each variable is in the likelihood and has a perfectly accounted for mean/intercept and variance/residual variance)

- Over-identified: have positive DF leftover (possible>estimated)

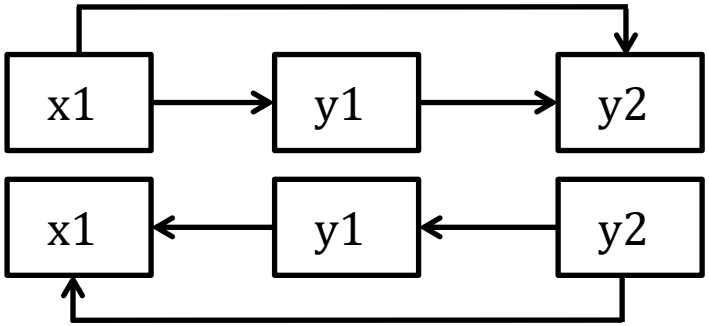


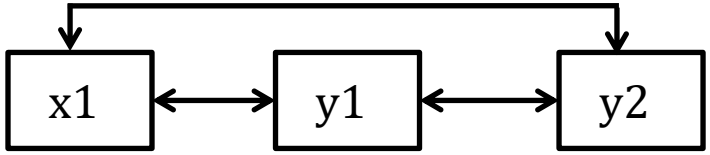
Right: also **DF=1**, but predicts no correlation of $x1$ with $y2$



Left two models all have **equivalent fit with DF=1** (for the 1 missing direct relationship)

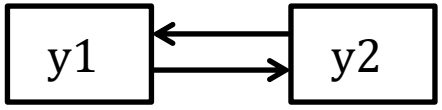
- Just-identified: have 0 df leftover (possible = estimated)





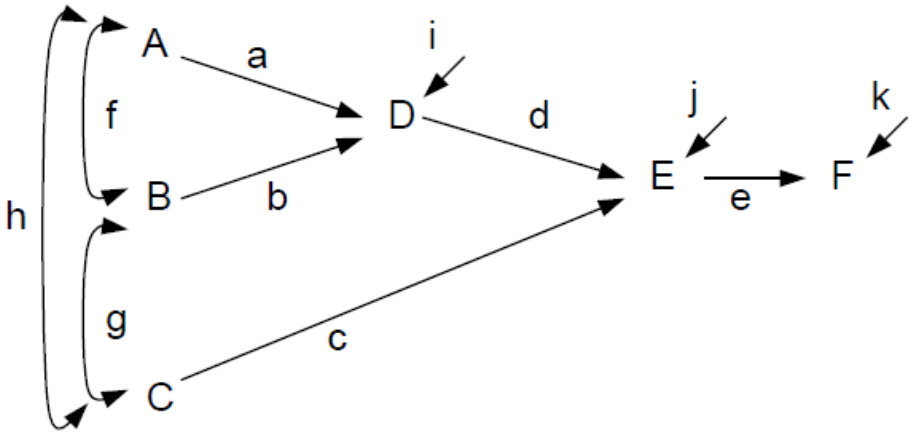
These 3 models all have **equivalent fit with DF=0** (for 0 missing direct relationships)

- Under-identified: have negative DF (possible < estimated)



This model is trying to estimate 2 paths using only 1 covariance (can't be solved)

Path Model Code in Mplus



By including the covariances between the **A, B, and C predictors**, this diagram indicates they **are in the likelihood**. This means that they can have missing data (under an assumption of missing at random), but they are also assumed to have a multivariate normal distribution.

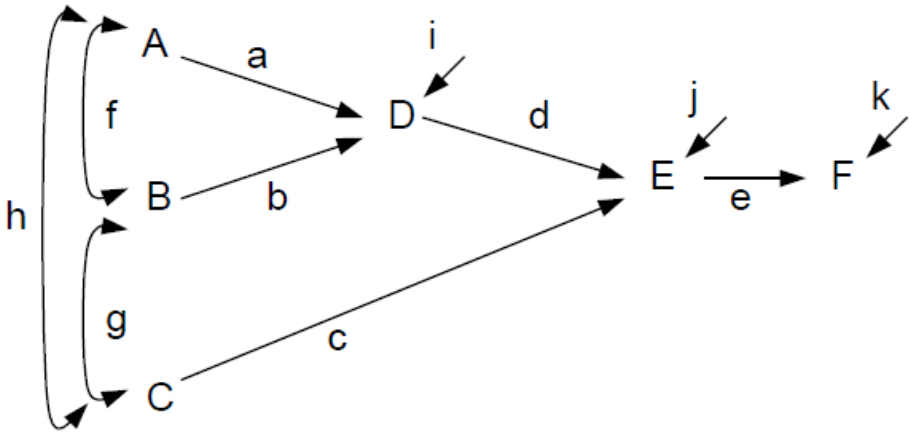
```
! Required code to estimate regression paths
  D ON A B;
  E ON D C;
  F ON E;

! Outcome intercepts and residual variances estimated by default
  [D E F]; D E F;

! To bring A, B, and C predictors into the likelihood,
! Request their covariances
  A B C WITH A B C;

! Predictor means and variances then estimated by default
  [A B C]; A B C;
```

Wright's Rules of Tracing for Path Analysis

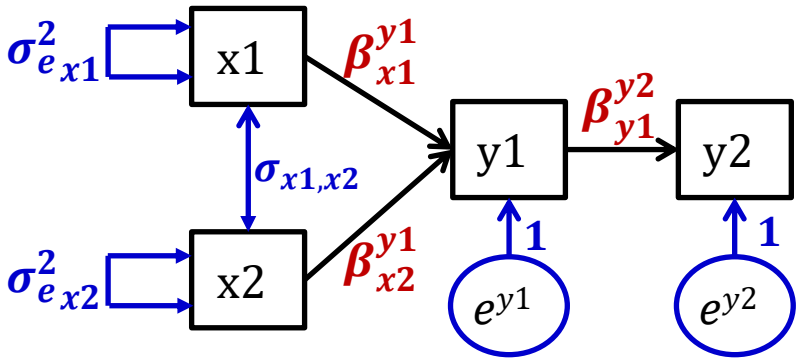


These rules below use correlations for convenience, but for covariances, the **variance** of a predictor variable that originates the path (or at any change in directions of directed arrows) gets included as a **multiplier** of the path:

Cov B to D: $(\boldsymbol{\sigma_B^2})\boldsymbol{b} + \boldsymbol{fa}$

- Total correlations between variables can result from more than one path with these rules: no loops (can't pass through same variable twice), no going forward then backward (common causes, not common outcomes), and only one curved arrow (covariance) is allowed from first to last variable
  - Correct B to D: $r_{BD} = b + fa$
  - Correct C to D: $r_{CD} = gb + ha$
  - Correct A to E: $r_{AE} = ad + fbd + hc$
  - Correct A to F: $r_{AF} = ade + fbde + hce$
  - Wrong A to B: $r_{AB} \neq abf$
  - Wrong C to D: $r_{CD} \neq cd$
  - Wrong A to C: $r_{AC} \neq fg$

Model-Predicted Covariances: Example



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

Each unique intercept will capture any leftover misfit to its variable's mean

- This model with all four outcomes in the likelihood has six covariances to be predicted by the model—**4 will be perfectly predicted given direct paths/covariances**:
  - $Cov(x1, x2) = \sigma_{x1,x2}$
  - $Cov(x1, y1) = (\sigma^2_{x1})\beta^{y1}_{x1} + (\sigma_{x1,x2})\beta^{y1}_{x2}$
  - $Cov(x2, y1) = (\sigma^2_{x2})\beta^{y1}_{x2} + (\sigma_{x1,x2})\beta^{y1}_{x1}$
  - $Cov(y1, y2) = (\sigma^2_{y1})\beta^{y2}_{y1}$

2 covariances are only predicted by the other direct paths/covariances, and **will not be perfect**:

$$Cov(x1, y2) = (\sigma^2_{x1})\beta^{y1}_{x1}\beta^{y2}_{y1} + (\sigma_{x1,x2})\beta^{y1}_{x2}\beta^{y2}_{y1}$$
$$Cov(x2, y2) = (\sigma^2_{x2})\beta^{y1}_{x2}\beta^{y2}_{y1} + (\sigma_{x1,x2})\beta^{y1}_{x1}\beta^{y2}_{y1}$$

- The <u>model-implied variances</u> of $y1$ and $y2$ are complex but perfect because of each e_i:
  - $Var(y1) = (\sigma^2_{x1})\beta^{y1}_{x1}\beta^{y1}_{x1} + (\sigma^2_{x2})\beta^{y1}_{x2}\beta^{y1}_{x2} + 2(\sigma_{x1,x2})\beta^{y1}_{x2}\beta^{y1}_{x1} + \sigma^2_{ey1}$
  - $Var(y2) = (\sigma^2_{x1})\beta^{y1}_{x1}\beta^{y1}_{x1}\beta^{y2}_{y1}\beta^{y2}_{y1} + (\sigma^2_{x2})\beta^{y1}_{x2}\beta^{y1}_{x2}\beta^{y2}_{y1}\beta^{y2}_{y1} + 2(\sigma_{x1,x2})\beta^{y1}_{x2}\beta^{y1}_{x1}\beta^{y2}_{y1}\beta^{y2}_{y1}$
    $+ (\sigma^2_{ey1})\beta^{y2}_{y1}\beta^{y2}_{y1} + \sigma^2_{ey2}$

Path Model Evaluation: Steps 1, 2, and 3

1. **Assess global absolute model fit**
   - Recall that variable means and variances are perfectly predicted (just-identified) → *misfit comes from mis-predicted covariances*
   - χ^2 is sensitive to large sample size, so pick at least one global fit index from each class; hope they agree (e.g., CFI, RMSEA)
2. **Identify localized model strain**
   - Global model fit means that the observed and predicted covariance matrices aren't too far off on the whole… says nothing about the specific matrix elements (reproduction of each covariance)
   - Consider normalized residuals and modification indices to try and "fix" the model – add missing relationships that should be there
3. **Revise the model until it fits**
   - Make sure all the parameters make sense (e.g., no negative variances)

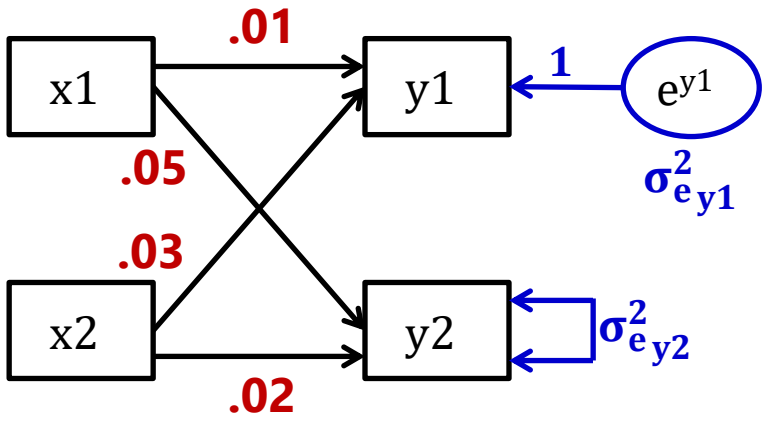
Good global and local fit? Great, but we're not done yet…

Step #4 in Model Evaluation

4. Inspect **parameter effect sizes** and significance
   - A good-fitting model does not necessarily imply a good model!
     - Can reproduce lack of covariance quite well and still not have anything useful – e.g., correlation of .2 → 4% shared variance?
     - **Effect size (R^2 for variance explained) is practical significance**

This example model could have "excellent fit" (testable because DF=1) but no significant paths…

Why? Good absolute fit just means it has successfully reproduced the (non)relationships among these variables—not whether there are relationships worth reproducing!



Complications in Path Models for Generalized Outcomes

- Fewer path model software options are available that include link functions and non-normal conditional distributions
  - I am most familiar with Mplus (and have been learning STATA GSEM)
  - These two vary in options for outcome types and estimation methods

- Differences compared to path models with MVN outcomes
  - No residual variances means:
    - Traditional measures of absolute fit are not available when using ML
    - Conventional standardized solutions may not be available
    - Residual (error) covariances must be included via random intercepts
  - Different estimation methods will generally not lead to the same result, even given complete data
    - Mplus: (Robust) full-info ML or limited-info WLSMV
    - STATA GSEM: Equation-wise ML (functions more like limited-info ML)

Mplus for Generalized Outcomes

- Link functions:
  - Logit (binary, cumulative, adjacent, or baseline) or probit (binary, cumulative, or baseline) for categorical outcomes; log for counts
- Distributions:
  - Multinomial (so binary, ordinal, or nominal outcomes)
  - Counts: Poisson and negative binomial (and zero-altered for each, negative binomial hurdle (can trick it into Poisson hurdle)
- Estimation for all outcomes using (robust) full-info ML
  - Quadrature or Montecarlo numeric integration
  - Missing observations assumed missing at random (conditionally random)
- For binary or ordinal outcomes, there is also the limited-info estimator WLSMV (i.e., "diagonally-weighted least squares)
  - Works fast, provides absolute fit indices, but assumes any missing observations are missing completely at random

STATA GSEM for Generalized Outcomes

- Right: Relative to Mplus, STATA v. 16 has many more options for distributions (rows) and link functions (columns)…

	identity	log	logit	probit	cloglog
Gaussian	D	x			
Bernoulli			D	x	x
beta			D	x	x
binomial			D	x	x
ordinal			D	x	x
multinomial			D		
Poisson		D			
negative binomial		D			
exponential		D			
Weibull		D			
gamma		D			
loglogistic		D			
lognormal		D			
pointmass	D				

D denotes the default.

`family(`*family*`)`	distribution family; default is `family(gaussian)`
`link(`*link*`)`	link function; default varies per family
`cloglog`	synonym for `family(bernoulli) link(cloglog)`
`exponential`	synonym for `family(exponential) link(log)`
`gamma`	synonym for `family(gamma) link(log)`
`logit`	synonym for `family(bernoulli) link(logit)`
`loglogistic`	synonym for `family(loglogistic) link(log)`
`lognormal`	synonym for `family(lognormal) link(log)`
`llogistic`	synonym for `family(llogistic) link(log)`
`lnormal`	synonym for `family(lnormal) link(log)`
`mlogit`	synonym for `family(multinomial) link(logit)`
`nbreg`	synonym for `family(nbreg mean) link(log)`
`ocloglog`	synonym for `family(ordinal) link(cloglog)`
`ologit`	synonym for `family(ordinal) link(logit)`
`oprobit`	synonym for `family(ordinal) link(probit)`
`poisson`	synonym for `family(poisson) link(log)`
`probit`	synonym for `family(bernoulli) link(probit)`
`regress`	synonym for `family(gaussian) link(identity)`
`weibull`	synonym for `family(weibull) link(log)`
`exposure(`$varname_e$`)`	include ln($varname_e$) with coefficient constrained to 1
`offset(`$varname_o$`)`	include $varname_o$ with coefficient constrained to 1

… But estimation is more problematic given missing data…

Estimation in STATA GSEM

- What ML estimation **STATA GSEM** uses is unclear…
  from v. 16 manual:
  - It's an "equation-wise deleter": it drops the exogenous predictors from joint normality assumption (treats them as given, so they are not in the likelihood)
  - "sem and gsem produce the same numeric solutions for the parameters and the SEs when both can fit the same model" → when all outcomes are MVN
  - "gsem will often be able to use more observations from the data than sem will, assuming you do not use sem with method MLMV"

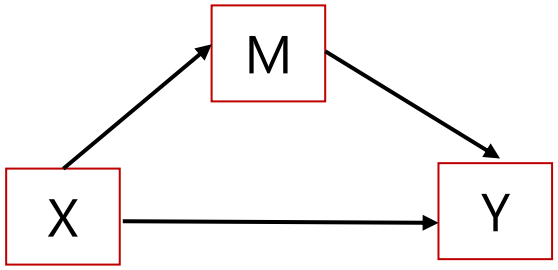
- What I've also figured out through trial and error:
  - It allows the same trick as Mplus—you *can* bring exogenous predictors into the likelihood as outcomes by listing their means, variances, or covariances as parameters → but that doesn't change which cases get used in each equation
  - If you ask for robust SEs, it changes to QLM (which is limited-info ML), and the estimates do not change (and neither does the model LL value)
  - The results from the same model with incomplete outcomes do not match those of Mplus when it uses full-info ML (then assuming missing at random, MAR)

- My conclusion: **STATA GSEM does not do truly full-info ML**, which means all variables are assumed missing completely at random (MCAR, not MAR)

Terminology: Mediation ≠ Moderation

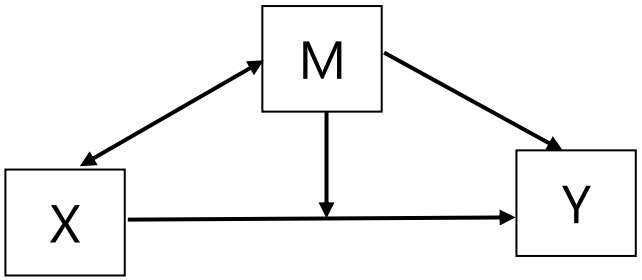
Mediation model (regression with better marketing):

- X **causes** M, M **causes** Y
- M is an outcome of X but a predictor of Y

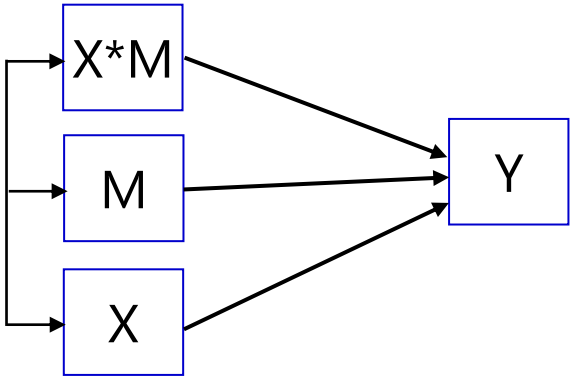


Moderation model:

- M adjusts the size of X→Y relationship
- M is a predictor of Y, and is **correlated** with X
- Moderation is represented by an **interaction** effect



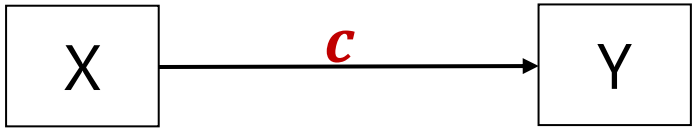
This figure does NOT depict an estimable model.

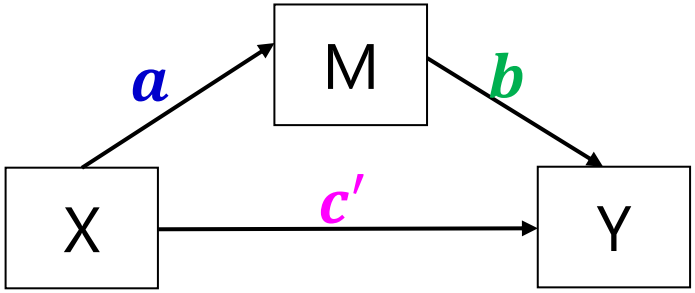


This is what is actually implied by above model.

Terminology: Mediation Effects

c = uncontrolled X to Y path
(Y regressed on X)





The big question in mediation:

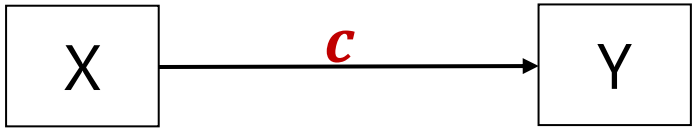
- Phrased as usual regression → *Is the effect of X predicting Y still significant after controlling for M?*
- Phrased as "mediation" → *Is the effect of X predicting Y significantly mediated by M?* **OR** *Is there a significant indirect effect of X through M in predicting Y?*
- Phrased either way, is $c \neq c'$?

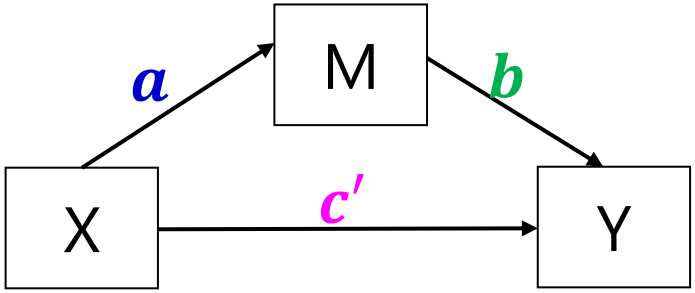
Direct Effects:

- a = X to M path (M on X;)
- b = M to Y path (Y on M;)
- c' = X to Y path controlled for M (Y on X;)
- $a * b$ = indirect effect of X to Y
- The estimates for $c - c'$ and $a * b$ will be equivalent in MVN observed variables (if same *N*)

Old versus New Rules for Mediation

c = uncontrolled X to Y path
(Y regressed on X)





- Baron & Kenny (1986, JPSP) rules were standard for a long time…
  - Simulation studies have found these rules to be way too conservative

- Old rule that can now be broken:
  - X must predict Y in the first place (c must be initially significant)
  - When not? Differential power for paths; suppressor effects of mediators
  - Mediation is really about whether $c \neq c'$, not whether each is significant

- Old rules that pry still hold:
  - X must predict M (a must be significant)
  - M must predict Y (b must be significant)

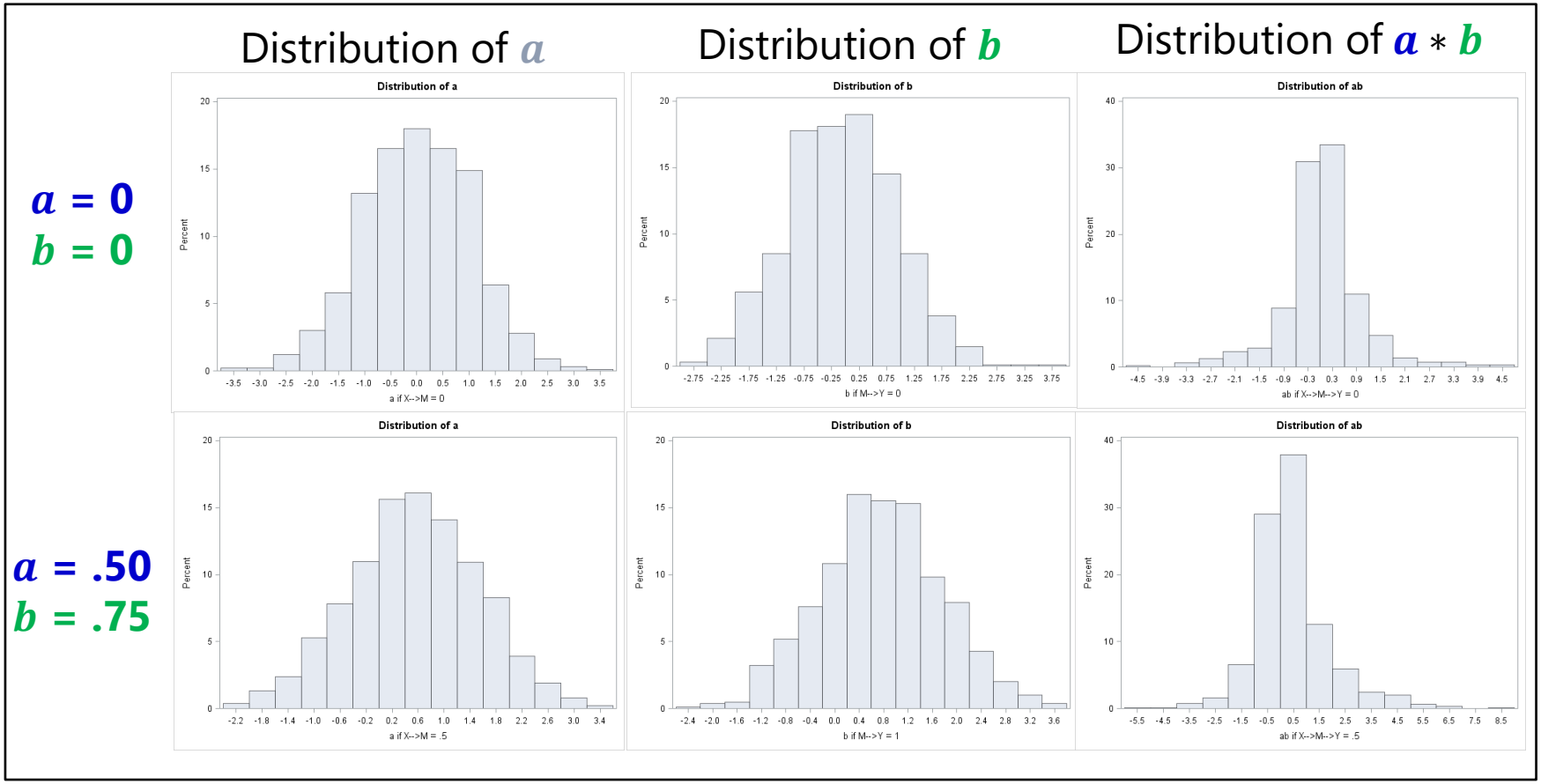
Testing Significance of Mediation

- Need to obtain a SE in order to test if $\boldsymbol{c} - \boldsymbol{c'} = \mathbf{0}$ or if $\boldsymbol{a} * \boldsymbol{b} = \mathbf{0}$
  - For $\boldsymbol{c} - \boldsymbol{c'}$ → "difference in coefficients SE"
  - For $\boldsymbol{a} * \boldsymbol{b}$ → "product of coefficients SE" → we'll start here

- Use "multivariate delta method" (second-derivative approximation shown here) to get SE for product of two random variables $\boldsymbol{a} * \boldsymbol{b}$
  - $SE_{a*b} = \sqrt{a^2 SE_b^2 + b^2 SE_a^2 + SE_a^2 SE_b^2}$
  - An equivalent formula to calculate SE_{a*b} that may have less rounding error because it avoids squaring $\boldsymbol{a}$ and $\boldsymbol{b}$ is $SE_{a*b} = \frac{ab\sqrt{t_a^2 + t_b^2 + 1}}{t_a t_b}$
  - This is known as the "Sobel test" and can be calculated by hand using the results of a simultaneous path model or separate regression models, also provided through MODEL INDIRECT/CONSTRAINT in Mplus, NLCOM in STATA SEM or GSEM, or TESTFUNC in SAS PROC CALIS

Testing Significance of Mediation

- One problem: we *shouldn't* use this SE for usual significance test
  - So, nope: $t_{indirect} = \frac{a*b}{SE_{a*b}}$ or $95\%\ CI = a * b \pm 1.96 * SE_{a*b}$
  - Why? Although the estimates for a and b will be normally distributed, the estimate of their product won't be, especially if a and b are near 0



Testing Significance of Mediation

- So what do we do? Another idea based on same premise:
  - For $\boldsymbol{a} * \boldsymbol{b}$ → find "distribution of the product SE" → $\boldsymbol{z_a} * \boldsymbol{z_b} = \frac{\boldsymbol{a}}{\boldsymbol{SE_a}} * \frac{\boldsymbol{b}}{\boldsymbol{SE_b}}$ in which the sampling distribution does not have a tractable form, but tables of critical values have been derived through simulation for the single mediator case (but may not generalize to complex models)
  - Implemented in PRODCLIN program for use with SAS, SPSS, and R

- A better solution: **bootstrap the data** to find the empirical SE and asymmetric CI for the indirect effect
  - Bootstrap = draw n samples with replacement from your **full data**, re-estimate mediation model and get $\boldsymbol{a} * \boldsymbol{b}$ for each bootstrap sample
  - Point estimate of $\boldsymbol{a} * \boldsymbol{b}$ is mean or median over n bootstrap samples
  - SE_{a*b} is standard deviation of estimated $\boldsymbol{a} * \boldsymbol{b}$ over n bootstrap samples
  - 95% CI can be computed as estimates at the 2.5 and 97.5 percentiles
  - Typically at least 500 or 1000 n bootstrap samples are used

Testing Significance of Mediation

- There are multiple kinds of bootstrap CIs possible in testing the significance of the $a * b$ indirect effect within MVN data
  - Regular bootstrap CI = "**percentile**" (as just described)
    - In Mplus, OUTPUT: CINTERVAL(bootstrap); in STATA SEM, vce(bootstrap)
  - **Bias-corrected bootstrap** CI = shifts CIs so median is sample estimate
    *** *Supposed to be best one*
    - In Mplus, OUTPUT: CINTERVAL(BCbootstrap); not sure about STATA SEM/GSEM
  - Accelerated bootstrap CI = ???
    - Not given in Mplus (as far as I know); not sure about STATA SEM

- For models with not simply MVN outcomes (i.e., non-normal mediators or outcomes, multilevel data), a different bootstrap approach can be implemented as a separate non-model step using any program's output
  - *Parametric*, *Monte Carlo*, or *empirical-M* bootstrap →
    Draw repeatedly from a and b parameter distributions instead of the data, then compute point estimates, SEs, and CIs from those distributions
  - See http://www.quantpsy.org/medn.htm for online calculators

Mediation with Non-Normal Variables

- The path models in this class (Example 4b, 5a, and 6a) have assumed every variable in the likelihood* is conditionally multivariate normal
  - * In the likelihood → is predicted by something or has an estimated mean, variance, or covariance (i.e., the missing data trick called "I used FIML")
  - In reality, one may have non-normal (NN) mediators or outcomes…
    - For mediation with a binary outcome, see Example 6b from same class

- Estimation gets tricky, because there is no closed-form ML anymore
  - NN outcomes → fit link function to Y, requires numeric integration
    - Becomes exponentially more complex with more non-normal variables
  - NN mediators → fit link function M, but estimation is even trickier
    - In Mplus, requires Monte Carlo integration (re-sampling approach)

- Interpretation gets tricky, because the paths are of different kinds
  - For example, X → M → binary Y: X → regular M, M → logit Y
  - For example, X → binary M → Y: X → logit M, regular M → Y
  - Oh, and there are no standard absolute model fit statistics in ML (no observed covariance matrix to compare the model predictions to)

Path Models and Mediation: Summary

- Path models are a very useful way to test many different types of multivariate hypotheses simultaneously:
  - Unique direct and indirect effects ("mediation")
  - Differences in effect size (via model constraints)
  - Mediation relationships (direct and indirect effects)

- Good fit is a pre-requisite to interpreting the model results, but good fit does *not* mean it is a good (useful) model
  - Good fit = model reproduces the covariance matrix of the variables (but it does not indicate how big or small those relationships are)
  - However – when all possible relationships are estimated (either as covariances or direct regressions), fit is perfect and irrelevant
    - Also known as "multivariate regression" with an "unstructured R matrix"

- Make sure you know what's happening to the predictor variables!
  - Are their means, variances, and covariances part of the likelihood? Then they have an assumed distribution (usually MVN), which may not make any sense!
  - Otherwise, they may result in dropped cases even when using "full-information" ML!

Structural Equation Modeling (SEM)

- The term "SEM" gets used to describe many different 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") if indicators are binary, ordinal, or nominal
    - "?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 the CATEGORICAL statement)
      - 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, variance)
  - The overall 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 likelihood: 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: 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 category frequencies** will be predicted perfectly
  - If each (normal) indicator has its own **residual variance**, then the indicator **total variance** 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 in Mplus
  - 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 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 **to see bivariate** relations first
    - 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 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 # estimated parameters)
  - Many estimated parameters (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)
  - Switching to Bayes estimation may fix at least some of this, but if not…

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

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)
   - See Cole & Preacher (2014) for an illustration

- 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???**
  - Total possible DF for actual 12 indicators = $\frac{12(12+1)}{2} + 12 = 90$
  - Estimated DF for actual 12 indicators = $12\lambda_i + 12\mu_i + 12\sigma_{e_i}^2 = 36$
  - 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" = $3\lambda_i + 3\mu_i + 3\sigma_{e_i}^2 = 9$
  - Remaining DF leftover = 9 − 9 = **0 = fit is "perfect" (just-identified)**

Option 1: Parceling Indicators

- So contrary to what others may say… **PARCELING <u>IS</u> 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 report this "parceling allocation variability"; see Sterba 2019 for more info)
  - **Be honest** that parceling is an intermediate choice between:
    - ASU completely (one sum score to replace a latent factor)
    - 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**

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 μ_i or $\sigma^2_{e_i}$**
  - 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**!
    - Mplus 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* 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 =`** $\mathbf{(1-\omega)} * \mathbf{Var}(\boldsymbol{sumscore_s})$**`;`**
    - 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 over 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 = random effect = central tendency 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:
  - $\boldsymbol{F_s = \lambda_{11} y_{1s} + \lambda_{21} y_{2s} + \lambda_{21} y_{3s}}$ → 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: $\boldsymbol{\rho} = \frac{\boldsymbol{\sigma_F^2}}{\boldsymbol{\sigma_F^2 + SE_{FS}^2}}$

    $\boldsymbol{\sigma_F^2}$ = factor variance from model solution
    $\boldsymbol{SE_{FS}^2}$ = error variance of EAP factor scores
  - **`Factor BY fscore@1; fscore* (ResVar); Factor*;`**
  - **`MODEL CONSTRAINT: Resvar =`** $\boldsymbol{(1-\rho) * (\sigma_F^2 + SE_{FS}^2);}$
    - 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 over 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));
```

Omega Reliability for *Sum* Scores

New/Additional Parameters

OMEGAP	**0.744**	**0.020**	**37.956**	**0.000**
OMEGAN	**0.775**	**0.014**	**56.803**	**0.000**

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

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

Option 2: Single-Indicator Models

- 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 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 means to make almost any idea an empirical question
  - Measurement models create latent constructs (= random effects) that better represent trait individual differences than 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