Multivariate Models Using Path Analysis

- Topics:
  - Path modeling diagram conventions and vocabulary
  - Path models for multivariate normal outcomes:
    - Model identification vocabulary
    - Tracing rules for predicted correlations, covariances, and variances
    - Testing fit of the variance–covariance matrix
  - Complications for path models with non-normal outcomes
  - Testing mediation through indirect effects

Path Models: General vs. Generalized

- The vast, vast majority of textbooks and resources for path models 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)
  - See Part 2 of Example 5; Mediation in Example 6a

- Software for path analysis involving **generalized outcomes** is harder to find and requires more complexity in estimation
  - **STATA GSEM**, **Mplus;** lavaan in R (binary or ordinal outcomes only)
  - See Example 6b: Mediation with two binary outcomes
  - Non-normal conditional distributions do not allow "direct" residual covariances, so covariances among outcomes must be specified using random intercepts (via latent factors)

Categorizing Estimation Options

- Path analysis software programs vary in their capabilities for **maximum likelihood estimation** (regular ML; no residual ML as “REML” available)
- For all MVN outcomes, the two options below provide the same result for complete cases, but differ in what happens to incomplete cases:
  - “**Full-information**” ML (FIML): uses all the individual data to estimate the model, and thus can include incomplete cases *to some extent*
    - Programs differ in options for incomplete predictors vs. outcomes
  - “**Limited-information**” ML: estimates the model using only a summary of the original data → variable means, variances, and covariances only
    - Must do listwise deletion of cases missing *any* variables in the model
- Another ML distinction is regular versus “**robust**”: Robust ML adjusts model fit statistics and parameter standard errors for deviations of multivariate non-normality using an estimated scaling factor (see Enders 2010 ch. 5)
  - Scaling factor = 1.000 = perfectly multivariate normal → same as regular ML!
  - Scaling factor > 1.000 = leptokurtosis (too-fat tails; fixes too big χ^2 in fit tests)
  - Scaling factor < 1.000 = platykurtosis (too-thin tails; fixes too small χ^2 in fit tests)
  - LRTs using robust ML with scaling factors are done differently (see next slide)
  - There are also **“robust” standard errors**, which don’t adjust fit statistics (I think)

Rescaled Likelihood Ratio Tests for MLR

- Compare nested models via a "**likelihood ratio test**" → **−2ΔLL** (MLR rescaled version—see Mplus documentation)
  - 1. Calculate **−2ΔLL** = $-2*(LL_{fewer} - LL_{more})$

    Fewer = simpler model
    More = more parameters

  - 2. Calculate **difference scaling correction** =

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

  - 3. Calculate **rescaled difference** = −2ΔLL / scaling correction
  - 4. Calculate **Δdf** = $\#parms_{more} - \#parms_{fewer}$
  - 5. **Compare rescaled difference to χ^2 with df = Δdf**
    - Add 1 parameter? $LL_{diff} > 3.84$, add 2 parameters: $LL_{diff} > 5.99$…
    - Absolute values of LL are meaningless (is relative fit only)
    - Process generalizes to many other kinds of models
- In R, the "anova" function will do LRTs for regular ML or MLR; I also have spreadsheets I made for this (in Example 4 in my SEM class, PSQF 6249)

Path Models: Pictures and Equations

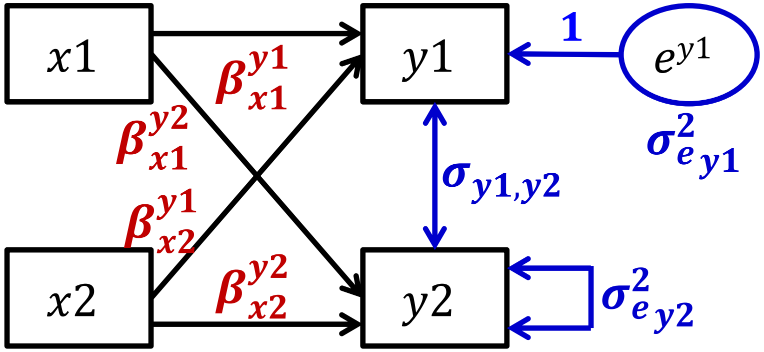
- So what are **path models**? "Truly" multivariate models for predicting 2+ outcomes simultaneously for the same unit of analysis
- Models most often expressed as a **diagram** using these conventions:
  - **Boxes** = observed variables; **ovals** = latent variables (in SEM) or residuals
  - **One-headed arrow** = fixed slope (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):

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

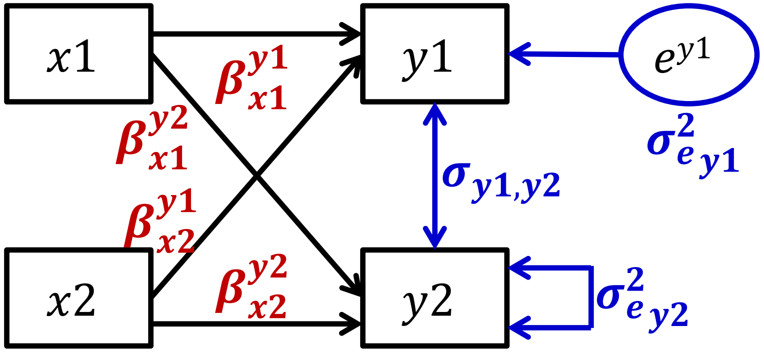
Unstructured R matrix for outcome residual variances and covariance(s): $\begin{bmatrix} \sigma^2_{e_{y1}} & \sigma_{12} \\ \sigma_{12} & \sigma^2_{e_{y2}} \end{bmatrix}$

The idea of a residual variable is either expressed using a separate oval (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



$x1$
$x2$
$y1$
$y2$
e^{y1}
$\boldsymbol{\beta}_{x1}^{y1}$
$\boldsymbol{\beta}_{x1}^{y2}$
$\boldsymbol{\beta}_{x2}^{y1}$
$\boldsymbol{\beta}_{x2}^{y2}$
$\boldsymbol{\sigma}_{y1,y2}$
$\boldsymbol{\sigma}_{e_{y1}}^2$
$\boldsymbol{\sigma}_{e_{y2}}^2$

- So why do both models at once?
  - To test differences in effect size (e.g., does $\boldsymbol{\beta}_{x1}^{y1} = \boldsymbol{\beta}_{x1}^{y2}$?)
  - 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 variables in their original scales:
  - **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]** + $\boldsymbol{\sigma_{e\,y1}^2}$

- **Standardized** → predicts $\boldsymbol{z}$**-scored** versions of variables instead:
  - Useful when comparing effects within a solution (are then on same scale)
  - Model parameters predict the variables' **correlations**
  - Standardized slope = $[\boldsymbol{\beta_{x1}^{y1}} * \boldsymbol{SD}(\boldsymbol{x1})] \,/\, \boldsymbol{SD}(\boldsymbol{y1})$ **= unique correlation**, but is not bounded by ±1 because the total SD is standardized, not the unique SD!
  - $\boldsymbol{R^2}$ for $\boldsymbol{y1} = \boldsymbol{1} -$ **standardized** $\boldsymbol{\sigma_{e\,y1}^2}$
  - Standardized solutions are usually only reported for path models with conditionally multivariate normal residuals (with estimated variances)

New (and Confusing) Terminology

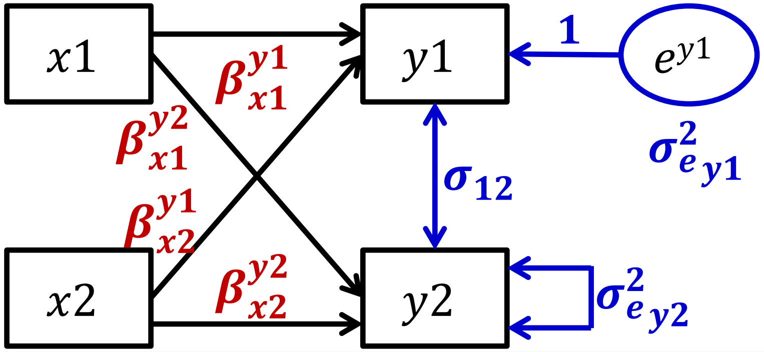
- **Predictors** are known as **exogenous** variables (X-ogenous to me)
- **Outcomes** are known as **endogenous** variables (IN-dogenous to me)
- Variables that are both at once are called **endogenous** variables

Our previous example model:

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

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

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



$x1$
$y1$
1
e^{y1}
$\boldsymbol{\beta_{x1}^{y1}}$
$\boldsymbol{\beta_{x1}^{y2}}$
$\boldsymbol{\beta_{x2}^{y1}}$
$\boldsymbol{\sigma_{12}}$
$\boldsymbol{\sigma_{e_{y1}}^2}$
$x2$
$\boldsymbol{\beta_{x2}^{y2}}$
$y2$
$\boldsymbol{\sigma_{e_{y2}}^2}$

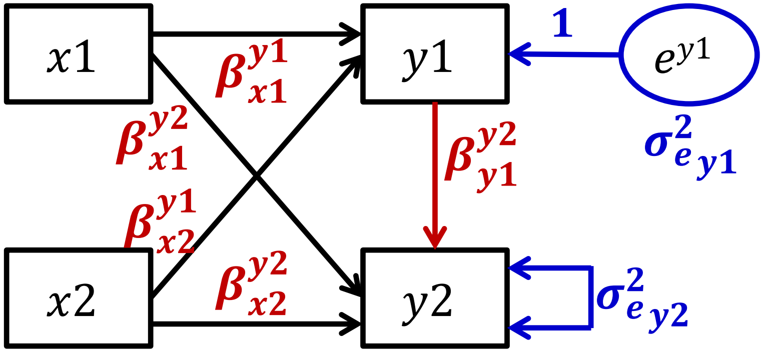
New Mplus code under MODEL:

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

Our modified example model:

$y1$ predicts $y2$ (still endogenous)

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



$x1$
$y1$
1
e^{y1}
$\boldsymbol{\beta_{x1}^{y1}}$
$\boldsymbol{\beta_{x1}^{y2}}$
$\boldsymbol{\beta_{y1}^{y2}}$
$\boldsymbol{\sigma_{e_{y1}}^2}$
$\boldsymbol{\beta_{x2}^{y1}}$
$x2$
$\boldsymbol{\beta_{x2}^{y2}}$
$y2$
$\boldsymbol{\sigma_{e_{y2}}^2}$

New Mplus code under MODEL:

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

New (and Confusing) Terminology

- Which 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** 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 both can do full-info ML
  - 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 missing at random (MAR), but they also then have distributional assumptions (MVN)
  - 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 in practice, that depends on what your software is doing!

- New, more comprehensive rule: **Is a variable part of 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 part of 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 structural equation models (which we will use to create a random intercept for non-normal outcomes)

- Need to know **Total DF** = possible and **Model DF** = remaining
  - In models in which all variables are in the likelihood as outcomes, **total DF =** $\boldsymbol{\frac{v(v+1)}{2}+v}$ 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 is 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 = # leftover

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 Mplus output labels switch from conditional to unconditional:
  - For a predictor that is part of the likelihood, the 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 re-create covariance)
- Estimated **regression path or residual covariance** of each predictor with each outcome (to re-create covariance)

- Note of terminology: if the "outcome" is not actually being predicted, then the <u>output labels switch from conditional to unconditional</u>:
  - For a predictor that is part of the likelihood, the 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

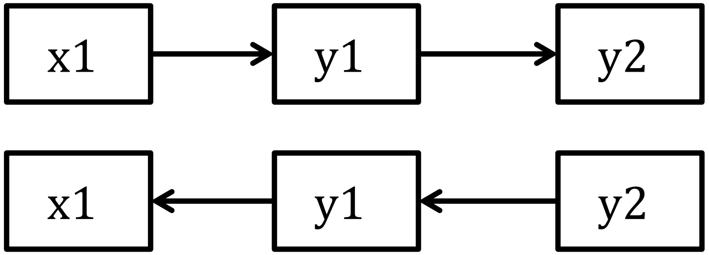
- Two things to know:
  - **Is the model estimable**—can all parameters be found? (= no redundancy)
  - **Is the absolute model fit testable**—can we determine if the parameters used adequately re-create the (outcome) means, variances, and covariances?
  - Comes from Model DF = # possible parameters − # 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)
    - Absolute model fit is then necessary before interpreting model results
      (is generally more of an issue for latent variable measurement models in SEM)

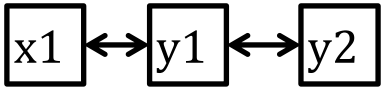
Model Identification Examples

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

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

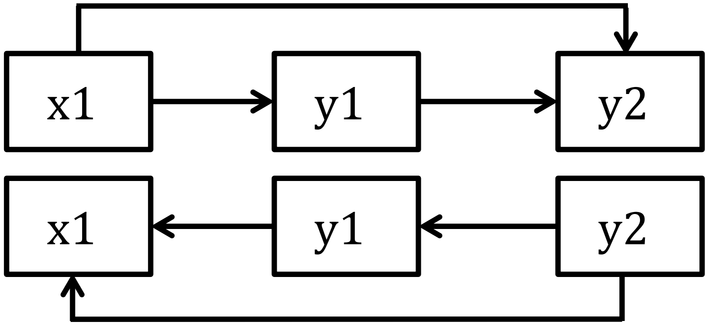


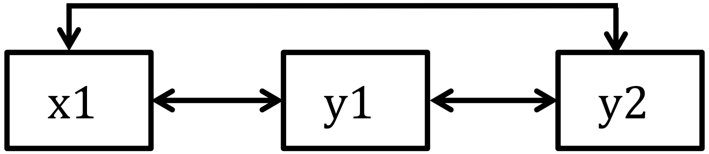
Right: also **DF=1**, but predicts no correlation of $x1$ with $y2$



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

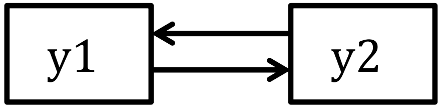
- Just-identified: have 0 DF leftover (possible = estimated)





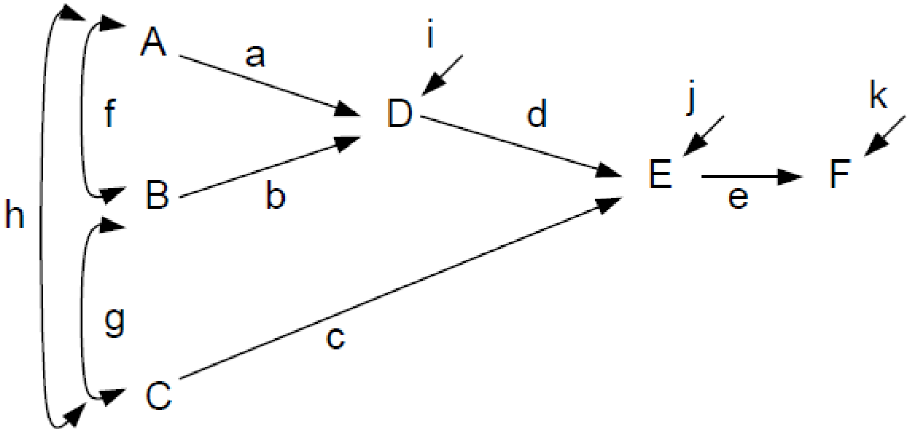
These 3 models all have **equivalent fit with DF=0** (for 0 missing direct relationships)

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



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

Wright's Rules of Tracing for Path Analysis



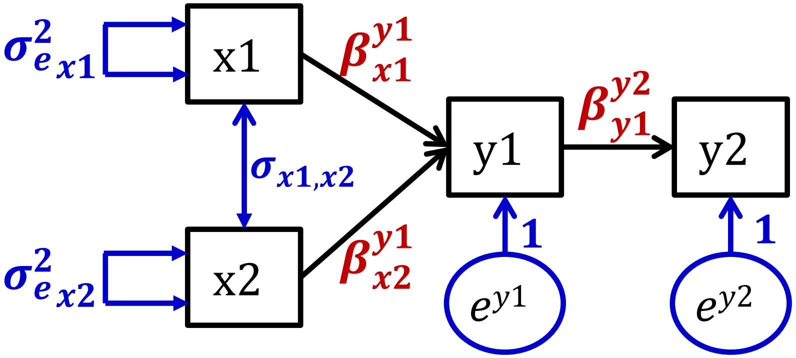
These rules below use correlations for convenience, but for covariances, the **variance** of a predictor variable that originates the path (or at any change in directions of directed arrows) gets included as a **multiplier** of the path:

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

- 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: $r_{AC}\neq fg$

Example borrowed from: http://web.pdx.edu/~newsomj/semclass/ho_path.pdf

Model-Predicted Covariances: Example



$$x1_i = \boldsymbol{\beta}_0^{x1} + \boldsymbol{e}_i^{x1}$$
$$x2_i = \boldsymbol{\beta}_0^{x2} + \boldsymbol{e}_i^{x2}$$
$$y1_i = \boldsymbol{\beta}_0^{y1} + \boldsymbol{\beta}_{x1}^{y1}(x1_i) + \boldsymbol{\beta}_{x2}^{y1}(x2_i) + \boldsymbol{e}_i^{y1}$$
$$y2_i = \boldsymbol{\beta}_0^{y2} + \boldsymbol{\beta}_{y1}^{y2}(y1_i) + \boldsymbol{e}_i^{y2}$$

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**:
  - $Cov(x1, x2) = \boldsymbol{\sigma}_{x1,x2}$
  - $Cov(x1, y1) = (\boldsymbol{\sigma}^2_{x1})\boldsymbol{\beta}^{y1}_{x1} + (\boldsymbol{\sigma}_{x1,x2})\boldsymbol{\beta}^{y1}_{x2}$
  - $Cov(x2, y1) = (\boldsymbol{\sigma}^2_{x2})\boldsymbol{\beta}^{y1}_{x2} + (\boldsymbol{\sigma}_{x1,x2})\boldsymbol{\beta}^{y1}_{x1}$
  - $Cov(y1, y2) = (\boldsymbol{\sigma}^2_{y1})\boldsymbol{\beta}^{y2}_{y1}$

2 covariances are only predicted by the other direct paths/covariances, and **will not be perfect**:
$Cov(x1, y2) = (\boldsymbol{\sigma}^2_{x1})\boldsymbol{\beta}^{y1}_{x1}\boldsymbol{\beta}^{y2}_{y1} + (\boldsymbol{\sigma}_{x1,x2})\boldsymbol{\beta}^{y1}_{x2}\boldsymbol{\beta}^{y2}_{y1}$
$Cov(x2, y2) = (\boldsymbol{\sigma}^2_{x2})\boldsymbol{\beta}^{y1}_{x2}\boldsymbol{\beta}^{y2}_{y1} + (\boldsymbol{\sigma}_{x1,x2})\boldsymbol{\beta}^{y1}_{x1}\boldsymbol{\beta}^{y2}_{y1}$

- The model-implied variances of $y1$ and $y2$ are complex but perfect because of each $\boldsymbol{e}_i$:
  - $Var(y1) = (\boldsymbol{\sigma}^2_{x1})\boldsymbol{\beta}^{y1}_{x1}\boldsymbol{\beta}^{y1}_{x1} + (\boldsymbol{\sigma}^2_{x2})\boldsymbol{\beta}^{y1}_{x2}\boldsymbol{\beta}^{y1}_{x2} + 2(\boldsymbol{\sigma}_{x1,x2})\boldsymbol{\beta}^{y1}_{x2}\boldsymbol{\beta}^{y1}_{x1} + \boldsymbol{\sigma}^2_{ey1}$
  - $Var(y2) = (\boldsymbol{\sigma}^2_{x1})\boldsymbol{\beta}^{y1}_{x1}\boldsymbol{\beta}^{y1}_{x1}\boldsymbol{\beta}^{y2}_{y1}\boldsymbol{\beta}^{y2}_{y1} + (\boldsymbol{\sigma}^2_{x2})\boldsymbol{\beta}^{y1}_{x2}\boldsymbol{\beta}^{y1}_{x2}\boldsymbol{\beta}^{y2}_{y1}\boldsymbol{\beta}^{y2}_{y1} + 2(\boldsymbol{\sigma}_{x1,x2})\boldsymbol{\beta}^{y1}_{x2}\boldsymbol{\beta}^{y1}_{x1}\boldsymbol{\beta}^{y2}_{y1}\boldsymbol{\beta}^{y2}_{y1}$
    $+(\boldsymbol{\sigma}^2_{ey1})\boldsymbol{\beta}^{y2}_{y1}\boldsymbol{\beta}^{y2}_{y1} + \boldsymbol{\sigma}^2_{ey2}$

Assessing the Fit of Path Models

- If your model is over-identified (have positive DF leftover), then you can **assess absolute fit of your model as "$\boldsymbol{H_0}$"**
- Software estimates two baseline models for you that it uses as comparisons in likelihood ratio tests (LRTs) and other fit statistics:
  - **Best** = "**Saturated**" or "**Unstructured**" model (as "H_1")
    - All outcome means, variances, and covariances are estimated, as well as all covariances of exogenous predictors with outcomes (count as DF)
    - Exogenous predictor means, variances, and their covariances are ignored
    - Output provides LRT as χ^2 for how much WORSE your H_0 model is
  - **Worst** = "**Independence**" or "**Null**" model
    - Each outcome gets a mean and variance, but all covariances forced = 0
    - Exogenous predictor means, variances, and their covariances are ignored, but their covariances with outcomes are also forced to 0 (count as DF)
    - Output provides LRT as χ^2 for how much WORSE the null model is than the saturated model—this has nothing to do with your H_0 model, but it does tell you if you have any covariances worth modeling!

4 Steps in Assessing Model Fit

1. **Global model fit**
   - *Does the model "work" overall: Does it reproduce the observed data?*
   - *Data = means, variances, and covariances*

2. Local model fit
   - *Are there any more specific problems (that cause global misfit)?*

3. Inspection of model parameters
   - *Are the estimates, SEs, and the item responses they predict plausible?*

4. Effect size
   - *How strong are the paths? How well are the outcomes predicted?*

Step 1: Indices of Global Model Fit

- Primary fit index: obtained model $\boldsymbol{\chi^2 = 2 * N * F_{ML}}$
  - χ^2 is evaluated based on model DF (# parameters left over)
  - Tests null hypothesis that $\boldsymbol{\Sigma} = \boldsymbol{S}$ (that model = data is perfect), so **significance is bad** (i.e., smaller χ^2, bigger *p*-value is better)
    - Is LRT ($-2\Delta LL$) of your H_0 model versus saturated best H_1 model
    - Btw, don't use "ratio rules" like $\chi^2/\mathrm{DF} > 2$ or $\chi^2/\mathrm{DF} > 3$
  - Just using χ^2 to index model fit is usually insufficient, however:
    - χ^2 depends largely on sample size (is overpowered with large N)
    - Is "unreasonable" null hypothesis (perfect fit, really??)
    - Btw, χ^2 is only possible given balanced data (as typical for path models)

- Because of these issues, additional fit indices are usually used in conjunction with the χ^2 test (that are like fit effect sizes)
  - **Absolute** Fit Indices (besides χ^2)—relative to "**saturated**" best model
  - **Comparative** (Incremental) Fit Indices—relative to "**null**" worst model
  - Cite a reference for any cut-offs you use… it's now more complicated!

Step 1: Indices of Global Model Fit

- Absolute Fit: **SRMR**
  - **Standardized Root Mean Square Residual**
  - Get difference of standardized $\mathbf{S} - \mathbf{\Sigma}$ → "residual" (leftover) matrix
  - Sum the squared residuals of the predicted correlation matrix across items, divide by number of matrix elements, then take square root:
  - $SRMR = \sqrt{\frac{2\sum_{i=1}^{I}\sum_{j=1}^{J}\left[\frac{s_{ij}-\sigma_{ij}}{s_{ii}s_{jj}}\right]^2}{I(I-1)}}$
  - Ranges from 0 to 1: **smaller is better**
  - Convention: ".08 or less" → good fit

- Less common variant: **RMR** (**Root Mean Square Residual**)

Step 1: Indices of Global Model Fit

Parsimony-Corrected: **RMSEA**

- **Root Mean Square Error of Approximation**
- Relies on a “non-centrality parameter” (NCP) for T (target H_0)
  - NCP indexes how far off your model is → adjusted χ^2 distribution
  - $NCP_T = \max(\chi^2_T - DF_T, 0)$ → scaled discrepancy $d_T = NCP_T/N$
  - $\text{RMSEA} = \sqrt{\frac{\max(\chi^2_T - DF_T, 0)}{DF_T * N}} = \sqrt{\frac{d}{DF_T}}$ → how far off per Model DF left

- RMSEA ranges from 0 to 1; **smaller is better**
  - Conventions: < .05 or .06 = “good”, .05 to .08 = “adequate”
  - In addition to point estimate, get 90% confidence interval (CI)
  - RMSEA penalizes for model complexity—it’s discrepancy in fit per DF left in model (but not sensitive to N, although CI can be)
  - Also get test of “close fit”: null hypothesis that $\text{RMSEA} \leq .05$

Step 1: Indices of Global Model Fit

Comparative (Incremental) Fit Indices (bigger is better)

- Fit evaluated relative to "null" (independence) model of 0 covariances
- Relative to that, your model fit should be great!
- Conventions: > .90 = "adequate", > .95 = "good"

- **CFI: Comparative Fit Index (ranges from 0 to 1)**
  - Also based on idea of NCP ($\chi^2_T - \mathrm{DF}_T$)
  - $CFI = \dfrac{\max(\chi^2_N - DF_N, 0) - \max(\chi^2_T - DF_T, 0)}{\max(\chi^2_N - DF_N, 0)}$

T = target model (H_0)
N = null model (no covariances)

- **TLI: Tucker-Lewis Index** (= **Non-Normed Fit Index**)
  - $TLI = \dfrac{\frac{\chi^2_N}{DF_N} - \frac{\chi^2_T}{DF_N}}{\frac{\chi^2_N}{DF_N} - 1}$ (so can go negative or > 1)

4 Steps in Model Evaluation

1. **Assess global model fit (summary)**
   - Recall that variable means and residual variances are usually just-identified → *so misfit comes from mis-predicted covariances*
   - χ^2 is sensitive to large N, so pick at least one global fit index from each class; hope they agree (e.g., CFI, RMSEA) that fit is "good"

- Conventions of "good" absolute model fit largely stem from simulation studies reported in Hu & Bentler (1999)
  - Cited >100,000 times! But no one study can cover everything…
    - Held indicator reliability relatively constant: standardized loadings .70-.80
    - Small-ish model of 15 indicators measuring 3 correlated factors
    - Complete data, generated using perfectly multivariate normal indicators
  - Research now suggests standards for what is "good" model fit will vary significantly as a function of these unaddressed features…
    - For more info, see Lectures 4 and 5 (and associated readings) from PSQF 6249

4 Steps in Model Evaluation: Step 2

2. **Identify local misfit: localized model strain**

- Global model fit means that the observed and predicted outcome covariance matrices aren't too far off on the whole… this says nothing about the specific covariances to be predicted
- Should inspect **normalized model residuals** for that → Local misfit
  - **RESIDUAL** output option in Mplus, residual () in R lavaan, or ESTAT RESIDUAL in STATA
  - "Normalized" is residual/SE → **works like a z-score**
  - Relatively large absolute values indicate "localized strain"
  - **Positive** residual → outcomes are **more** related than you predicted
    - More than just your model creating a covariance
  - **Negative** residual → outcomes are **less** related than you predicted
    - Not as related as the model said they should be
- **Evidence of localized strain tells you where the problems are, but not what to do about them…**

4 Steps in Model Evaluation: Step 2

2. Identify localized model strain, continued…

- Parallel approach: **Modification Indices** (*aka*, cheat codes)
  - LaGrange Multiplier: decrease in χ^2 by adding the listed model parameter (e.g., residual covariance or direct path)
    - Usually only pay attention if > 3.84 for DF=1 (for p < .05)
    - Get expected parameter estimate for what's to be added – but should only pay attention if its effect size is meaningful
    - Also only pay attention if you can INTERPRET AND DEFEND IT
  - Implement these ONE AT A TIME, because one addition to the model can alter the rest of the model substantially

- Keep in mind that these "manipulation indices" can only try to repair your current model; they will never suggest a new model!
  - More of an issue in latent variable measurement models, though

Model Evaluation: Steps 1, 2, and 3

1. **Assess global absolute model fit**
   - Recall that variable means and variances are perfectly predicted (just-identified) → *so misfit comes from badly recreated covariances*
   - χ^2 is sensitive to large sample size, so pick at least one global fit index from each class (e.g., CFI, RMSEA); use cutoffs with caveats
2. **Identify localized model strain**
   - Global model fit means that the observed and recreated variable covariance matrices aren't too far off on the whole… this doesn't guarantee each specific covariance is recreated well
   - Consider normalized residuals and modification indices to try and "fix" 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)

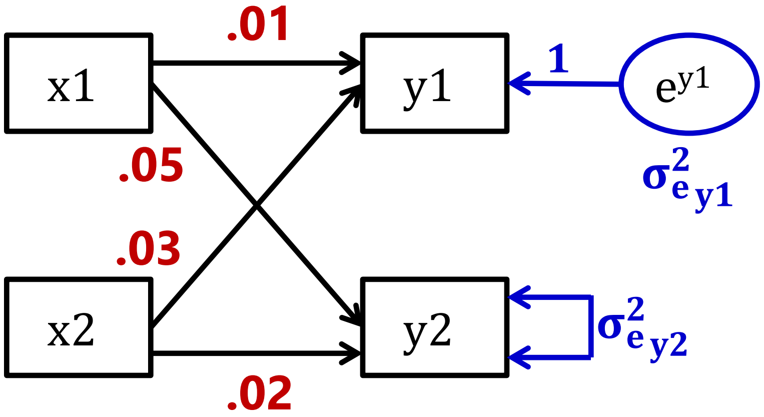
Good global and local fit? Great, but we're not done yet…

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

- There are fewer path model software options available that include link functions and non-normal conditional distributions
  - I am most familiar with Mplus (and still learning STATA GSEM)
  - These two vary in options for outcome types and estimation methods
  - I did not find anything directly comparable in R (lavaan currently only has limited-information estimation for ordinal outcomes, and no other link functions)
    - But please let me know if anyone knows better!
- Differences compared to path models with MVN outcomes
  - No residual variances means:
    - Traditional measures of absolute fit are not available when using full-info ML
    - Conventional standardized solutions may not be available
    - Residual covariances must be introduced via random intercepts (latent factors)
  - 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
- For binary or ordinal outcomes, there are also many limited-info estimators using weighted least squares
  - More on this… (but mostly for your reference for now)

What is WLSMV Estimation in Mplus?

- **WLSMV:** "Weighted Least Square parameter estimates use a diagonal weight matrix and a Mean- and Variance-adjusted χ^2 test"
  - Called "diagonally-weighted least squares" by non-Mplus people
  - Also available in lavaan in R (for path models and structural equation models)

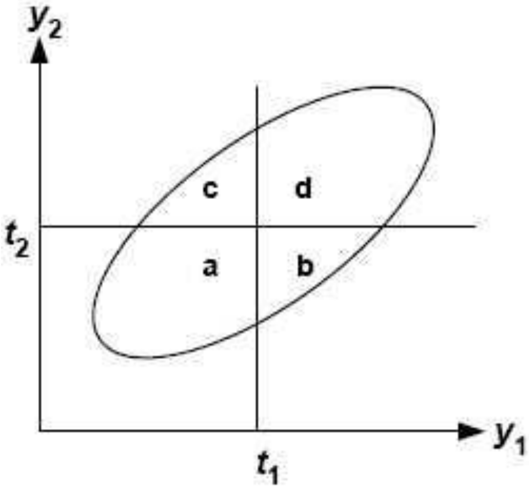
- Translation: **WLSMV** is a **limited-information** estimator that uses a different summary of responses instead → **a "linked" covariance matrix**

- Absolute fit can then be assessed in regular ways, because what is trying to be reproduced is a type of covariance matrix (that has residual variances)
  - So residual covariances can again be included directly (i.e., like in an R matrix)
  - So indices of local fit (RESIDUAL output option) are again available
  - But the observed covariance matrix would have correlations with restricted range for binary or ordinal outcomes… so what does it do instead?

WLSMV Estimation

Data	y_2 = 0	y_2 = 1
y_1 = 0	a	c
y_1 = 1	b	d

Use the observed proportions as the area under the curve of each section of the bivariate distribution to determine what the correlation would be →



- WLSMV first estimates correlation matrix of underlying continuous responses (probit scale only; logit scale is not available)
    - For **binary** responses → "**tetrachoric** correlation matrix"
    - For **ordinal** (polytomous) responses → "**polychoric** correlation matrix"
- The model then tries to find parameters to predict this new correlation matrix
- The diagonal W "weight" part then tries to emphasize reproducing latent variable correlations that are relatively well-determined more than those that aren't
    - The full weight matrix is of order z*z, where z is number of elements to estimate
    - The "diagonal" part means it only uses the *preciseness of the estimates themselves*, not the covariances among the "preciseness-es" (much easier, and not a whole lot of info lost)
- The "MV" corrects the χ^2 test for bias arising from this weighting process

More about WLSMV Estimation

- Works much faster than ML when you have many latent variables in the model (because no rectangling via quadrature is required)
- Does assume missing data are **missing completely at random,** whereas full-info ML assumes only *missing at random* (conditionally random)
- Because a covariance matrix (on probit scale) is used as the input data, we get absolute fit indices as in path models with MNV outcomes
  - People tend not to be as strict with cut-off values (is an active area of research)
  - Research suggests RMSEA tends to vary by number of outcome categories

- Model coefficients will be on the **probit scale** instead of logit scale
- Two different model variants in Mplus via the **PARAMETERIZATION** option on the **ANALYSIS** command, where a 1 is needed for identification
  - "**Delta**" (default): total y_i^* variance = 1 = "marginal parameterization"
  - "**Theta**": residual variance = 1 instead = "conditional parameterization"
    - **IN SEM, I USE THIS ONE TO HELP SIMPLIFY IRT CONVERSIONS**

Model Comparisons with WLSMV using DIFFTEST in Mplus

- Not the same process! Model DF is NOT calculated in usual way, and model fit is not compared in the usual way
  - Absolute χ^2 model fit values are meaningless—they are not comparable!
  - Difference in model χ^2 are not distributed as χ^2
- Here's how you do nested model comparisons in WLSMV:
  - Step 1: Estimate model with *more* parameters, adding this command:
    - SAVEDATA: DIFFTEST=more.dat; → Saves needed derivatives
  - Step 2: Estimate model with *fewer* parameters, adding this command:
    - ANALYSIS: DIFFTEST=fewer.dat; → Uses those derivatives to do $\Delta\chi^2$ test
  - Step 2 model output will have a new χ^2 difference test in it that you can use, with df difference to compare to a χ^2 distribution

Options for Residual Covariances

- Additional relationships between outcomes can be included:
  - Via **residual covariances** (the same as in MVN models) when using **WLSMV** because the model is being estimated on the tetrachoric /polychoric correlation matrix (so the residuals of the underlying probit can covary, even if item residual variances are not being estimated)
  - Residual covariances are not allowed when using maximum likelihood
    - Instead, you can specify a "**random intercept**" (in WLSMV or ML)
- An example using Mplus WLSMV to demonstrate both ways:

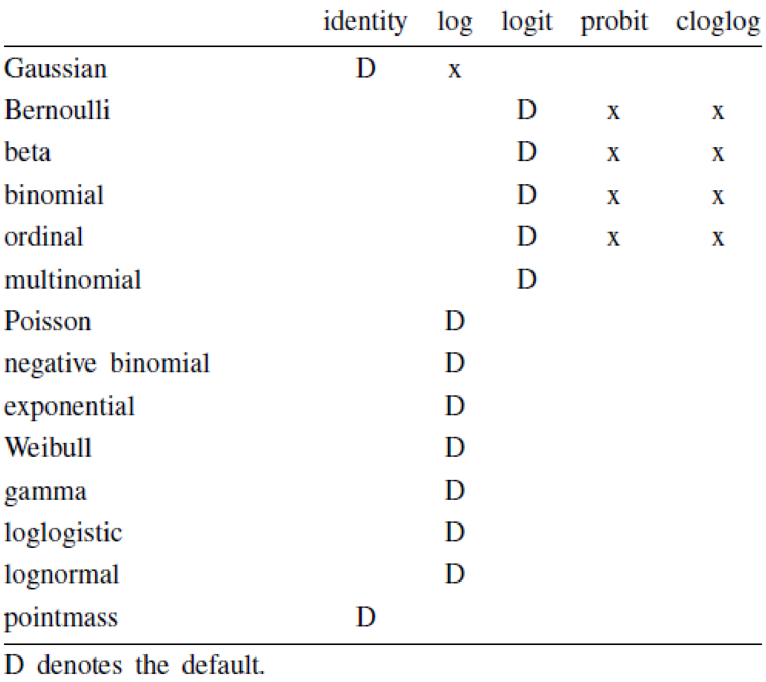
```
! Residual covariance
! directly;
dv2 WITH dv3*;
```

```
! Random intercept as a latent
! variable to create residual
! covariance indirectly;
RandInt BY dv2@1 dv3@1;

! Shut off mean, estimate variance;
[RandInt@0]; RandInt*;
```

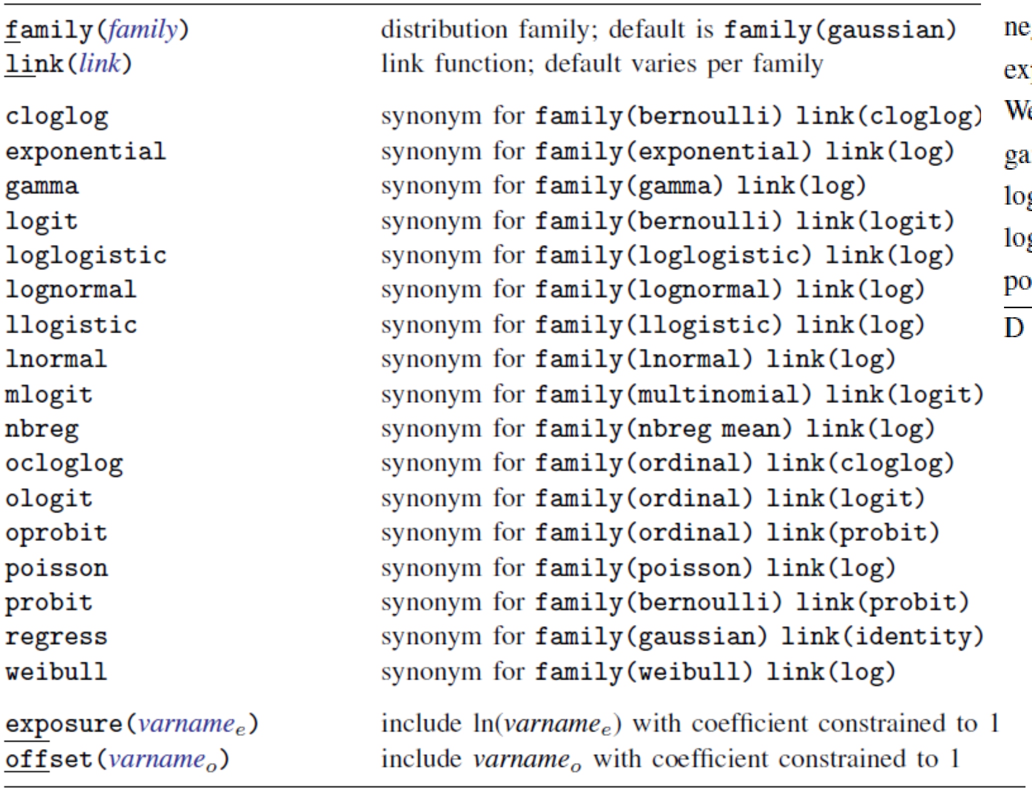
STATA GSEM for Generalized Outcomes

- 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…

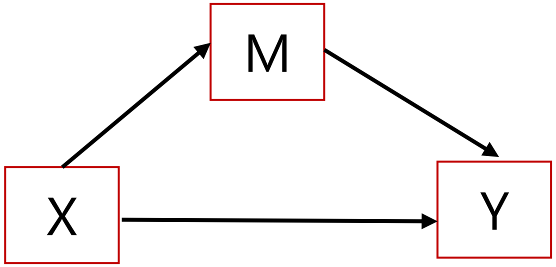
Estimation in STATA GSEM

- 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

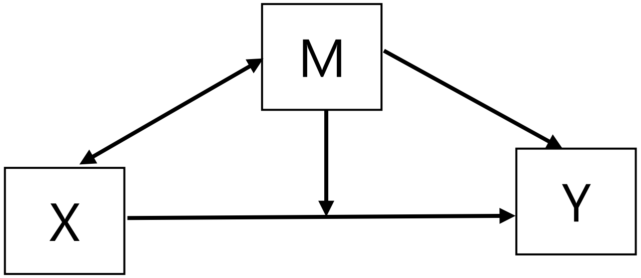
Mediation model → regression with better marketing:

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

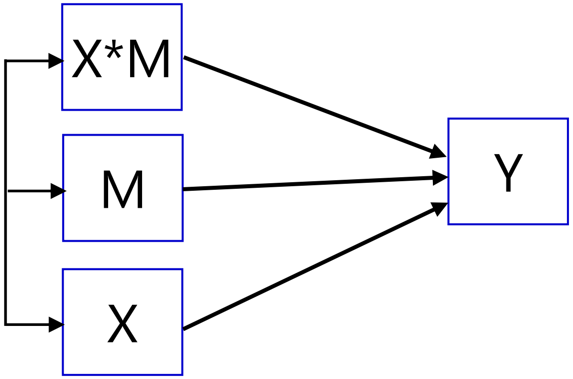


Moderation model:

- 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



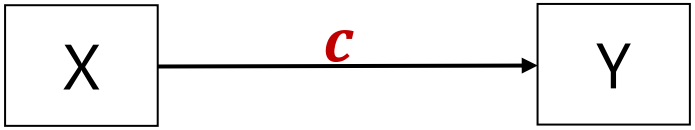
This figure does NOT depict an estimable model.



This is what is actually implied by above model.

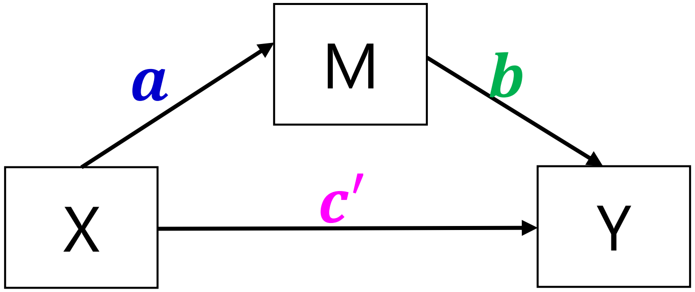
Terminology: Mediation Effects

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 $c \neq c'$?



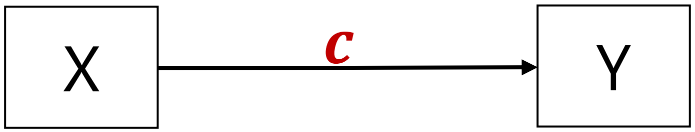
Direct Effects:

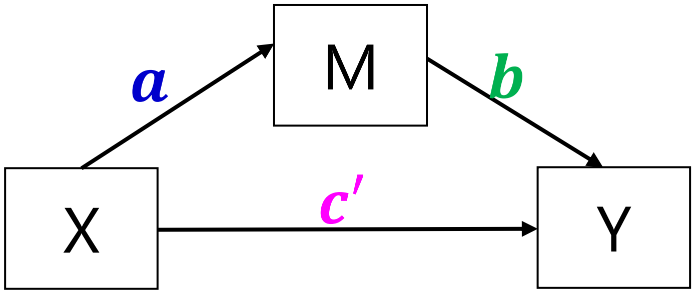
- a = X to M path (M on X;)
- b = M to Y path (Y on M;)
- c' = X to Y path controlled for M (Y on X;)
- $a * b$ = indirect effect of X to Y

- The estimates for $c - c'$ and $a * b$ will be equivalent using MVN observed variables (if same *N*)
  - Otherwise, $a * b$ should be used to get the indirect effect instead

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" → not generalizable
  - 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, TESTFUNC in SAS PROC CALIS, or user-defined new terms in R LAVAAN

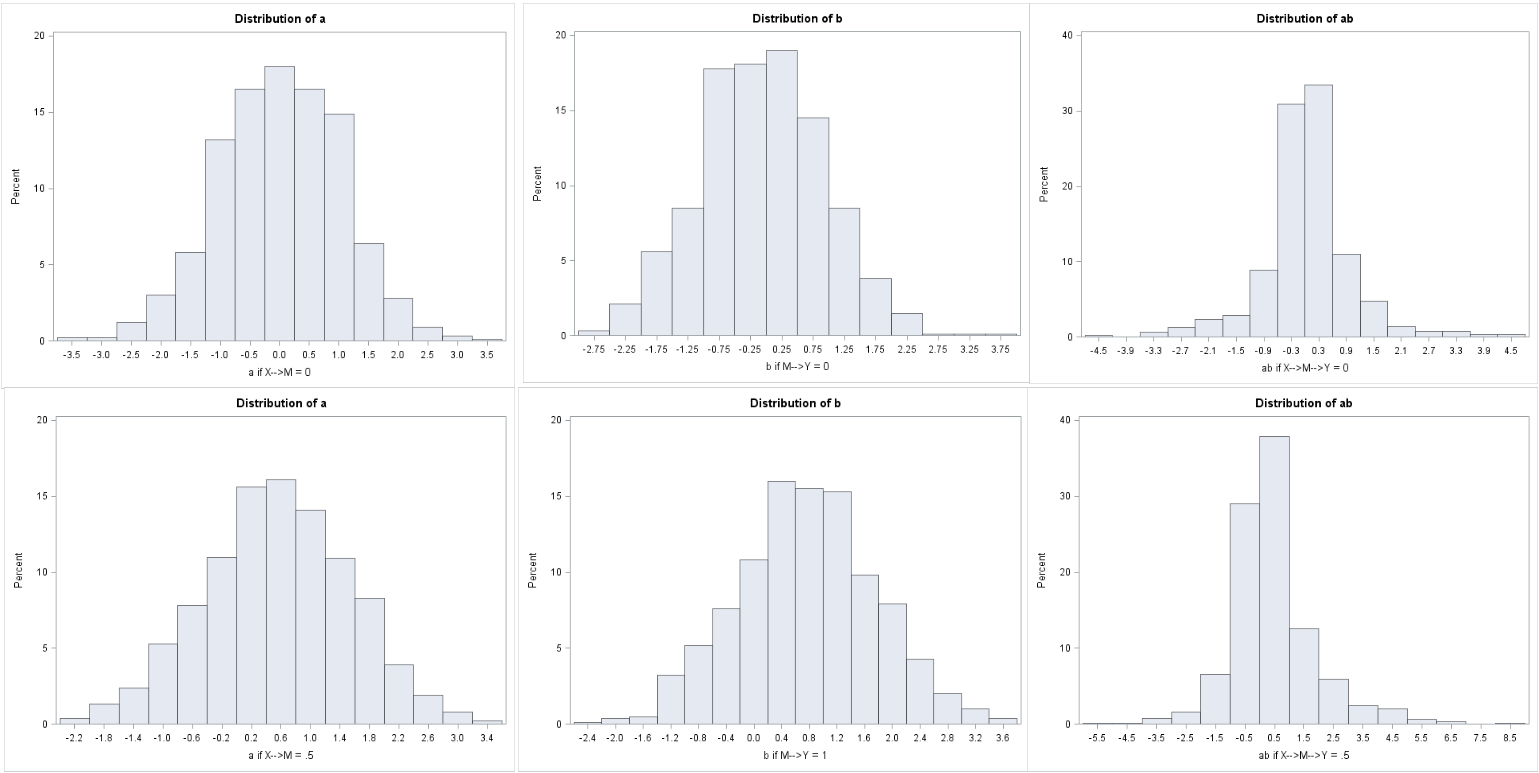
Testing Significance of Mediation

- 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

Distribution of ***a*** | Distribution of ***b*** | Distribution of ***a*** $*$ ***b***

a = 0
b = 0

a = .50
b = .75



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 $\boldsymbol{a} * \boldsymbol{b}$ 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 $\boldsymbol{a}$ and $\boldsymbol{b}$ 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

- All the path models shown so far (in Example 5 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…

- Estimation gets tricky, because there is no closed-form ML anymore
  - NN outcomes → link function for Y, so may require numeric integration
    - In Mplus, see this page or these slides for more info
  - NN mediators → link function for M, but interpretation differs by estimator
    - In Mplus, see p. 551 of their user guide

- Interpretation gets tricky, because the paths are then of different kinds
  - e.g., X → M → binary Y: X → regular M, M → logit/probit Y
  - e.g., X → binary M → Y: X → logit/probit M, probit/regular? M → Y
  - Fewer easy options for other kinds of mediations (e.g., nominal, count)

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 and/or Wald tests for difference)
  - Differences in mediation relationships (comparisons 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 sense!
  - Otherwise, they may result in dropped cases even when using "full-information" ML!