Structural Equation Modeling with Missing Data

Paul Johnson[1], Chong Xing[1], and Po-Yi Chen

[1]Center for Research Methods and Data Analysis

2019

Outline

The SEM Community

"The missing data problem has long been an issue for data analysis of all kinds, and structural equation modeling (SEM) was, in the early days, not exempt from such problems in those early days .... In those early days, new statistical procedures first focused on well-behaved data (e.g., normal distributions, no missing values), and only later began to address problem data (e.g., non-normal data, missing values)."

Graham and Coffman (2012, p. 277)

Outline

Reminder Missing Mechanism

MCAR **m**issing **c**ompletely at **r**andom - the cause of missingness (Y_{miss}) on a variable (Y) is uncorrelated with all other variables (including itself)

MAR **m**issing **a**t **r**andom - the cause of Y_{miss} is some other observed variables, but not Y

MNAR **m**issing **n**ot **a**t **r**andom - the cause of Y_{miss} is Y itself (or other unobserved variables)

MCAR and MAR can be managed!

but MNAR is, mostly, an unsolved problem

SEM Estimation

- In SEM model fitting, we often are able to describe the estimator as the comparison of the observed covariance matrix with a theoretical or "model implied" covariance.
- In the presence of missing values, several methodological avenues appear
    - Create an approximate sample covariance matrix with "pairwise complete" variables. Proceed as if there are no missings after that
    - FIML: Full Information Maximum Likelihood on covariance structures within data subgroups.
- Other Strategies not based on "covariance structures"
    - Individual level "casewise" Maximum Likelihood (referred to as IRT by some SEM practitioners)
    - Multiple Imputation
    - Bayesian estimation

Outline

Method 1: Listwise Deletion

Listwise Deletion Dropping any case with any missing values on variables to be included in the analysis model

If data is actually **MAR** then listwise deletion can lead to poor parameter estimates

1. Parameter estimates, such as factor loadings can be biased
2. Will distort other elements in the SEM as well (Enders & Bandalos, 2001)

Outline

Method 2: Covar Matrix from Pairwise Complete

Pairwise complete covariance: Estimate covariance for pairs of variables using "complete cases".

Problems:

1. Not be positive definite (not consistent with Normal assumption)
2. Even under **MCAR**, Type-I error rate for χ^2 global model fit test may be inflated
3. Under **MAR**, factor correlations and regression coefficients can be underestimated

Outline

We have to understand ML first

- FIML is a general stats term, but in SEM it means something particular
- We need to take a detour to explain ML in covariance structures analysis
- Once we lay out the terminology for ML and the fitting function known as F_{ML}, we can demonstrate why FIML works as it does.

Maximum Likelihood

- Maximum Likelihood (Fisher, 1922; Aldrich, 1997): Choose parameters (collectively referred to as $\boldsymbol{\theta}$) to make the data as likely as possible to have occurred from a designated mechanism
- $\mathbf{S}$ is a sample-based covariance matrix–a data summary
- $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ SEM model-implied covariance: based on loadings, regression coefficients, variance parameters

Where MVN comes into the story

- Each case's data is assumed to be a draw from an MVN random process.
  - The covariance matrix is called $\boldsymbol{\Sigma}$ and the vector of expected values is called $\boldsymbol{\mu}$.
- Stack together all of the indicator variables

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{p+q} \end{bmatrix}$$

The total number of indicators is $p + q$

p # of indicators for endogenous (outcome) latent variables
q # of indicators for the exogenous (predictor) latent variables)

KU

SEM reformulation: Covariance Structures

- The ML objective function can be mathematically re-arranged as the *minimization* of the following "Fit" function (see Brown, 1974, p. 13; Bollen, 1989, p. 107).

$$F_{\mathbf{ML}} = log|\mathbf{\Sigma}(\boldsymbol{\theta})| - log|\mathbf{S}| + tr(\mathbf{S}\{\mathbf{\Sigma}(\boldsymbol{\theta})\}^{-1}) - (p+q)$$

$\mathbf{S}$ and $\mathbf{\Sigma}(\boldsymbol{\theta})$ are $(p+q) \times (p+q)$ symmetric matrices.
$\mathbf{\Sigma}(\boldsymbol{\theta})$ is the "model-implied" covariance matrix.
$\mathbf{\Sigma}(\boldsymbol{\theta})^{-1}$ is the inverse of the model-implied covariance matrix

Covariance Structure: Worked Example

- It is pretty easy to see where $\mathbf{\Sigma}$ comes from in a simple CFA model. It is just tedious to write it out.
- But we will write out one example, nevertheless.
- Suppose there are 4 indicators with 2 latent variables, ξ_1 and ξ_2

$$\begin{bmatrix} x_{1i} \\ x_{2i} \\ x_{3i} \\ x_{4i} \end{bmatrix} = \begin{bmatrix} \lambda_{11} & 0 \\ \lambda_{21} & 0 \\ 0 & \lambda_{32} \\ 0 & \lambda_{42} \end{bmatrix} \begin{bmatrix} \xi_{1i} \\ \xi_{2i} \end{bmatrix} + \begin{bmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \\ \varepsilon_{i3} \\ \varepsilon_{i4} \end{bmatrix}$$

$$\mathbf{x} = \mathbf{\Lambda}\xi + \varepsilon$$

- We need to calculate $Var(\mathbf{x})$, which is referred to as $\mathbf{\Sigma}$ or $\mathbf{\Sigma}(\boldsymbol{\theta})$ in these notes.

KU

Covariance Structure: Worked Example ...

- Let the variance matrices of ξ and ε be

$$Var(\xi) = \begin{bmatrix} \sigma^2_{\xi_1} & \sigma_{\xi_1\xi_2} \\ \sigma_{\xi_1\xi_2} & \sigma^2_{\xi_2} \end{bmatrix} \quad Var(\varepsilon) = \begin{bmatrix} \sigma^2_{\varepsilon_1} & 0 & 0 & 0 \\ 0 & \sigma^2_{\varepsilon_2} & 0 & 0 \\ 0 & 0 & \sigma^2_{\varepsilon_3} & 0 \\ 0 & 0 & 0 & \sigma^2_{\varepsilon_4} \end{bmatrix}$$

- The law of variance states, for uncorrelated ξ and ε:

$$Var(\mathbf{x}) = Var(\boldsymbol{\Lambda}\boldsymbol{\xi} + \boldsymbol{\varepsilon})$$

$$= Var(\boldsymbol{\Lambda}\boldsymbol{\xi}) + Var(\boldsymbol{\varepsilon})$$

$$= \boldsymbol{\Lambda} Var(\boldsymbol{\xi})\boldsymbol{\Lambda}^T + Var(\boldsymbol{\varepsilon})$$

Covariance Structure: Worked Example ...

- Which works out to be a 4×4 matrix

$$Var(\mathbf{x}) = \mathbf{\Sigma} = \begin{bmatrix} \sigma^2_{\xi_1}\lambda^2_{11} + \sigma^2_{\varepsilon_1} & \sigma^2_{\varepsilon_1}\lambda_{21}\lambda_{11} & \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{32} & \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{42} \\ \sigma^2_{\xi_1}\lambda_{11}\lambda_{21} & \sigma^2_{\varepsilon_1}\lambda^2_{21} + \sigma^2_{\varepsilon_2} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{32} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{42} \\ \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{32} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{32} & \sigma^2_{\xi_2}\lambda^2_{32} + \sigma^2_{\varepsilon_3} & \sigma^2_{\xi_2}\lambda_{32}\lambda_{42} \\ \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{42} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{42} & \sigma^2_{\xi_2}\lambda_{32}\lambda_{42} & \sigma^2_{\xi_2}\lambda^2_{42} + \sigma^2_{\varepsilon_4} \end{bmatrix}$$

- That's $\mathbf{\Sigma}(\theta)$, the model-implied covariance matrix
- The **parameter vector** $\boldsymbol{\theta} = (\lambda_{11}, \ldots \lambda_{42}, \sigma^2_{\xi_1}, \sigma^2_{\xi_2}, \sigma_{\xi_1\xi_2}, \sigma_{\varepsilon_1}, \ldots, \sigma_{\varepsilon_4})$
- Estimation process iteratively revises $\boldsymbol{\theta}$ to minimize F_{ML} (equivalent, maximize lnL).

KU

Estimation: Make Σ as close to S as possible

$$\Sigma(\theta) = \begin{bmatrix} \sigma^2_{\xi_1}\lambda^2_{11} + \sigma^2_{\varepsilon_1} & \sigma^2_{\varepsilon_1}\lambda_{21}\lambda_{11} & \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{32} & \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{42} \\ \sigma^2_{\xi_1}\lambda_{11}\lambda_{21} & \sigma^2_{\varepsilon_1}\lambda^2_{21} + \sigma^2_{\varepsilon_2} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{32} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{42} \\ \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{32} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{32} & \sigma^2_{\xi_2}\lambda^2_{32} + \sigma^2_{\varepsilon_3} & \sigma^2_{\xi_2}\lambda_{32}\lambda_{42} \\ \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{42} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{42} & \sigma^2_{\xi_2}\lambda_{32}\lambda_{42} & \sigma^2_{\xi_2}\lambda^2_{42} + \sigma^2_{\varepsilon_4} \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} 21 & 4 & -1 & 5 \\ 4 & 9 & 2 & 1 \\ -1 & 2 & 8 & 3 \\ 5 & 1 & 4 & 6 \end{bmatrix}$$

- The key observation: Inside $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, one cannot "freely" adjust the separate cells. Changing σ_{ξ_1} simultaneously alters many cells.
- Optimizing will not generally match $\mathbf{S}$ exactly.

Maximum Likelihood

$$F_{\mathbf{ML}} = log|\boldsymbol{\Sigma}(\boldsymbol{\theta})| - log|\mathbf{S}| + tr(\mathbf{S}\{\boldsymbol{\Sigma}(\boldsymbol{\theta})\}^{-1})) - (p+q)$$

- Calculations are iterative, estimating F_{ML} and adapting $\hat{\theta}$. Various optimizer algorithms have been tested (EM, Newton, Quasi-Newton, Annealing, etc)
- Because the N rows of data have been compressed into a $(p+q) \times (p+q)$ matrix $\mathbf{S}$, the magnitude of the calculation is, more or less, *independent of the sample size*! It only depends on the number of parameters in $\boldsymbol{\theta}$.
- Covariance structure interpretation:$\mathbf{S}$ has a Wishart Distribution (SEM sometimes calls it MWL, Maximum Wishart likelihood)
    - $\hat{\boldsymbol{\theta}}$ from the last (or converged) iteration are the ML estimates (e.g., factor loadings and correlations)

Maximum Likelihood ...

Reminder ML estimate properties:

- Small sample properties: unknown
- Large sample properties
  - Consistency: as the sample size increases ($N \to \infty$), the parameter estimate approaches the true value ($\hat{\theta} \to \theta$)
  - Asymptotic Normality: as $N \to \infty$, the estimates are normally distributed
  - Asymptotic Efficiency: $N \to \infty$, no other estimator has smaller variance

What's the Saturated Model?

$$\Sigma(saturated) = \begin{bmatrix} a & b & c & d \\ b & e & f & g \\ c & f & h & j \\ d & g & j & l \end{bmatrix}$$

The values in $\mathbf{\Sigma}$ are unrestricted, so $\mathbf{\Sigma}$ can exactly match S. In that case, the Fit function reduces:

$$F_{\mathbf{satruated}} = log|\mathbf{\Sigma}(\boldsymbol{\theta})| - log|\mathbf{S}| + tr(\mathbf{S}\{\mathbf{\Sigma}(\boldsymbol{\theta})\}^{-1})) - (p+q)$$

$$= 0 + (p+q) - (p+q) = 0$$

- The Likelihood value at $\hat{\theta}$ is $\frac{1}{2}(N-1) \cdot F_{ML}$
- The likelihood-ratio test comparing the fitted and saturated model is very frequently considered a "goodness of fit" index.

$$-2ln\left(\frac{L_{fitted}}{L_{saturated}}\right)$$

which, as $N \to \infty$, converges in distribution to χ^2_{p+q}

SEM: Adapt Covariance Analysis to Missing Data Patterns

- The earliest discussion of this in SEM (possibly still most understandable) is Arbuckle (1996, p. 248), who attributes the idea to Finkbeiner (1979).
- The log likelihood for a case i is re-written, with case-specific values of the expected value $\boldsymbol{\mu}_i$ and variance $\boldsymbol{\Sigma}_i$.

$$logL_i = K_i - \frac{1}{2}log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_i)$$

- Obviously, this is not directly meaningful for cases with missing scores.
  - it is impossible to calculate a variance matrix estimate for an individual case, but
  - we can calculate covariance estimates for data sub-groups that have complete information on some variables.

A little example

- In a 3-variable model, suppose person i has complete data:
  $$\boldsymbol{x}_i = \begin{bmatrix} 6 \\ 4 \\ 7 \end{bmatrix}$$
- We assumed those variables are MVN with these parameters
  $$\boldsymbol{\mu_i} = \begin{bmatrix} \mu_{i1} \\ \mu_{i2} \\ \mu_{i3} \end{bmatrix}, \sum = \begin{bmatrix} \sigma_{11} & \sigma_{21} & \sigma_{31} \\ \sigma_{21} & \sigma_{22} & \sigma_{32} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix}$$

FIML Estimation with Missing Data

- Their individual log-likelihood is based on the full set of data:

$$log(L_i) = K_i - \frac{1}{2}log(|\begin{bmatrix}\sigma_{11} & \sigma_{21} & \sigma_{31}\\ \sigma_{21} & \sigma_{22} & \sigma_{32}\\ \sigma_{31} & \sigma_{32} & \sigma_{33}\end{bmatrix}^{-1}|)$$

$$-\frac{1}{2}\left(\left(\begin{bmatrix}6\\4\\7\end{bmatrix} - \begin{bmatrix}\mu_1\\ \mu_2\\ \mu_3\end{bmatrix}\right)^T \begin{bmatrix}\sigma_{11} & \sigma_{21} & \sigma_{13}\\ \sigma_{21} & \sigma_{22} & \sigma_{32}\\ \sigma_{31} & \sigma_{32} & \sigma_{33}\end{bmatrix}^{-1} \left(\begin{bmatrix}6\\4\\7\end{bmatrix} - \begin{bmatrix}\mu_1\\ \mu_2\\ \mu_3\end{bmatrix}\right)\right)$$

KU

FIML Estimation with Missing Data

- If data are missing, say variable 2, the dimension of the multivariate normal density function is reduced and missing dimensions are "skipped"

$$log(L_i) = K_i - \frac{1}{2}log(|\begin{bmatrix}\sigma_{11} & \sigma_{13}\\ \sigma_{13} & \sigma_{33}\end{bmatrix}^{-1}|)$$

$$-\frac{1}{2}\left(\left(\begin{bmatrix}6\\7\end{bmatrix} - \begin{bmatrix}\mu_1\\ \mu_3\end{bmatrix}\right)^T \begin{bmatrix}\sigma_{11} & \sigma_{13}\\ \sigma_{31} & \sigma_{33}\end{bmatrix}^{-1} \left(\begin{bmatrix}6\\7\end{bmatrix} - \begin{bmatrix}\mu_1\\ \mu_3\end{bmatrix}\right)\right)$$

FIML Estimation with Missing Data

- We don't ever need to make N fully unique covariance structures, because we group subjects according to a missing data pattern.
  - variance $\mathbf{\Sigma}_i$, and importantly $\mathbf{\Sigma_i}^{-1}$ will be the same for cases i in a given missing data pattern.
- We use cases within a data pattern to create the empirical covariance matrix that is employed among those cases.
  The mis-fit function F_{ML} is now divided into a sum of mis-fit functions, one for each sub-group of missing information.
- Nevertheless, it is safe to think of maximizing a likelihood with N independent pieces

$$logL(\boldsymbol{\mu}, \mathbf{\Sigma}) = \sum_{i=1}^{N} logL_i$$

FIML Works: Another Jorgensen & Lang Slide

Example: Compare MI to FIML

- The MCAR simulation randomly removed Job Performance data (unrelated to anything)
- MAR simulation deleted JP data for low IQ (reason for missingness included in model)
- MNAR deleted JP data for low JP scores (reason for JP missingness not recoverable because it is due to JP itself)

Parameter	Population value	Multiple imputation	Maximum likelihood
	MCAR simulation		
μ_{IQ}	100.00	99.98	100.02
μ_{JP}	12.00	11.99	11.99
σ^2_{IQ}	169.00	169.34	168.25
σ^2_{JP}	9.00	9.08	8.96
$\sigma_{IQ,JP}$	19.50	19.51	19.48
	MAR simulation		
μ_{IQ}	100.00	100.00	100.01
μ_{JP}	12.00	12.00	12.01
σ^2_{IQ}	169.00	168.46	168.50
σ^2_{JP}	9.00	9.23	8.96
$\sigma_{IQ,JP}$	19.50	19.43	19.15
	MNAR simulation		
μ_{IQ}	100.00	100.02	100.00
μ_{JP}	12.00	14.13	14.12
σ^2_{IQ}	169.00	170.37	169.11
σ^2_{JP}	9.00	3.42	3.33
$\sigma_{IQ,JP}$	19.50	8.51	8.55

KU

Summary table from Jorgensen & Lang notes

Example: Compare FIML to LD

- The same simulation, but comparing FIML (which yield the same results as MI) to Listwise Deletion (LD)
- Whereas FIML/MI yield bias for parameters affected by missingness (JP), LD yields bias for all estimates
- LD only unbiased for MCAR (most missing data are not MCAR)

Parameter	Population value	Maximum likelihood	Listwise deletion
	MCAR simulation		
μ_{IQ}	100.00	100.02	100.00
μ_{JP}	12.00	11.99	11.99
σ^2_{IQ}	169.00	168.25	166.94
σ^2_{JP}	9.00	8.96	8.94
$\sigma_{IQ,JP}$	19.50	19.48	19.31
	MAR simulation		
μ_{IQ}	100.00	100.01	110.35
μ_{JP}	12.00	12.01	13.18
σ^2_{IQ}	169.00	168.50	61.37
σ^2_{JP}	9.00	8.96	7.49
$\sigma_{IQ,JP}$	19.50	19.15	6.99
	MNAR simulation		
μ_{IQ}	100.00	100.00	105.19
μ_{JP}	12.00	14.12	14.38
σ^2_{IQ}	169.00	169.11	141.41
σ^2_{JP}	9.00	3.33	3.25
$\sigma_{IQ,JP}$	19.50	8.55	7.14

KU

Standard Errors

- The standard errors come from the **information matrix**, which comes from the second derivative (magnitude of the curvature) of the log-likelihood function.
- Steeper functions with sharp peaks give smaller standard errors, have less MLE uncertainty

Find the unique elements of Σ and S

- S and Σ are symmetric (same above and below the diagonal):

$$\Sigma = \begin{bmatrix} \sigma^2_{\xi_1}\lambda^2_{11}+\sigma^2_{\varepsilon_1} & \sigma^2_{\varepsilon_1}\lambda_{21}\lambda_{11} & \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{32} & \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{42} \\ \sigma^2_{\xi_1}\lambda_{11}\lambda_{21} & \sigma^2_{\varepsilon_1}\lambda^2_{21}+\sigma^2_{\varepsilon_2} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{32} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{42} \\ \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{32} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{32} & \sigma^2_{\xi_2}\lambda^2_{32}+\sigma^2_{\varepsilon_3} & \sigma^2_{\xi_2}\lambda_{32}\lambda_{42} \\ \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{42} & \sigma_{\xi_1\xi_2}\lambda_{21}\lambda_{42} & \sigma^2_{\xi_2}\lambda_{32}\lambda_{42} & \sigma^2_{\xi_2}\lambda^2_{42}+\sigma^2_{\varepsilon_4} \end{bmatrix}$$

$$S = \begin{bmatrix} 21 & 4 & -1 & 5 \\ 4 & 9 & 2 & 1 \\ -1 & 2 & 8 & 3 \\ 5 & 1 & 4 & 6 \end{bmatrix}$$

- Pick out the unique elements and stack them in vectors

$$\sigma = [\ \sigma^2_{\xi_1}\lambda^2_{11}+\sigma^2_{\varepsilon_1} \quad \sigma^2_{\xi_1}\lambda_{11}\lambda_{21} \quad \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{32} \quad \sigma_{\xi_1\xi_2}\lambda_{11}\lambda_{42} \quad \ldots \quad \sigma^2_{\xi_2}\lambda^2_{42}+\sigma^2_{\varepsilon_4}\ ]^T$$

$$s = (\ 21, \quad 4, \quad -1, \quad 5, \quad 9, \quad 2, \quad 1, \quad 8, \quad 4, \quad 6\ )^T$$

Find the unique elements of Σ and S ...

- One can view the estimation process as a minimization of the weighted sum of squares of σ and s

$$(s-\sigma)^T W (s-\sigma)$$

- This makes the estimation into a problem of Generalized Least Squares (Brown, 1974, 1984)
- As sample size goes to infinity, this GLS based calculation is equivalent to the Wishart-based ML described above.
- Getting the "right" weight matrix is part of the analysis. Some procedures used with incomplete data call for a full weight symmetric weight matrix
- Some call only for the diagonal elements (hence "diagonally weighted least squares").

Outline

Numeric Data

- With MVN numeric data,
  - The widespread preference is for FIML when possible. The summary statistics work!
  - Allison (2003 ,2012) argues forcefully in favor of FIML, even when the data departs from MVN to a significant degree.

Categorical Data

- FIML not widely available
- Create the polychoric correlation matrix with pairwise complete data, then treat like ordinary SEM calculation.
- "Casewise" maximum likelihood analysis for categorical data not generally available

What to do with categorical data

- There is a very strong temptation to "pretend" the data is numeric and proceed with FIML analysis. *Very Strong temptation*.
- Putting that aside, these are the options
  1. listwise deletion of missings
  2. Improvise a pairwise-complete correlation matrix
  3. Multiple Imputation followed by SEM analysis of each data set
  4. Individual-level "casewise" maximum likelihood analysis. View the discrete outcomes from an IRT point of view.
  5. Go Bayesian! The IRT movement is powerful and it is pulling multivariate discrete data analysis along with it.

What to do with categorical data?

	Pro	Con
listwise	fast, convenient	inaccurate if not MCAR
pairwise	convenient	widely criticized
MI	Rubin's theorems, software improving	Unfamiliar, time consuming, limited post-hoc analysis tools
casewise ML	Theoretically desirable	Slow! Unavailable for larger models
Bayesian	Theoretically desirable	Unfamiliar, hard to learn, Slow!

What does Mplus do?

If missing cases are found, Mplus now defaults to

- Numeric Data: FIML.
  - Can explicitly ask for listwise deletion
- Categorical data:
  - WLSMV estimator defaults: Pairwise-Complete Covariance Matrix (since Version 6)
  - MLR estimator: casewise Individual Level maximum likelihood (IRT-based)
  - Can explicitly ask for listwise deletion

What does Lavaan do?

- Numeric data: same as Mplus, listwise deletion and FIML estimator available
- Categorical data:
  - defaults to listwise deletion
  - pairwise complete covariance matrix available
  - No individual-level "casewise" ML (yet)

Smart money on Bayesian for Long Term

- Currently, the implementation is problematic, but
- Long term answer is likely a Bayesian approach that treats the missing values as additional parameters to be estimated in same process that estimates the rest of the model.

References

Aldrich, J. (1997). R. a. fisher and the making of maximum likelihood 1912-1922. *Statistical Science*, *12*, 162-176. doi:10.1214/ss/1030037906

Allison, P. D. (2003). Missing data techniques for structural equation modeling. *Journal of Abnormal Psychology*, *112*, 54. doi:10.1037/0021-843x.112.4.545

Arbuckle, J. L. (1996). Advanced structural equation modeling: Issues and techniques. In G. A. Marcoulides & R. E. Schumacker (Eds.), (p. 243-277). Mahwah, NJ: Lawrence Erlbaum. doi:10.4324/9781315827414

Bollen, K. A. (1989). *Structural equations with latent variables*. New York: Wiley. (OCLC: 18834634)

References ...

Enders, C. K., & Bandalos, D. L. (2001). The relative performance of full information maximum likelihood estimation for missing data in structural equation modeling. *Structural Equation Modeling*, *8*, 430-457. doi:10.1207/s15328007sem0803_5

Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society A*, *2222*, 309-368. doi:10.1007/978-1-4612-0919-5_2

Graham, J. W., & Coffman, D. L. (2012). Structural equation modeling with missing data. In R. H. Hoyle (Ed.), *Handbook of structural equation modeling* (p. 277-295). New York: Guilford.

Session

```
sessionInfo()
```

```
R version 3.6.0 (2019-04-26)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 19.04

Matrix products: default
BLAS:   /usr/lib/x86_64-linux-gnu/atlas/libblas.so.3.10.3
LAPACK: /usr/lib/x86_64-linux-gnu/atlas/liblapack.so.3.10.3

locale:
 [1] LC_CTYPE=en_US.UTF-8          LC_NUMERIC=C
      LC_TIME=en_US.UTF-8
 [4] LC_COLLATE=en_US.UTF-8        LC_MONETARY=en_US.UTF-8
      LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8          LC_NAME=C                  LC_ADDRESS=C
[10] LC_TELEPHONE=C                LC_MEASUREMENT=en_US.UTF-8
     LC_IDENTIFICATION=C

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

loaded via a namespace (and not attached):
[1] compiler_3.6.0 tools_3.6.0
```