Introductory Structural Equation Modeling

Ed Merkle
Department of Psychological Sciences
University of Missouri

CRMDA Stats Camp 2019

Contents

1 Introduction to SEM **5**
1.1 From regression to structural equation modeling 5
1.2 The model-implied covariance matrix (the essence of SEM) 19
1.3 Matrix representation in a CFA model 23
1.4 The implied covariance matrix for the full SEM model 34
1.5 Model parameters and model matrices 38
1.6 Model estimation 44
1.7 Model evaluation 47
1.8 Model respecification/comparison 49
1.9 Bayesian SEM 53
1.10 Reporting results 57
1.11 Further reading 58

2 Advanced topics **60**
2.1 Meanstructures 60
2.2 Multiple groups 63
2.3 Measurement invariance 65

2.4 Missing data 71
2.5 Nonnormal data and alternative estimators 74
2.6 Handling categorical endogenous variables 84
2.7 Growth models 99

3 Appendices **104**
3.1 Rules about variances and covariances 104
3.2 Estimator ML, MLM and MLR: robust standard errors and scaled test statistics 111

acknowledgment

These slides are based on similar slides by Yves Rosseel.

1 Introduction to SEM

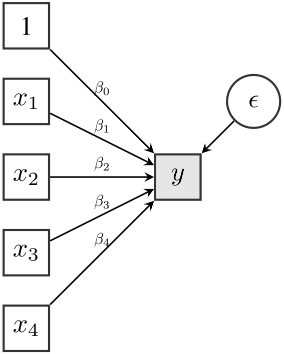
1.1 From regression to structural equation modeling

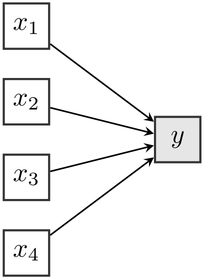
overview

- Regression: Predict one observed variable from other observed variables. (correlation)
- Factor analysis: Do sets of observed variables measure the same unobserved trait? (latent variable)
- SEM: Predict one latent variable from other latent variables.

- All of these can be represented visually (*path diagrams*), which can be helpful for understanding.

univariate linear regression





$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \epsilon_i \quad (i = 1, 2, \ldots, n)$$

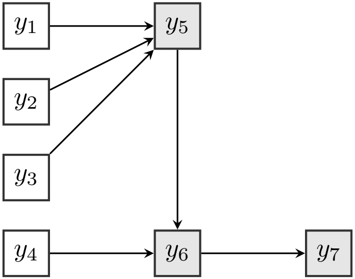
(squares are observed; circles are latent/unobserved)

extension: multivariate regression

$$\begin{tikzcd}
\boxed{x_1} \arrow[dr] \arrow[ddr] & \\
\boxed{x_2} \arrow[r] \arrow[dr] & \boxed{y_1} \\
\boxed{x_3} \arrow[ur] \arrow[r] & \boxed{y_2} \\
\boxed{x_4} \arrow[uur] \arrow[ur] &
\end{tikzcd}$$

further extension: path analysis

- testing models of *causal* relationships among observed variables
- all variables are observed (manifest)
- system of regression equations

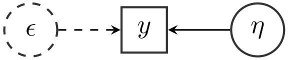


exogenous/endogenous

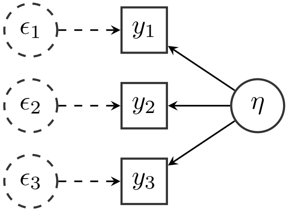
- all variables in the model can be classified as *exogenous* or *endogenous*
  - exogenous: the variable arises outside of the model (latent variable receives no arrows)
  - endogenous: the variable is explained within the model
  - on the previous slide, y_1 to y_4 are exogenous and the remaining are endogenous
- during SEM estimation, exogenous variables are often handled differently from endogenous variables

latent variables

- in the social sciences and beyond, observed variables have measurement error
- single indicator measurement model



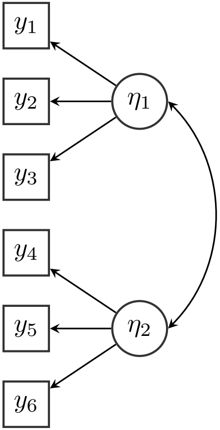
- multiple indicator measurement model



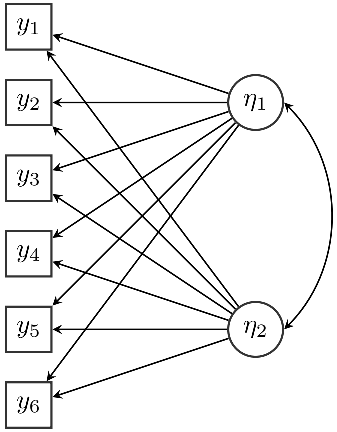
factor analysis

- factor analysis: formally representing the relationship between one or more latent variables and their (observed) indicators (say, 3 indicators per latent)
  - confirmatory factor analysis: studying whether observed variables correspond to a specific number of latent variables in a specific arrangement (more common, in traditional applications)
  - exploratory factor analysis: determining the number of latent variables, as well as which observed variables go with which latent variables (less common, though interest in "big data" applications)

confirmatory factor analysis (CFA)



exploratory factor analysis (EFA)

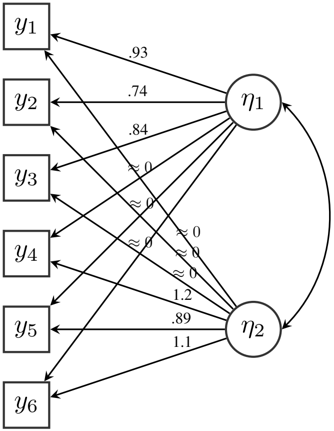


exploratory factor analysis (EFA)

- when we estimate an EFA model, there is not a unique set of best estimates (*rotational indeterminacy*)
- this means that we cannot uniquely tell which observed variables go with which latent variables
- we first estimate a "just-identified" model (details to come) and then apply a *rotation* after the estimation. the rotation chooses the set of estimates that is simplest to interpret.
  - *Orthogonal* rotation: Latent variables (factors) are uncorrelated with one another.
  - *Oblique* rotation: Latent variables are allowed to be correlated with one another (preferred, as orthogonal is a special case)

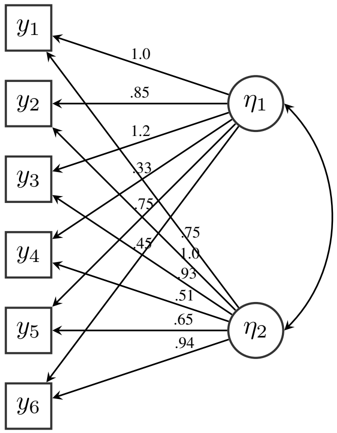
exploratory factor analysis (EFA)

simple estimates:



exploratory factor analysis (EFA)

complicated estimates:

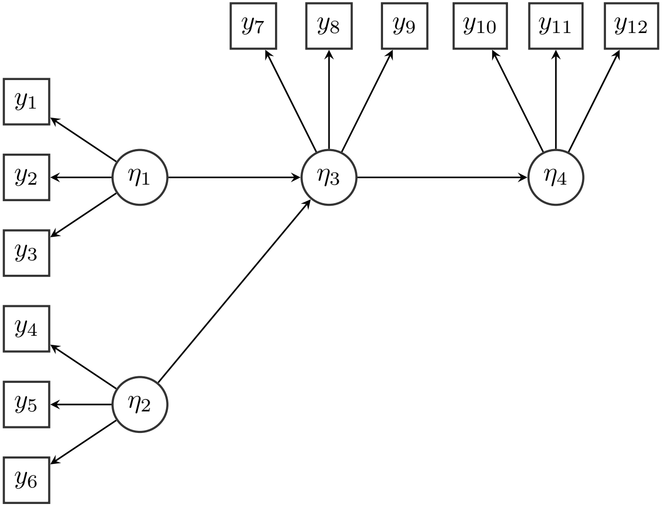


exploratory factor analysis (EFA)

- there are many ways to define “simplest,” leading to many rotation criteria (technically, an infinite number)
- many researchers have a personal preference for a rotation criterion, without clear consensus
- current research on this topic involves “regularization” methods
- strategy: try the software default and a few others, examining similarities/differences between solutions
- see the `rotations()` function from R package GPArotation (used in other packages, like `psych` and `mirt`)

structural equation modeling (SEM)

- path analysis with latent variables



1.2 The model-implied covariance matrix (the essence of SEM)

- the goal of SEM is to test an a priori specified theory (which often can be depicted as a path diagram)
- we may have several alternative models, each one with its own path diagram
- each path diagram can be converted to a SEM:
  - measurement model (relate latent variables to observed indicators)
  - structural model (regressions among latent/observed variables)
- each diagram has ‘model-based’ implications
  - for the model-implied covariance matrix: $\hat{\boldsymbol{\Sigma}}$
  - for the model-implied mean vector: $\hat{\boldsymbol{\mu}}$
  - $\ldots$
- different diagrams lead to (potentially) different implications; some implications may provide a bad description of your data

example model-implied covariance matrix (1)

- suppose we have three observed variables, y_1, y_2 and y_3; to explain why they are correlated, we may postulate the following model:

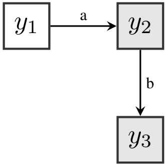
$$\begin{tikzcd}
\boxed{y_1} \arrow[r, "a"] \arrow[dr, "b"] & \boxed{y_2} \\
& \boxed{y_3}
\end{tikzcd}$$

- if we use the following values for the model parameters: $a = 3$ and $b = 5$, $\sigma^2(y_1) = 10$, $\sigma^2(\epsilon_2) = 20$ and $\sigma^2(\epsilon_3) = 30$, then we find (see Appendix):

$$\hat{\boldsymbol{\Sigma}} = \begin{bmatrix} 10 & & \\ 30 & 110 & \\ 50 & 150 & 280 \end{bmatrix}$$

example model-implied covariance matrix (2)

- but if we change the path diagram (and keep the parameter values fixed), the model-implied covariance matrix will also change:



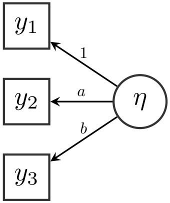
we find

$$\hat{\mathbf{\Sigma}} = \begin{bmatrix} 10 & & \\ 30 & 110 & \\ 150 & 550 & 2780 \end{bmatrix}$$

- two models are said to be *equivalent*, if they imply the same covariance matrix (different parameter values leading to the same predictions)

example model-implied covariance matrix (3)

- we can alternatively say that the correlations among the three observed variables are explained by a common 'factor':



we find (using $\sigma^2(\epsilon_1) = 10$, $\sigma^2(\epsilon_2) = 20$, $\sigma^2(\epsilon_3) = 30$, $\sigma^2(\eta) = 1$):

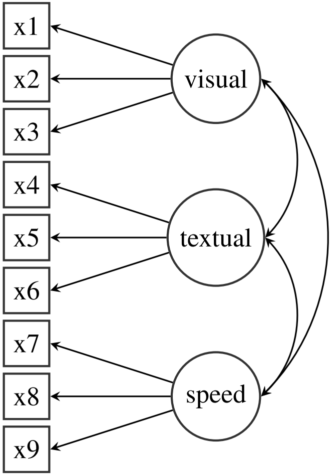
$$\hat{\mathbf{\Sigma}} = \begin{bmatrix} 11 & & \\ 4 & 36 & \\ 5 & 20 & 55 \end{bmatrix}$$

1.3 Matrix representation in a CFA model

classic CFA example

- well-known dataset based on Holzinger & Swineford (1939) data
- also analyzed by Jöreskog (1969), included in *lavaan*
- 9 observed 'indicators' measuring three latent factors:
  - a 'visual' factor measured by x1, x2 and x3
  - a 'textual' factor measured by x4, x5 and x6
  - a 'speed' factor measured by x7, x8 and x9
- N=301
- we assume the three factors are correlated

diagram of the model



data

```
      x1   x2    x3    x4   x5    x6    x7   x8    x9
1  3.333 7.75 0.375 2.333 5.75 1.286 3.391 5.75 6.361
2  5.333 5.25 2.125 1.667 3.00 1.286 3.783 6.25 7.917
3  4.500 5.25 1.875 1.000 1.75 0.429 3.261 3.90 4.417
4  5.333 7.75 3.000 2.667 4.50 2.429 3.000 5.30 4.861
5  4.833 4.75 0.875 2.667 4.00 2.571 3.696 6.30 5.917
6  5.333 5.00 2.250 1.000 3.00 0.857 4.348 6.65 7.500
7  2.833 6.00 1.000 3.333 6.00 2.857 4.696 6.20 4.861
8  5.667 6.25 1.875 3.667 4.25 1.286 3.391 5.15 3.667
9  4.500 5.75 1.500 2.667 5.75 2.714 4.522 4.65 7.361
10 3.500 5.25 0.750 2.667 5.00 2.571 4.130 4.55 4.361
11 3.667 5.75 2.000 2.000 3.50 1.571 3.739 5.70 4.306
12 5.833 6.00 2.875 2.667 4.50 2.714 3.696 5.15 4.139
13 5.667 4.50 4.125 2.667 4.00 2.286 5.870 5.20 5.861
14 6.000 5.50 1.750 4.667 4.00 1.571 5.130 4.70 4.444
15 5.833 5.75 3.625 5.000 5.50 3.000 4.000 4.35 5.861
16 4.667 4.75 2.375 2.667 4.25 0.714 4.087 3.80 5.139
...
301 4.333 6.00 3.375 3.667 5.75 3.143 4.087 6.95 5.167
```

- data are complete (and rounded above)
- under normality, the data can be summarized by the covariance matrix ($\mathbf{S}$) and the mean vector ($\mathbf{m}$)

observed covariance matrix: S

- p is the number of observed variables: $p = 9$
- observed covariance matrix (divided by N-1, vs N):

```
      x1    x2   x3   x4   x5   x6    x7   x8   x9
x1 1.36
x2 0.41  1.38
x3 0.58  0.45 1.28
x4 0.51  0.21 0.21 1.35
x5 0.44  0.21 0.11 1.10 1.66
x6 0.46  0.25 0.24 0.90 1.01 1.20
x7 0.09 -0.10 0.09 0.22 0.14 0.14  1.18
x8 0.26  0.11 0.21 0.13 0.18 0.17  0.54 1.02
x9 0.46  0.24 0.37 0.24 0.30 0.24  0.37 0.46 1.02
```

- we want to 'explain' the observed correlations/covariances via three latent variables (factors) and a corresponding factor structure
- we will 'rewrite' the $p(p+1)/2 = 45$ elements in the covariance matrix as a function of a smaller number of 'free parameters' in the CFA model, summarized by a number of (typically sparse) matrices

matrix representation

- The path diagrams can be formally written out via matrices, with multiple sets of matrices (*representations*) available. These representations are different ways of writing the same model.
- Historically, the representations have corresponded to different pieces of software. Popular representations include “LISREL” and “RAM.” We will focus on LISREL here.

the standard CFA model: matrix representation

- the LISREL representation uses three matrices for CFA
- the LAMBDA matrix contains the ‘factor structure’:

$$\mathbf{\Lambda} = \begin{bmatrix} x & 0 & 0 \\ x & 0 & 0 \\ x & 0 & 0 \\ 0 & x & 0 \\ 0 & x & 0 \\ 0 & x & 0 \\ 0 & 0 & x \\ 0 & 0 & x \\ 0 & 0 & x \end{bmatrix}$$

- the variances/covariances of the latent variables are summarized in the PSI matrix:

$$\mathbf{\Psi} = \begin{bmatrix} x & & \\ x & x & \\ x & x & x \end{bmatrix}$$

- what we cannot explain by the set of common factors (the ‘residual variance’ of the model) is written in the (typically diagonal) matrix THETA:

$$\mathbf{\Theta} = \begin{bmatrix} x & & & & & & & & \\ & x & & & & & & & \\ & & x & & & & & & \\ & & & x & & & & & \\ & & & & x & & & & \\ & & & & & x & & & \\ & & & & & & x & & \\ & & & & & & & x & \\ & & & & & & & & x \end{bmatrix}$$

- note that we have only 24 parameters (of which 21 are estimable)

the standard CFA model: the model implied covariance matrix

- in the standard CFA model, the ‘implied’ covariance matrix is:

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Psi}\boldsymbol{\Lambda}' + \boldsymbol{\Theta}$$

- all parameters are included in three model matrices (parameters are the parts of the model we can vary so that our model predictions better match the data)
- simple matrix multiplication (and addition) gives us the model implied covariance matrix
- for identification purposes, some parameters need to be fixed to a constant (typically to 1). otherwise, different sets of parameter values will yield the same predictions.

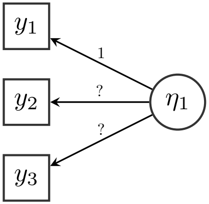
setting the metric of the latent variables: UVI or ULI

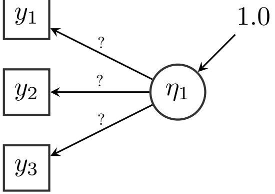
- to identify the model, we need to set the scale/metric of the latent variable
- example: say that the latent variable is “verbal ability.” What is the mean and variance of this unobserved variable?
  - the mean and variance require a unit of measurement for verbal ability. Because no units exist here (like fahrenheit or celsius), we must define our own units by setting the mean and variance.
  - we usually set the mean to 0, with the variance being set in one of two ways

setting the metric of the latent variables: UVI or ULI

1. *Unit Loading Identification* (ULI):
   the factor loading of one (often the first) of the indicators is fixed to 1.0; this indicator is called the *reference* indicator

2. *Unit Variance Identification* (UVI):
   the variance of the factor is fixed to 1.0





- in many models, it does not matter
- in multigroup SEM analysis, we usually use ULI

estimation

- estimation problem: choose the free parameters, so that the estimated implied covariance matrix ($\hat{\boldsymbol{\Sigma}}$) is 'as close as possible' to the observed covariance matrix $\mathbf{S}$
  - maximum likelihood estimation (ML)
  - generalized (weighted) least-squares estimation (GLS, WLS)
  - Bayesian approaches
- this is where we need computers: the models have lots of parameters, and we cannot hope to obtain good parameter values by hand or by intuition

1.4 The implied covariance matrix for the full SEM model

- for general SEM (beyond factor analysis), the LISREL representation requires an additional matrix ($\mathbf{B}$):

  $$\mathbf{\Sigma} = \mathbf{\Lambda}(\mathbf{I} - \mathbf{B})^{-1}\mathbf{\Psi}(\mathbf{I} - \mathbf{B})'^{-1}\mathbf{\Lambda}' + \mathbf{\Theta}$$

  where $\mathbf{B}$ summarizes the regressions among the latent variables

- we need this extended model for
  - second-order CFA
  - MIMIC models
  - other models with directional relationships (single-headed arrows) between latent variables

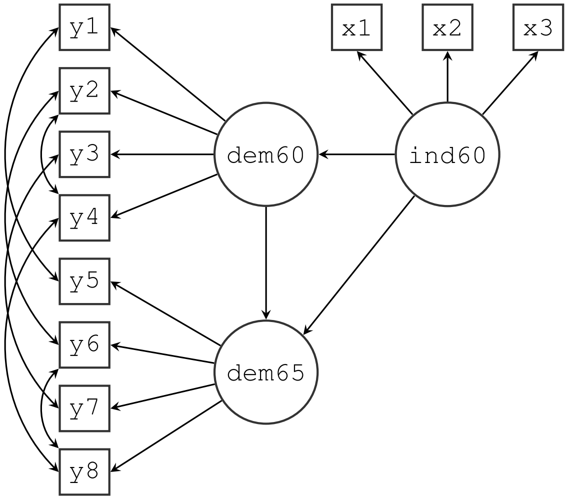
- in LISREL parlance, this is the ‘all-y’ model (other model variants specially handle exogenous and/or categorical variables)

SEM example: Political Democracy

- Industrialization and Political Democracy dataset (N=75)
- This dataset is used throughout Bollen's 1989 book (see pages 12, 17, 36 in chapter 2, pages 228 and following in chapter 7, pages 321 and following in chapter 8).
- The dataset contains various measures of political democracy and industrialization in developing countries:

```
y1: Expert ratings of the freedom of the press in 1960
y2: The freedom of political opposition in 1960
y3: The fairness of elections in 1960
y4: The effectiveness of the elected legislature in 1960
y5: Expert ratings of the freedom of the press in 1965
y6: The freedom of political opposition in 1965
y7: The fairness of elections in 1965
y8: The effectiveness of the elected legislature in 1965
x1: The gross national product (GNP) per capita in 1960
x2: The inanimate energy consumption per capita in 1960
x3: The percentage of the labor force in industry in 1960
```

model diagram



selection of the output

```
                   Estimate  Std.err  Z-value  P(>|z|)   Std.lv  Std.all
Latent variables:
  ind60 =~
    x1                1.000                               0.670    0.920
    x2                2.180    0.139   15.742    0.000    1.460    0.973
    x3                1.819    0.152   11.967    0.000    1.218    0.872
  dem60 =~
    y1                1.000                               2.223    0.850
    y2                1.257    0.182    6.889    0.000    2.794    0.717
    y3                1.058    0.151    6.987    0.000    2.351    0.722
    y4                1.265    0.145    8.722    0.000    2.812    0.846
  dem65 =~
    y5                1.000                               2.103    0.808
    y6                1.186    0.169    7.024    0.000    2.493    0.746
    y7                1.280    0.160    8.002    0.000    2.691    0.824
    y8                1.266    0.158    8.007    0.000    2.662    0.828

Regressions:
  dem60 ~
    ind60             1.483    0.399    3.715    0.000    0.447    0.447
  dem65 ~
    ind60             0.572    0.221    2.586    0.010    0.182    0.182
    dem60             0.837    0.098    8.514    0.000    0.885    0.885
...
```

1.5 Model parameters and model matrices

31 free model parameters

```
> coef(fit)

   ind60=~x2    ind60=~x3    dem60=~y2    dem60=~y3    dem60=~y4    dem65=~y6
       2.180        1.819        1.257        1.058        1.265        1.186
   dem65=~y7    dem65=~y8  dem60~ind60  dem65~ind60  dem65~dem60       y1~~y5
       1.280        1.266        1.483        0.572        0.837        0.624
      y2~~y4       y2~~y6       y3~~y7       y4~~y8       y6~~y8       x1~~x1
       1.313        2.153        0.795        0.348        1.356        0.082
      x2~~x2       x3~~x3       y1~~y1       y2~~y2       y3~~y3       y4~~y4
       0.120        0.467        1.891        7.373        5.067        3.148
      y5~~y5       y6~~y6       y7~~y7       y8~~y8 ind60~~ind60 dem60~~dem60
       2.351        4.954        3.431        3.254        0.448        3.956
dem65~~dem65
       0.172
```

model matrices: free parameters

```
> inspect(fit)

$lambda
   ind60 dem60 dem65
x1     0     0     0
x2     1     0     0
x3     2     0     0
y1     0     0     0
y2     0     3     0
y3     0     4     0
y4     0     5     0
y5     0     0     0
y6     0     0     6
y7     0     0     7
y8     0     0     8
```

```
$theta
   x1 x2 x3 y1 y2 y3 y4 y5 y6 y7 y8
x1 18
x2  0 19
x3  0  0 20
y1  0  0  0 21
y2  0  0  0  0 22
y3  0  0  0  0  0 23
y4  0  0  0  0 13  0 24
y5  0  0  0 12  0  0  0 25
y6  0  0  0  0 14  0  0  0 26
y7  0  0  0  0  0 15  0  0  0 27
y8  0  0  0  0  0  0 16  0 17  0 28

$psi
      ind60 dem60 dem65
ind60 29
dem60  0    30
dem65  0     0    31

$beta
      ind60 dem60 dem65
ind60     0     0     0
dem60     9     0     0
dem65    10    11     0
```

model matrices: estimated values

```
> inspect(fit, "est")

$lambda
   ind60 dem60 dem65
x1 1.000 0.000 0.000
x2 2.180 0.000 0.000
x3 1.819 0.000 0.000
y1 0.000 1.000 0.000
y2 0.000 1.257 0.000
y3 0.000 1.058 0.000
y4 0.000 1.265 0.000
y5 0.000 0.000 1.000
y6 0.000 0.000 1.186
y7 0.000 0.000 1.280
y8 0.000 0.000 1.266

$theta
   x1    x2    x3    y1    y2    y3    y4    y5    y6    y7    y8
x1 0.082
x2 0.000 0.120
x3 0.000 0.000 0.467
y1 0.000 0.000 0.000 1.891
y2 0.000 0.000 0.000 0.000 7.373
y3 0.000 0.000 0.000 0.000 0.000 5.067
y4 0.000 0.000 0.000 0.000 1.313 0.000 3.148
y5 0.000 0.000 0.000 0.624 0.000 0.000 0.000 2.351
```

```
y6 0.000 0.000 0.000 0.000 2.153 0.000 0.000 0.000 4.954
y7 0.000 0.000 0.000 0.000 0.000 0.795 0.000 0.000 0.000 3.431
y8 0.000 0.000 0.000 0.000 0.000 0.000 0.348 0.000 1.356 0.000 3.254

$psi
      ind60 dem60 dem65
ind60 0.448
dem60 0.000 3.956
dem65 0.000 0.000 0.172

$beta
      ind60 dem60 dem65
ind60 0.000 0.000     0
dem60 1.483 0.000     0
dem65 0.572 0.837     0
```

manually computing the model-implied covariance matrix (optional)

```
# make the model matrices available in R's workspace
attach(inspect(fit, "est"))

# compute (I - B)^(-1)
IB <- diag(nrow(beta)) - beta
IB.inv <- solve(IB)

# compute the model-implied model matrix (using formula around slide 35)
Sigma.hat <- lambda %*% IB.inv %*% psi %*% t(IB.inv) %*% t(lambda) + theta

# print the matrix
round(Sigma.hat, 3)

      x1    x2    x3    y1     y2     y3     y4    y5     y6     y7     y8
x1 0.530 0.978 0.815 0.665  0.836  0.703  0.841 0.814  0.965  1.041  1.030
x2 0.978 2.252 1.778 1.450  1.822  1.534  1.834 1.774  2.103  2.270  2.245
x3 0.815 1.778 1.950 1.209  1.520  1.279  1.530 1.479  1.754  1.893  1.873
y1 0.665 1.450 1.209 6.834  6.211  5.228  6.251 5.143  5.358  5.782  5.721
y2 0.836 1.822 1.520 6.211 15.179  6.570  9.169 5.679  8.887  7.267  7.190
y3 0.703 1.534 1.279 5.228  6.570 10.597  6.612 4.780  5.667  6.911  6.051
y4 0.841 1.834 1.530 6.251  9.169  6.612 11.054 5.716  6.777  7.313  7.584
y5 0.814 1.774 1.479 5.143  5.679  4.780  5.716 6.773  5.243  5.658  5.598
y6 0.965 2.103 1.754 5.358  8.887  5.667  6.777 5.243 11.171  6.709  7.994
y7 1.041 2.270 1.893 5.782  7.267  6.911  7.313 5.658  6.709 10.671  7.163
y8 1.030 2.245 1.873 5.721  7.190  6.051  7.584 5.598  7.994  7.163 10.341
```

1.6 Model estimation

- we seek those values for $\boldsymbol{\theta}$ that minimize the difference between what we observe in the data, $\mathbf{S}$, and what the model implies, $\boldsymbol{\Sigma}(\boldsymbol{\theta})$
- the final estimated values are denoted by $\hat{\boldsymbol{\theta}}$, and the estimated model-implied covariance matrix can be written as $\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}(\hat{\boldsymbol{\theta}})$
- there are many ways to quantify this 'difference', leading to different discrepancy measures

observed covariance matrix

```
   x1     x2     x3     x4     x5     x6     x7     x8     x9
x1  1.358
x2  0.407  1.382
x3  0.580  0.451  1.275
x4  0.505  0.209  0.208  1.351
x5  0.441  0.211  0.112  1.098  1.660
x6  0.455  0.248  0.244  0.896  1.015  1.196
x7  0.085 -0.097  0.088  0.220  0.143  0.144  1.183
x8  0.264  0.110  0.212  0.126  0.181  0.165  0.535  1.022
x9  0.458  0.244  0.374  0.243  0.295  0.236  0.373  0.457  1.015
```

model-implied covariance matrix

```
   x1    x2    x3    x4    x5    x6    x7    x8    x9
x1 1.358
x2 0.448 1.382
x3 0.590 0.327 1.275
x4 0.408 0.226 0.298 1.351
x5 0.454 0.252 0.331 1.090 1.660
x6 0.378 0.209 0.276 0.907 1.010 1.196
x7 0.262 0.145 0.191 0.173 0.193 0.161 1.183
x8 0.309 0.171 0.226 0.205 0.228 0.190 0.453 1.022
x9 0.284 0.157 0.207 0.188 0.209 0.174 0.415 0.490 1.015
```

Model estimation

- the most popular discrepancy measure is based on maximum likelihood and assumed multivariate normality:

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

- in practice, we replace $\boldsymbol{\Sigma}$ by $\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}(\hat{\boldsymbol{\theta}})$
- an alternative is (weighted) least squares, for some weight matrix $\mathbf{W}$ (which is a sort of standardization):

$$F_{WLS}(\boldsymbol{\theta}) = (\mathbf{s} - \boldsymbol{\sigma})'\mathbf{W}^{-1}(\mathbf{s} - \boldsymbol{\sigma})$$

  where $\mathbf{s}$ and $\boldsymbol{\sigma}$ are the unique elements of $\mathbf{S}$ and $\boldsymbol{\Sigma}$ respectively

1.7 Model evaluation

evaluation of global fit – chi-square test statistic

- the chi-square test statistic is the primary test of our model. this statistic can usually be obtained regardless of discrepancy measure.
- if the chi-square test statistic is NOT significant, we have a good fit of the model
- roughly, H0: the model fits the data; H1: the model does not fit the data
- non-significant chi-squares become rare as the sample size grows

evaluation of global fit – fit indices

- (some) rules of thumb: CFI/TLI > 0.95, RMSEA < 0.05, SRMR < 0.06
- there is a lot of controversy about the use (and misuse) of these fit indices
- a classic reference here is Hu & Bentler (1999)

admissibility of the results

- are the parameter values valid? Often a sign of a bad-fitting model
  - negative (residual) variances
  - correlations larger than one
- do the regression coefficients, factor loadings, covariances have the expected sign (positive or negative)?
- are all free parameters significant?
- are there any excessively large standard errors?

1.8 Model respecification/comparison

- if the fit of a model is not good, we can adapt (respecify) the model
  - change the number of factors
  - allow for indicators to be related to more than one factor (cross-loadings)
  - allow for correlated residual errors among the observed indicators
  - allow for correlated disturbances among the endogenous latent variables
  - remove problematic indicators . . .
- ideally, all changes should have a sound theoretical justification
- this is a place where people get into trouble: if we continually change the model until it fits, then results are less likely to generalize/replicate

comparison of models

- we can also compare two models to one another (compare alternate theories with one another), using statistics similar to those used for global fit evaluation
- comparisons are easier if one model is *nested* within the other
  - roughly, model A is nested within model B if it is a special case of model B (i.e., model B can predict everything that model A can predict and more)
  - model A is often called the "reduced" model, with model B being the "full" model
  - model B will always fit the observed data better than model A
  - nesting is often evident from path diagrams (but not always)

comparison of models – nested

- when models are nested, we can compare them via a chi-square difference test (also called “likelihood ratio test”)
  - H0: model A is as good as model B (therefore, prefer model A because it is simpler)
  - H1: model A is not as good as model B
  - the chi-square statistic is the difference between the models’ global chi-squares, with degrees of freedom being the corresponding difference in global degrees of freedom

comparison of models – non-nested

- when models are not nested, we typically compare them via AIC or BIC
- each model receives a criterion value, and the model with the lowest value is deemed "best"
- differences between fit indices are also sometimes considered, though there are no strict results about what constitutes a small/large difference
- Merkle, You, & Preacher (2016) implemented a chi-square test of non-nested models that works similarly to the nested test, available via package *nonnest2*

1.9 Bayesian SEM

- the model estimation methods considered so far all converge to a point; they seek the parameter values that best reproduce the observed covariance matrix (and possibly mean vector)
- alternatively, we can use Bayesian methods to estimate the *posterior distribution* of model parameters (not point estimates; distributional estimates)

Bayesian SEM

- idea of Bayesian methods
  - start with prior beliefs (distributions) about model parameters
  - observe data
  - update prior beliefs with the data, leading to posterior beliefs (distributions)
- advantages: easier to interpret intervals; fewer asymptotic arguments are needed; easier to estimate some complicated models

Bayesian SEM

- idea of Bayesian estimation
  - instead of finding the “best” estimates, we are looking for the posterior distribution of estimates
  - this is typically accomplished by drawing random samples from the posterior distribution via *Markov chain Monte Carlo*
  - once we draw thousands of posterior samples, we can summarize them by computing means, standard deviations, etc.

Bayesian SEM

- software for Bayesian SEM
  - flexible (but sometimes difficult) model specification: BUGS, JAGS, Stan
  - easier, less flexible model specification: Mplus (stand alone), blavaan (translate from lavaan to JAGS or Stan, then convert results back to lavaan)
- Bayesian SEM is relatively new: research opportunities; best practices are still under development

1.10 Reporting results

- see Boomsma (2000)
- report enough information so that the analysis can be replicated
  - always report the observed covariance matrix (or the correlation matrix + standard deviations)
  - better to include the full dataset (either as an electronic appendix or via a website)
  - code also useful, because it is difficult to write down every single modeling detail in a manuscript

1.11 Further reading

Kline, R. B. (2015). Principles and practice of structural equation modeling (Fourth Edition). New York: Guilford Press.

…The companion website supplies data, syntax, and output for the book's examples–now including files for Amos, EQS, LISREL, Mplus, Stata, and R (lavaan).

Brown, T. A. (2015). Confirmatory Factor Analysis for Applied Research (Second Edition) New York: Guilford Press.

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

Hancock, G. R., & Mueller, R. O. (Eds.). (2013). Structural equation modeling: A second course (Second Edition). Greenwich, CT: Information Age Publishing, Inc.

Boomsma, A. (2000). Reporting Analyses of Covariance Structures. *Structural Equation Modeling: A Multidisciplinary Journal, 7*, 461–483.

SEM in R, using lavaan

Beaujean, A. A. (2014). Latent variable modeling using R: A step-by-step guide. Routledge.

Finch, W.H., and French, B.F. (2015). Latent Variable Modeling with R. Routledge.

Little, T.D. (2013). Longitudinal Structural Equation Modeling (Methodology in the Social Sciences). The Guilford Press.

2 Advanced topics

2.1 Meanstructures

- traditionally, SEM has focused on covariance structure analysis, but we can also include the means (fitting only the covariance matrix, vs fitting both the covariance matrix and means)
- typical situations where we would include the means are:
  - multiple group analysis
  - growth curve models
  - analysis of non-normal data, and/or missing data

adding the means

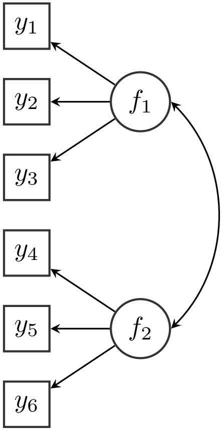
- we have more data: the p-dimensional mean vector (if we have raw data, we will always have this)
- we have more parameters:
  - means/intercepts for the observed variables
  - means/intercepts for the latent variables (often fixed to zero)

adding the means

- if no restrictions are imposed on the means, the fit will be identical to the non-meanstructure fit
- we add p datapoints (the mean vector)
- we add p free parameters (the intercepts of the observed variables)
- we fix the latent means to zero
- the number of degrees of freedom does not change

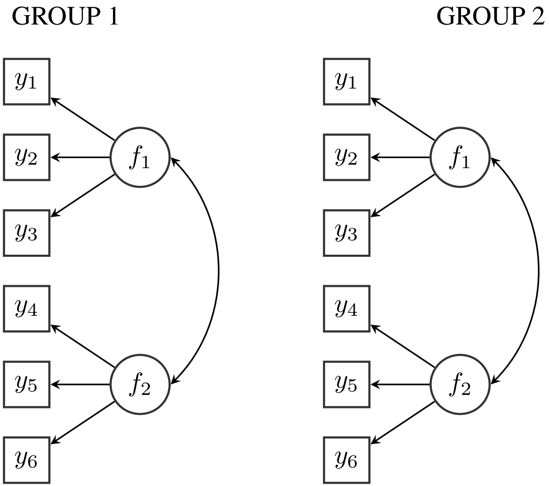
2.2 Multiple groups

single group analysis (CFA)



- factor means typically fixed to zero

multiple group analysis (CFA)



- we can potentially compare the means of the latent variables

2.3 Measurement invariance

- latent variable means get at interesting questions; we want to know whether the “true” trait of interest varies, as opposed to the observed means that are influenced by noisy measurement
- we can only compare the means of the latent variables across groups if ‘measurement invariance’ across groups has been established
- example: students complain that a fourth-grade test is biased, favoring girls over boys. In support of the students’ complaint, the average score for girls was 5 points higher than the average score for boys.

- we can consider two possibilities here:
  - the test is not biased; boys in the class just have lower ability (= mean of latent variable) than girls. (*measurement invariance is satisfied*)
  - the test is biased; a boy and girl with the same level of ability would systematically receive different scores on the test. (*bias/lack of invariance*)
- if we could get everyone to take a second test that we knew to be unbiased, we could begin to distinguish between these possibilities

- formal testing for measurement invariance involves a fixed sequence of model comparison tests
- one typical sequence involves 3 steps:
    1. Model 1: configural invariance. The same model/factor structure is imposed on all groups.
    2. Model 2: weak invariance. The factor loadings are constrained to be equal across groups.
    3. Model 3: strong invariance. The factor loadings and intercepts are constrained to be equal across groups, so we can compare the latent variable means.
- other sequences involve more steps; for example 'strict invariance' implies constraining the residual variances too
- model comparison tests: compare the current model with the previous one (either using a chi-square difference test or looking at other fit measures)
- *partial* measurement invariance: most but not all parameters are equivalent across groups. (i.e., two tests are invariant but one is not)

- references:
  - Vandenberg, R.J. and Lance, C.E. (2000). A Review and Synthesis of the Measurement Invariance Literature: Suggestions, Practices, and Recommendations for Organizational Research. *Organizational Research Methods, 3*, 4–69.
  - Cheung, G.W., and Rensvold, R.B. (2000). Evaluating goodness-of-fit indices for testing measurement invariance. *Structural Equation Modeling, 9*, 233-255.
  - Byrne, B.M., Shavelson, R.J and Muthén, B. (1989). Testing for the equivalence of factor covariance and mean structures: The issue of partial measurement invariance. *Psychological Bulletin, 105*, 456-466.

measurement invariance tests – all together

```
> library(semTools)
> measurementInvariance(HS.model, data = HolzingerSwineford1939,
                        group = "school", strict = FALSE)

Measurement invariance models:

Model 1 : fit.configural
Model 2 : fit.loadings
Model 3 : fit.intercepts
Model 4 : fit.means

Chi Square Difference Test

               Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
fit.configural 48 7484.4 7706.8 115.85
fit.loadings   54 7480.6 7680.8 124.04      8.192       6     0.2244
fit.intercepts 60 7508.6 7686.6 164.10     40.059       6  4.435e-07 ***
fit.means      63 7543.1 7710.0 204.61     40.502       3  8.338e-09 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
Fit measures:

                  cfi rmsea cfi.delta rmsea.delta
fit.configural 0.923 0.097        NA          NA
fit.loadings   0.921 0.093     0.002       0.004
fit.intercepts 0.882 0.107     0.038       0.015
fit.means      0.840 0.122     0.042       0.015
```

2.4 Missing data

missing data

- missing data are especially problematic in SEM because each person supplies us with many variables
- if a person is missing one variable, we don't want to throw out his/her other, observed variables
- Rubin defines three missing data mechanisms that provide guidance for how to handle the missing data

missing data mechanisms

- MCAR: missing completely at random
  - listwise deletion is ok (data are lost, but the estimates are still unbiased)
- MAR: missing at random
  - what caused the data to be missing does not depend upon the missing data itself, but may depend on the non-missing data
  - listwise deletion is NOT ok: estimates are biased
  - alternatives: full information ML (FIML), multiple imputation, . . .
- MNAR: not missing at random
  - in general, almost all standard statistical methods are no longer valid
  - we can only try to understand the missingness mechanism at hand, and take this into account when modeling the data

missing data examples

- examples of missingness mechanisms, using a high school reading test (Collins, Schafer, & Kam, 2001)
  - MCAR: some students miss the test due to drivers' education class. this is a "completely random" set of missing students, because "drivers' education class" is unrelated to reading ability.
  - MAR: special education students are more likely to be absent for the test, as compared to other students. data are MAR *if* we include "student status" (special education/not special education) as an additional variable in our analysis.
  - MNAR: students with lower reading ability skip the test day more than other students. if we have no other information about these absent students, then data are MNAR.
  - as in the above example, we can sometimes move from MNAR to MAR by including extra *auxiliary variables* thought to explain missingness (esp relevant to multiple imputation)

2.5 Nonnormal data and alternative estimators

what if the data are NOT normally distributed?

- in the real world, data may never be normally distributed
- two types:
  - categorical and/or limited-dependent outcomes: binary, ordinal, nominal, counts, censored (WLSMV, logit/probit)
  - continuous outcomes, not normally distributed: skewed, too flat/too peaked (kurtosis), . . .
- three strategies to deal with continuous non-normal data
  1. asymptotically distribution-free estimation
  2. ML estimation with ‘robust’ standard errors, and a ‘robust’ test statistic for model evaluation
  3. bootstrapping

robust method 1: asymptotically distribution-free (ADF) estimation

- the ADF estimator (Browne, 1984) makes no assumption of normality and is part of a larger family of estimators called weighted least squares (WLS) estimators:

  $$F_{WLS} = (\mathbf{s} - \hat{\boldsymbol{\sigma}})^\top \mathbf{W}^{-1} (\mathbf{s} - \hat{\boldsymbol{\sigma}})$$

  where $\mathbf{s}$ and $\hat{\boldsymbol{\sigma}}$ are vectors containing the non-duplicated elements in the sample ($\mathbf{S}$) and model-implied ($\hat{\boldsymbol{\Sigma}}$) covariance matrix respectively

- the weight matrix $\mathbf{W}$ utilized with the ADF estimator is the asymptotic covariance matrix: a matrix of the covariances of the observed sample variances and covariances

- unfortunately, empirical research has shown that the ADF method breaks down unless the sample size is huge (e.g., N > 5000)

- in lavaan:

```
fit <- cfa(HS.model, data = HolzingerSwineford1939,
           estimator = "WLS")
```

robust method 2: robust ML

1. **parameter estimates: vanilla ML**
   - if ML is used, the parameter estimates are still consistent (if the model is identified and correctly specified)
2. **'robust' standard errors**
   - if data are non-normal, the standard errors tend to be too small (as much as 25-50%)
   - 'robust' standard errors correct for non-normality (see Appendix)
3. **'robust' scaled (chi-square) test statistic**
   - if data are non-normal, the usual model (chi-square) test statistic tends to be too large
   - the **Satorra-Bentler scaled test statistic** rescales the value of the ML-based chi-square test statistic by an amount that reflects the degree of kurtosis (see Appendix)

robust ML in lavaan

- robust standard errors

```
fit <- cfa(HS.model, data = HolzingerSwineford1939,
           se = "robust")
```

- Satorra-Bentler scaled test statistic

```
fit <- cfa(HS.model, data = HolzingerSwineford1939,
           test = "Satorra-Bentler")
```

- robust standard errors + scaled test statistic

```
fit <- cfa(HS.model, data = HolzingerSwineford1939,
           se = "robust", test = "Satorra-Bentler")
```

- estimator MLM = robust standard errors + scaled test statistic

```
fit <- cfa(HS.model, data = HolzingerSwineford1939,
           estimator = "MLM")
```

- alternative: estimator MLR (also for missing data)

```
fit <- cfa(HS.model, data = HolzingerSwineford1939,
           estimator = "MLR", missing = "ml")
```

output: robust standard errors and scaled test statistic

```
> fit <- cfa(HS.model,
             data = HolzingerSwineford1939,
             estimator = "MLM")

> summary(fit, fit.measures=TRUE)
lavaan (0.6-1) converged normally after  35 iterations

  Number of observations                           301

  Estimator                                         ML       Robust
  Minimum Function Test Statistic               85.306       80.872
  Degrees of freedom                                24           24
  P-value (Chi-square)                           0.000        0.000
  Scaling correction factor                                   1.055
    for the Satorra-Bentler correction

Model test baseline model:

  Minimum Function Test Statistic              918.852      789.298
  Degrees of freedom                                36           36
  P-value                                        0.000        0.000

Full model versus baseline model:

  Comparative Fit Index (CFI)                    0.931        0.925
  Tucker-Lewis Index (TLI)                       0.896        0.887
```

mimic option

```
> cfa(HS.model, data = HolzingerSwineford1939,
      estimator = "MLM", mimic = "EQS")
...
  Estimator                                         ML       Robust
  Minimum Function Test Statistic               85.022       81.141
...

> cfa(HS.model, data = HolzingerSwineford1939,
      estimator = "MLM", mimic = "Mplus")

...
  Estimator                                         ML       Robust
  Minimum Function Test Statistic               85.306       81.908
...

> cfa(HS.model, data = HolzingerSwineford1939,
      estimator = "MLM", mimic = "lavaan")
...
  Estimator                                         ML       Robust
  Minimum Function Test Statistic               85.306       80.872
...
```

robust method 3: bootstrapping

1. **parameter estimates: vanilla ML**
2. **bootstrapping standard errors**
   - for the standard errors, we can use the usual nonparametric bootstrap:
     (a) take a bootstrap sample (random selection of cases with replacement)
     (b) fit the model using this bootstrap sample
     (c) extract the t estimated values of the free parameters
     (d) repeat steps 1–3 R times (typically, $R > 1000$)
   - collect all these values in a matrix of size $R \times t$
   - the bootstrap standard errors are the square root of the diagonal elements of the covariance matrix of this $R \times t$ matrix

3. **bootstrapping the test statistic**

   - for the test statistic, we cannot use the usual nonparametric bootstrap, because it reflects not only non-normality and sampling variability, but also model misfit
   - the original sample must first be transformed so that the sample covariance matrix corresponds to the model-implied covariance matrix
   - in the SEM literature, this model-based bootstrap procedure is known as the **Bollen-Stine bootstrap**
   - the standard p value of the chi-square test can be replaced by a bootstrap p value: the proportion of test statistics from the bootstrap samples that exceed the value of the test statistic from the original (parent) sample

bootstrapping in lavaan

- bootstrapping standard errors:

```
fit <- cfa(HS.model, data = HolzingerSwineford1939,
           se = "bootstrap", verbose = TRUE, bootstrap = 1000)
```

- bootstrapping the test statistic

```
fit <- cfa(HS.model, data = HolzingerSwineford1939,
           test = "bootstrap", verbose = TRUE, bootstrap = 1000)
```

- when we use se = “bootstrap”, the parameterEstimates() output will contain bootstrap based confidence intervals

using bootstrapLavaan() to compute the Bollen-Stine p-value (optional)

```
fit <- cfa(HS.model, data = HolzingerSwineford1939, se = "none")

# get the test statistic for the original sample

T.orig <- fitMeasures(fit, "chisq")

# bootstrap to get bootstrap test statistics
# we only generate 10 bootstrap sample in this example; in practice
# you may wish to use a much higher number

T.boot <- bootstrapLavaan(fit,
                          R = 10,
                          type = "bollen.stine",
                          FUN = fitMeasures,
                          fit.measures = "chisq")

# compute a bootstrap based p-value

pvalue.boot <- sum(T.boot > T.orig)/length(T.boot)
```

2.6 Handling categorical endogenous variables

categorical exogenous variables

- categorical exogenous covariates; e.g. gender, country
- we simply need to construct 'dummy variables' and proceed as usual
- just like in ordinary regression

categorical endogenous variables

- need special treatment
- binary data, ordinal (ordered) data
- censored data, limited dependent data
- count data
- nominal (unordered) data

two approaches for handling categorical data in a SEM framework

- limited information approach
  - only univariate and bivariate information is used (think contingency tables)
  - estimation often proceeds in two or three stages; the first stages use maximum likelihood, the last stage uses (weighted) least squares
  - mainly developed in the SEM literature
  - perhaps the best known implementation is in Mplus (WLSMV)

two approaches for handling categorical data in a SEM framework

- full information approach
  - all information is used
  - most practical: marginal maximum likelihood estimation
  - requires numerical integration (number of dimensions = number of latent variables)
  - mainly developed in the IRT literature (and GLMM literature)
  - only recently incorporated in modern SEM software

a limited information approach: the WLSMV estimator

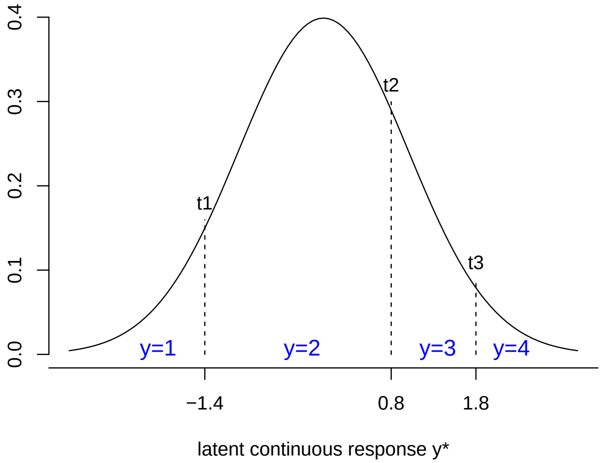
- developed by Bengt Muthén, in a series of papers; the seminal paper is

  Muthén, B. (1984). A general structural equation model with dichotomous, ordered categorical, and continuous latent variable indicators. *Psychometrika, 49*, 115–132

- this approach has been the 'gold standard' in the SEM literature for almost three decades
- first available in LISCOMP (Linear Structural Equations using a Comprehensive Measurement Model), distributed by SSI, 1987 – 1997
- follow up program: Mplus (Version 1: 1998), currently version 8
- other authors (Jöreskog 1994; Lee, Poon, Bentler 1992) have proposed similar approaches (implemented in LISREL and EQS respectively)

stage 1 – estimating the thresholds (1)

- an observed variable y can often be viewed as a partial observation of a latent continuous response $y^\star$; eg ordinal variable with $K = 4$ response categories:



stage 1 – estimating the thresholds (2)

- estimating the thresholds: maximum likelihood using univariate data
  - if no exogenous variables, this is just

```
# generate ‘ordered’ data with 4 categories
Y <- sample(1:4, size = 100, replace = TRUE)

prop <- table(Y)/sum(table(Y))
cprop <- c(0, cumsum(prop))

th <- qnorm(cprop)
```

  - in the presence of exogenous covariates, this is just ordered probit regression

```
library(MASS)
X1 <- rnorm(100); X2 <- rnorm(100); X3 <- rnorm(100)

fit <- polr(ordered(Y) ~ X1 + X2 + X3, method = "probit")
fit$zeta
```

stage 2 – estimating tetrachoric, polychoric, . . . , correlations

- estimate tetrachoric/polychoric/. . . correlation from bivariate data:
  - tetrachoric (binary – binary)
  - polychoric (ordered – ordered)
  - polyserial (ordered – numeric)
  - biserial (binary – numeric)
  - pearson (numeric – numeric)
- ML estimation is available (see eg. Olsson 1979 and 1982)
  - two-step: first estimate thresholds using univariate information only; then, keeping the thresholds fixed, estimate the correlation
  - one-step: estimate thresholds and correlation simultaneously
- if exogenous covariates are involved, the correlations are based on the residual values of $y^\star$ (eg bivariate probit regression)

stage 3 – estimating the SEM model

- third stage uses weighted least squares:

$$F_{WLS} = (\mathbf{s} - \hat{\boldsymbol{\sigma}})^\top \mathbf{W}^{-1} (\mathbf{s} - \hat{\boldsymbol{\sigma}})$$

  where $\mathbf{s}$ and $\hat{\boldsymbol{\sigma}}$ are vectors containing all relevant sample-based and model-based statistics respectively

- $\mathbf{s}$ contains: thresholds, correlations, optionally regression slopes of exogenous covariates, optionally variances and means of continuous variables
- the weight matrix $\mathbf{W}$ is (a consistent estimator of) the asymptotic covariance matrix of the sample statistics ($\mathbf{s}$)
- robust version: WLSMV
  - use the diagonal of $\mathbf{W}$ only for estimation (DWLS)
  - use the full matrix for inference (standard errors and test statistic)
  - ‘MV’ stands for the Satterthwaite’s mean and variance corrected test statistic

using categorical variables in lavaan

- declare them as 'ordered' (using the `ordered()` function, which is part of base R) in your data.frame before you run the analysis;
  for example, if you need to declare four variables (say, item1, item2, item3, item4) as ordinal in your data.frame (called 'Data'), you can use something like:

```
Data[,c("item1","item2","item3","item4")] <-
    lapply(Data[,c("item1","item2","item3","item4")], ordered)
```

- use the `ordered=` argument when using one of the fitting functions; for example, if you have four binary or ordinal variables (say, item1, item2, item3, item4), you can use:

```
fit <- cfa(myModel, data = myData, ordered = c("item1","item2",
                                               "item3","item4"))
```

- by default, lavaan will use robust WLS (DWLS + robust standard errors and a scaled-shifted test statistic; this is equivalent to estimator=WLSMV in Mplus)

example

```
# binary version of Holzinger & Swineford
HS9 <- HolzingerSwineford1939[,c("x1","x2","x3","x4","x5",
                                 "x6","x7","x8","x9")]
HSbinary <- as.data.frame( lapply(HS9, cut, 2, labels = FALSE) )

# single factor model
model <- ' visual  =~ x1 + x2 + x3
           textual =~ x4 + x5 + x6
           speed   =~ x7 + x8 + x9 '

# binary CFA
fit <- cfa(model, data=HSbinary, ordered = names(HSbinary))

summary(fit, fit.measures = TRUE)
```

output

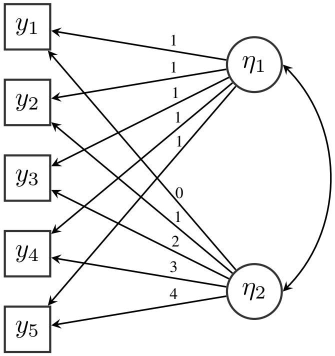
```
lavaan (0.6-1) converged normally after  35 iterations

  Number of observations                           301

  Estimator                                       DWLS      Robust
  Minimum Function Test Statistic               30.918      38.546
  Degrees of freedom                                24          24
  P-value (Chi-square)                           0.156       0.030
  Scaling correction factor                                  0.866
  Shift parameter                                            2.861
    for simple second-order correction (Mplus variant)

Model test baseline model:

  Minimum Function Test Statistic              582.533     469.769
  Degrees of freedom                                36          36
  P-value                                        0.000       0.000

User model versus baseline model:

  Comparative Fit Index (CFI)                    0.987       0.966
  Tucker-Lewis Index (TLI)                       0.981       0.950

Root Mean Square Error of Approximation:

  RMSEA                                          0.031       0.045
  90 Percent Confidence Interval          0.000  0.059         0.014  0.070
  P-value RMSEA <= 0.05                          0.847         0.596

Weighted Root Mean Square Residual:

  WRMR                                           0.829         0.829

Parameter Estimates:

  Information                                 Expected
  Standard Errors                           Robust.sem

Latent Variables:
                   Estimate  Std.Err  Z-value  P(>|z|)
  visual =~
    x1                1.000
    x2                0.900    0.188    4.788    0.000
    x3                0.939    0.197    4.766    0.000
  textual =~
    x4                1.000
    x5                0.976    0.118    8.241    0.000
    x6                1.078    0.125    8.601    0.000
  speed =~
    x7                1.000
    x8                1.569    0.461    3.403    0.001
    x9                1.449    0.409    3.541    0.000

Covariances:
```

```
                   Estimate  Std.Err  Z-value  P(>|z|)
  visual ~~
    textual           0.303    0.061    4.981    0.000
    speed             0.132    0.049    2.700    0.007
  textual ~~
    speed             0.076    0.046    1.656    0.098

Intercepts:
                   Estimate  Std.Err  Z-value  P(>|z|)
    x1                0.000
    x2                0.000
    x3                0.000
    x4                0.000
    x5                0.000
    x6                0.000
    x7                0.000
    x8                0.000
    x9                0.000
    visual            0.000
    textual           0.000
    speed             0.000

Thresholds:
                   Estimate  Std.Err  Z-value  P(>|z|)
    x1|t1            -0.388    0.074   -5.223    0.000
    x2|t1            -0.054    0.072   -0.748    0.454
    x3|t1             0.318    0.074    4.309    0.000
    x4|t1             0.180    0.073    2.473    0.013
    x5|t1             -0.257    0.073   -3.506    0.000
    x6|t1              1.024    0.088   11.641    0.000
    x7|t1              0.231    0.073    3.162    0.002
    x8|t1              1.128    0.092   12.284    0.000
    x9|t1              0.626    0.078    8.047    0.000

Variances:
                   Estimate  Std.Err  Z-value  P(>|z|)
    x1                 0.592
    x2                 0.670
    x3                 0.640
    x4                 0.303
    x5                 0.336
    x6                 0.191
    x7                 0.778
    x8                 0.453
    x9                 0.534
    visual             0.408    0.112    3.651    0.000
    textual            0.697    0.101    6.883    0.000
    speed              0.222    0.094    2.363    0.018

Scales y*:
                   Estimate  Std.Err  Z-value  P(>|z|)
    x1                 1.000
    x2                 1.000
    x3                 1.000
    x4                 1.000
    x5                 1.000
    x6                 1.000
    x7                 1.000
    x8                 1.000
    x9                 1.000

> inspect(fit, "sampstat")
$cov
x1     x2     x3     x4     x5     x6     x7     x8     x9
x1  1.000
x2  0.284  1.000
x3  0.415  0.389  1.000
x4  0.364  0.328  0.232  1.000
x5  0.319  0.268  0.138  0.688  1.000
x6  0.422  0.322  0.206  0.720  0.761  1.000
x7 -0.048  0.061  0.041  0.200  0.023 -0.029  1.000
x8  0.159  0.105  0.439 -0.029 -0.059  0.183  0.464  1.000
x9  0.165  0.210  0.258  0.146  0.183  0.230  0.335  0.403  1.000

$mean
x1 x2 x3 x4 x5 x6 x7 x8 x9
 0  0  0  0  0  0  0  0  0

$th
 x1|t1  x2|t1  x3|t1  x4|t1  x5|t1  x6|t1  x7|t1  x8|t1  x9|t1
-0.388 -0.054  0.318  0.180 -0.257  1.024  0.231  1.128  0.626
```

2.7 Growth models

- the SEM framework accommodates many longitudinal models, where a single measure is administered repeatedly over time (vs multiple measures at once)
- latent variables can be used as "random effects" associated with various model parameters (similar to multilevel/hierarchical models)
- a latent variable no longer represents a trait of interest; it more generally represents something that varies across observations (e.g., a slope describing change over time)

- in the model below, everyone is measured at five equally-spaced time points; η_1 is a random intercept and η_2 is a random slope



growth models

- the latent variable η_2 represents a slope, so the loading parameters must represent something about time
- this is where the model starts to look different from traditional SEMs: we fix loading parameters to correspond to time points at which people are measured
- note this only works well when everyone is measured at the same time points (though see *definition variables* in Mplus)

growth vs multilevel

- the growth model from the previous slide corresponds exactly to a multilevel model with a random slope and intercept, where everyone is measured at exactly the same time
- if you have seen R package *lme4*, the command would be something like

  ```
  fit <- lmer(y ~ time + (time | person))
  ```

  where the data are in long format for *lme4*: each row is a single observation at a single time (as opposed to each row containing all observations for one person), `time` repeats the time points 0–4 for each person, and `person` is a "person id" column
- models can get much more complex than this; see, e.g.
  McArdle, J. J. (2009). Latent variable modeling of differences and changes with longitudinal data. *Annual Review of Psychology, 60*, 577–605.
  Kievit et al. (2018). Developmental cognitive neuroscience using latent change score models: A tutorial and applications. *Developmental Cognitive Neuroscience*.

(from Kievit et al, 2018)

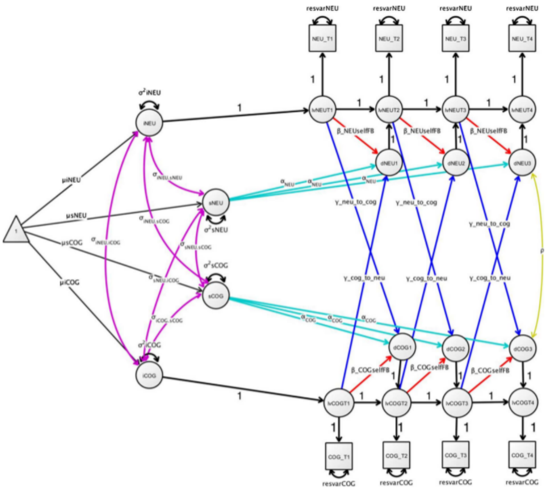
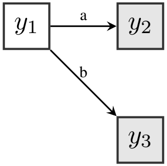


Fig. 5. Bivariate Dual Change Score Model. This more complex latent change score model captures both the stable change over time in the form of slopes (sCOG and sNEU), as well as more fine-grained residual changes. Note this model incorporates latent variables at each timepoint – See Newsom (2015, p. 135) for more detail.

3 Appendices

3.1 Rules about variances and covariances

- consider the following path diagram:



- the two corresponding linear equations are:

$$\begin{cases} y_2 = a\,y_1 + \epsilon_2 \\ y_3 = b\,y_1 + \epsilon_3 \end{cases}$$

- the two equations imply how the three variables are related to each other: there is a relationship between y_2 and y_1, and y_3 and y_1; there is also a relationship between y_2 and y_3 (because, there is a common cause)
- to quantify the (linear) relationship between two variables, we use the *co-variance*
- suppose we have already estimated the values of the structural coefficients a and b, can we derive:
    1. the *model-implied covariances* for each pair of variables in the set (y_1, y_2, y_3)?
    2. the *variances* of the (endogenous) variables y_2 and y_3?
    3. what about the variance of the exogenous variable y_1?
- the *model-implied* variance covariance matrix $\hat{\mathbf{\Sigma}}$:

$$\begin{bmatrix} \sigma^2(y_1) & & \\ \sigma(y_2, y_1) & \sigma^2(y_2) & \\ \sigma(y_3, y_1) & \sigma(y_3, y_2) & \sigma^2(y_3) \end{bmatrix}$$

rules about variances and covariances (1)

- suppose X and Y are random variables, and a and b are constants.
- some simple rules for variances:
  - $\sigma^2(a) = 0$
  - $\sigma^2(a + X) = \sigma^2(X)$
  - $\sigma^2(aX) = a^2 \sigma^2(X)$
  - $\sigma^2(X + Y) = \sigma^2(X) + \sigma^2(Y) + 2\,\sigma(X, Y)$
- some simple rules for covariances:
  - $\sigma(a, b) = 0$
  - $\sigma(a, X) = 0$
  - $\sigma(X, Y) = \sigma(Y, X)$
  - $\sigma(X + a, Y + b) = \sigma(X, Y)$
  - $\sigma(aX, bY) = a\,b\,\sigma(X, Y)$

rules about variances and covariances (2)

- given two linear combinations X and Y:

$$X = a_1X_1 + a_2X_2 + \ldots + a_pX_p \quad \text{and} \quad Y = b_1Y_1 + b_2Y_2 + \ldots + b_qY_q$$

- the general formula for the variance of a linear combination is given by

$$\begin{aligned} \sigma^2(X) &= \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j \sigma(X_i, X_j) \\ &= \sum_{i=1}^{p} a_i^2 \sigma^2(X_i) + \sum_{i=1}^{p}\sum_{j \neq i}^{p} a_i a_j \sigma(X_i, X_j) \end{aligned}$$

- the covariance between these two linear combinations is given by

$$\sigma(X, Y) = \sum_{i=1}^{p}\sum_{j=1}^{q} a_i b_j \sigma(X_i, Y_j)$$

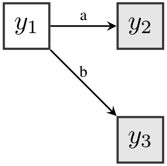
applying the rules

- following the rules for the covariances, we find:
  - $\sigma(y_2, y_3) = a\,b\,\sigma(y_1, y_1) + a\,\sigma(y_1, \epsilon_3) + b\,\sigma(y_1, \epsilon_2) + \sigma(\epsilon_2, \epsilon_3)$
  - $\sigma(y_2, y_1) = a\,\sigma(y_1, y_1) + \sigma(y_1, \epsilon_2)$ and $\sigma(y_3, y_1) = b\,\sigma(y_1, y_1) + \sigma(y_1, \epsilon_3)$
  - but $\sigma(y_1, y_1) = \sigma^2(y_1)$, $\sigma(y_1, \epsilon_3) = 0$, $\sigma(y_1, \epsilon_2) = 0$, and we also assume (here) that $\sigma(\epsilon_2, \epsilon_3) = 0$
- following the rules for variances, we find:
  - $\sigma^2(y_1) = \sigma^2(y_1)$
  - $\sigma^2(y_2) = a^2\sigma^2(y_1) + \sigma^2(\epsilon_2)$ and $\sigma^2(y_3) = b^2\sigma^2(y_1) + \sigma^2(\epsilon_3)$
- the *model-implied* variance covariance matrix for our two equations is

$$\begin{bmatrix} \sigma^2(y_1) & & \\ a\,\sigma^2(y_1) & a^2\,\sigma^2(y_1) + \sigma^2(\epsilon_2) & \\ b\,\sigma^2(y_1) & a\,b\,\sigma^2(y_1) & b^2\,\sigma^2(y_1) + \sigma^2(\epsilon_3) \end{bmatrix}$$

the model-implied covariance matrix for our two-equation model

- for example, if $a = 3$ and $b = 5$, $\sigma^2(y_1) = 10$, $\sigma^2(\epsilon_2) = 20$ and $\sigma^2(\epsilon_3) = 30$, then for this model:

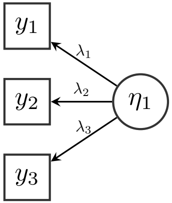


we find

$$\hat{\mathbf{\Sigma}} = \begin{bmatrix} 10 & & \\ 30 & 110 & \\ 50 & 150 & 280 \end{bmatrix}$$

the model-implied covariance in a one-factor CFA

- consider this simple one-factor CFA with three indicators:



- the model-implied covariance is again a function of the model parameters:

$$\begin{bmatrix} \lambda_1^2\sigma^2(\eta_1) + \sigma^2(\epsilon_1) & & \\ \lambda_1\lambda_2\sigma^2(\eta_1) & \lambda_2^2\sigma^2(\eta_1) + \sigma^2(\epsilon_2) & \\ \lambda_1\lambda_3\sigma^2(\eta_1) & \lambda_2\lambda_3\sigma^2(\eta_1) & \lambda_3^2\sigma^2(\eta_1) + \sigma^2(\epsilon_3) \end{bmatrix}$$

where we have assumed that the ϵ's are uncorrelated.

3.2 Estimator ML, MLM and MLR: robust standard errors and scaled test statistics

estimator ML

- ML is the default estimator in all software packages for SEM
- the likelihood function is derived from the multivariate normal distribution (the 'normal' tradition) or the Wishart distribution (the 'Wishart' tradition)
- standard errors are usually based on the covariance matrix that is obtained by inverting the expected information matrix

$$n\text{Cov}(\hat{\theta}) = A^{-1} = (\Delta' W \Delta)^{-1}$$

  - Δ is a jacobian matrix and W is a function of Σ^{-1}
  - if no meanstructure:

$$\Delta = \partial\hat{\Sigma}/\partial\hat{\theta}'$$
$$W = \frac{1}{2}D'(\hat{\Sigma}^{-1} \otimes \hat{\Sigma}^{-1})D$$

- an alternative is to use the *observed* information matrix

$$\begin{aligned} n\text{Cov}(\hat{\theta}) &= A^{-1} \\ &= [-\text{Hessian}]^{-1} \\ &= \left[-\partial F(\hat{\theta})/(\partial\hat{\theta}\partial\hat{\theta}')\right]^{-1} \end{aligned}$$

  where $F(\theta)$ is the function that is minimized

- overall model evaluation is based on the likelihood-ratio (LR) statistic (chi-square test): T_{ML}
  - (minus two times the) difference between loglikelihood of user-specified model H_0 and unrestricted model H_1
  - equals (in lavaan 0.5) $2 \times n$ times the minimum value of $F(\theta)$
  - T_{ML} follows (under regularity conditions) a chi-square distribution

estimator MLM

- parameter estimates are standard ML estimates
- standard errors are robust to non-normality
  - standard errors are computed using a sandwich-type estimator:

$$\begin{aligned} n\text{Cov}(\hat{\theta}) &= A^{-1}BA^{-1} \\ &= (\Delta' W \Delta)^{-1}(\Delta' W \Gamma W \Delta)(\Delta' W \Delta)^{-1} \end{aligned}$$

  - A is usually the expected information matrix (but not in Mplus)
  - references: Huber (1967), Browne (1984), Shapiro (1983), Bentler (1983), . . .

- chi-square test statistic is robust to non-normality
  - test statistic is ‘scaled’ by a correction factor

$$T_{SB} = T_{ML}/c$$

  - the scaling factor c is computed by:

$$c = tr\left[U\Gamma\right]/\text{df}$$

    where

$$U = (W^{-1} - W^{-1}\Delta(\Delta' W^{-1}\Delta)^{-1}\Delta' W^{-1})$$

  - correction method described by Satorra & Bentler (1986, 1988, 1994)
- estimator MLM: for complete data only

estimator MLR

- parameter estimates are standard ML estimates
- standard errors are robust to non-normality
  - standard errors are computed using a (different) sandwich approach:

$$\begin{aligned} n\text{Cov}(\hat{\theta}) &= A^{-1}BA^{-1} \\ &= A_0^{-1}B_0A_0^{-1} = C_0 \end{aligned}$$

    where

$$A_0 = -\sum_{i=1}^{n} \frac{\partial \log L_i}{\partial\hat{\theta}\,\partial\hat{\theta}'} \quad \text{(observed information)}$$

    and

$$B_0 = \sum_{i=1}^{n} \left(\frac{\partial \log L_i}{\partial\hat{\theta}}\right) \times \left(\frac{\partial \log L_i}{\partial\hat{\theta}}\right)'$$

  - for both complete and incomplete data

- Huber (1967), Gourieroux, Monfort & Trognon (1984), Arminger & Schoenberg (1989)

- chi-square test statistic is robust to non-normality
  - test statistic is 'scaled' by a correction factor

$$T_{MLR} = T_{ML}/c$$

  - the scaling factor c is (usually) computed by

$$c = tr\,[M]$$

    where

$$M = C_1(A_1 - A_1\Delta(\Delta' A_1 \Delta)^{-1}\Delta' A_1)$$

  - A_1 and C_1 are computed under the unrestricted (H_1) model
  - correction method described by Yuan & Bentler (2000)

- information matrix (A) can be observed or expected

- for complete data, the MLR and MLM corrections are asymptotically equivalent