Multilevel Structural Equation Modeling with lavaan (part 1)

Yves Rosseel + Axel Mayer

Tübingen, 17 September 2017
## Contents

1  **SEM Refresher**  
   1.1 From regression to structural equation modeling  
   1.2 The model-implied covariance matrix (the essence of SEM)  
   1.3 Matrix representation in a CFA model  
   1.4 The implied covariance matrix for the full SEM model  
   1.5 Model estimation  
   1.6 Model evaluation  
   1.7 Model respecification  
   1.8 Further reading  

2  **Introduction to lavaan**  

3  **Multilevel regression**  

4  **Multilevel SEM**  
   4.1 Introduction  
   4.2 History
4.3 Frameworks (and software) for multilevel SEM .......... 61
4.4 The two-level SEM model with random intercepts .......... 69
4.5 Loglikelihood of a two-level SEM ........................... 71
4.6 The status of a latent variable in a two-level SEM .......... 86
4.7 The status of observed covariates in a two-level SEM .......... 98
4.8 Evaluating model fit .......................................... 103
4.9 Example: two-level CFA ....................................... 105
4.10 Example: two-level SEM ....................................... 121

5 Alternative ways to analyze multilevel data with SEM .... 128
5.1 The ‘wide data’ approach ...................................... 130
5.2 The ‘survey’ (design-based) approach ........................ 138
5.3 MUthén ML (MUML) estimator using multiple group SEM ... 142

6 Last slide ................................. 149
1 SEM Refresher

1.1 From regression to structural equation modeling

univariate linear regression

\[ y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \epsilon_i \quad (i = 1, 2, \ldots, n) \]
multivariate regression
path analysis

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

- in the social sciences, observed variables are not without measurement error

- single indicator measurement model

- multiple indicator measurement model
confirmatory factor analysis (CFA)

- factor analysis: representing the relationship between one or more latent variables and their (observed) indicators
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 (relationship latent variables and indicators)
  - structural equations (regressions among latent/observed variables)

- each diagram has ‘model-based’ implications
  - for the model-implied covariance matrix: $\hat{\Sigma}$
  - for the model-implied mean vector: $\hat{\mu}$
  - ...

- different diagrams lead to (potentially) different implications; some implications may not fit with your data
1.3 Matrix representation in a CFA model

classic example CFA

- well-known dataset; based on Holzinger & Swineford (1939) data
- also analyzed by Jöreskog (1969)
- 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
observed covariance matrix: $S$

- $p$ is the number of observed variables: $p = 9$

- observed covariance matrix (elements divided by $N-1$):

\[
\begin{array}{cccccccc}
  & x1 & x2 & x3 & x4 & x5 & x6 & x7 & x8 & x9 \\
 x1 & 1.36 \\
 x2 & 0.41 & 1.38 \\
 x3 & 0.58 & 0.45 & 1.28 \\
 x4 & 0.51 & 0.21 & 0.21 & 1.35 \\
 x5 & 0.44 & 0.21 & 0.11 & 1.10 & 1.66 \\
 x6 & 0.46 & 0.25 & 0.24 & 0.90 & 1.01 & 1.20 \\
 x7 & 0.09 & -0.10 & 0.09 & 0.22 & 0.14 & 0.14 & 1.18 \\
 x8 & 0.26 & 0.11 & 0.21 & 0.13 & 0.18 & 0.17 & 0.54 & 1.02 \\
 x9 & 0.46 & 0.24 & 0.37 & 0.24 & 0.30 & 0.24 & 0.37 & 0.46 & 1.02 \\
\end{array}
\]

- we want to ‘explain’ the observed correlations/covariances by postulating a number of 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 a smaller number of ‘free parameters’ in the CFA model, summarized in a number of (typically sparse) matrices
the standard CFA model: matrix representation

- the classic LISREL representation uses three matrices (for CFA)

- the LAMBDA matrix contains the ‘factor structure’:

  \[
  \Lambda = \begin{bmatrix}
  x & 0 & 0 \\
  x & 0 & 0 \\
  x & 0 & 0 \\
  0 & x & 0 \\
  0 & x & 0 \\
  0 & x & 0 \\
  0 & 0 & x \\
  0 & 0 & x \\
  0 & 0 & x \\
  \end{bmatrix}
  \]

- the variances/covariances of the latent variables are summarized in the PSI matrix:
\[ \Psi = \begin{bmatrix} x \\ x & x \\ x & x & x \end{bmatrix} \]

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

\[ \Theta = \begin{bmatrix} x \\ x \\ x \\ x \\ x \\ x \\ x \\ x \\ x \\ x \\ x \\ x \end{bmatrix} \]

- 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:

\[ \Sigma = \Lambda \Psi \Lambda' + \Theta \]

• all parameters are included in three model matrices
• simple matrix multiplication (and addition) gives us the model implied covariance matrix
• for identification purposes, some parameters need to be fixed to a constant
• estimation problem: choose the ‘free’ parameters, so that the estimated implied covariance matrix (\( \hat{\Sigma} \)) is ‘as close as possible’ to the observed covariance matrix \( S \)
  
  – generalized (weighted) least-squares estimation (GLS, WLS)
  – maximum likelihood estimation (ML)
  – Bayesian approaches
setting the metric of the latent variables: UVI of 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
1.4 The implied covariance matrix for the full SEM model

- in the LISREL representation, we need an additional matrix (B):

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

where B summarizes the regressions among the latent variables

- we need this extended model for

  - second-order CFA
  - MIMIC models
  - SEM models

- in LISREL parlance, this the ‘all-y’ model
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:

  \[
  \begin{align*}
  y_1 & : \text{Expert ratings of the freedom of the press in 1960} \\
  y_2 & : \text{The freedom of political opposition in 1960} \\
  y_3 & : \text{The fairness of elections in 1960} \\
  y_4 & : \text{The effectiveness of the elected legislature in 1960} \\
  y_5 & : \text{Expert ratings of the freedom of the press in 1965} \\
  y_6 & : \text{The freedom of political opposition in 1965} \\
  y_7 & : \text{The fairness of elections in 1965} \\
  y_8 & : \text{The effectiveness of the elected legislature in 1965} \\
  x_1 & : \text{The gross national product (GNP) per capita in 1960} \\
  x_2 & : \text{The inanimate energy consumption per capita in 1960} \\
  x_3 & : \text{The percentage of the labor force in industry in 1960}
  \end{align*}
\]
selection of the output

| Latent variables: | Estimate | Std.err | Z-value | P(>|z|) | Std.lv | Std.all |
|------------------|----------|---------|---------|---------|--------|---------|
| **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 ~ dem60 | 0.837   | 0.098   | 8.514   | 0.000   | 0.885  | 0.885   |

...
1.5 Model estimation

• we seek those values for $\theta$ that minimize the difference between what we observe in the data, $S$, and what the model implies, $\Sigma(\theta)$

• the final estimated values are denoted by $\hat{\theta}$, and the estimated model-implied covariance matrix can be written as $\hat{\Sigma} = \Sigma(\hat{\theta})$

• there are many ways to quantify this ‘difference’, leading to different discrepancy measures

• the most used discrepancy measure is based on maximum likelihood:

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

• in practice, we replace $\Sigma$ by $\hat{\Sigma} = \Sigma(\hat{\theta})$

• an alternative is (weighted) least squares, for some weight matrix $W$:

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

where $s$ and $\sigma$ are the unique elements of $S$ and $\Sigma$ respectively
1.6 Model evaluation

evaluation of global fit – chi-square test statistic

• the chi-square test statistic is the primary test of our model

• if the chi-square test statistic is NOT significant, we have a good fit of the model

• this becomes increasingly difficult if 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 good reference is still Hu & Bentler (1999)

admissibility of the results

• are the parameter values valid?
1.7 Model respecification

• 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

• of course, we may let the data speak for itself, and have a look at the modification indices (a more exploratory approach)
1.8 Further reading


…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).


SEM in R, using lavaan


2 Introduction to lavaan

what is lavaan?

• lavaan is an R package for latent variable analysis:
  – user-friendly syntax, many convenient extractor functions
  – general mean/covariance structure modeling
  – support for continuous, binary and ordinal data

• under development, future plans:
  – multilevel SEM (0.6), mixture/latent-class SEM, Bayesian SEM (?)

• the long-term goal of lavaan is
  1. to implement all the state-of-the-art capabilities that are currently available in commercial packages
  2. to provide a modular and extensible platform that allows for easy implementation and testing of new statistical and modeling ideas
installing lavaan, finding documentation

- **lavaan** depends on the R project for statistical computing:
  
  http://www.r-project.org

- to install **lavaan**, simply start up an R session and type:
  
  > install.packages("lavaan")

- more information about **lavaan**:
  
  http://lavaan.org

- the lavaan paper:
  

- **lavaan** discussion group (mailing list)
  
  https://groups.google.com/d/forum/lavaan
installing a development version of lavaan

• first method: type in R:

  > install.packages("lavaan", repos = "http://www.da.ugent.be",
                   type = "source")

• second method, using the devtools package:

  > library(devtools)
  > install_github("yrosseel/lavaan")

• third method: if no internet, but you have a lavaan *.tar.gz file

  > install.packages("c:/temp/lavaan_0.6-1.tar.gz", NULL, type = "source")

where you need to adapt the first string to point to the directory where the lavaan *.tar.gz file is located
the lavaan ecosystem

- **blavaan** (Ed Merkle, Yves Rosseel)
  Bayesian SEM (using jags) with a lavaan interface

- **lavaan.survey** (Daniel Oberski)
  survey weights, clustering, strata, and finite sampling corrections in SEM

- **Onyx** (Timo von Oertzen, Andreas M. Brandmaier, Siny Tsang)
  interactive graphical interface for SEM (written in Java)

- **semTools** (maintainer: Terrence Jorgensen)
  collection of useful functions for SEM

- **simsem** (maintainer: Terrence Jorgensen)
  simulation of SEM models
the lavaan ecosystem (2)

- **semPlot** (Sacha Epskamp)
  
  visualizations of SEM models

- **EffectLiteR** (Axel Mayer, Lisa Dietzfelbinger)
  
  using SEM to estimate average and conditional effects

- **nlsem** (Nora Umbach and many others)
  
  Estimation of structural equation models with nonlinear effects and underlying nonnormal distributions

- many others
  
  bmem, coefficientalpha, eqs2lavaan, fSRM, influence.SEM, MI-IVsem, profileR, RAMpath, regsem, RMediation, RSA, rsem, stremo, faoutlier, gimme, lavaan.shiny, matrixpls, MBESS, NlsyLinks, nonnest2, piecewiseSEM, pscore, psytabs, qgraph, sesem, sirt, TAM, userfriendlyscience, ...
the lavaan model syntax – a simple regression

```r
library(lavaan)
myData <- read.csv("c:/temp/myData.csv")

myModel <- ' y ~ x1 + x2 + x3 + x4 '

# fit model
fit <- sem(model = myModel, 
data = myData)

# show results
summary(fit, nd = 4)
```

• to ‘see’ the intercept, use either

```r
fit <- sem(model = myModel, data = myData, meanstructure = TRUE)
```

or include it explicitly in the syntax:

```r
myModel <- ' y ~ 1 + x1 + x2 + x3 + x4 '
```
lavaan output

lavaan (0.5–20) converged normally after 32 iterations

Number of observations 100

Estimator ML
Minimum Function Test Statistic 0.000
Degrees of freedom 0
Minimum Function Value 0.0000000000000

Parameter Estimates:

Information Expected
Standard Errors Standard

Regressions:

|     | Estimate | Std.Err | Z-value | P(>|z|) |
|-----|----------|---------|---------|---------|
| y ~ |          |         |         |         |
| x1  |  5.7733  |  0.5105 |  11.3087|  0.0000 |
| x2  | -1.3214  |  0.4792 |  -2.7574|  0.0058 |
| x3  |  1.1350  |  0.4459 |   2.5451|  0.0109 |
| x4  |  0.2707  |  0.4658 |   0.5812|  0.5611 |

Variances:

|     | Estimate | Std.Err | Z-value | P(>|z|) |
|-----|----------|---------|---------|---------|
| y   | 2075.0999| 293.4634|   7.0711|  0.0000 |
the lavaan model syntax – multivariate regression

```
myModel <- 'y1 ~ x1 + x2 + x3 + x4
y2 ~ x1 + x2 + x3 + x4`
```
the lavaan model syntax – path analysis

myModel <- ' x5 ~ x1 + x2 + x3 
x6 ~ x4 + x5 
x7 ~ x6 '
the lavaan model syntax – mediation analysis

\[
\text{myModel} \leftarrow ' \\
Y \sim b \times M + c \times X \\
M \sim a \times X \\
\text{indirect} \leftarrow a \times b \\
\text{total} \leftarrow c + (a \times b) ,
\]

\[
\text{fit} \leftarrow \text{sem(model = myModel,} \\
\quad \text{data = myData,} \\
\quad \text{se = "bootstrap")}
\]

\text{summary(fit)}
output

...  

Parameter estimates:

<table>
<thead>
<tr>
<th>Information</th>
<th>Observed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Errors</td>
<td>Bootstrap</td>
</tr>
<tr>
<td>Number of requested bootstrap draws</td>
<td>1000</td>
</tr>
<tr>
<td>Number of successful bootstrap draws</td>
<td>1000</td>
</tr>
</tbody>
</table>

| Regression  | Estimate | Std.err | Z-value | P(>|z|) |
|-------------|----------|---------|---------|--------|
| Y ~ M (b)   | 0.597    | 0.098   | 6.068   | 0.000  |
| X (c)       | 2.594    | 1.210   | 2.145   | 0.032  |
| M ~ X (a)   | 2.739    | 0.999   | 2.741   | 0.006  |

<table>
<thead>
<tr>
<th>Variances:</th>
<th>Y</th>
<th>108.700</th>
<th>17.747</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M</td>
<td>105.408</td>
<td>16.556</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Defined parameters:</th>
<th>Yves Rosseel</th>
</tr>
</thead>
<tbody>
<tr>
<td>indirect</td>
<td>1.636</td>
</tr>
<tr>
<td>total</td>
<td>4.230</td>
</tr>
</tbody>
</table>
the lavaan model syntax – using cfa() or sem()

```
HS.model <- ' visual =~ x1 + x2 + x3
textual =~ x4 + x5 + x6
  speed =~ x7 + x8 + x9
',

fit <- cfa(model = HS.model,
          data = HolzingerSwineford1939)

summary(fit, fit.measures = TRUE,
        standardized = TRUE)
```
**the lavaan model syntax – using lavaan()**

```r
HS.model <- '  
  # latent variables  
  visual =~ 1*x1 + x2 + x3  
  textual =~ 1*x4 + x5 + x6  
  speed =~ 1*x7 + x8 + x9  
  
  # factor (co)variances  
  visual ~~ visual; visual ~~ textual  
  visual ~~ speed; textual ~~ textual  
  textual ~~ speed; speed ~~ speed  
  
  # residual variances  
  x1 ~~ x1; x2 ~~ x2; x3 ~~ x3  
  x4 ~~ x4; x5 ~~ x5; x6 ~~ x6  
  x7 ~~ x7; x8 ~~ x8; x9 ~~ x9  
',  
  
fit <- lavaan(model = HS.model,  
  data = HolzingerSwineford1939)  
summary(fit, fit.measures = TRUE,  
  standardized = TRUE)
```
lavaan (0.5-20) converged normally after 35 iterations

Number of observations 301

Estimator ML
Minimum Function Test Statistic 85.306
Degrees of freedom 24
P-value (Chi-square) 0.000

Model test baseline model:

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

User model versus baseline model:

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

Loglikelihood and Information Criteria:

Loglikelihood user model (H0) -3737.745
Loglikelihood unrestricted model (H1) -3695.092

Number of free parameters 21
Akaike (AIC)  7517.490
Bayesian (BIC)  7595.339
Sample-size adjusted Bayesian (BIC)  7528.739

Root Mean Square Error of Approximation:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSEA</td>
<td>0.092</td>
</tr>
<tr>
<td>90 Percent Confidence Interval</td>
<td>0.071 0.114</td>
</tr>
<tr>
<td>P-value RMSEA &lt;= 0.05</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Standardized Root Mean Square Residual:

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRMR</td>
<td>0.065</td>
</tr>
</tbody>
</table>

Parameter Estimates:

<table>
<thead>
<tr>
<th>Information</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Errors</td>
<td>Standard</td>
</tr>
</tbody>
</table>

Latent Variables:

| Latent Variable | Estimate | Std.Err | Z-value | P(>|z|) |
|-----------------|----------|---------|---------|--------|
| visual =~       |          |         |         |        |
| x1              | 1.000    |         |         |        |
| x2              | 0.554    | 0.100   | 5.554   | 0.000  |
| x3              | 0.729    | 0.109   | 6.685   | 0.000  |
| textual =~      |          |         |         |        |
| x4              | 1.000    |         |         |        |
| x5              | 1.113    | 0.065   | 17.014  | 0.000  |
\begin{align*}
  \text{x6} & \quad 0.926 \quad 0.055 \quad 16.703 \quad 0.000 \\
  \text{speed} \sim ~
  \text{x7} & \quad 1.000 \\
  \text{x8} & \quad 1.180 \quad 0.165 \quad 7.152 \quad 0.000 \\
  \text{x9} & \quad 1.082 \quad 0.151 \quad 7.155 \quad 0.000 \\

  \text{Covariances:} & \\
  \begin{array}{l l l l l}
    \text{Estimate} & \text{Std.Err} & \text{Z-value} & \text{P(>|z|)} \\
    \text{visual} \sim \text{textual} & 0.408 & 0.074 & 5.552 & 0.000 \\
    \text{visual} \sim \text{speed} & 0.262 & 0.056 & 4.660 & 0.000 \\
    \text{textual} \sim \text{speed} & 0.173 & 0.049 & 3.518 & 0.000 \\
  \end{array}

  \text{Variances:} & \\
  \begin{array}{l l l l l}
    \text{Estimate} & \text{Std.Err} & \text{Z-value} & \text{P(>|z|)} \\
    \text{x1} & 0.549 & 0.114 & 4.833 & 0.000 \\
    \text{x2} & 1.134 & 0.102 & 11.146 & 0.000 \\
    \text{x3} & 0.844 & 0.091 & 9.317 & 0.000 \\
    \text{x4} & 0.371 & 0.048 & 7.779 & 0.000 \\
    \text{x5} & 0.446 & 0.058 & 7.642 & 0.000 \\
    \text{x6} & 0.356 & 0.043 & 8.277 & 0.000 \\
    \text{x7} & 0.799 & 0.081 & 9.823 & 0.000 \\
    \text{x8} & 0.488 & 0.074 & 6.573 & 0.000 \\
    \text{x9} & 0.566 & 0.071 & 8.003 & 0.000 \\
    \text{visual} & 0.809 & 0.145 & 5.564 & 0.000 \\
    \text{textual} & 0.979 & 0.112 & 8.737 & 0.000 \\
    \text{speed} & 0.384 & 0.086 & 4.451 & 0.000 \\
  \end{array}
\end{align*}
3 Multilevel regression

different types of data with non-independent observations

- clustered data (family members, teeth in a mouth)
- dyadic data (romantic couples)
- hierarchical data (students within schools within regions)
- matched data (case-control studies)
- survey data (nested sampling)
- longitudinal data (blood pressure of patients measured every week)
- repeated measures (within-subjects design)
- …
balanced versus unbalanced data

• when the data is balanced, we have the same number of units within each cluster

• typical examples of balanced data:
  
  – dyadic data: always two units per cluster
  – repeated measures data: everyone has scores for the same set of conditions
  – longitudinal data where the number of observations (over time) is the same for all individuals (often called panel data)
  – hierarchical data where a fixed number of units was sampled for each cluster

• when the data is unbalanced, we have different cluster sizes
  
  – this may be due to missing values
  – in hierarchical data, the number of units for each cluster may vary considerably from cluster to cluster
wide versus long data

• when data is arranged in ‘wide’ format, each row corresponds to a single cluster
  – we may end up with many columns (one for each measure/variable, for each unit)
  – rows are independent
  – unbalanced data can be handled by filling in missing values for the smaller clusters

• when data is arranged in ‘long’ format, each row corresponds to a single unit
  – the columns contain the variables for that unit (only)
  – multiple rows belong to the same clusters
  – rows are not independent
  – higher-level variables (for example school characteristics) are duplicated for each unit
example wide format

<table>
<thead>
<tr>
<th>cluster.id</th>
<th>y1</th>
<th>m1</th>
<th>x1</th>
<th>y2</th>
<th>m2</th>
<th>x2</th>
<th>y3</th>
<th>m3</th>
<th>x3</th>
<th>schoolsize</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>60</td>
<td>28</td>
<td>36</td>
<td>6</td>
<td>4</td>
<td>22</td>
<td>large</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>24</td>
<td>14</td>
<td>10</td>
<td>18</td>
<td>6</td>
<td>20</td>
<td>38</td>
<td>22</td>
<td>medium</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>26</td>
<td>2</td>
<td>2</td>
<td>32</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>medium</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>36</td>
<td>14</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>small</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>14</td>
<td>10</td>
<td>16</td>
<td>28</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>22</td>
<td>small</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>24</td>
<td>20</td>
<td>16</td>
<td>42</td>
<td>18</td>
<td>2</td>
<td>2</td>
<td>28</td>
<td>large</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>22</td>
<td>0</td>
<td>14</td>
<td>32</td>
<td>6</td>
<td>2</td>
<td>18</td>
<td>18</td>
<td>medium</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>34</td>
<td>16</td>
<td>16</td>
<td>14</td>
<td>8</td>
<td>28</td>
<td>large</td>
</tr>
</tbody>
</table>
### example long format

<table>
<thead>
<tr>
<th>cluster.id</th>
<th>y</th>
<th>m</th>
<th>x</th>
<th>schoolsize</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>16</td>
<td>4</td>
<td>60 large</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>28</td>
<td>36</td>
<td>6 large</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>22</td>
<td>12</td>
<td>large</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>24</td>
<td>14</td>
<td>10 medium</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>18</td>
<td>6</td>
<td>20 medium</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>38</td>
<td>22</td>
<td>medium</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>26</td>
<td>2</td>
<td>2 medium</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>32</td>
<td>4</td>
<td>8 medium</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>10 medium</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>4</td>
<td>36</td>
<td>14 small</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>2</td>
<td>2</td>
<td>0 small</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>10 small</td>
</tr>
<tr>
<td>13</td>
<td>5</td>
<td>14</td>
<td>10</td>
<td>16 small</td>
</tr>
<tr>
<td>14</td>
<td>5</td>
<td>28</td>
<td>2</td>
<td>4 small</td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>8</td>
<td>22</td>
<td>6 small</td>
</tr>
<tr>
<td>16</td>
<td>6</td>
<td>24</td>
<td>20</td>
<td>16 large</td>
</tr>
<tr>
<td>17</td>
<td>6</td>
<td>42</td>
<td>18</td>
<td>2 large</td>
</tr>
<tr>
<td>18</td>
<td>6</td>
<td>2</td>
<td>28</td>
<td>18 large</td>
</tr>
<tr>
<td>19</td>
<td>7</td>
<td>22</td>
<td>0</td>
<td>14 medium</td>
</tr>
<tr>
<td>20</td>
<td>7</td>
<td>32</td>
<td>6</td>
<td>2 medium</td>
</tr>
<tr>
<td>21</td>
<td>7</td>
<td>18</td>
<td>18</td>
<td>10 medium</td>
</tr>
<tr>
<td>22</td>
<td>8</td>
<td>0</td>
<td>8</td>
<td>34 large</td>
</tr>
<tr>
<td>23</td>
<td>8</td>
<td>16</td>
<td>16</td>
<td>14 large</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>8</td>
<td>28</td>
<td>18 large</td>
</tr>
</tbody>
</table>
ignoring the dependency structure

• we could treat the sample as a simple random sample with $N$ independent observations

• in the multilevel context, this is often called a ‘disaggregated analysis’, as higher-level variables (e.g., school characteristics) are assigned to the individual level

• although (still) widely used, ignoring the clustering in the data may have severe consequences:
  
  – wrong standard errors
  
  – inflated type I error rates

• what about reviewers?

  – the ‘tolerance’ for ignoring the clustering in data is now almost non-existing in most fields
solutions

- because clustered data is everywhere, a wide spectrum of ‘solutions’ have been proposed:
  - avoiding the clustering (only pick one individual per cluster)
  - aggregating the data (may lead to ecological fallacies)
  - cluster-robust standard errors (clustering is just a nuisance)
  - fixed-effects approach (school as a fixed factor)
  - mixed-effects approach (school as a random factor)
  - ...

- some solutions are naive, and some may lead to wrong conclusions
the many faces of mixed-effects models

- mixed-effects models have been developed in a variety of disciplines, with varying names and terminology:
  - random-effects (ANOVA) models (statistics, econometrics)
  - linear mixed models (statistics)
  - variance components models (statistics)
  - hierarchical linear models (education, Bayesian)
  - multilevel models (sociology, education)
  - contextual-effects models (sociology)
  - random-coefficient models (econometrics)
  - repeated-measures models, repeated measures ANOVA (statistics, psychology)
  - ...

- the different terminology is still a source of much confusion
**multilevel regression**

- multilevel regression is the application of mixed-effects statistical models to analyze hierarchical (or multilevel) data

- this branch of statistics was mainly developed in the educational sciences, and in quantitative sociology

- Blalock (1984) introduced ‘contextual effect models’ in sociology

- school effectiveness researchers realized early on (’70s, ’80s) that taking the cluster structure into account was important
  
  - a regression analysis per school was one solution, but this ignored the fact that many regression coefficients (across schools) should be similar; this similarity should be used (‘borrowing strength’)
  
  - on the other hand, requiring regression coefficients in all schools to be the same, was regarded as too restrictive
  
  - clearly, some intermediate form of analysis was needed
• this led to the idea of random coefficient models, but it left open the problem of combining predictors of different levels

• Burstein (and others) suggested in the early ’80s to proceed in two stages:
  – in a first stage, a regression analysis was done for each school
  – in a second stage, the resulting regression coefficients were entered as outcome variables in a regression, where the predictors were cluster variables
  – this became known as the ‘slopes-as-outcomes’ approach

• in the mid ’80s, it became clear that the models that educational researchers were looking for had been around for quite some time in other branches of statistics (e.g., linear mixed models)

• a number of authors published a series of papers that would eventually lead to what we now call today ‘multilevel regression’ (Mason et al., 1983; Aitkin and Longford, 1986; de Leeuw and Kreft, 1986; Goldstein, 1986; Raudenbush and Bryk, 1986)
• some important textbooks paved the way for a wide adoption of multilevel regression in the social and behavioural sciences:


4  Multilevel SEM

4.1  Introduction

• limitations of the multilevel regression model:
  – (mostly) univariate perspective (multivariate is possible but awkward)
  – no measurement models (latent variables)
  – no mediators (only strictly dependent or independent variables)
  – no reciprocal effects, no goodness-of-fit measures, . . .

• two evolutions since the late 1980s:
  – the multilevel regression framework was extended to include measurement errors and latent variables (cfr. HLM and MLwiN software)
  – the traditional SEM framework started to incorporate random intercepts and random slopes

• the boundaries between SEM and multilevel regression have gradually disappeared
4.2 History


  - full modeling of within and between covariance matrices
  - provided a computer program for ML estimation
  - balanced data only, no level-2 variables, no meanstructure
  - structured case is described in Schmidt & Wisenbaker (1986)


LISREL also offers great possibilities for conducting such multilevel analyses. It has been shown by Schmidt (1969) that maximum likelihood estimates can be derived of the within-class and between-class covariance matrices, and these can...
be parameterized in LISREL models, to allow separate estimates of parameters at the two levels [...] A great problem, of course, is that there in most studies tend to be few classes (or other higher level units) only, which precludes the possibility of obtaining any stable estimates at the class level. We would like to suggest, however, that in the least within-class analyses should be performed to guard against the possibility that results obtained in non-hierarchical analyses can in fact he accounted for by effects at the class level, which may be more or less artifactual.

- his work was also picked up by Leigh Burstein
  - Burstein worked at the Graduate School of Education (UCLA)

  - he reformulated Schmidt’s fitting function so that it could be estimated using existing software for multiple-group SEM (e.g., LISCOMP)
Goldstein & McDonald

  - very general formulation, including multilevel SEM
  - univariate perspective (multivariate vector = 1st level)
  - can handle missing data, hierarchical data, cross-classified data
  - expression of the likelihood, IGLS algorithm is suggested

  - multivariate perspective, within-and-between formulation
  - likelihood expression + a computationally tractable (re)expression
  - both for balanced and unbalanced clusters

Muthén


  – re-expresses the within-part of the likelihood as a sum over different cluster sizes

  – in the balanced case, this leads to a multiple-group SEM fitting function with two groups


  – derivations of Muthén (1989)

  – suggestion: we can use the balanced solution even in the unbalanced case (using an estimate of the average cluster size): estimator = MUML


• standard SEM software could be used (at least for the balanced case)
Lee

  
  – statistically more rigorous development of multilevel SEM theory: ML and GLS estimation, inference, goodness-of-fit statistics, constraints
  
  – suggested using Fisher scoring and Gauss-Newton for optimization
  
  – no level-2 variables

• Poon & Lee (1992): within-part as sum over different cluster sizes

• Yau, Lee & Poon (1993): three-level setting

• Lee & Poon (1998): using the EM algorithm (by treating the the latent random vectors at the cluster level as missing data)


  – Chapter 9: Bayesian methods for analyzing various two-level SEMs
Bentler

  - extend the EM algorithm of Lee & Poon (1998) to handle level-2 predictors
  - clever way to avoid a large number of matrix inversions
  - often considered to be the state-of-the-art algorithm for estimating 2-level SEMs with continuous responses
  - no missing data, no random slopes

- perhaps the last technical paper on (continuous) two-level SEM (in the frequentist framework)
4.3 Frameworks (and software) for multilevel SEM

overview

• two-level SEM with random intercepts
  – Mplus (type = twolevel), LISREL, EQS, lavaan

• the gllamm framework: gllamm, (related approach: Latent Gold)

• the Mplus framework: Mplus

• the case-wise likelihood based approach (e.g., Mehta & Neale, 2005)
  – Mplus (type = random), Mx, OpenMx (definition variables)
  – in principle: both continuous and categorical outcomes; random slopes
  – xxM?

• the Bayesian framework
  – Mplus
  – (Open)BUGS, JAGS, Stan
two-level SEM with random intercepts

- an extension of single-level SEM to incorporate random intercepts
- extensive technical literature, starting from the late 1980s (until about 2004)
- available in Mplus, EQS, LISREL, lavaan, ...
- this is by far the most widely used framework in the applied literature

- advantages:
  - fast, simple, well-understood, plenty of examples
  - well-documented

- disadvantages:
  - continuous outcomes only
  - no random slopes
the Mplus framework

• the Mplus framework has added many extensions to the two-level within/between approach in the last 17 years
  – EM algorithm can handle random slopes and missing data
  – categorical outcomes (with numerical quadrature)
  – multilevel (robust) (D)WLS
  – combination multilevel with complex survey data, mixture modeling, …

• advantages:
  – superb implementation
  – user-friendly, familiar (‘multivariate’) approach

• disadvantages:
  – NO technical documentation (about the extensions)
  – black box software
the gllamm framework

- Sophia Rabe-Hesketh, Anders Skrondal and Andrew Pickles

- see http://www.gllamm.org/

- an extension of generalized linear mixed models to include (continuous and discrete) latent variables (including a structural part)

- advantages:
  - very well documented, open-source code (written in Stata)
  - handles a wide range of outcome types (normal, categorical, ...)
  - very general, very flexible

- disadvantages:
  - not easy to specify (complex) models, univariate perspective
  - needs Stata
  - very, very slow (even in the continuous case)
lavaan

• multilevel SEM development just started (jan 2017)

• implemented in the development version (0.6-1):
  – standard two-level ‘within-and-between’ approach
  – continuous responses only, no missing data (for now)
  – no random slopes (for now)
  – using quasi-newton optimization (for now)

• future plans: many
  – gllamm framework (but more user-friendly)
  – case-wise likelihood approach
  – hybrids
lavaan syntax setup for two-level SEM

\[ \Sigma_B \]

\underline{Between} \\
\underline{Within} \\

\[ \Sigma_W \]

```
model <- ' 
  level: 1 
    # here comes the within level 
  level: 2 
    # here comes the between level ,

fit <- sem(myModel, myData, 
  cluster = "school")
```
useful literature

• the relationship between SEM and multilevel regression:


• books:


4.4 The two-level SEM model with random intercepts

- we assume two-level data with individuals (students) nested within clusters (schools)
- in this framework, we decompose the total score of each variable into two parts: a within part, and a between part (Cronbach & Webb, 1979):

\[
\begin{align*}
  y_{ji} & = (y_{ji} - \bar{y}_g) + \bar{y}_g \\
  y_T & = y_W + y_B
\end{align*}
\]

where \( j = 1, \ldots, J \) is an index for the clusters, and \( i = 1, \ldots, n_j \) is an index for the units within a cluster; \( \bar{y}_j \) is the cluster mean of cluster \( j \)

- both components are treated as unknown (latent) variables
- the two parts are orthogonal and additive; one of the parts can be zero
- the total covariance (at the population level) can be decomposed as

\[
\text{Cov}(y) = \Sigma_T = \Sigma_W + \Sigma_B
\]
two-level SEM: specifying a model for each level

- for a two-level CFA model, we can use

\[ \Sigma_W = \Lambda_W \Psi_W \Lambda_W' + \Theta_W \]

and

\[ \Sigma_B = \Lambda_B \Psi_B \Lambda_B' + \Theta_B \]

- if we add a structural (regression) part, we need to add the \((I - B)^{-1}\) term to the matrix formulation (as in regular SEM)

- no meanstructure is needed for the within part (as the level-1 variables are cluster-centered)

- a meanstructure \(\mu_B\) can be added for the between part of the model

- in addition, we can add level-2 covariates \((z_j)\) to the model
4.5 Loglikelihood of a two-level SEM

notation

- number of clusters: $J$, number of units per cluster: $n_j$
- data for cluster $j$:
  \[ \mathbf{v}_j = [\mathbf{z}_j, \mathbf{y}_{j1}, \mathbf{y}_{j2}, \ldots, \mathbf{y}_{jn_j}]^T \]
- model implied matrices/vectors: $\Sigma_{zz}$, $\Sigma_{zy}$, $\Sigma_w$, $\Sigma_b$ and $\mu_b = [\mu_z, \mu_y]^T$
- expectation of $\mathbf{v}_j$:
  \[ E[\mathbf{v}_j] = \hat{\mathbf{v}}_j = [\mu_z, \mu_y, \mu_y, \ldots, \mu_y]^T \]
- covariance matrix for $\mathbf{v}_j$:
  \[
  \text{Cov}[\mathbf{v}_j] = \mathbf{V}_j = \begin{bmatrix}
  \Sigma_{zz} & 1_{n_j}^T \otimes \Sigma_{zy} \\
  1_{n_j} \otimes \Sigma_{yz} & \Sigma_{yy}
  \end{bmatrix}
  \]
  where
  \[
  \Sigma_{yy} = \mathbf{I}_{n_j} \otimes \Sigma_w + 1_{n_j} 1_{n_j}^T \otimes \Sigma_b
  \]
loglikelihood

- assuming multivariate normality, we can write the loglikelihood for cluster $j$ as follows:

$$\text{loglik}_j = -\frac{O_j}{2} \ln(2\pi) - \frac{1}{2} \ln |V_j| - \frac{1}{2} (v_j - \hat{v}_j)^T V_j^{-1} (v_j - \hat{v}_j)$$

where $O_j$ is the length of $v_j$, usually $p_z + (n_j \times p_y)$

- the total likelihood over all $J$ clusters:

$$\text{loglik} = \sum_{j=1}^{J} \text{loglik}_j$$

- we can find ML estimates by minimizing the objective function $F_{ML}$ which is minus two times the loglikelihood function, ignoring the constant:

$$F_{ML} = \sum_{j=1}^{J} \ln |V_j| + (v_j - \hat{v}_j)^T V_j^{-1} (v_j - \hat{v}_j)$$
**objective function (optional)**

- the original objective function:

\[
F_{ML} = \sum_{j=1}^{J} \ln |V_j| + (v_j - \hat{v}_j)^T V_j^{-1} (v_j - \hat{v}_j)
\]

- for large clusters, the size of \( V_j \) can be formidable

- we should exploit the block-diagonal structure of \( V \)

- we define:

\[
\Sigma_{b.z} = (\Sigma_b - \Sigma_{yz} \Sigma_{zz}^{-1} \Sigma_{zy})
\]
• version 1: McDonald & Goldstein (1989), per cluster, using $\Sigma_{b.z}$:

$$F_{ML} = \sum_{j=1}^J \left[ \ln |\Sigma_{zz}| + (n_j - 1) \ln |\Sigma_{w}| + \ln |\Sigma_{w} + n_j \cdot \Sigma_{b.z}| ight.$$  

$$+ \text{tr} \left[ (\Sigma_{zz}^{-1} + n_j \Sigma_{zz}^{-1} \Sigma_{zy} (n_j \Sigma_{b.z} + \Sigma_{w})^{-1} \Sigma_{yz} \Sigma_{zz}^{-1}) (z_j - \mu_z)(z_j - \mu_z)^T \right]$$  

$$+ 2n_j \text{tr} \left[ -\Sigma_{zz}^{-1} \Sigma_{zy} (n_j \Sigma_{b.z} + \Sigma_{w})^{-1} (\bar{y}_j - \mu_y)(z_j - \mu_z)^T \right]$$  

$$+ \text{tr} \left[ \Sigma_{w}^{-1} \mathbf{Y}_j^{(c)T} \mathbf{Y}_j^{(c)} \right]$$  

$$- n_j \text{tr} \left[ \Sigma_{w}^{-1} (\bar{y}_j - \mu_y)(\bar{y}_j - \mu_y)^T \right]$$  

$$+ n_j \text{tr} \left[ (n_j \Sigma_{b.z} + \Sigma_{w})^{-1} (\bar{y}_j - \mu_y)(\bar{y}_j - \mu_y)^T \right]$$
• version 2: lavaan = McDonald & Goldstein (1989), per cluster size,

\[
F_{ML} = (N - J) \left( \ln |\Sigma_w| + \text{tr} \left[ \Sigma_w^{-1} S_{pw} \right] \right) + \\
\sum_{s=1}^{S} n_s \cdot \left[ (\ln |\Sigma_{zz}| + \ln |\Sigma_w + n_j \cdot \Sigma_{b,z}|) + \\
\text{tr} \left[ (\Sigma_{zz}^{-1} + n_j \Sigma_{zz}^{-1} \Sigma_{zy} (n_j \Sigma_{b,z} + \Sigma_w)^{-1} \Sigma_{yz} \Sigma_{zz}^{-1}) (z_j - \mu_z) (z_j - \mu_z)^T \right] \\
+ 2n_j \text{tr} \left( -\Sigma_{zz}^{-1} \Sigma_{zy} (n_j \Sigma_{b,z} + \Sigma_w)^{-1} (\bar{y}_j - \mu_y) (z_j - \mu_z)^T \right) \\
+ n_j \text{tr} \left( (n_j \Sigma_{b,z} + \Sigma_w)^{-1} (\bar{y}_j - \mu_y) (\bar{y}_j - \mu_y)^T \right) \right]
\]

where \( S_{pw} \) is the pooled within-clusters covariance matrix:

\[
S_{pw} = \frac{\sum_{j=1}^{J} \sum_{i=1}^{n_j} (y_{ji} - \bar{y}_j) (y_{ji} - \bar{y}_j)^T}{N - J}
\]
optimization techniques for two-level SEM (optional)

- quasi-newton methods (lavaan, 0.6-1)
- Fisher scoring, Gauss-Newton (LISREL)
- Expectation-Maximization (EM) (Mplus, EQS)
  - many variants exist
- hybrid optimization schemes (EM + quasi-newton)
Example: Mplus ex9.6 (simulated data)

- data: 110 clusters, 1000 observations, cluster sizes: 5, 10, 15
- 4 measures at the within level $y_1, y_2, y_3, y_4$
- 2 covariates at the within level $x_1, x_2$
- 1 covariate at the between level $w$
- reading in the data:

```r
> names(Data) <- c("y1", "y2", "y3", "y4", "x1", "x2", "w", "clus")
> head(Data)

   y1     y2    y3     y4    x1  x2    w clus
1 2.20 1.86 1.74 2.24 1.14 -0.8 0.14  
2 1.93 2.13 0.08 2.51 1.94 -0.1 0.14  
3 0.32 0.97 -0.84 0.56 -0.7 0.76 -0.1  
4 0.07 -1.74 -2.31 -1.51 0.64 0.31 0.14  
5 -1.21 0.45 0.37 -1.79 -0.3 0.31 -0.1  
6 0.29 -1.82 0.56 -2.09 1.36 0.31 0.31  
```

Yves Rosseel
Multilevel Structural Equation Modeling with lavaan (part 1)
library(lavaan)

model <- '
  level: 1
    fw =~ y1 + y2 + y3 + y4
    fw ~ x1 + x2

  level: 2
    fb =~ y1 + y2 + y3 + y4
    # optional
    y1 ~ 0*y1
    y2 ~ 0*y2
    y3 ~ 0*y3
    y4 ~ 0*y4
    fb ~ w
',

fit <- sem(model, data = Data,
    cluster = "clus",
    fixed.x = FALSE)
> summary(fit)

lavaan (0.6–1.1156) converged normally after 27 iterations

Number of observations 1000
Number of clusters [clus] 110

Estimator ML
Model Fit Test Statistic 4.228
Degrees of freedom 17
P-value (Chi-square) 0.999

Parameter Estimates:

Information Observed
Standard Errors Standard

Level 1 [within]:

Latent Variables:

<table>
<thead>
<tr>
<th>fw =~</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>y1</td>
<td>1.000</td>
<td>0.033</td>
<td>30.735</td>
<td>0.000</td>
</tr>
<tr>
<td>y2</td>
<td>0.999</td>
<td>0.033</td>
<td>30.735</td>
<td>0.000</td>
</tr>
<tr>
<td>y3</td>
<td>0.995</td>
<td>0.033</td>
<td>29.804</td>
<td>0.000</td>
</tr>
<tr>
<td>y4</td>
<td>1.017</td>
<td>0.033</td>
<td>30.364</td>
<td>0.000</td>
</tr>
</tbody>
</table>
### Regressions:

|       | Estimate | Std.Err | z-value | P(>|z|) |
|-------|----------|---------|---------|---------|
| fw ~  |          |         |         |         |
| x1    | 0.973    | 0.042   | 23.287  | 0.000   |
| x2    | 0.510    | 0.038   | 13.422  | 0.000   |

### Covariances:

|       | Estimate | Std.Err | z-value | P(>|z|) |
|-------|----------|---------|---------|---------|
| x1    |          |         |         |         |
| x2    | 0.032    | 0.032   | 1.014   | 0.311   |

### Intercepts:

|       | Estimate | Std.Err | z-value | P(>|z|) |
|-------|----------|---------|---------|---------|
| .y1   | 0.000    |         |         |         |
| .y2   | 0.000    |         |         |         |
| .y3   | 0.000    |         |         |         |
| .y4   | 0.000    |         |         |         |
| x1    | 0.007    | 0.031   | 0.222   | 0.825   |
| x2    | 0.014    | 0.032   | 0.440   | 0.660   |
| .fw   | 0.000    |         |         |         |

### Variances:

|       | Estimate | Std.Err | z-value | P(>|z|) |
|-------|----------|---------|---------|---------|
| .y1   | 0.981    | 0.057   | 17.151  | 0.000   |
| .y2   | 0.948    | 0.056   | 17.015  | 0.000   |
| .y3   | 1.070    | 0.060   | 17.700  | 0.000   |
| .y4   | 1.014    | 0.059   | 17.182  | 0.000   |
| .fw   | 0.980    | 0.071   | 13.888  | 0.000   |
| Variable | Estimate | Std. Err | z-value | P(>|z|) |
|----------|----------|----------|---------|---------|
| x1       | 0.985    | 0.044    | 22.361  | 0.000   |
| x2       | 1.017    | 0.045    | 22.361  | 0.000   |

**Level 2 [clus]:**

**Latent Variables:**

| fb =˜ | Estimate | Std. Err | z-value | P(>|z|) |
|-------|----------|----------|---------|---------|
| y1    | 1.000    |          |         |         |
| y2    | 0.960    | 0.073    | 13.078  | 0.000   |
| y3    | 0.924    | 0.074    | 12.452  | 0.000   |
| y4    | 0.949    | 0.075    | 12.631  | 0.000   |

**Regressions:**

| fb ˜ | Estimate | Std. Err | z-value | P(>|z|) |
|------|----------|----------|---------|---------|
| w    | 0.344    | 0.078    | 4.429   | 0.000   |

**Intercepts:**

| .y1  | Estimate | Std. Err | z-value | P(>|z|) |
|------|----------|----------|---------|---------|
| .y2  | -0.083   | 0.074    | -1.128  | 0.259   |
| .y3  | -0.077   | 0.071    | -1.081  | 0.280   |
| .y4  | -0.045   | 0.071    | -0.637  | 0.524   |
| .w   | -0.030   | 0.072    | -0.418  | 0.676   |
| .fb  | 0.006    | 0.086    | 0.070   | 0.944   |
| .fb  | 0.000    |          |         |         |
### Variances:

|     | Estimate | Std.Err | z-value | P(>|z|) |
|-----|----------|---------|---------|---------|
| .y1 | 0.000    |         |         |         |
| .y2 | 0.000    |         |         |         |
| .y3 | 0.000    |         |         |         |
| .y4 | 0.000    |         |         |         |
| .fb | 0.361    | 0.078   | 4.643   | 0.000   |
| w   | 0.815    | 0.110   | 7.416   | 0.000   |

```r
> fitMeasures(fit)
```

<table>
<thead>
<tr>
<th></th>
<th>npar</th>
<th>fmin</th>
<th>chisq</th>
<th>df</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>26.000</td>
<td>3.913</td>
<td>4.228</td>
<td>17.000</td>
</tr>
<tr>
<td>pvalue</td>
<td>baseline.chisq</td>
<td>baseline.df</td>
<td>baseline.pvalue</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.999</td>
<td>3280.729</td>
<td>25.000</td>
<td>0.000</td>
</tr>
<tr>
<td>cfi</td>
<td>tli</td>
<td>nnfi</td>
<td>rfi</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.000</td>
<td>1.006</td>
<td>1.006</td>
<td>0.998</td>
</tr>
<tr>
<td>nfi</td>
<td>pnfi</td>
<td>ifi</td>
<td>rni</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.999</td>
<td>0.679</td>
<td>1.004</td>
<td>1.004</td>
</tr>
<tr>
<td>logl</td>
<td>unrestricted.logl</td>
<td>aic</td>
<td>bic</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-9527.429</td>
<td>-9525.315</td>
<td>19106.857</td>
<td>19234.459</td>
</tr>
<tr>
<td>ntotal</td>
<td>bic2</td>
<td>rmsea</td>
<td>rmsea.ci.lower</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1000.000</td>
<td>19151.882</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>rmsea.ci.upper</td>
<td>rmsea.pvalue</td>
<td>srmr</td>
<td>srmr_within</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.000</td>
<td>1.000</td>
<td>0.025</td>
<td>0.005</td>
</tr>
<tr>
<td>srmr_between</td>
<td>0.020</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```r
> lavInspect(fit, "h1")

$within
$within$cov
   y1  y2  y3  y4  x1  x2
y1 3.163
y2 2.185 3.124
y3 2.170 2.167 3.238
y4 2.201 2.217 2.207 3.257
x1 0.949 0.949 0.954 0.996 0.978
x2 0.566 0.555 0.535 0.552 0.030 1.022

$within$mean
   y1  y2  y3  y4  x1  x2
0.000 0.000 0.000 0.000 0.007 0.014

$clus
$clus$cov
   y1  y2  y3  y4  w
y1 0.503
y2 0.492 0.475
y3 0.469 0.455 0.423
y4 0.484 0.464 0.444 0.472
w  0.325 0.273 0.300 0.291 0.830

$clus$mean
   y1  y2  y3  y4  w
```
$-0.066$ $-0.059$ $-0.029$ $-0.011$ $0.007$

$> \text{lavInspect(fit, "implied")}$

$\text{\$within}$

$\text{\$within$cov}$

\[
\begin{array}{cccccc}
y1 & y2 & y3 & y4 & x1 & x2 \\
y1 & 3.190 & & & & \\
y2 & 2.207 & 3.152 & & & \\
y3 & 2.198 & 2.195 & 3.256 & & \\
y4 & 2.247 & 2.245 & 2.235 & 3.300 & \\
x1 & 0.975 & 0.974 & 0.970 & 0.992 & 0.985 \\
x2 & 0.550 & 0.550 & 0.548 & 0.560 & 0.032 & 1.017 \\
\end{array}
\]

$\text{\$within$mean}$

\[
\begin{array}{cccccc}
y1 & y2 & y3 & y4 & x1 & x2 \\
0.014 & 0.014 & 0.014 & 0.014 & 0.007 & 0.014 \\
\end{array}
\]

$\text{\$clus}$

$\text{\$clus$cov}$

\[
\begin{array}{cccc}
y1 & y2 & y3 & y4 & w \\
y1 & 0.458 & & & \\
y2 & 0.439 & 0.421 & & \\
y3 & 0.423 & 0.406 & 0.391 & \\
y4 & 0.434 & 0.417 & 0.401 & 0.412 & \\
w & 0.281 & 0.269 & 0.259 & 0.266 & 0.815 \\
\end{array}
\]
$clus$mean

 y1  y2  y3  y4    w 
-0.081 -0.075 -0.043 -0.028  0.006

> lavInspect(fit, "icc")

 y1  y2  y3  y4   x1   x2 
 0.137 0.132 0.115 0.127 0.000 0.000
4.6 The status of a latent variable in a two-level SEM

- when a latent variable, representing a hypothetical construct, is introduced in a two-level model, we need to carefully reflect on the ‘status’ of this latent variable
  - are the indicators measured at the within or the between level?
  - is the construct of (theoretical) interest at the within level, the between level, or both?
  - how can we interpret the ‘meaning’ of the construct at the within/between level?

- based on the answers on these questions, we need to create the latent variable in a different way at the within and/or the between level

- this is (still today) a big source of confusion (and bad practices) in the educational sciences
different types of latent variables

- we will discuss five different construct types:
  
  1. within-only construct
     
        - in this case, if we have no other level-2 variables, we may as well use a single-level SEM based on a pooled within-cluster covariance matrix
  
  2. between-only construct
  
  3. shared between-level construct
  
  4. configural (or contextual) construct
  
  5. shared and configural construct

- reference:

within-only construct

- indicators of the latent variable are measured at the within level
- level at which construct is of interest: within level only
- interpretation at the within level: construct explains the covariances between its indicators measured at the within level
- interpretation at the between level: not relevant
- although the construct only ‘exists’ at the within level, we may still observe ‘spurious’ between-level variation in the sample
- example: construct represents ‘lactose intolerance’
  - items inquire about the degree of severity of physical reactions after consuming products containing lactose
  - construct can not be a school-level characteristic, although we may observe differences (on average) across schools
diagram and lavaan syntax

\[
\text{model} \leftarrow ' \\
\text{level: 1} \\
fw = \sim y1 + y2 + y3 + y4 \\
\text{level: 2} \\
y1 \sim y1 + y2 + y3 + y4 \\
y2 \sim y2 + y2 + y3 \\
y3 \sim y3 + y4 \\
y4 \sim y4 '
\]

\[
\text{fw} = \sim y1 + y2 + y3 + y4 \\
\text{level: 2} \\
y1 \sim y1 + y2 + y3 + y4 \\
y2 \sim y2 + y2 + y3 \\
y3 \sim y3 + y4 \\
y4 \sim y4 
\]
between-only construct

• indicators of the latent variable are measured at the between level

• level at which construct is of interest: between level only

• interpretation at the within level: not relevant (does not ‘exist’ at the within level)

• interpretation at the between level: construct explains the covariances between its indicators measured at the between level

• example: construct reflects self-reported ‘leadership style’ measured by a questionnaire filled in by the school principles
diagram and lavaan syntax

\[
\text{model} \leftarrow \'
\begin{align*}
\text{level: 1} \\
# \text{perhaps other level-1 variables} \\
\text{level: 2} \\
\text{fb} = ~ y_1 + y_2 + y_3 + y_4
\end{align*}
\]
shared (or reflective) between-level construct

- indicators of the latent variable are measured at the within level
- level at which construct is of interest: between level only
- interpretation at the within level: none
- interpretation at the between level: construct represents a characteristic of the cluster
- example: construct reflects ‘instructional quality’ (a classroom characteristic) as perceived by students
  - each student in each classroom was asked to judge the ‘instructional quality’ of the teacher of that classroom
  - we are interested in the ‘average’ responses of the individual students within each classroom
  - responses within each classroom should be highly correlated (high agreement) if indeed ‘instructional quality’ is a shared construct
diagram and lavaan syntax

\[
\begin{align*}
\text{level: 1} \\
y_1 & \sim y_1 + y_2 + y_3 + y_4 \\
y_2 & \sim y_2 + y_2 + y_3 \\
y_3 & \sim y_3 + y_4 \\
y_4 & \sim y_4 \\
\text{level: 2} \\
fs & = y_1 + y_2 + y_3 + y_4
\end{align*}
\]
configural (or formative) construct

- indicators of the latent variable are measured at the within level
- level at which construct is of interest: both within and between level
- interpretation at the within/between level: construct explains the covariances of the within/between part of its indicators
- the configural construct (at the between level) represents the aggregate of the measurements of individuals within a cluster
- example: reading motivation:
  - at the individual level (within cluster)
  - at the school level (average student motivation within a school)
- the cluster itself is not seen as the source/reason for variability of an individual construct
- therefore, between-cluster loadings are fixed to be the same as within-cluster loadings (cross-level measurement invariance)
**diagram and lavaan syntax**

```r
model <- '
  level: 1
    fw =~ a*y1 + b*y2 + c*y3 + d*y4
  level: 2
    fb =~ a*y1 + b*y2 + c*y3 + d*y4
',
```

---

Yves Rosseel

Multilevel Structural Equation Modeling with lavaan (part 1)
shared + configural construct

• indicators of the latent variable are measured at the within level

• level at which construct is of interest: within and between level

• interpretation at the within level: construct explains the covariances of the within part of its indicators

• interpretation at the between level: both the configural construct and the shared construct explain the covariances of the within/between part of its indicators

• example: reading motivation for each child in a classroom is rated by the classroom teacher (using multiple items)
  – some teachers tend to rate more positively as compared to others
  – the ‘shared’ construct reflects the rater effect
  – the ‘configural’ construct reflects the average reading motivation in a classroom
diagram and lavaan syntax

model <- '  
  level: 1  
    fw = a*y1 + b*y2 + c*y3 + d*y4  
  level: 2  
    fb = a*y1 + b*y2 + c*y3 + d*y4  
    fs = y1 + y2 + y3 + y4  
    # fb and fs must be orthogonal  
    fs ~ 0*fb  
  ,
4.7 The status of observed covariates in a two-level SEM

- when observed covariates are added in a two-level model, we again need to carefully reflect on the ‘status’ of these covariates
  - are the covariates measured at the within or the between level?
  - if they are measured at the within level, does it make sense to split this covariate into a within and a between part?

- based on the answers on these questions, we can make a distinction between three types of covariates:
  1. within-only covariates
  2. between-only covariates
  3. level-1 covariates with a within and a between part
adding a within-only covariate

model <- '  
  level: 1  
  \( fw = \sim a*y1 + b*y2 + c*y3 + d*y4 \)  
  \( fw \sim x1 \)  
  level: 2  
  \( fb = \sim a*y1 + b*y2 + c*y3 + d*y4 \)  
,'
adding a between-only covariate

\[
\text{model} \leftarrow ' \\
\text{level: 1} \\
\quad \text{fw} \sim a \times y_1 + b \times y_2 + c \times y_3 + d \times y_4 \\
\text{level: 2} \\
\quad \text{fb} \sim a \times y_1 + b \times y_2 + c \times y_3 + d \times y_4 \\
\quad \text{fb} \sim z_1
\]
adding a level-1 covariate with a within and a between part

```r
model <- ' 
  level: 1
    fw =~ a*y1 + b*y2 + c*y3 + d*y4
    fw ~ x1
  level: 2
    fb =~ a*y1 + b*y2 + c*y3 + d*y4
    fb ~ x1',
```

Yves Rosseel

Multilevel Structural Equation Modeling with lavaan (part 1)
adding a level-1 covariate with a within and a between part (2)

• decomposition of a level-1 covariate (say, $x_1$) into its within part and between part:
  
  – the level-1 covariate is centered using cluster/group-mean centering
  
  – the cluster/group means are treated as (unknown, latent) population parameters that need to be estimated
  
  – this implies that we assume a random intercept for the level-1 covariate

• note that this is not the same as creating aggregated versions of the level-1 covariates manually, and adding them to the datafile so they can be used in the between part of the model, see:

4.8 Evaluating model fit

- if no random slopes are involved, we can fit an unrestricted (saturated) model: we estimate all the elements of $\Sigma_W$, $\Sigma_B$ and $\mu_B$

- then, we can compute the standard $\chi^2$ goodness-of-fit test statistic as:

$$T = -2(L_0 - L_1)$$

where $L_0$ and $L_1$ are the loglikelihood of the restricted (user-specified) model (h0) and the unrestricted model (h1) respectively

- under various optimal conditions, this statistic follows a chi-square distribution

- the degrees of freedom are computed as in a two-group SEM model: the difference between the number of (non-redundant) sample statistics for each level, and the number of free model parameters

- in principle, fit measures like CFI/TLI, RMSEA, SRMR, ... can be computed in a similar way as in a single-level SEM
evaluating fit (2)

• unfortunately, a recent simulation study showed that CFI, TLI, and RMSEA were not sensitive to Level-2 model misspecification:


• there seems to be a growing sentiment that ‘global’ fit indices may not be very useful in a multilevel setting

• an alternative approach is to assess the fit per level:
  – we could compute the SRMR for each level
  – we could fit a single-level model separately for each level, and look at the traditional fit measures to judge the model fit for that level
4.9 Example: two-level CFA

- we use an example from this book (Chapter 14):
  

- the (simulated) data are the scores on six intelligence measures of 399 children from 60 (large) families, patterned after a real dataset collected by Van Peet, A.A.J. (1992)

- the six intelligence measures are: word list, cards, matrices, figures, animals, and occupations

- the data have a two-level structure, with children nested within families

- if intelligence is strongly influenced by shared genetic and environmental influences in the families, we may expect strong between-family effects

- the ICCs of the 6 measures range from 0.36 to 0.49
exploring the data

> FamIQData <- read.table("FamIQData.dat")
> names(FamIQData) <- c("family", "child", "wordlist", "cards", "matrices", +    "figures", "animals", "occupats")
> summary(FamIQData)

<table>
<thead>
<tr>
<th></th>
<th>family</th>
<th>child</th>
<th>wordlist</th>
<th>cards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>1.00</td>
<td>1.00</td>
<td>12.00</td>
<td>11.00</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>16.00</td>
<td>2.00</td>
<td>27.00</td>
<td>26.50</td>
</tr>
<tr>
<td>Median</td>
<td>33.00</td>
<td>4.00</td>
<td>30.00</td>
<td>30.00</td>
</tr>
<tr>
<td>Mean</td>
<td>31.78</td>
<td>4.04</td>
<td>29.95</td>
<td>29.84</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>48.00</td>
<td>6.00</td>
<td>33.00</td>
<td>33.00</td>
</tr>
<tr>
<td>Max.</td>
<td>60.00</td>
<td>12.00</td>
<td>45.00</td>
<td>44.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>matrices</th>
<th>figures</th>
<th>animals</th>
<th>occupats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min.</td>
<td>15.00</td>
<td>17.00</td>
<td>15.00</td>
<td>15.00</td>
</tr>
<tr>
<td>1st Qu.</td>
<td>26.00</td>
<td>27.00</td>
<td>27.00</td>
<td>27.00</td>
</tr>
<tr>
<td>Median</td>
<td>30.00</td>
<td>30.00</td>
<td>30.00</td>
<td>30.00</td>
</tr>
<tr>
<td>Mean</td>
<td>29.73</td>
<td>30.08</td>
<td>30.11</td>
<td>30.01</td>
</tr>
<tr>
<td>3rd Qu.</td>
<td>33.00</td>
<td>33.00</td>
<td>34.00</td>
<td>33.00</td>
</tr>
<tr>
<td>Max.</td>
<td>46.00</td>
<td>44.00</td>
<td>46.00</td>
<td>43.00</td>
</tr>
</tbody>
</table>

> # various cluster sizes
> table(table(FamIQData$family))

4 5 6 7 8 9 10 11 12
3 16 11 12 11 4 1 1 1
analytic procedure

• fitting a two-level model is often a stepwise procedure; below are the steps used by Joop Hox

• model 0: as a preliminary step, an EFA was carried out on the pooled within-clusters covariance matrix $S_{PW}$
  – it was concluded that a 2-factor model fitted well at the within level
  – not shown here

• model 1: a two-factor model at the within level + a null model at the between level
  – a null model implies: zero variances and covariances for all (6) variables
  – if this model fits well, we would conclude that there is no between family structure at all: we may as well continue with a single-level analysis
• model 2: a two-factor model at the within level + an independence model at the between level
  – independence model implies: estimated variances but zero covariances
  – if this model holds, there is family-level variance, but no substantively interesting structural model

• model 3: a two-factor model at the within level + a saturated model at the between level
  – the factors at the within-level in this model correspond to what we have called ‘within-only’ constructs

• models 4a and 4b: in his book, Joop Hox goes on and fits a model with a one-factor model for the between part (4a), and a model with a two-factor model for the between part (4b)
  – the two-factor model seems no improvement over the one-factor model
  – model 4a (with a general factor at the between level) is kept as the final model
model 1: a 2-factor within model + null between model
lavaan syntax

```r
> model1 <- ' level: 1
+ numeric =~ wordlist + cards + matrices
+ perception =~ figures + animals + occupats
+ level: 2
+ wordlist ~~ 0*wordlist
+ cards ~~ 0*cards
+ matrices ~~ 0*matrices
+ figures ~~ 0*figures
+ animals ~~ 0*animals
+ occupats ~~ 0*occupats
+ ' 
> fit1 <- sem(model1, data = FamIQData, cluster = "family",
+ std.lv = TRUE, verbose = TRUE)
> summary(fit1)
```

partial R output

```
lavaan (0.6-1.1120) converged normally after 74 iterations

  Number of observations  399

  Estimator  ML
  Model Fit Test Statistic  323.428
  Degrees of freedom  29
  P-value (Chi-square)  0.000
```
### Level 1 [within]:

#### Latent Variables:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| numeric =~ |
| wordlist | 4.346   | 0.217   | 20.037  | 0.000   |
| cards    | 4.286   | 0.212   | 20.194  | 0.000   |
| matrices | 4.000   | 0.213   | 18.748  | 0.000   |
| perception =~ |
| figures  | 4.104   | 0.220   | 18.651  | 0.000   |
| animals  | 4.446   | 0.207   | 21.465  | 0.000   |
| occupats | 4.298   | 0.212   | 20.272  | 0.000   |

#### Covariances:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| numeric ~~ |
| perception | 0.676 | 0.034 | 19.770 | 0.000 |

### Level 2 [family]:

#### Variances:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| .wordlist | 0.000   |         |         |
| .cards   | 0.000   |         |         |
| .matrices | 0.000  |         |         |
| .figures | 0.000   |         |         |
| .animals | 0.000   |         |         |
| .occupats | 0.000  |         |         |
model 2: a 2-factor within model + independence between model

```
  wordlist  cards  matrices  figures  animals  occupats
    ↓        ↓        ↓        ↓        ↓        ↓
    Between

  wordlist  cards  matrices  figures  animals  occupats
    ↓        ↓        ↓        ↓        ↓        ↓
    Within

  numeric
```

```
  perception
  ↓
  numeric
```
lavaan syntax model 2

> model2 <- '  
+   level: 1  
+     numeric ~ wordlist + cards + matrices  
+     perception ~ figures + animals + occupats  
+   level: 2  
+     wordlist ~ wordlist  
+     cards ~ cards  
+     matrices ~ matrices  
+     figures ~ figures  
+     animals ~ animals  
+     occupats ~ occupats  
+   '

> fit2 <- sem(model2, data = FamIQData, cluster = "family",  
+   std.lv = TRUE, verbose = TRUE)  
> summary(fit2)

partial R output

lavaan (0.6-1.1120) converged normally after 63 iterations

Number of observations 399

Estimator ML
Model Fit Test Statistic 176.985
Degrees of freedom 23
P-value (Chi-square) 0.000
Level 1 [within]:

Latent Variables:

|          | Estimate | Std.Err | z-value | P(>|z|) |
|----------|----------|---------|---------|---------|
| numeric =~ |          |         |         |         |
| wordlist | 4.148    | 0.235   | 17.636  | 0.000   |
| cards    | 4.081    | 0.228   | 17.878  | 0.000   |
| matrices | 3.848    | 0.226   | 17.050  | 0.000   |
| perception =~ |          |         |         |         |
| figures  | 3.984    | 0.235   | 16.970  | 0.000   |
| animals  | 4.265    | 0.233   | 18.269  | 0.000   |
| occupats | 3.785    | 0.235   | 16.091  | 0.000   |

Covariances:

|          | Estimate | Std.Err | z-value | P(>|z|) |
|----------|----------|---------|---------|---------|
| numeric ~~ |          |         |         |         |
| perception | 0.644    | 0.038   | 17.023  | 0.000   |

Level 2 [family]:

Variances:

|           | Estimate | Std.Err | z-value | P(>|z|) |
|-----------|----------|---------|---------|---------|
| .wordlist | 1.557    | 0.649   | 2.398   | 0.016   |
| .cards    | 1.792    | 0.673   | 2.662   | 0.008   |
| .matrices | 2.036    | 0.696   | 2.925   | 0.003   |
| .figures  | 2.198    | 0.748   | 2.939   | 0.003   |
| .animals  | 1.148    | 0.608   | 1.888   | 0.059   |
| .occupats | 2.870    | 0.879   | 3.265   | 0.001   |
model 3: a 2-factor within model, with saturated between part

![Diagram showing a 2-factor within model with saturated between part. The variables include wordlist, cards, matrices, figures, animals, and occupats. The diagram is divided into between and within parts, with numeric and perception nodes connecting the variables.]
lavaan syntax model 3

```r
> model3 <- ' 
+   level: 1 
+   numeric =~ wordlist + cards + matrices 
+   perception =~ figures + animals + occupats 
+   level: 2 
+   # saturated 
+   wordlist =~ cards + matrices + figures + animals + occupats 
+   cards =~ matrices + figures + animals + occupats 
+   matrices =~ figures + animals + occupats 
+   figures =~ animals + occupats 
+   animals =~ occupats
+ ' 
> fit3 <- sem(model3, data = FamIQData, cluster = "family", 
+      std.lv = TRUE, verbose = TRUE) 
> summary(fit3)
```

partial R output

```
lavaan (0.6-1.1120) converged normally after 145 iterations

  Number of observations      399

  Estimator                   ML
  Model Fit Test Statistic    6.495
  Degrees of freedom          8
  P-value (Chi-square)         0.592
```
Level 1 [within]:

Latent Variables:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| numeric = |         |         |         |
| wordlist  | 3.155   | 0.203   | 15.555  | 0.000   |
| cards     | 3.156   | 0.196   | 16.125  | 0.000   |
| matrices  | 3.032   | 0.200   | 15.196  | 0.000   |
| perception = |       |         |         |
| figures   | 3.091   | 0.205   | 15.063  | 0.000   |
| animals   | 3.192   | 0.195   | 16.403  | 0.000   |
| occupats  | 2.774   | 0.183   | 15.141  | 0.000   |

Covariances:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| numeric ~ |         |         |         |
| perception | 0.386  | 0.058   | 6.690   | 0.000   |

Level 2 [family]:

Covariances:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| .wordlist ~ |         |         |         |
| .cards    | 9.272   | 2.422   | 3.827   | 0.000   |
| .matrices | 8.515   | 2.267   | 3.757   | 0.000   |
| .figures  | 8.410   | 2.188   | 3.843   | 0.000   |
| .animals  | 9.700   | 2.326   | 4.170   | 0.000   |

[...]
model 4a: a 2-factor within model + general factor between

![Diagram showing the model 4a structure with factors and variables.]
lavaan syntax model 4a

```r
> model4a <- '
+   level: 1
+       numeric =~ wordlist + cards + matrices
+     perception =~ figures + animals + occupats
+   level: 2
+     general =~ wordlist + cards + matrices +
+     figures + animals + occupats
+ '
> fit4a <- sem(model4a, data = FamIQData, cluster = "family",
+     std.lv = TRUE, verbose = TRUE)
> summary(fit4a)
```

partial R output

```
lavaan (0.6-1.1120) converged normally after 82 iterations

Number of observations 399

Estimator ML
Model Fit Test Statistic 11.706
Degrees of freedom 17
P-value (Chi-square) 0.818

Level 1 [within]:

Latent Variables:
```

Yves Rosseel
Multilevel Structural Equation Modeling with lavaan (part 1)
| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| numeric =~
| wordlist | 3.175   | 0.202   | 15.716  | 0.000   |
| cards    | 3.144   | 0.194   | 16.184  | 0.000   |
| matrices | 3.054   | 0.199   | 15.354  | 0.000   |
| perception =~
| figures  | 3.095   | 0.204   | 15.162  | 0.000   |
| animals  | 3.188   | 0.194   | 16.438  | 0.000   |
| occupats | 2.782   | 0.183   | 15.216  | 0.000   |

Covariances:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| numeric ~ perception | 0.382 | 0.057   | 6.738   | 0.000 |

Level 2 [family]:

Latent Variables:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| general =~
| wordlist | 3.057   | 0.393   | 7.786   | 0.000 |
| cards    | 3.054   | 0.389   | 7.844   | 0.000 |
| matrices | 2.632   | 0.381   | 6.910   | 0.000 |
| figures  | 2.806   | 0.398   | 7.042   | 0.000 |
| animals  | 3.204   | 0.383   | 8.366   | 0.000 |
| occupats | 3.439   | 0.415   | 8.296   | 0.000 |

[...]
4.10 Example: two-level SEM

• we use an example from this book (Chapter 15):


• based on a study by Schijf and Dronker (1991): they collected data from 1559 pupils (1382 after listwise deletion) in 58 schools

• pupil variables: father’s occupational status (focc), father’s education (feduc), mother’s education (meduc), the result of the GALO school achievement test (galo), and the teacher’s advice about secondary education (advice)

• at the school level, we have one variable: the school’s denomination (denom) coded as 1=Protestant, 2=Nondenominational, 3=Catholic

• the main research question is whether the school’s denomination affects the GALO score and the teacher’s advice, after the other variables have been accounted for
modeling strategy

- a latent variable is constructed to reflect the socio-economic status (ses) using the variables focc, meduc and feduc as indicators
  - we will construct a configural latent variable for ses at the between level (using equality constraints for the loadings)

- preliminary analysis (using the pooled within-clusters covariance matrix only) revealed that a residual correlation is needed between the indicators focc and feduc at the within level

- a secondary question is whether the effect of ses on advice is direct or indirect
  - we label the various regression paths, and compute product terms to compute the indirect effect
  - both at the within and the between level
**the model**

![Diagram of the model]

**Between**

**Within**

---

Yves Rosseel

Multilevel Structural Equation Modeling with lavaan (part 1)
exploring the data

```r
> Galo <- read.table("Galo.dat")
> names(Galo) <- c("school", "sex", "galo", "advice", "feduc", "meduc", +     "focc", "denom")
> Galo[Galo == 999] <- NA
> summary(Galo)

> table(table(Galo$school))
```

```
  school  sex  galo  advice  
   Min. :1.00 Min. :1.000 Min. : 53.0 Min. :0.000
  1st Qu.:16.00 1st Qu.:1.000 1st Qu.: 94.0 1st Qu.:2.000
  Median :30.00 Median :2.000 Median :103.0 Median :2.000
  Mean :29.87 Mean :1.509 Mean :102.3 Mean :3.121
  3rd Qu.:43.00 3rd Qu.:2.000 3rd Qu.:111.0 3rd Qu.:4.000
  Max. :58.00 Max. :2.000 Max. :143.0 Max. :6.000
  NA's :7

  feduc  meduc  focc  denom  
   Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000
  1st Qu.:1.000 1st Qu.:1.000 1st Qu.:2.000 1st Qu.:2.000
  Median :4.000 Median :2.000 Median :3.000 Median :2.000
  Mean :4.002 Mean :2.966 Mean :3.336 Mean :2.007
  3rd Qu.:6.000 3rd Qu.:5.000 3rd Qu.:5.000 3rd Qu.:2.000
  Max. :9.000 Max. :9.000 Max. :6.000 Max. :3.000
  NA's :89 NA's :61 NA's :117

> table(table(Galo$school))

10 12 13 14 19 20 21 22 23 24 25 26 27 28 29 30 32 33 34 35 36 37 42 46
  1 2 1 3 1 2 3 3 1 6 3 3 4 2 1 4 1 4 5 1 2 2 1 2
```
lavaan syntax

> model <- ' 
+   level: within 
+   wses ˜ a*focc + b*meduc + c*feduc 
+   # residual correlation 
+   focc ˜˜ feduc 
+ 
+   advice ˜ wc*wses + wb*galo 
+   galo ˜ wa*wses 
+ 
+   level: between 
+   bses ˜ a*focc + b*meduc + c*feduc 
+   feduc ˜˜ 0*feduc 
+ 
+   advice ˜ bc*bses + bb*galo 
+   galo ˜ ba*bses + denom 
+ 
+   # defined parameters 
+   wi := wa * wb 
+   bi := ba * bb 
+ ' 
> fit <- sem(model, data = Galo, cluster = "school", fixed.x = FALSE, 
+             verbose = TRUE, std.lv = TRUE, h1 = TRUE) 
> summary(fit)
partial R output

lavaan (0.6-1.1120) converged normally after 93 iterations

<table>
<thead>
<tr>
<th>Used</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1382</td>
<td>1559</td>
</tr>
</tbody>
</table>

Estimator: ML
Model Fit Test Statistic: 23.855
Degrees of freedom: 15
P-value (Chi-square): 0.068

[...]
Level 1 [within]:

Latent Variables:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| wses =~   |         |         |         |
| focc      | 0.748   | 0.038   | 19.489  | 0.000   |
| meduc     | 1.282   | 0.047   | 27.566  | 0.000   |
| feduc     | 1.674   | 0.057   | 29.163  | 0.000   |

Regressions:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| advice ~  |         |         |         |
| wses      | 0.120   | 0.027   | 4.492   | 0.000   |
| galo      | 0.086   | 0.002   | 44.614  | 0.000   |
| galo ~    |         |         |         |
| wses      | 4.202   | 0.370   | 11.352  | 0.000   |
Level 2 [school]:

Latent Variables:

|  | Estimate | Std.Err | z-value | P(>|z|) |
|---|----------|---------|---------|---------|
| bses =~ | | | | |
| focc (a) | 0.748 | 0.038 | 19.489 | 0.000 |
| meduc (b) | 1.282 | 0.047 | 27.566 | 0.000 |
| feduc (c) | 1.674 | 0.057 | 29.163 | 0.000 |

Regressions:

|  | Estimate | Std.Err | z-value | P(>|z|) |
|---|----------|---------|---------|---------|
| advice ~ | | | | |
| bses (bc) | 0.267 | 0.070 | 3.822 | 0.000 |
| galo (bb) | 0.063 | 0.011 | 5.524 | 0.000 |
| galo ~ | | | | |
| bses (ba) | 4.945 | 0.608 | 8.137 | 0.000 |
| denom | 2.648 | 0.823 | 3.218 | 0.001 |

Defined Parameters:

|  | Estimate | Std.Err | z-value | P(>|z|) |
|---|----------|---------|---------|---------|
| wi | 0.360 | 0.032 | 11.181 | 0.000 |
| bi | 0.314 | 0.067 | 4.660 | 0.000 |
5 Alternative ways to analyze multilevel data with SEM

• some alternative ways to analyze multilevel data with SEM:

1. the ‘wide data’ approach: we arrange data in the wide format, and then use single-level SEM to analyze our model

2. the ‘survey’ approach: we analyze the data (in long format) as if there where no clusters, but we use cluster-robust standard errors

3. the two-stage approach: multilevel software (e.g., MLwiN) is used to estimate the (saturated) within and between covariance matrix; analysis by multigroup SEM (Goldstein, 1987)

4. the pseudo-balanced approach: we pretend the data is balanced, and use a special estimator to fit a multigroup SEM (MUML)

5. …
why should you know about these alternatives?

- they may enhance your understanding of:
  - SEM
  - multilevel regression
  - multilevel SEM
  - the relationships between the different modeling frameworks

- depending on your data, model and research questions, they may be easier to set up, have less convergence problems, and the results may be easier to interpret and report

- in some cases, they may save the day
5.1 The ‘wide data’ approach

• wonderful paper about this:


• first approach: using classic SEM to mimic multilevel regression models
  – the random intercepts and random slopes are represented by latent variables
  – the factor loadings of the random intercept are fixed to 1.0
  – the factor loadings of the random slope are fixed to the values of the predictor
  – only feasible if the predictor has a limited number of possible values (e.g. binary, or timepoint 1, 2, 3, or 4)
  – most importantly: only if the values for the predictor are the same for all units (‘balanced design’)

Yves Rosseel
Multilevel Structural Equation Modeling with lavaan (part 1)
– typical example: growth curve model
– advantage: single-level analysis, model fit (although care is needed to specify the saturated model), flexible error structure, ...

• second approach: calculate a model-implied covariance matrix (and mean vector) for each individual
  – needs special software (like OpenMx or Mplus)
  – predictor can be continuous, design does not need to be balanced

• because we are in the SEM context, we can extend these approaches to include latent variables, mediators, ...

• can be useful if:
  – the cluster sizes are (very) small
  – the number of variables (per unit) is relatively small
  – the data is (almost) balanced
  – the wide data still has many more rows \( N \) then columns \( P \)
example: a growth curve model with 4 time-points

- random intercept and random slope

\[ y_t = \text{(initial time at time 1)} + \text{(growth per unit time)} \times \text{time} + \text{error} \]

\[ y_t = \text{intercept} + \text{slope} \times \text{time} + \text{error} \]
R code: using SEM in wide format

```r
> library(lavaan)
> head(Demo.growth[,c("t1","t2","t3","t4")], n = 4)

   t1    t2    t3    t4
1 1.73  2.14  2.77  2.52
2-1.98-4.40-6.02-7.03
3 0.32-1.27 1.56 2.87
4 0.78  3.53  3.14  5.36

> model.slope <- ' + int =~ 1* t1 + 1* t2 + 1* t3 + 1* t4
+ slope =~ 0* t1 + 1* t2 + 2* t3 + 3* t4
+
+ # intercepts (fixed effects)
+ int  ~ 1
+ slope ~ 1
+
+ # random intercept, random slope
+ int  ~ int
+ slope ~ slope
+ int  ~ slope
+
+ # force same variance for all (compound symmetry)
+ t1 ~ v1*t1
+ t2 ~ v1*t2
```
+ t3 ~ v1*t3
+ t4 ~ v1*t4
+
> fit.slope <- lavaan(model.slope, data = Demo.growth)
> summary(fit.slope)

lavaan (0.6-1.1156) converged normally after 24 iterations

Number of observations                       400

Estimator                  ML
Model Fit Test Statistic    9.678
Degrees of freedom          8
P-value (Chi-square)         0.288

Parameter Estimates:

Information        Expected
Standard Errors     Standard

Latent Variables:

  Estimate  Std.Err  z-value  P(|z|)
int =~
  t1       1.000
  t2       1.000
  t3       1.000
  t4       1.000
slope =~
|   | Estimate | Std.Err | z-value | P(>|z|) |
|---|----------|---------|---------|---------|
| int ~ slope | 0.627 | 0.069 | 9.129 | 0.000 |

**Intercepts:**

|   | Estimate | Std.Err | z-value | P(>|z|) |
|---|----------|---------|---------|---------|
| int | 0.617 | 0.077 | 8.029 | 0.000 |
| slope | 1.005 | 0.042 | 24.013 | 0.000 |
| t1 | 0.000 | 0.000 | 0.000 |
| t2 | 0.000 | 0.000 | 0.000 |
| t3 | 0.000 | 0.000 | 0.000 |
| t4 | 0.000 | 0.000 | 0.000 |

**Variances:**

|   | Estimate | Std.Err | z-value | P(>|z|) |
|---|----------|---------|---------|---------|
| int | 1.928 | 0.169 | 11.439 | 0.000 |
| slope | 0.576 | 0.050 | 11.540 | 0.000 |
| t1 (v1) | 0.622 | 0.031 | 20.000 | 0.000 |
| t2 (v1) | 0.622 | 0.031 | 20.000 | 0.000 |
| t3 (v1) | 0.622 | 0.031 | 20.000 | 0.000 |
| t4 (v1) | 0.622 | 0.031 | 20.000 | 0.000 |
**R code: using lmer**

```r
> # wide to long
> id <- rep(1:400, each = 4)
> score <- lav_matrix_vecr(Demo.growth[,1:4])
> time <- rep(0:3, times = 400)
> growth.long <- data.frame(id = id, score = score, time = time)
> head(growth.long)

```

<table>
<thead>
<tr>
<th>id</th>
<th>score</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.725645</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2.142401</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2.773172</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>2.515956</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>-1.984160</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>-4.400603</td>
<td>1</td>
</tr>
</tbody>
</table>

```r
> library(lme4)
> fit.lmer <- lmer(score ~ 1 + time + (1 + time | id), data = growth.long, + REML = FALSE)
> summary(fit.lmer, correlation = FALSE)
```

**Linear mixed model fit by maximum likelihood ['lmerMod']**

**Formula:** score ~ 1 + time + (1 + time | id)

**Data:** growth.long
AIC    BIC    logLik deviance df.resid
5523.7  5556.0  -2755.9   5511.7     1594

Scaled residuals:

   Min     1Q  Median     3Q    Max
-2.62395 -0.51865 -0.00867 0.51881 2.83705

Random effects:

   Groups    Name     Variance  Std.Dev.  Corr
      id   (Intercept)  1.9279    1.3885
         time         0.5765    0.7592  0.59
   Residual                       0.6223    0.7889
Number of obs: 1600, groups: id, 400

Fixed effects:

   Estimate Std. Error    t value
(Intercept)    0.61716     0.07687      8.029
  time         1.00519     0.04186     24.013
5.2 The ‘survey’ (design-based) approach

- literature:


- only if all variables (and constructs) are at the within-level only

- we treat the clustering as a (sampling) nuisance

- the parameter estimates are ‘aggregated’: they consistently estimate parameters aggregated over any clusters and strata and no explicit modeling of the effects of clusters and strata is involved

- standard errors are design-based

- allows for incorporation of clustering, stratification, unequal probability weights, finite population correction, and multiple imputation
setting up a two-factor model in lavaan with clustered data

> library(lavaan.survey)

> # step 1:
> # fit model ignoring clustering using estimator = "MLM"
> model <- '
+ numeric =~ wordlist + cards + matrices
+ perception =~ figures + animals + occupats
+
> fit.naive <- sem(model, data = FamIQData, std.lv = TRUE,
+ estimator = "MLM")

> # step 2:
> # create a survey design object with family clustering
> survey.design <- svydesign(ids = ~family, prob = ~1, data = FamIQData)

> # step 3:
> # refit, taking survey.design into account
> fit.survey <- lavaan.survey(lavaan.fit = fit.naive,
+ survey.design = survey.design)
> summary(fit.survey)
R output

lavaan (0.6-1.1120) converged normally after 33 iterations

Number of observations 399

Estimator          ML      Robust
Model Fit Test Statistic  9.786  7.671
Degrees of freedom     8      8
P-value (Chi-square)    0.280  0.466
Scaling correction factor
 for the Satorra-Bentler correction

Parameter Estimates:

Information Expected
Standard Errors Robust.sem

Latent Variables:

| Latent Variable | Estimate | Std.Err | z-value | P(>|z|) |
|-----------------|----------|---------|---------|---------|
| numeric =~      |          |         |         |         |
| wordlist        | 4.346    | 0.338   | 12.857  | 0.000   |
| cards           | 4.286    | 0.303   | 14.129  | 0.000   |
| matrices        | 4.000    | 0.263   | 15.229  | 0.000   |
| perception =~   |          |         |         |         |
| figures         | 4.104    | 0.259   | 15.861  | 0.000   |
| animals         | 4.446    | 0.254   | 17.479  | 0.000   |
| occupats        | 4.298    | 0.292   | 14.702  | 0.000   |
### Covariances:

|                | Estimate | Std.Err | z-value | P(>|z|) |
|----------------|----------|---------|---------|---------|
| numeric ~ perception | 0.676    | 0.041   | 16.384  | 0.000   |

### Intercepts:

|                | Estimate | Std.Err | z-value | P(>|z|) |
|----------------|----------|---------|---------|---------|
| .wordlist      | 29.947   | 0.450   | 66.587  | 0.000   |
| .cards         | 29.845   | 0.449   | 66.469  | 0.000   |
| .matrices      | 29.734   | 0.437   | 67.984  | 0.000   |
| .figures       | 30.075   | 0.458   | 65.704  | 0.000   |
| .animals       | 30.108   | 0.451   | 66.731  | 0.000   |
| .occupats      | 30.008   | 0.481   | 62.360  | 0.000   |
| numeric        | 0.000    |         |         |         |
| perception     | 0.000    |         |         |         |

### Variances:

|                | Estimate | Std.Err | z-value | P(>|z|) |
|----------------|----------|---------|---------|---------|
| .wordlist      | 7.262    | 0.899   | 8.076   | 0.000   |
| .cards         | 6.801    | 0.813   | 8.370   | 0.000   |
| .matrices      | 8.382    | 0.748   | 11.211  | 0.000   |
| .figures       | 9.260    | 0.840   | 11.020  | 0.000   |
| .animals       | 5.433    | 0.649   | 8.375   | 0.000   |
| .occupats      | 6.923    | 0.831   | 8.333   | 0.000   |
| numeric        | 1.000    |         |         |         |
| perception     | 1.000    |         |         |         |
5.3 MUthén ML (MUML) estimator using multiple group SEM

• $S_{PW}$ and $S_{B}^{*}$ are the input matrices for a two-group SEM analysis with $N - G$ and $G$ observations respectively.

• the within part of the model is specified both for the within ‘group’ and the between ‘group’ with equality constraints to ensure that the same within structure is estimated for both groups.

• the between part of the model is only specified for the between ‘group’ but with a scale factor of $\sqrt{s}$ hard-wired in the lambda matrix.

• although tricky to set up, this results in ML estimates for a two-level SEM model in the *balanced* case.

• if the data is unbalanced, we can still use this approach (called MUML in Mplus) as a pseudo-ML method; here, we replace $s$ by

$$ s^{*} = \left[ N^2 - \sum_{j=1}^{J} n_j^2 \right] / (N(J - 1)) $$
setting up model 4a using MUML in lavaan

```r
> Data <- read.table("FamIQData.dat")
> names(Data) <- c("family", "child", "wordlist", "cards", "matrices",
+ "figures", "animals", "occupats")
> ov.names <- c("wordlist","cards","matrices","figures","animals","occupats")
> # descriptives
> cluster.size <- as.integer(table(Data$family))
> nClusters <- length(cluster.size)
> N <- sum(cluster.size)
> # between sample statistics
> DataB <- with(Data, aggregate(Data[,ov.names],
+ by = list(family), FUN = mean))[,ov.names]
> # weighted mean
> B.mean <- colSums(DataB * cluster.size/N)
> # weighted cov
> DataBc <- as.matrix(sweep(DataB, 2, STATS=B.mean))
> B.cov <- crossprod(DataBc * cluster.size, DataBc) / (nClusters - 1)
> # note: Hox divides by nClusters, not nClusters - 1
> B.cov1 <- crossprod(DataBc * cluster.size, DataBc) / nClusters
> # within
> cluster.idx <- as.integer(as.factor(Data[,"family"]))
> DataW <- Data[,ov.names] - DataB[cluster.idx,]
> W.mean <- colMeans(DataW)
> W.cov <- cov(DataW) * (N - 1) / (N - nClusters)
> # s (average sample size)
> s <- (N^2 - sum(cluster.size^2)) / (N * (nClusters - 1))
> # sqrt(s)
```
> # 6.642708
> # B.cov.star
> B.cov.star <- (B.cov - W.cov)/s
> # intra-class correlations
> diag(B.cov.star)/(diag(B.cov.star) + diag(W.cov))
> # 'MULE' approach, creating 'phantom' latent variables for each
> # observed variable at the between level + scaling
> model <- '  +    # within model
>    numeric =~ c(11,11)*wordlist + c(12,12)*cards + c(13,13)*matrices
>    percept =~ c(14,14)*figures + c(15,15)*animals + c(16,16)*occupats
>    numeric =~ c(f1,f1)*numeric; percept =~ c(f2,f2)*percept
>    numeric =~ c(f12,f12)*percept
>    +
>    wordlist =~ c(r1,r1)*wordlist; cards =~ c(r2,r2)*cards
>    matrices =~ c(r3,r3)*matrices; occupats =~ c(r4,r4)*occupats
>    figures =~ c(r5,r5)*figures; animals =~ c(r6,r6)*animals
>    +
>    # between model
>    +
>    # scaling
>    bwordlist =~ c(0,6.642708)*wordlist
>    bcards =~ c(0,6.642708)*cards
>    bmatrices =~ c(0,6.642708)*matrices
>    boccupats =~ c(0,6.642708)*occupats
+ bfigures  =~ c(0, 6.642708) * figures
+ banimals  =~ c(0, 6.642708) * animals
+
+ # between model
+ bgeneral  =~ c(0, NA) * bwordlist + c(0, NA) * bcards + c(0, NA) * bmatrices +
+     c(0, NA) * bfigures + c(0, NA) * banimals + c(0, NA) * boccupats
+     bgeneral  =~ c(0, 1) * bgeneral
+
+ # residual variances between
+ bwordlist =~ c(0, NA) * bwordlist; bcards    =~ c(0, NA) * bcards
+     bmatrices =~ c(0, NA) * bmatrices; boccupats =~ c(0, NA) * boccupats
+     bfigures =~ c(0, NA) * bfigures; banimals =~ c(0, NA) * banimals
+
> fit <- lavaan(model, sample.cov = list(within = W.cov, between = B.cov),
+       sample.cov.rescale = FALSE, meanstructure = TRUE,
+       sample.nobs = list(N-nClusters, nClusters), std.lv = TRUE)
> summary(fit)
partial R output

lavaan (0.6–1.1120) converged normally after 56 iterations

Number of observations per group
within 339
between 60

Estimator ML
Model Fit Test Statistic 11.340
Degrees of freedom 29
P-value (Chi-square) 0.999

Chi-square for each group:
within 6.601
between 4.739

Group 1 [within]:

Latent Variables:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| numeric =~
| wordlist (11) | 3.168 | 0.201 | 15.764 | 0.000 |
| cards (12) | 3.147 | 0.194 | 16.247 | 0.000 |
| matrices (13) | 3.059 | 0.199 | 15.409 | 0.000 |
| percept =~
| figures (14) | 3.098 | 0.204 | 15.178 | 0.000 |
### Latent Variables:

|                  | Estimate | Std.Err | z-value | P(>|z|) |
|------------------|----------|---------|---------|---------|
| numeric =~        |          |         |         |         |
| wordlist (11)    | 3.168    | 0.201   | 15.764  | 0.000   |
| cards (12)       | 3.147    | 0.194   | 16.247  | 0.000   |
| matrices (13)    | 3.059    | 0.199   | 15.409  | 0.000   |
| percept =~       |          |         |         |         |
| figures (14)     | 3.098    | 0.204   | 15.178  | 0.000   |
| animals (15)     | 3.193    | 0.192   | 16.645  | 0.000   |
| occupats (16)    | 2.775    | 0.182   | 15.276  | 0.000   |
| bwordlist =~     |          |         |         |         |
| wordlist         | 6.643    |         |         |         |
| bcards =~        |          |         |         |         |
| cards            | 6.643    |         |         |         |
| bmatrices =~     |          |         |         |         |
matrices 6.643
boccupats =~
occupats 6.643
bfigures =~
figures 6.643
banimals =~
animals 6.643
bgeneral =~
bwordlist 1.145 0.148 7.759 0.000
bcards 1.153 0.147 7.848 0.000
bmatrices 1.008 0.145 6.979 0.000
bfigures 1.075 0.152 7.088 0.000
banimals 1.209 0.144 8.402 0.000
boccupats 1.287 0.155 8.291 0.000

[...]  

# note: scaling of between factor loadings is different  
# but the Z-values are very close compared to ML
6 Last slide

- be careful with a small number of clusters (may lead to biased results)


- topics not discussed in this (part of the) workshop:
  - construct reliability in the multilevel setting
  - random slopes, centering
  - categorical outcomes
  - missing data
  - the glamm framework

- when will lavaan 0.6 be officially released?