Multilevel Structural Equation Modeling with lavaan

Yves Rosseel
Department of Data Analysis
Ghent University

Zürich 2–3 November 2017

Contents

1 Multilevel regression

1.1 Brief overview

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

```
  cluster.id y1 m1 x1 y2 m2 x2 y3 m3 x3 schoolsize
1          1 16  4 60 28 36  6  4 22 12      large
2          2 24 14 10 18  6 20 38 28 22     medium
3          3 26  2  2 32  4  8  4  4 10     medium
4          4  4 36 14  2  2  0  8  8 10      small
5          5 14 10 16 28  2  4  8 22  6      small
6          6 24 20 16 42 18  2  2 28 18      large
7          7 22  0 14 32  6  2 18 18 10     medium
8          8  0  8 34 16 16 14  8 28 18      large
```

example long format

```
   cluster.id  y  m  x schoolsize
1           1 16  4 60      large
2           1 28 36  6      large
3           1  4 22 12      large
4           2 24 14 10     medium
5           2 18  6 20     medium
6           2 38 28 22     medium
7           3 26  2  2     medium
8           3 32  4  8     medium
9           3  4  4 10     medium
10          4  4 36 14      small
11          4  2  2  0      small
12          4  8  8 10      small
13          5 14 10 16      small
14          5 28  2  4      small
15          5  8 22  6      small
16          6 24 20 16      large
17          6 42 18  2      large
18          6  2 28 18      large
19          7 22  0 14     medium
20          7 32  6  2     medium
21          7 18 18 10     medium
22          8  0  8 34      large
23          8 16 16 14      large
24          8  8 28 18      large
```

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:
  - Goldstein, H. (1987). *Multilevel Statistical Models.* London: Edward Arnold. (Cfr. MLwiN software)
  - Raudenbush, S.W. & Bryk A.S. (1992) *Hierarchical Linear Models: Applications and Data Analysis Methods.* Thousand Oaks, Calif.: Sage. (Cfr. HLM software)
  - Hox, J. (1995). *Applied Multilevel Analysis.* Amsterdam: TT-Publikaties.
  - Kreft, I.G.G. & De Leeuw, J. (1998) *Introducing Multilevel Modeling.* Sage, London.
  - Snijders, T. & Bosker, R. (1999). *Multilevel Analysis: An Introduction to Basic and Advanced Multilevel Modeling.* Thousand Oaks, Calif.: Sage.

1.2 Example: the "High School and Beyond" data

- the following example is borrowed from Raudenbusch and Bryk (2001)
- the data are from the 1982 "High School and Beyond" survey, and pertain to 7185 U.S. high-school students from 160 schools (70 catholic, 90 public)
- these are the variables that we will use:
  - **school** an ordered factor designating the school that the student attends.
  - **ses** a numeric vector of socio-economic scores
  - **mAch** a numeric vector of Mathematics achievement scores
  - **meanses** a numeric vector of mean ses for the school
  - **sector** a factor with levels Public and Catholic
  - **cses** a numeric vector of centered ses values where the centering is with respect to the meanses for the school
- the aim of the analysis is to determine how students' math achievement scores are related to their family socioeconomic status

- but this relationship may very well vary among schools
- if there is indeed variation among schools, can we find any school characteristics that 'explain' this variation? the two school characteristics that we will use are:
  - **sector**: public school or Catholic school
  - **meanses**: the average SES of students in the school
- dataset is included in the R package 'mlmRev'

'

exploring the data

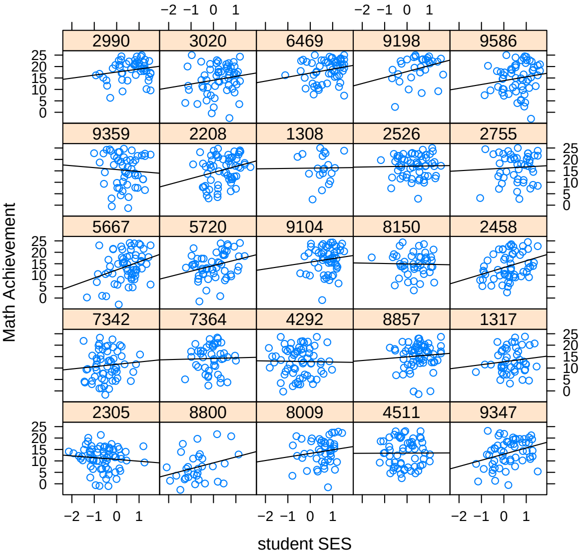
```
> library(mlmRev)
> summary(Hsb82)

    school      minrty          sx              ses                  mAch
 2305   :  67   No :5211   Male  :3390   Min.   :-3.758000   Min.   :-2.832
 5619   :  66   Yes:1974   Female:3795   1st Qu.:-0.538000   1st Qu.: 7.275
 4292   :  65                            Median : 0.002000   Median :13.131
 8857   :  64                            Mean   : 0.000143   Mean   :12.748
 4042   :  64                            3rd Qu.: 0.602000   3rd Qu.:18.317
 3610   :  64                            Max.   : 2.692000   Max.   :24.993
 (Other):6795
   meanses                sector          cses
 Min.   :-1.1939459   Public  :3642   Min.   :-3.6507
 1st Qu.:-0.3230000   Catholic:3543   1st Qu.:-0.4479
 Median : 0.0320000                   Median : 0.0160
 Mean   : 0.0001434                   Mean   : 0.0000
 3rd Qu.: 0.3269123                   3rd Qu.: 0.4694
 Max.   : 0.8249825                   Max.   : 2.8561
```

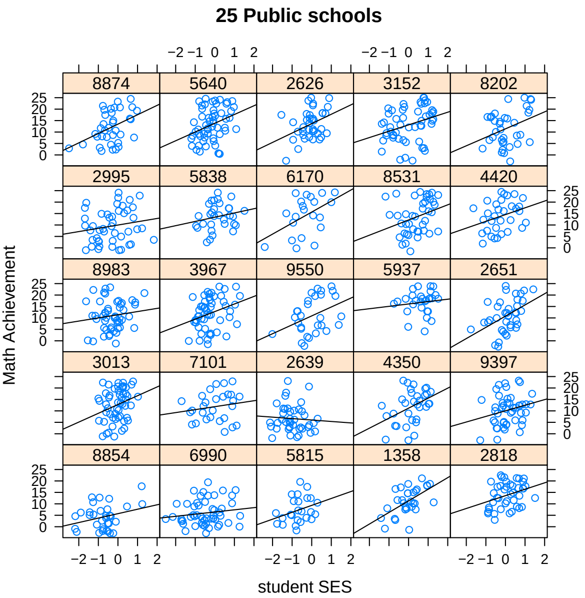
exploring the data (2)

```
> head(Hsb82, n = 8)

  school minrty     sx    ses   mAch   meanses sector        cses
1   1224     No Female -1.528  5.876 -0.434383 Public -1.09361702
2   1224     No Female -0.588 19.708 -0.434383 Public -0.15361702
3   1224     No   Male -0.528 20.349 -0.434383 Public -0.09361702
4   1224     No   Male -0.668  8.781 -0.434383 Public -0.23361702
5   1224     No   Male -0.158 17.898 -0.434383 Public  0.27638298
6   1224     No   Male  0.022  4.583 -0.434383 Public  0.45638298
7   1224     No Female -0.618 -2.832 -0.434383 Public -0.18361702
8   1224     No   Male -0.998  0.523 -0.434383 Public -0.56361702
```

```
> tail(Hsb82, n = 8)

     school minrty     sx    ses   mAch   meanses   sector       cses
7178   9586     No Female  1.212 15.260 0.6211525 Catholic  0.5908475
7179   9586     No Female  1.022 22.780 0.6211525 Catholic  0.4008475
7180   9586    Yes Female  1.612 20.967 0.6211525 Catholic  0.9908475
7181   9586     No Female  1.512 20.402 0.6211525 Catholic  0.8908475
7182   9586     No Female -0.038 14.794 0.6211525 Catholic -0.6591525
7183   9586     No Female  1.332 19.641 0.6211525 Catholic  0.7108475
7184   9586     No Female -0.008 16.241 0.6211525 Catholic -0.6291525
7185   9586     No Female  0.792 22.733 0.6211525 Catholic  0.1708475
```

25 Catholic schools




25 Public schools
8874
5640
2626
3152
8202
2995
5838
6170
8531
4420
8983
3967
9550
5937
2651
3013
7101
2639
4350
9397
8854
6990
5815
1358
2818
Math Achievement
student SES
−2 −1 0 1 2
0 5 10 15 20 25

model 1: a random-effects one-way ANOVA

- this is often called the 'empty' model, since it contains no predictors, but simply reflects the nested structure
- no level-1 predictors, no level-2 predictors
- in the 'multilevel' notation we specify a model for each level
- model for the first (student) level:

$$y_{ij} = \alpha_{0i} + \epsilon_{ij}$$

- model for the second (school) level:

$$\alpha_{0i} = \gamma_{00} + u_{0i}$$

- the combined model and the Laird-Ware form:

$$\begin{aligned} y_{ij} &= \gamma_{00} + u_{0i} + \epsilon_{ij} \\ &= \beta_0 + b_{0i} + \epsilon_{ij} \end{aligned}$$

- this is an example of a random-effects one-way ANOVA model with one fixed effect (the intercept, β_0) representing the general population mean of math achievement, and two random effects:
  - b_{0i} representing the deviation of math achievement in school i from the general mean
  - ϵ_{ij} representing the deviation of individual j's math achievement in school i from the school mean
- there are two variance components for this model:
  - $\text{Var}(b_{0i}) = d^2$: the variance among school means
  - $\text{Var}(\epsilon_{ij}) = \sigma^2$: the variance among individuals in the same school
- since b_{0i} and ϵ_{ij} are assumed to be independent, the variation in math scores among individuals can be decomposed into these two variance components:

$$\text{Var}(y_{ij}) = d^2 + \sigma^2$$

R code

```
> fit.model1 <- lmer(mAch ~ 1 + (1 | school), data = Hsb82, REML = FALSE)
> summary(fit.model1)

Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: mAch ~ 1 + (1 | school)
   Data: Hsb82

     AIC      BIC   logLik deviance df.resid
 47121.8  47142.4 -23557.9  47115.8     7182

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.06262 -0.75365  0.02676  0.76070  2.74184

Random effects:
 Groups   Name        Variance Std.Dev.
 school   (Intercept)  8.553   2.925
 Residual             39.148   6.257
Number of obs: 7185, groups:  school, 160

Fixed effects:
            Estimate Std. Error t value
(Intercept)  12.6371     0.2436   51.87
```

intra-class correlation

- the *intra-class correlation coefficient* is the proportion of variation in individuals' scores due to differences among schools:

$$\frac{d^2}{\mathsf{Var}(y_{ij})} = \frac{d^2}{d^2 + \sigma^2} = \rho$$

- ρ may also be interpreted as the correlation between the math scores of two individuals from the same school:

$$\mathsf{Cor}(y_{ij}, y_{ij'}) = \frac{\mathsf{Cov}(y_{ij}, y_{ij'})}{\sqrt{\mathsf{Var}(y_{ij}) \times \mathsf{Var}(y_{ij'})}} = \frac{d^2}{d^2 + \sigma^2} = \rho$$

```
> d2 <- as.numeric(VarCorr(fit.model1)$school)
> s <- as.numeric(attr(VarCorr(fit.model1), "sc"))
> rho <- d2/(d2 + s^2); rho

[1] 0.1793109
```

- about 18 percent of the variation in students' match-achievement scores is “attributable” to differences among schools

model 2: a random-effects one-way ANCOVA

- 1 level-1 predictor (SES, centered within school), no level-2 predictors
- random intercept, no random slopes
- model for the first (student) level:

$$y_{ij} = \alpha_{0i} + \alpha_{1i}\,\text{cses}_{ij} + \epsilon_{ij}$$

- model for the second (school) level:

$$\alpha_{0i} = \gamma_{00} + u_{0i} \qquad \text{(the random intercept)}$$
$$\alpha_{1i} = \gamma_{10} \qquad \text{(the constant slope)}$$

- the combined model and the Laird-Ware form:

$$\begin{aligned} y_{ij} &= (\gamma_{00} + u_{0i}) + \gamma_{10}\,\text{cses}_{ij} + \epsilon_{ij} \\ &= \gamma_{00} + \gamma_{10}\,\text{cses}_{ij} + u_{0i} + \epsilon_{ij} \\ &= \beta_0 + \beta_1 x_{1ij} + b_{0i} + \epsilon_{ij} \end{aligned}$$

- the fixed-effect coefficients β_0 and β_1 represent the average within-schools population intercept and slope respectively
  - note: because SES is centered within schools, the intercept β_0 represents the 'average' level of math achievement in the population
- the model has two variance-covariance components:
  - $\mathsf{Var}(b_{0i}) = d^2$: the variance among school intercepts
  - $\mathsf{Var}(\epsilon_{ij}) = \sigma^2$: the error variance around the within-school regressions

R code

```
> fit.model2 <- lmer(mAch ~ 1 + cses + (1 | school), data = Hsb82, REML = FALSE)
> summary(fit.model2, correlation = FALSE)

Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: mAch ~ 1 + cses + (1 | school)
   Data: Hsb82

     AIC      BIC   logLik deviance df.resid
 46728.4  46755.9 -23360.2  46720.4     7181

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.09692 -0.73195  0.01945  0.75738  2.91422

Random effects:
 Groups   Name        Variance Std.Dev.
 school   (Intercept)  8.612   2.935
 Residual             37.005   6.083
Number of obs: 7185, groups:  school, 160

Fixed effects:
            Estimate Std. Error t value
(Intercept)  12.6362     0.2437   51.85
cses          2.1912     0.1086   20.17
```

model 3: a random-coefficients regression model

- 1 level-1 predictor (SES, centered within school), no level-2 predictors
- random intercept and random slopes
- model for the first (student) level:

$$y_{ij} = \alpha_{0i} + \alpha_{1i}\,\text{cses}_{ij} + \epsilon_{ij}$$

- model for the second (school) level:

$$\alpha_{0i} = \gamma_{00} + u_{0i} \qquad \text{(the random intercept)}$$
$$\alpha_{1i} = \gamma_{10} + u_{1i} \qquad \text{(the random slope)}$$

- the combined model and the Laird-Ware form:

$$\begin{aligned} y_{ij} &= (\gamma_{00} + u_{0i}) + (\gamma_{10} + u_{1i})\,\text{cses}_{ij} + \epsilon_{ij} \\ &= \gamma_{00} + \gamma_{10}\,\text{cses}_{ij} + u_{0i} + u_{1i}\,\text{cses}_{ij} + \epsilon_{ij} \\ &= \beta_0 + \beta_1 x_{1ij} + b_{0i} + b_{0i} z_{1ij} + \epsilon_{ij} \end{aligned}$$

- the fixed-effect coefficients β_0 and β_1 again represent the average within-schools population intercept and slope respectively
- the model has four variance-covariance components:
  - Var$(b_{0i}) = d_0^2$: the variance among school intercepts
  - Var$(b_{1i}) = d_1^2$: the variance among school slopes
  - Cov$(b_{0i}, b_{1i}) = d_{01}$: the covariance between within-school intercepts and slopes
  - Var$(\epsilon_{ij}) = \sigma^2$: the error variance around the within-school regressions

R code

```
> fit.model3 <- lmer(mAch ~ 1 + cses + (1 + cses | school), data = Hsb82,
+                    REML = FALSE)
> summary(fit.model3, correlation = FALSE)

Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: mAch ~ 1 + cses + (1 + cses | school)
   Data: Hsb82

     AIC      BIC   logLik deviance df.resid
 46723.0  46764.3 -23355.5  46711.0     7179

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.09688 -0.73199  0.01794  0.75445  2.89901

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 school   (Intercept)  8.6204  2.9361
          cses         0.6782  0.8236   0.02
 Residual             36.7000  6.0581
Number of obs: 7185, groups:  school, 160

Fixed effects:
            Estimate Std. Error t value
(Intercept)  12.6363     0.2437   51.85
cses          2.1932     0.1278   17.15
```

model 4: random intercept + level-2 predictor

- we drop the level-1 predictor (cses), but add a level-2 predictor (meanses)
- random intercept, no random slopes
- model for the first (student) level:

$$y_{ij} = \alpha_{0i} + \epsilon_{ij}$$

- model for the second (school) level:

$$\alpha_{0i} = \gamma_{00} + \gamma_{01}\,\text{meanses}_i + u_{0i}$$

- the combined model and the Laird-Ware form:

$$\begin{aligned} y_{ij} &= \gamma_{00} + \gamma_{01}\,\text{meanses}_i + u_{0i} + \epsilon_{ij} \\ &= \beta_0 + \beta_1 x_{1ij} + b_{0i} + \epsilon_{ij} \end{aligned}$$

- note that in the Laird-Ware notation, we use a double index for the fixed-effects x_{ij}, even if the variabele (meanses) does not change over students

- this model has two fixed effects (β_0 and β_1) and two random effects (b_{0i} and ϵ_{ij})
- the interpretation of b_{0i} has changed: whereas in the previous model it had been the deviation of school i's mean from the grand mean, it now represents the residual $(\alpha_{0i} - \gamma_{00} - \gamma_{10} \text{ meanses}_j)$; correspondingly, the variance component d^2 is now a conditional variance (conditional on the school mean SES)
- in Raudenbush and Bryk, this is called a 'Regression with means-as-outcomes' model, because the school's mean (α_{0i}) is predicted by the means SES of the school
- note in the following output that the residual variance between schools is substantially smaller than the original

R code

```
> fit.model4 <- lmer(mAch ~ 1 + meanses + (1 | school), data = Hsb82,
+                    REML = FALSE)
> summary(fit.model4, correlation = FALSE)

Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: mAch ~ 1 + meanses + (1 | school)
   Data: Hsb82

     AIC      BIC   logLik deviance df.resid
 46967.1  46994.6 -23479.6  46959.1     7181

Scaled residuals:
     Min       1Q   Median       3Q      Max
-3.13480 -0.75252  0.02331  0.76833  2.78413

Random effects:
 Groups   Name        Variance Std.Dev.
 school   (Intercept)  2.593   1.610
 Residual             39.157   6.258
Number of obs: 7185, groups:  school, 160

Fixed effects:
            Estimate Std. Error t value
(Intercept)  12.6849     0.1483   85.52
meanses       5.8629     0.3591   16.32
```

model 5: intercepts-and-slopes-as-outcomes model

- we expand the model by including two level-2 predictors: meanses and sector; the slopes are again allowed to vary randomly
- model for the first (student) level:

$$y_{ij} = \alpha_{0i} + \alpha_{1i}\,\text{cses}_{ij} + \epsilon_{ij}$$

- model for the second (school) level:

$$\alpha_{0i} = \gamma_{00} + \gamma_{01}\,\text{meanses}_i + \gamma_{02}\,\text{sector}_i + u_{0i}$$
$$\alpha_{1i} = \gamma_{10} + \gamma_{11}\,\text{meanses}_i + \gamma_{12}\,\text{sector}_i + u_{1i}$$

- the combined model and the Laird-Ware form:

$$\begin{aligned} y_{ij} =& (\gamma_{00} + \gamma_{01}\,\text{meanses}_i + \gamma_{02}\,\text{sector}_i + u_{0i}) + \\ & (\gamma_{10} + \gamma_{11}\,\text{meanses}_i + \gamma_{12}\,\text{sector}_i + u_{1i})\,\text{cses}_{ij} + \epsilon_{ij} \\ =& \gamma_{00} + \gamma_{01}\,\text{meanses}_i + \gamma_{02}\,\text{sector}_i + \gamma_{10}\,\text{cses}_{ij} + \\ & \gamma_{11}\,\text{meanses}_i\,\text{cses}_{ij} + \gamma_{12}\,\text{sector}_i\,\text{cses}_{ij} + u_{0i} + u_{1i}\,\text{cses}_{ij} + \epsilon_{ij} \\ =& \beta_0 + \beta_1 x_{1ij} + \beta_2 x_{2ij} + \beta_3 x_{3ij} + \beta_4 (x_{1ij} x_{3ij}) + \beta_5 (x_{2ij} x_{3ij}) + \\ & b_{0i} + b_{1i} z_{1ij} + \epsilon_{ij} \end{aligned}$$

R code

```
> fit.model5 <- lmer(mAch ~ 1 + meanses*cses + sector*cses + (1 + cses | school),
+                    data = Hsb82, REML = FALSE)
> summary(fit.model5, correlation = FALSE)

Linear mixed model fit by maximum likelihood  ['lmerMod']
Formula: mAch ~ 1 + meanses * cses + sector * cses + (1 + cses | school)
   Data: Hsb82

     AIC      BIC   logLik deviance df.resid
 46516.4  46585.2 -23248.2  46496.4     7175
```

```
Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.1610 -0.7244  0.0168  0.7549  2.9581

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 school   (Intercept)  2.31667 1.5221
          cses         0.06507 0.2551   0.48
 Residual             36.72118 6.0598
Number of obs: 7185, groups:  school, 160

Fixed effects:
                    Estimate Std. Error t value
(Intercept)          12.1279     0.1974   61.44
meanses               5.3317     0.3655   14.59
cses                  2.9457     0.1540   19.13
sectorCatholic        1.2269     0.3033    4.05
meanses:cses          1.0427     0.2960    3.52
cses:sectorCatholic  -1.6440     0.2373   -6.93
```

do we need the random slopes?

```
> fit.model5bis <- lmer(mAch ~ 1 + meanses*cses + sector*cses + (1 | school),
+                          data = Hsb82, REML = FALSE)
> anova(fit.model5bis, fit.model5)

Data: Hsb82
Models:
fit.model5bis: mAch ~ 1 + meanses * cses + sector * cses + (1 | school)
fit.model5: mAch ~ 1 + meanses * cses + sector * cses + (1 + cses | school)
              Df   AIC   BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
fit.model5bis  8 46513 46568 -23249    46497
fit.model5    10 46516 46585 -23248    46496 1.0016      2     0.6061
```

- no: apparently, the level-2 predictors do a sufficiently good job of accounting for differences in slopes
- statistical note: using a LRT for comparing two models with a different random structure is conservative; better approaches exist (e.g. in the R package RLRsim)

summary of the models

```
> # model 1:
> lmer(mAch ~ 1 + (1 | school), data = Hsb82, REML = FALSE)

> # model 2:
> lmer(mAch ~ 1 + cses + (1 | school), data = Hsb82, REML = FALSE)

> # model 3:
> lmer(mAch ~ 1 + cses + (1 + cses | school), data = Hsb82, REML = FALSE)

> # model 4:
> lmer(mAch ~ 1 + meanses + (1 | school), data = Hsb82, REML = FALSE)

> # model 5:
> lmer(mAch ~ 1 + meanses*cses + sector*cses + (1 + cses | school),
+      data = Hsb82, REML = FALSE)

> # model 5bis:
> lmer(mAch ~ 1 + meanses*cses + sector*cses + (1 | school),
+      data = Hsb82, REML = FALSE)
```

2 Multilevel SEM

2.1 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

2.2 History

Schmidt, W. H. (1969)

- Schmidt, W.H. (1969). Covariance structure analysis of the multivariate random effects model (Doctoral dissertation, University of Chicago, Department of Education).
  - 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)
- Gustafsson, J.E., & Lindström, B. (1979). Analyzing ATI Data By Structural Analysis of Covariance Matrices. (Paper presented at the Annual Meeting of the AERA, San Fransisco, April 8–12, 1979) – Examples 7 + 8

> 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, L. (1980). The analysis of multilevel data in educational research and evaluation. *Review of research in education, 8*, 158–233.
  - Burstein worked at the Graduate School of Education (UCLA)
- also cited in Muthén, B.O. (1989). Latent variable modeling in heterogeneous populations. *Psychometrika, 54*, 557–585
  - he reformulated Schmidt's fitting function so that it could be estimated using existing software for multiple-group SEM (e.g., LISCOMP)

Goldstein & McDonald

- Goldstein, H., & McDonald, R.P. (1988). A general model for the analysis of multilevel data. *Psychometrika, 53*, 455–467.
  - very general formulation, including multilevel SEM
  - univariate perspective (multivariate vector = 1st level)
  - can handle missing data, hierchical data, cross-classified data
  - expression of the likelihood, IGLS algorithm is suggested
- McDonald, R.P., & Goldstein, H. (1989). Balanced versus unbalanced designs for linear structural relations in two-level data. *British Journal of mathematical and statistical psychology, 42*, 215–232.
  - multivariate perspective, within-and-between formulation
  - likelihood expression + a computationally tractable (re)expression
  - both for balanced and unbalanced clusters
- McDonald, R.P. (1993). A general model for two-level data with responses missing at random. *Psychometrika*, 58, 575–585.

Muthén

- Muthén, B.O. (1989). Latent variable modeling in heterogeneous populations. *Psychometrika, 54*, 557–585
  - 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
- Muthén, B.O. (1990). Mean and covariance structure analysis of hierarchical data. Department of Statistics, UCLA. (unpublished technical report)
  - 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
  - more discussion in Muthén, B.O. (1994). Multilevel covariance structure analysis. *Sociological methods & research, 22*, 376–398.
- standard SEM software could be used (at least for the balanced case)

Lee

- Lee, S.Y. (1990). Multilevel analysis of structural equation models. *Biometrika, 4*, 763–772.
  - 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)
- Lee, S.Y. (2007). *Structural equation modeling: A Bayesian approach*. John Wiley & Sons.
  - Chapter 9: Bayesian methods for analyzing various two-level SEMs

Bentler

- Liang, J., & Bentler, P.M. (2004). An EM algorithm for fitting two-level structural equation models. *Psychometrika, 69*, 101–122.
  - earlier work: Benter & Liang (2000?), Bentler & Liang (2003), Liang & Bentler (2003)
  - 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)

2.3 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

$\mathbf{\Sigma}_B$

Between

Within

$\mathbf{\Sigma}_W$

```
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:

Curran, P.J. (2003). Have multilevel models been structural equation models all along? *Multivariate Behavioral Research, 38*, 529–569.

Bauer, D.J. (2003). Estimating Multilevel Linear Models as Structural Equation Models. *Journal of Educational and Behavioral Statistics, 28*, 135–167.

Mehta, P.D., and Neale, M.C. (2005). People are variables too: Multilevel structural equations modeling. *Psychological methods, 10*, 259–284.

- books:

Hox, J.J., Moerbeek, M., & van de Schoot, R. (2010). *Multilevel analysis: Techniques and applications*. Routledge.

Heck, R.H., & Thomas, S.L. (2015). *An introduction to multilevel modeling techniques: MLM and SEM approaches using Mplus*. Routledge.

Skrondal, A., & Rabe-Hesketh, S. (2004). *Generalized latent variable modeling: Multilevel, longitudinal, and structural equation models.* CRC Press.

Lee, Sik-Yum (2007). *Structural equation modeling: A Bayesian approach.* John Wiley & Sons.

2.4 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):

$$\mathbf{y}_{ji} = (\mathbf{y}_{ji} - \bar{\mathbf{y}}_g) + \bar{\mathbf{y}}_g$$
$$\mathbf{y}_T = \mathbf{y}_W + \mathbf{y}_B$$

  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; $\bar{\mathbf{y}}_j$ 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

$$\text{Cov}(\mathbf{y}) = \mathbf{\Sigma}_T = \mathbf{\Sigma}_W + \mathbf{\Sigma}_B$$

two-level SEM: specifying a model for each level

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

$$\mathbf{\Sigma}_W = \mathbf{\Lambda}_W \mathbf{\Psi}_W \mathbf{\Lambda}'_W + \mathbf{\Theta}_W$$

and

$$\mathbf{\Sigma}_B = \mathbf{\Lambda}_B \mathbf{\Psi}_B \mathbf{\Lambda}'_B + \mathbf{\Theta}_B$$

- if we add a structural (regression) part, we need to add the $(\boldsymbol{I} - \boldsymbol{B})^{-1}$ 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 $\boldsymbol{\mu}_B$ can be added for the between part of the model
- in addition, we can add level-2 covariates ($\mathbf{z}_j$) to the model

2.5 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: $\mathbf{\Sigma}_{zz}$, $\mathbf{\Sigma}_{zy}$, $\mathbf{\Sigma}_w$, $\mathbf{\Sigma}_b$ and $\boldsymbol{\mu}_b = [\boldsymbol{\mu}_z, \boldsymbol{\mu}_y]^T$
- expectation of $\mathbf{v}_j$:

$$\mathrm{E}[\mathbf{v}_j] = \hat{\mathbf{v}}_j = [\boldsymbol{\mu}_z, \boldsymbol{\mu}_{\boldsymbol{y}}, \boldsymbol{\mu}_{\boldsymbol{y}}, \ldots, \boldsymbol{\mu}_{\boldsymbol{y}}]^T$$

- covariance matrix for $\mathbf{v}_j$:

$$\mathrm{Cov}[\mathbf{v}_j] = \mathbf{V}_j = \begin{bmatrix} \mathbf{\Sigma}_{zz} & \mathbf{1}_{n_j}^T \otimes \mathbf{\Sigma}_{zy} \\ \mathbf{1}_{n_j} \otimes \mathbf{\Sigma}_{yz} & \mathbf{\Sigma}_{yy} \end{bmatrix}$$

where

$$\mathbf{\Sigma}_{yy} = \mathbf{I}_{n_j} \otimes \mathbf{\Sigma}_w + \mathbf{1}_{n_j}\mathbf{1}_{n_j}^T \otimes \mathbf{\Sigma}_b$$

loglikelihood

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

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

where O_j is the length of $\mathbf{v}_j$, 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:

$$F_{ML} = \sum_{j=1}^{J} \ln|\mathbf{V}_j| + (\mathbf{v}_j - \hat{\mathbf{v}}_j)^T \mathbf{V}_j^{-1}(\mathbf{v}_j - \hat{\mathbf{v}}_j)$$

objective function (optional)

- the original objective function:

$$F_{ML} = \sum_{j=1}^{J} \ln |\mathbf{V}_j| + (\mathbf{v}_j - \hat{\mathbf{v}}_j)^T \mathbf{V}_j^{-1} (\mathbf{v}_j - \hat{\mathbf{v}}_j)$$

- for large clusters, the size of $\mathbf{V}_j$ can be formidable
- we should exploit the block-diagonal structure of $\mathbf{V}$
- we define:

$$\boldsymbol{\Sigma}_{b.z} = (\boldsymbol{\Sigma}_b - \boldsymbol{\Sigma}_{yz} \boldsymbol{\Sigma}_{zz}^{-1} \boldsymbol{\Sigma}_{zy})$$

- version 1: McDonald & Goldstein (1989), per cluster, using $\boldsymbol{\Sigma}_{b.z}$:

$$
\begin{aligned}
F_{ML} = \sum_{j=1}^{J} \Bigg[& \ln|\boldsymbol{\Sigma}_{zz}| + (n_j - 1)\ln|\boldsymbol{\Sigma}_w| + \ln|\boldsymbol{\Sigma}_w + n_j \cdot \boldsymbol{\Sigma}_{b.z}| \\
& + \text{tr}\left[\left(\boldsymbol{\Sigma}_{zz}^{-1} + n_j \boldsymbol{\Sigma}_{zz}^{-1}\boldsymbol{\Sigma}_{zy}(n_j\boldsymbol{\Sigma}_{b.z} + \boldsymbol{\Sigma}_w)^{-1}\boldsymbol{\Sigma}_{yz}\boldsymbol{\Sigma}_{zz}^{-1}\right)(\mathbf{z}_j - \boldsymbol{\mu}_z)(\mathbf{z}_j \right. \\
& + 2n_j\,\text{tr}\left[-\boldsymbol{\Sigma}_{zz}^{-1}\boldsymbol{\Sigma}_{zy}(n_j\boldsymbol{\Sigma}_{b.z} + \boldsymbol{\Sigma}_w)^{-1}(\bar{\mathbf{y}}_j - \boldsymbol{\mu}_y)(\mathbf{z}_j - \boldsymbol{\mu}_z)^T\right] \\
& + \text{tr}\left[\boldsymbol{\Sigma}_w^{-1}\mathbf{Y}_j^{(c)T}\mathbf{Y}_j^{(c)}\right] \\
& - n_j\text{tr}\left[\boldsymbol{\Sigma}_w^{-1}(\bar{\mathbf{y}}_j - \boldsymbol{\mu}_y)(\bar{\mathbf{y}}_j - \boldsymbol{\mu}_y)^T\right] \\
& + n_j\text{tr}\left[(n_j\boldsymbol{\Sigma}_{b.z} + \boldsymbol{\Sigma}_w)^{-1}(\bar{\mathbf{y}}_j - \boldsymbol{\mu}_y)(\bar{\mathbf{y}}_j - \boldsymbol{\mu}_y)^T\right] \Bigg]
\end{aligned}
$$

- version 2: lavaan = McDonald & Goldstein (1989), per cluster size,

$$F_{ML} = (N-J)\left(\ln|\boldsymbol{\Sigma}_w| + \text{tr}\left[\boldsymbol{\Sigma}_w^{-1} S_{pw}\right]\right) + \sum_{s=1}^{S} n_s \cdot \Bigg[(\ln|\boldsymbol{\Sigma}_{zz}| + \ln|\boldsymbol{\Sigma}_w + n_j \cdot \boldsymbol{\Sigma}_{b.z}|) + \text{tr}\left[\left(\boldsymbol{\Sigma}_{zz}^{-1} + n_j \boldsymbol{\Sigma}_{zz}^{-1}\boldsymbol{\Sigma}_{zy}(n_j\boldsymbol{\Sigma}_{b.z} + \boldsymbol{\Sigma}_w)^{-1}\boldsymbol{\Sigma}_{yz}\boldsymbol{\Sigma}_{zz}^{-1}\right)(\mathbf{z}_j - \boldsymbol{\mu}_z)(\mathbf{z}_j - \boldsymbol{\mu}_z)^T\right] + 2n_j\,\text{tr}\left(-\boldsymbol{\Sigma}_{zz}^{-1}\boldsymbol{\Sigma}_{zy}(n_j\boldsymbol{\Sigma}_{b.z} + \boldsymbol{\Sigma}_w)^{-1}(\bar{\mathbf{y}}_j - \boldsymbol{\mu}_y)(\mathbf{z}_j - \boldsymbol{\mu}_z)^T\right) + n_j\text{tr}\left((n_j\boldsymbol{\Sigma}_{b.z} + \boldsymbol{\Sigma}_w)^{-1}(\bar{\mathbf{y}}_j - \boldsymbol{\mu}_y)(\bar{\mathbf{y}}_j - \boldsymbol{\mu}_y)^T\right) \Bigg]$$

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

$$S_{pw} = \frac{\sum_{j=1}^{J}\sum_{i=1}^{n_j}(\mathbf{y}_{ji} - \bar{\mathbf{y}}_j)(\mathbf{y}_{ji} - \bar{\mathbf{y}}_j)^T}{N-J}$$

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)

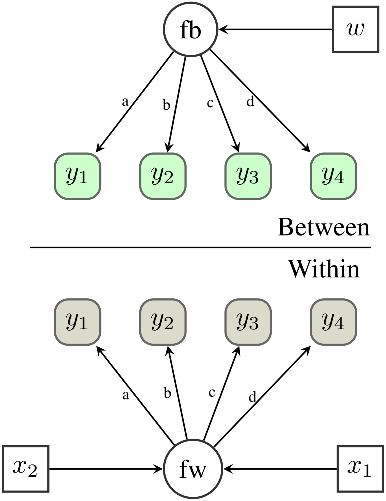
Example: Mplus ex9.6 (simulated data)

- 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 wihtin level x_1, x_2
- 1 covariate at the between level w
- reading in the data:

```
> Data <- read.table("http://statmodel.com/usersguide/chap9/ex9.6.dat")
> names(Data) <- c("y1", "y2", "y3", "y4", "x1", "x2", "w", "clus")
> head(Data)
```

```
         y1        y2        y3        y4        x1        x2         w clus
1  2.203250  1.858861  1.738477  2.244863  1.142800 -0.796987 -0.149501    1
2  1.934917  2.127876  0.083120  2.509436  1.949033 -0.122764 -0.149501    1
3  0.321955  0.977231 -0.835405  0.558367 -0.716481 -0.767064 -0.149501    1
4  0.073154 -1.743092 -2.310271 -1.514332 -2.649131  0.637570 -0.149501    1
5 -1.214906  0.452618  0.372610 -1.790372 -0.262916  0.302564 -0.149501    1
6  0.298330 -1.820272  0.561335 -2.090582 -0.944963  1.363045  0.319335    2
```

diagram + lavaan syntax




```
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 (0.6-1.1173) converged normally after  27 iterations

  Number of observations                          1000
  Number of clusters [clus]                        110

  Estimator                                         ML
  Model Fit Test Statistic                       3.863
  Degrees of freedom                                17
  P-value (Chi-square)                           1.000

Parameter Estimates:

  Information                                 Observed
  Observed information based on                Hessian
  Standard Errors                             Standard


Level 1 [within]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  fw =~
    y1                1.000
    y2                0.999    0.033   30.735    0.000
    y3                0.995    0.033   29.804    0.000
    y4                1.017    0.033   30.364    0.000
```

```
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    0.032    0.215    0.830
    x2                0.014    0.032    0.436    0.663
   .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
    x1              0.985    0.044   22.361    0.000
    x2              1.017    0.045   22.361    0.000


Level 2 [clus]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  fb =~
    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:
                   Estimate  Std.Err  z-value  P(>|z|)
  fb ~
    w               0.344    0.078    4.429    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1             -0.083    0.076   -1.095    0.274
   .y2             -0.077    0.074   -1.047    0.295
   .y3             -0.045    0.073   -0.617    0.537
   .y4             -0.030    0.074   -0.405    0.686
    w               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

> fitMeasures(fit)

            npar                fmin               chisq                  df
          26.000               3.913               3.863              17.000
          pvalue      baseline.chisq         baseline.df     baseline.pvalue
           1.000            3280.729              25.000               0.000
             cfi                 tli                nnfi                 rfi
           1.000               1.006               1.006               0.998
             nfi                pnfi                 ifi                 rni
           0.999               0.679               1.004               1.004
            logl  unrestricted.logl                 aic                 bic
       -9527.429           -9525.497           19106.857           19234.459
          ntotal                bic2               rmsea      rmsea.ci.lower
        1000.000           19151.882               0.000               0.000
  rmsea.ci.upper        rmsea.pvalue                srmr         srmr_within
           0.000               1.000               0.022               0.004
    srmr_between
           0.017
```

```
> lavInspect(fit, "h1")

$within
$within$cov
   y1    y2    y3    y4    x1    x2
y1 3.191
y2 2.216 3.144
y3 2.203 2.197 3.257
y4 2.233 2.242 2.236 3.284
x1 0.966 0.962 0.969 1.011 0.985
x2 0.566 0.553 0.535 0.552 0.032 1.017

$within$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 0.456
y2 0.440 0.432
y3 0.419 0.406 0.387
y4 0.433 0.418 0.398 0.425
w  0.296 0.247 0.273 0.264 0.815

$clus$mean
    y1     y2     y3     y4      w
```

```
-0.067 -0.062 -0.030 -0.013  0.006

> lavInspect(fit, "implied")

$within
$within$cov
   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

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


$clus
$clus$cov
   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
```

```
$clus$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
0.125 0.121 0.106 0.115 0.000 0.000
```

2.6 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

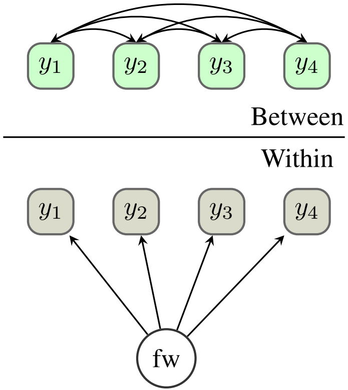
- reference:

Stapleton, L.M., Yang, J.S., & Hancock, G.R. (2016). Construct meaning in multilevel settings. *Journal of Educational and Behavioral Statistics, 41*, 481–520.

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

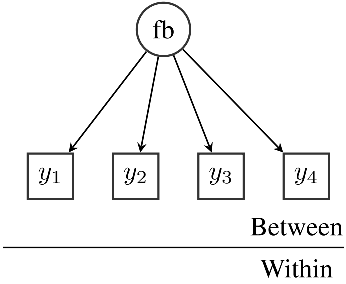



```
model <- '

  level: 1

    fw =~ y1 + y2 + y3 + y4

  level: 2

    y1 ~~ y1 + y2 + y3 + y4
    y2 ~~ y2 + y2 + y3
    y3 ~~ y3 + y4
    y4 ~~ 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

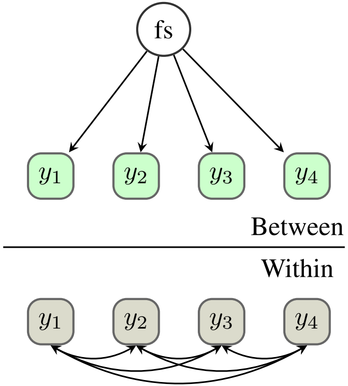



```
model <- '
  level: 1
    # perhaps other level-1 variables
  level: 2
    fb =~ y1 + y2 + y3 + y4
'
```

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

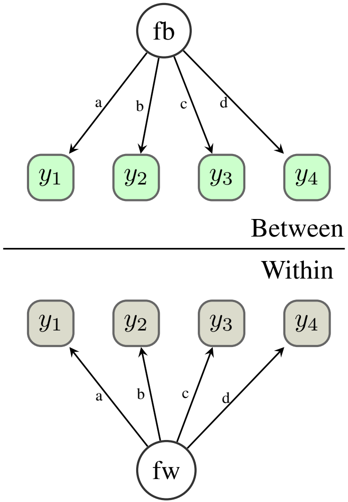



```
model <- '

  level: 1

    y1 ~~ y1 + y2 + y3 + y4
    y2 ~~ y2 + y2 + y3
    y3 ~~ y3 + y4
    y4 ~~ y4

  level: 2

    fs =~ y1 + y2 + y3 + y4

'
```

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

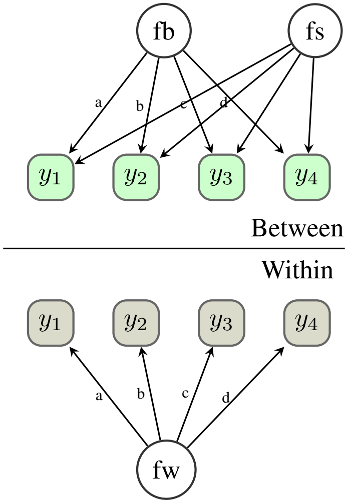



```
model <- '

  level: 1

    fw =~ a*y1 + b*y2 + c*y3 + d*y4

  level: 2

    fb =~ a*y1 + b*y2 + c*y3 + d*y4

'
```

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

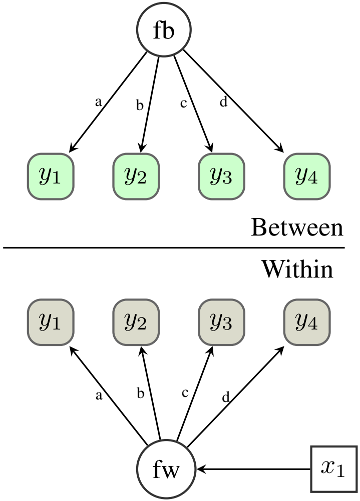



```
model <- '
  level: 1
    fw =~ a*y1 + b*y2 + c*y3 + d*y4
  level: 2
    fb =~ a*y1 + b*y2 + c*y3 + d*y4
    fs =~ y1 + y2 + y3 + y4
    # fb and fs must be orthogonal
    fs ~~ 0*fb
'
```

2.7 The status of observed covariates in a two-level SEM

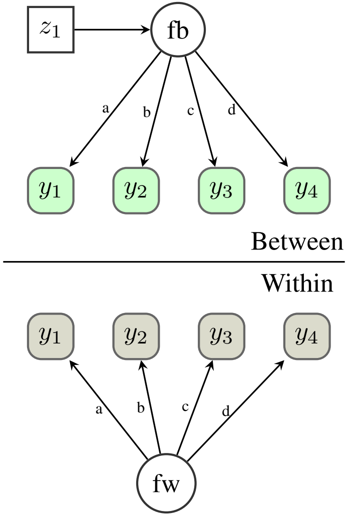
- when observed covariates are added in a two-level model, we again need to carefully reflect on the 'status' of these covariates
  - are the covariates measured at the within or the between level?
  - if they are measured at the within level, does it make sense to split this covariate into a within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



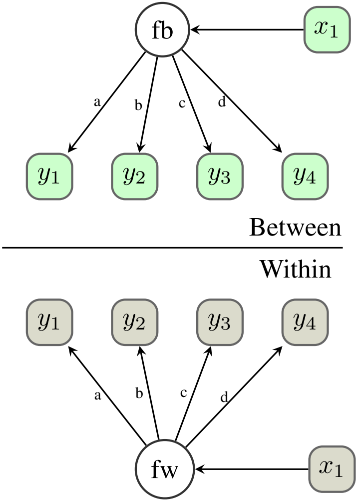

```
model <- '
  level: 1
    fw =~ a*y1 + b*y2 + c*y3 + d*y4
    fw ~ x1
  level: 2
    fb =~ a*y1 + b*y2 + c*y3 + d*y4
'
```

adding a between-only covariate




```
model <- '
  level: 1
    fw =~ a*y1 + b*y2 + c*y3 + d*y4
  level: 2
    fb =~ a*y1 + b*y2 + c*y3 + d*y4
    fb ~ z1
'
```

adding a level-1 covariate with a within and a between part




```
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
'
```

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:

Lüdtke, O., Marsh, H.W., Robitzsch, A., Trautwein, U., Asparouhov, T., & Muthén, B. (2008). The multilevel latent covariate model: a new, more reliable approach to group-level effects in contextual studies. *Psychological methods, 13*, 203–229.

2.8 Evaluating model fit

- if no random slopes are involved, we can fit an unrestricted (saturated) model: we estimate all the elements of $\boldsymbol{\Sigma}_W$, $\boldsymbol{\Sigma}_B$ and $\boldsymbol{\mu}_B$
- then, we can compute the standard 'χ^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:

  Hsu, H.Y., Kwok, O.M., Lin, J.H., & Acosta, S. (2015). Detecting misspecified multilevel structural equation models with common fit indices: a Monte Carlo study. *Multivariate behavioral research, 50*, 197–215.

- 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

2.9 Example: two-level CFA

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

  Hox, J.J., Moerbeek, M., & van de Schoot, R. (2010). *Multilevel analysis: Techniques and applications*. Routledge.

- 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

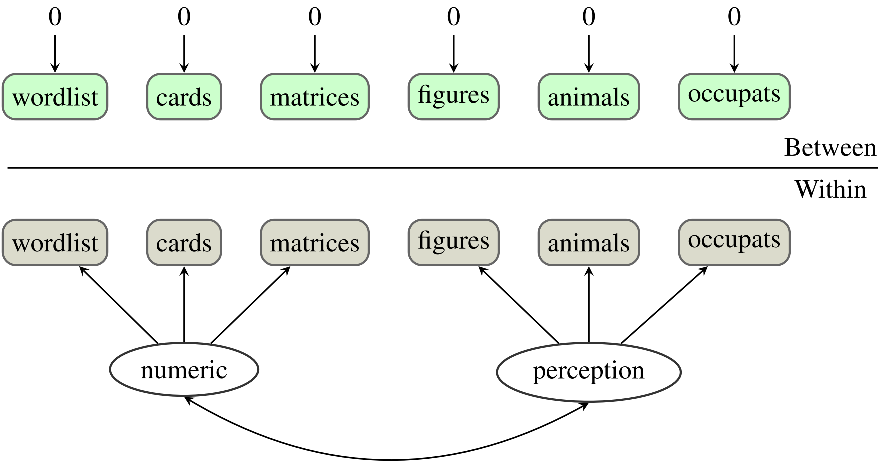
```
> FamIQData <- read.table("FamIQData.dat")
> names(FamIQData) <- c("family", "child", "wordlist", "cards", "matrices",
+                       "figures", "animals", "occupats")
> summary(FamIQData)

     family          child          wordlist         cards      
 Min.   : 1.00   Min.   : 1.00   Min.   :12.00   Min.   :11.00  
 1st Qu.:16.00   1st Qu.: 2.00   1st Qu.:27.00   1st Qu.:26.50  
 Median :33.00   Median : 4.00   Median :30.00   Median :30.00  
 Mean   :31.78   Mean   : 4.04   Mean   :29.95   Mean   :29.84  
 3rd Qu.:48.00   3rd Qu.: 6.00   3rd Qu.:33.00   3rd Qu.:33.00  
 Max.   :60.00   Max.   :12.00   Max.   :45.00   Max.   :44.00  
    matrices        figures         animals         occupats    
 Min.   :15.00   Min.   :17.00   Min.   :15.00   Min.   :15.00  
 1st Qu.:26.00   1st Qu.:27.00   1st Qu.:27.00   1st Qu.:27.00  
 Median :30.00   Median :30.00   Median :30.00   Median :30.00  
 Mean   :29.73   Mean   :30.08   Mean   :30.11   Mean   :30.01  
 3rd Qu.:33.00   3rd Qu.:33.00   3rd Qu.:34.00   3rd Qu.:33.00  
 Max.   :46.00   Max.   :44.00   Max.   :46.00   Max.   :43.00  

> # 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 $\mathbf{S}_{PW}$
  - 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

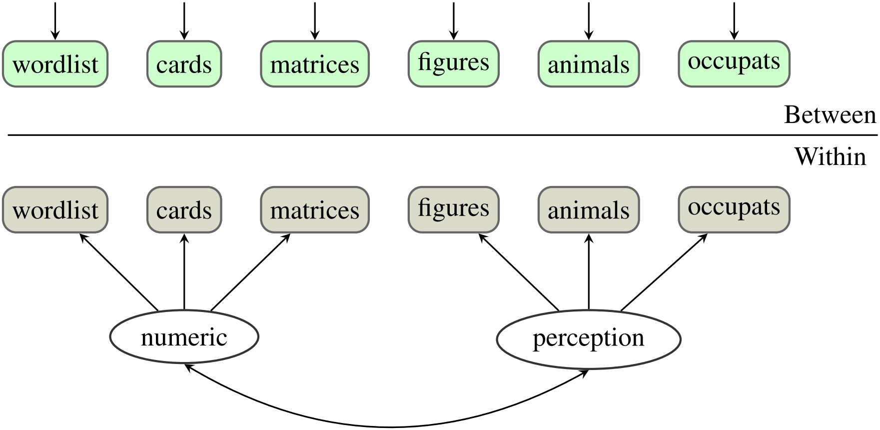


lavaan syntax

```
> 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 = FALSE)
> # summary(fit1)
> fit1
```

```
lavaan (0.6-1.1173) converged normally after  53 iterations

  Number of observations                           399
  Number of clusters [family]                       60

  Estimator                                         ML
  Model Fit Test Statistic                     323.649
  Degrees of freedom                                29
  P-value (Chi-square)                           0.000
```

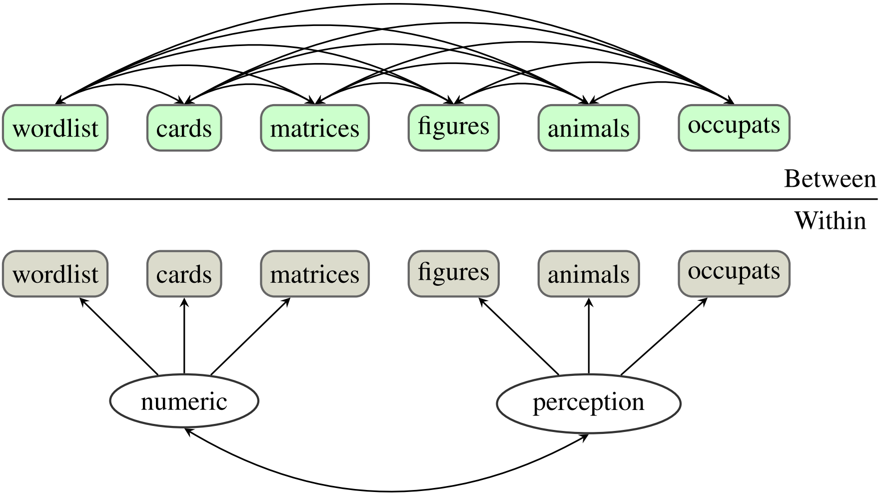
model 2: a 2-factor within model + independence between model



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 = FALSE)
> # summary(fit2)
> fit2

lavaan (0.6-1.1173) converged normally after  50 iterations

  Number of observations                           399
  Number of clusters [family]                       60

  Estimator                                         ML
  Model Fit Test Statistic                     177.206
  Degrees of freedom                                23
  P-value (Chi-square)                           0.000
```

model 3: a 2-factor within model, with saturated between part



lavaan syntax model 3

```
> 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 = FALSE)
> summary(fit3)
```

```
lavaan (0.6-1.1173) converged normally after 174 iterations

  Number of observations                           399
  Number of clusters [family]                       60

  Estimator                                         ML
  Model Fit Test Statistic                       6.716
  Degrees of freedom                                 8
  P-value (Chi-square)                           0.568
```

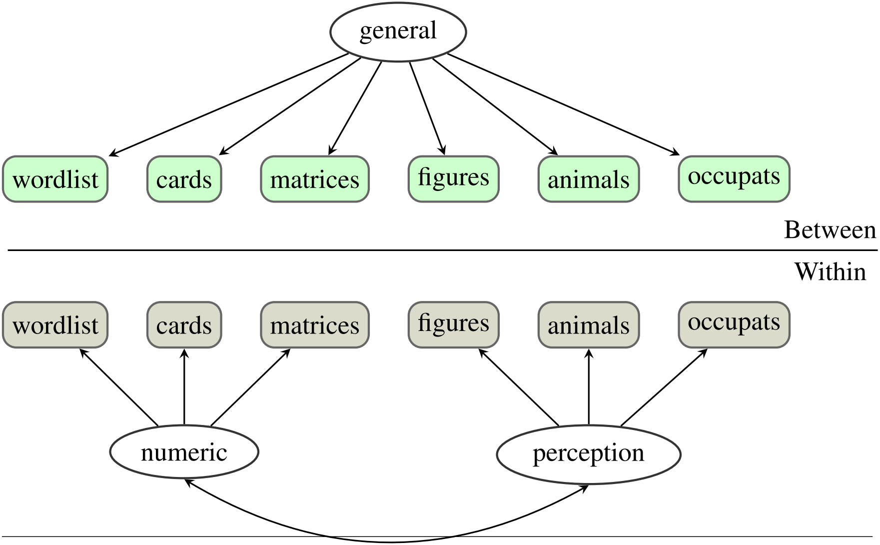
```
Parameter Estimates:

  Information                                     Observed
  Observed information based on                    Hessian
  Standard Errors                                 Standard


Level 1 [within]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  numeric =~
    wordlist          3.155    0.203   15.558    0.000
    cards             3.156    0.196   16.113    0.000
    matrices          3.032    0.199   15.207    0.000
  perception =~
    figures           3.091    0.205   15.069    0.000
    animals           3.192    0.195   16.397    0.000
    occupats          2.774    0.183   15.139    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  numeric ~~
    perception        0.386    0.058    6.691    0.000

Intercepts:
                   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
    numeric          0.000
    perception       0.000

Variances:
                  Estimate  Std.Err  z-value  P(>|z|)
   .wordlist         6.234    0.739    8.433    0.000
   .cards            5.344    0.693    7.705    0.000
   .matrices         6.443    0.714    9.025    0.000
   .figures          6.856    0.757    9.053    0.000
   .animals          4.851    0.696    6.968    0.000
   .occupats         5.338    0.604    8.835    0.000
    numeric          1.000
    perception       1.000


Level 2 [family]:

Covariances:
                  Estimate  Std.Err  z-value  P(>|z|)
 .wordlist ~~
   .cards            9.272    2.225    4.168    0.000
   .matrices         8.515    2.077    4.100    0.000
   .figures          8.410    2.053    4.097    0.000
```

```
   .animals          9.700    2.195    4.419    0.000
   .occupats        10.428    2.357    4.425    0.000
 .cards ~~
   .matrices         7.997    2.018    3.964    0.000
   .figures          8.424    2.035    4.140    0.000
   .animals         10.000    2.203    4.540    0.000
   .occupats        10.418    2.337    4.457    0.000
 .matrices ~~
   .figures          7.733    1.902    4.067    0.000
   .animals          8.022    1.966    4.081    0.000
   .occupats         9.000    2.142    4.203    0.000
 .figures ~~
   .animals          8.980    2.177    4.125    0.000
   .occupats         9.750    2.333    4.179    0.000
 .animals ~~
   .occupats        11.080    2.489    4.451    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .wordlist        29.890    0.470   63.547    0.000
   .cards           29.892    0.465   64.308    0.000
   .matrices        29.732    0.439   67.746    0.000
   .figures         30.047    0.459   65.476    0.000
   .animals         30.135    0.471   63.956    0.000
   .occupats        29.967    0.509   58.891    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
```

```
.wordlist         10.727    2.456    4.368    0.000
.cards            10.558    2.397    4.404    0.000
.matrices          9.097    2.123    4.285    0.000
.figures          10.051    2.321    4.330    0.000
.animals          10.956    2.466    4.442    0.000
.occupats         13.473    2.874    4.688    0.000
```

model 4a: a 2-factor within model + general factor between



lavaan syntax model 4a

```
> 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 = FALSE)
> summary(fit4a)
```

```
lavaan (0.6-1.1173) converged normally after  64 iterations

  Number of observations                           399
  Number of clusters [family]                       60

  Estimator                                         ML
  Model Fit Test Statistic                      11.927
  Degrees of freedom                                17
  P-value (Chi-square)                           0.805

Parameter Estimates:

  Information                                 Observed
  Observed information based on                Hessian
```

```
  Standard Errors                                 Standard

Level 1 [within]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  numeric =~
    wordlist          3.175    0.202   15.710    0.000
    cards             3.144    0.194   16.167    0.000
    matrices          3.054    0.199   15.349    0.000
  perception =~
    figures           3.095    0.204   15.147    0.000
    animals           3.188    0.194   16.438    0.000
    occupats          2.782    0.183   15.215    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  numeric ~~
    perception        0.382    0.057    6.740    0.000

Intercepts:
                   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
    numeric           0.000
    perception        0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .wordlist          6.194    0.737    8.406    0.000
   .cards             5.403    0.692    7.804    0.000
   .matrices          6.417    0.714    8.992    0.000
   .figures           6.847    0.757    9.049    0.000
   .animals           4.881    0.696    7.009    0.000
   .occupats          5.324    0.603    8.823    0.000
    numeric           1.000
    perception        1.000


Level 2 [family]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  general =~
    wordlist          3.057    0.393    7.785    0.000
    cards             3.054    0.389    7.843    0.000
    matrices          2.632    0.381    6.904    0.000
    figures           2.806    0.398    7.048    0.000
    animals           3.204    0.383    8.371    0.000
    occupats          3.439    0.415    8.292    0.000
```

```
Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .wordlist         29.891    0.468   63.847    0.000
   .cards            29.890    0.466   64.097    0.000
   .matrices         29.749    0.435   68.462    0.000
   .figures          30.044    0.458   65.536    0.000
   .animals          30.134    0.471   64.041    0.000
   .occupats         29.967    0.508   59.012    0.000
    general           0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .wordlist          1.253    0.569    2.201    0.028
   .cards             1.323    0.586    2.258    0.024
   .matrices          1.935    0.669    2.891    0.004
   .figures           2.158    0.714    3.022    0.003
   .animals           0.656    0.487    1.347    0.178
   .occupats          1.581    0.624    2.536    0.011
    general           1.000
```

2.10 Example: two-level SEM

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

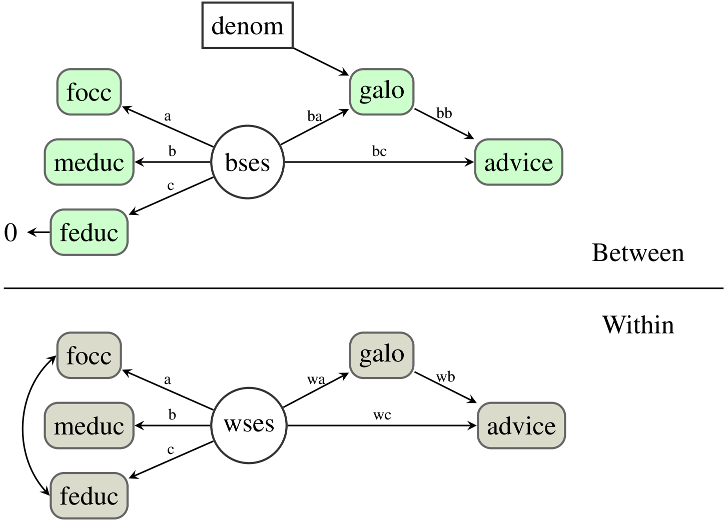
  Hox, J.J., Moerbeek, M., & van de Schoot, R. (2010). *Multilevel analysis: Techniques and applications*. Routledge.

- 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 (indirectly) 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
- in addtion, it was decided to fix the residual variance of `feduc` to zero at the between 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


denom
focc
galo
a
ba
bb
meduc
b
bses
bc
advice
c
0
feduc
Between
Within
focc
galo
a
wa
wb
meduc
b
wses
wc
advice
c
feduc

exploring the data

```
> Galo <- read.table("Galo.dat")
> names(Galo) <- c("school", "sex", "galo", "advice", "feduc", "meduc",
+                  "focc", "denom")
> Galo[Galo == 999] <- NA
> Galo$denom1 <- ifelse(Galo$denom == 1, 1, 0)
> Galo$denom2 <- ifelse(Galo$denom == 2, 1, 0)
> summary(Galo)
```

```
    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                    
    denom1            denom2      
 Min.   :0.0000   Min.   :0.0000  
```

```
 1st Qu.:0.0000   1st Qu.:0.0000
 Median :0.0000   Median :1.0000
 Mean   :0.1501   Mean   :0.6928
 3rd Qu.:0.0000   3rd Qu.:1.0000
 Max.   :1.0000   Max.   :1.0000

> 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 + denom1 + denom2
+
+     # defined parameters
+     wi := wa * wb
+     bi := ba * bb
+ '
> fit <- sem(model, data = Galo, cluster = "school", std.lv = TRUE)
> summary(fit)
```

```
lavaan (0.6-1.1173) converged normally after 105 iterations

                                                  Used       Total
```

```
  Number of observations                          1382        1559
  Number of clusters [school]                       58

  Estimator                                         ML
  Model Fit Test Statistic                      26.221
  Degrees of freedom                                19
  P-value (Chi-square)                           0.124

Parameter Estimates:

  Information                                 Observed
  Observed information based on                Hessian
  Standard Errors                             Standard


Level 1 [within]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  wses =~
    focc       (a)    0.748    0.038   19.558    0.000
    meduc      (b)    1.282    0.047   27.570    0.000
    feduc      (c)    1.674    0.057   29.205    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  advice ~
    wses      (wc)    0.119    0.027    4.489    0.000
```

```
    galo      (wb)    0.086    0.002   44.740    0.000
  galo ~
    wses      (wa)    4.200    0.371   11.325    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
 .focc ~~
   .feduc             0.257    0.086    2.986    0.003

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .focc              0.000
   .meduc             0.000
   .feduc             0.000
   .advice            0.000
   .galo              0.000
    wses              0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .focc              1.186    0.065   18.132    0.000
   .meduc             2.021    0.120   16.900    0.000
   .feduc             1.582    0.167    9.462    0.000
   .advice            0.574    0.022   25.512    0.000
   .galo            125.024    5.123   24.403    0.000
    wses              1.000
```

```
Level 2 [school]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  bses =~
    focc      (a)     0.748    0.038   19.558    0.000
    meduc     (b)     1.282    0.047   27.570    0.000
    feduc     (c)     1.674    0.057   29.205    0.000

Regressions:
                   Estimate  Std.Err  z-value  P(>|z|)
  advice ~
    bses     (bc)     0.274    0.069    3.958    0.000
    galo     (bb)     0.062    0.011    5.443    0.000
  galo ~
    bses     (ba)     5.121    0.591    8.672    0.000
    denom1           -5.152    1.602   -3.216    0.001
    denom2           -0.511    1.267   -0.403    0.687

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .focc              3.251    0.108   30.201    0.000
   .meduc             2.839    0.178   15.965    0.000
   .feduc             3.862    0.228   16.948    0.000
   .advice           -3.238    1.164   -2.782    0.005
   .galo            103.333    1.335   77.379    0.000
    bses              0.000
```

```
Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .feduc             0.000
   .focc              0.032    0.016    2.010    0.044
   .meduc             0.021    0.025    0.844    0.398
   .advice            0.015    0.008    1.905    0.057
   .galo              5.745    2.057    2.793    0.005
    bses              1.000

Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    wi                0.360    0.032   11.146    0.000
    bi                0.317    0.068    4.698    0.000
```

3 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 safe the day

3.1 The ‘wide data’ approach

- wonderful paper about this:

  Bauer, D.J. (2003). Estimating Multilevel Linear Models as Structural Equation Models. *Journal of Educational and Behavioral Statistics, 28*, 135–167.

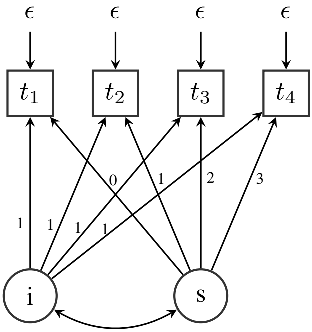
- first approach: using classic SEM to mimic multilevel regression models
  - the random intercepts and random slopes are represented by latent variables
  - the factor loadings of the random intercept are fixed to 1.0
  - the factor loadings of the random slope are fixed to the values of the predictor
  - 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’)

- 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 = (initial time at time 1) + (growth per unit time)*time + error
- y_t = intercept + slope*time + error

R code: using SEM in wide format

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

         t1        t2        t3        t4
1  1.7256454  2.142401  2.773172  2.515956
2 -1.9841595 -4.400603 -6.016556 -7.029618
3  0.3195183 -1.269117  1.560016  2.868530
4  0.7769485  3.531371  3.138211  5.363741
```

```
> 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 (0.6-1.1173) 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
  Information saturated (h1) model          Structured
  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 =~
    t1                0.000
    t2                1.000
    t3                2.000
    t4                3.000

Covariances:
                   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
   .t2                0.000
   .t3                0.000
   .t4                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

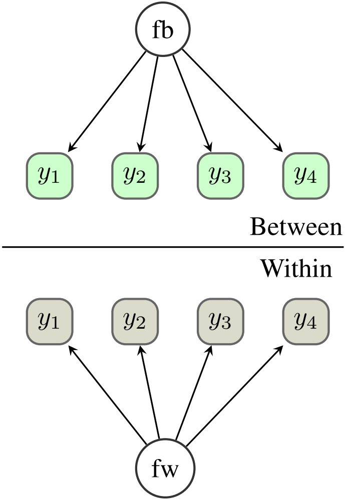
```
> # wide to long
> id    <- rep(1:400,  each = 4)
> score <- lav_matrix_vecr(Demo.growth[,1:4])
> time  <- rep(0:3, times = 400)
> growth.long <- data.frame(id = id, score = score, time = time)
> head(growth.long)

  id     score time
1  1  1.725645    0
2  1  2.142401    1
3  1  2.773172    2
4  1  2.515956    3
5  2 -1.984160    0
6  2 -4.400603    1
```

```
> 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
```

example 2: 1-factor model, cluster size = 3

- model in the multilevel SEM framework:



multilevel SEM syntax

```
> longData <- read.table("FCovRIcovWB.dat")
> names(longData) <- c("y1", "y2","y3", "y4", "x", "clus")
> model.long <- '
+     level: 1
+         fw =~ y1 + y2 + y3 + y4
+     level: 2
+         fb =~ y1 + y2 + y3 + y4
+         y1 ~~ 0*y1
+         y2 ~~ 0*y2
+         y3 ~~ 0*y3
+         y4 ~~ 0*y4
+ '
> fit.long <- sem(model.long, data = longData, cluster = "clus",
+                 fixed.x = FALSE)
> summary(fit.long)
```

```
lavaan (0.6-1.1173) converged normally after  28 iterations

  Number of observations                          1200
  Number of clusters [clus]                        400

  Estimator                                         ML
  Model Fit Test Statistic                       6.432
  Degrees of freedom                                 8
  P-value (Chi-square)                           0.599
```

```
Parameter Estimates:

  Information                                     Observed
  Observed information based on                    Hessian
  Standard Errors                                 Standard


Level 1 [within]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  fw =~
    y1                1.000
    y2                1.039    0.059   17.552    0.000
    y3                0.942    0.052   18.136    0.000
    y4                0.985    0.058   17.024    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1                0.000
   .y2                0.000
   .y3                0.000
   .y4                0.000
    fw                0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1                0.491    0.027   17.880    0.000
```

```
    .y2              0.473    0.028   16.995    0.000
    .y3              0.481    0.026   18.443    0.000
    .y4              0.521    0.028   18.506    0.000
     fw              0.558    0.051   10.910    0.000


Level 2 [clus]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  fb =~
    y1               1.000
    y2               0.886    0.098    9.087    0.000
    y3               0.977    0.095   10.328    0.000
    y4               0.871    0.098    8.847    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1              -0.040    0.038   -1.045    0.296
   .y2              -0.049    0.037   -1.335    0.182
   .y3              -0.034    0.037   -0.906    0.365
   .y4              -0.034    0.037   -0.926    0.354
    fb               0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1               0.000
   .y2               0.000
```

```
   .y3                 0.000
   .y4                 0.000
    fb                 0.241    0.046    5.294    0.000
```

wide-format syntax

```
> wideData <- matrix(lav_matrix_vecr(longData[,1:5]), 400, 15, byrow = TRUE)
> wideData <- as.data.frame(wideData)
> names(wideData) <- paste(rep(c("y1","y2","y3","y4","x"), 3),
+                          rep(1:3, each = 5), sep = ".")
> model.wide <- '
+     # WITHIN #
+
+     # within factors, common loadings, common (zero) means, common variance
+     fw1 =~ 1*y1.1 + lw2*y2.1 + lw3*y3.1 + lw4*y4.1
+     fw2 =~ 1*y1.2 + lw2*y2.2 + lw3*y3.2 + lw4*y4.2
+     fw3 =~ 1*y1.3 + lw2*y2.3 + lw3*y3.3 + lw4*y4.3
+     fw1 ~~ fvw*fw1
+     fw2 ~~ fvw*fw2
+     fw3 ~~ fvw*fw3
+
+     # uncorrelated fw1, fw2, fw3
+     fw1 ~~ 0*fw2 + 0*fw3; fw2 ~~ 0*fw3
+
+     # within intercepts (fixed to zero)
+     y1.1 + y2.1 + y3.1 + y4.1 ~ 0*1
+     y1.2 + y2.2 + y3.2 + y4.2 ~ 0*1
+     y1.3 + y2.3 + y3.3 + y4.3 ~ 0*1
+
+     # common residual variances
+     y1.1 ~~ rw1*y1.1; y1.2 ~~ rw1*y1.2; y1.3 ~~ rw1*y1.3
+     y2.1 ~~ rw2*y2.1; y2.2 ~~ rw2*y2.2; y2.3 ~~ rw2*y2.3
```

```
+      y3.1 ~~ rw3*y3.1; y3.2 ~~ rw3*y3.2; y3.3 ~~ rw3*y3.3
+      y4.1 ~~ rw4*y4.1; y4.2 ~~ rw4*y4.2; y4.3 ~~ rw4*y4.3
+
+      # BETWEEN #
+
+      # between version of y1,y2,y3,y4
+      by1 =~ 1*y1.1 + 1*y1.2 + 1*y1.3
+      by2 =~ 1*y2.1 + 1*y2.2 + 1*y2.3
+      by3 =~ 1*y3.1 + 1*y3.2 + 1*y3.3
+      by4 =~ 1*y4.1 + 1*y4.2 + 1*y4.3
+
+      # between intercepts
+      by1 + by2 + by3 + by4 ~ 1
+
+      # optional: zero residual variances
+      by1 ~~ 0*by1; by2 ~~ 0*by2; by3 ~~ 0*by3; by4 ~~ 0*by4
+
+      # between factor
+      fb =~ by1 + by2 + by3 + by4
+
+      # not correlated with the within lvs
+      fb ~~ 0*fw1 + 0*fw2 + 0*fw3
+
+ '
> fit.wide <- sem(model.wide, data = wideData, information = "observed")
> summary(fit.wide)

lavaan (0.6-1.1173) converged normally after  27 iterations
```

```
  Number of observations                           400

  Estimator                                         ML
  Model Fit Test Statistic                      69.728
  Degrees of freedom                                74
  P-value (Chi-square)                           0.619

Parameter Estimates:

  Information                                 Observed
  Observed information based on                Hessian
  Standard Errors                             Standard

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  fw1 =~
    y1.1              1.000
    y2.1     (lw2)    1.039    0.059   17.552    0.000
    y3.1     (lw3)    0.942    0.052   18.136    0.000
    y4.1     (lw4)    0.985    0.058   17.024    0.000
  fw2 =~
    y1.2              1.000
    y2.2     (lw2)    1.039    0.059   17.552    0.000
    y3.2     (lw3)    0.942    0.052   18.136    0.000
    y4.2     (lw4)    0.985    0.058   17.024    0.000
  fw3 =~
    y1.3              1.000
```

```
    y2.3      (lw2)    1.039    0.059   17.552    0.000
    y3.3      (lw3)    0.942    0.052   18.136    0.000
    y4.3      (lw4)    0.985    0.058   17.024    0.000
  by1 =~
    y1.1               1.000
    y1.2               1.000
    y1.3               1.000
  by2 =~
    y2.1               1.000
    y2.2               1.000
    y2.3               1.000
  by3 =~
    y3.1               1.000
    y3.2               1.000
    y3.3               1.000
  by4 =~
    y4.1               1.000
    y4.2               1.000
    y4.3               1.000
  fb =~
    by1                1.000
    by2                0.886    0.098    9.087    0.000
    by3                0.977    0.095   10.328    0.000
    by4                0.871    0.098    8.847    0.000

Covariances:
                    Estimate  Std.Err  z-value  P(>|z|)
  fw1 ~~
```

```
    fw2               0.000
    fw3               0.000
  fw2 ~~
    fw3               0.000
  fw1 ~~
    fb                0.000
  fw2 ~~
    fb                0.000
  fw3 ~~
    fb                0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1.1              0.000
   .y2.1              0.000
   .y3.1              0.000
   .y4.1              0.000
   .y1.2              0.000
   .y2.2              0.000
   .y3.2              0.000
   .y4.2              0.000
   .y1.3              0.000
   .y2.3              0.000
   .y3.3              0.000
   .y4.3              0.000
    by1              -0.040    0.038   -1.045    0.296
    by2              -0.049    0.037   -1.335    0.182
    by3              -0.034    0.037   -0.906    0.365
```

```
    by4                -0.034    0.037   -0.926    0.354
    fw1                 0.000
    fw2                 0.000
    fw3                 0.000
    fb                  0.000

Variances:
                     Estimate  Std.Err  z-value  P(>|z|)
    fw1       (fvw)     0.558    0.051   10.910    0.000
    fw2       (fvw)     0.558    0.051   10.910    0.000
    fw3       (fvw)     0.558    0.051   10.910    0.000
   .y1.1      (rw1)     0.491    0.027   17.880    0.000
   .y1.2      (rw1)     0.491    0.027   17.880    0.000
   .y1.3      (rw1)     0.491    0.027   17.880    0.000
   .y2.1      (rw2)     0.473    0.028   16.995    0.000
   .y2.2      (rw2)     0.473    0.028   16.995    0.000
   .y2.3      (rw2)     0.473    0.028   16.995    0.000
   .y3.1      (rw3)     0.481    0.026   18.443    0.000
   .y3.2      (rw3)     0.481    0.026   18.443    0.000
   .y3.3      (rw3)     0.481    0.026   18.443    0.000
   .y4.1      (rw4)     0.521    0.028   18.506    0.000
   .y4.2      (rw4)     0.521    0.028   18.506    0.000
   .y4.3      (rw4)     0.521    0.028   18.506    0.000
    by1                 0.000
    by2                 0.000
    by3                 0.000
    by4                 0.000
    fb                  0.241    0.046    5.294    0.000
```

(optional) wide-format syntax saturated model

```
> model.sat <- '
+     # WITHIN #
+
+     # common variances
+     y1.1 ~~ vw1*y1.1; y1.2 ~~ vw1*y1.2; y1.3 ~~ vw1*y1.3
+     y2.1 ~~ vw2*y2.1; y2.2 ~~ vw2*y2.2; y2.3 ~~ vw2*y2.3
+     y3.1 ~~ vw3*y3.1; y3.2 ~~ vw3*y3.2; y3.3 ~~ vw3*y3.3
+     y4.1 ~~ vw4*y4.1; y4.2 ~~ vw4*y4.2; y4.3 ~~ vw4*y4.3
+
+     # common covariances
+     y1.1 ~~ cw12*y2.1 + cw13*y3.1 + cw14*y4.1; y2.1 ~~ cw23*y3.1 + cw24*y4.1; y3
+     y1.2 ~~ cw12*y2.2 + cw13*y3.2 + cw14*y4.2; y2.2 ~~ cw23*y3.2 + cw24*y4.2; y3
+     y1.3 ~~ cw12*y2.3 + cw13*y3.3 + cw14*y4.3; y2.3 ~~ cw23*y3.3 + cw24*y4.3; y3
+
+     # within means (fixed to zero)
+     y1.1 + y2.1 + y3.1 + y4.1 ~ 0*1
+     y1.2 + y2.2 + y3.2 + y4.2 ~ 0*1
+     y1.3 + y2.3 + y3.3 + y4.3 ~ 0*1
+     # BETWEEN #
+
+     # between version of y1,y2,y3,y4
+     by1 =~ 1*y1.1 + 1*y1.2 + 1*y1.3
+     by2 =~ 1*y2.1 + 1*y2.2 + 1*y2.3
+     by3 =~ 1*y3.1 + 1*y3.2 + 1*y3.3
+     by4 =~ 1*y4.1 + 1*y4.2 + 1*y4.3
+
```

```
+     # between intercepts
+     by1 + by2 + by3 + by4 ~ 1
+
+     # between variances
+     by1 ~~ by1; by2 ~~ by2; by3 ~~ by3; by4 ~~ by4
+
+     # between covariances
+     by1 ~~ by2 + by3 + by4
+     by2 ~~ by3 + by4
+     by3 ~~ by4
+ '
> fit.sat <- sem(model.sat, data = wideData)
> lavTestLRT(fit.sat, fit.wide)
```

```
Chi Square Difference Test

         Df   AIC   BIC  Chisq Chisq diff Df diff Pr(>Chisq)
fit.sat  66 12565 12660 62.948
fit.wide 74 12555 12619 69.728     6.7799       8     0.5606
```

3.2 The 'survey' (design-based) approach

- literature:

  Oberski, D.L. (2014). lavaan.survey: An R package for complex survey analysis of structural equation models. *Journal of Statistical Software, 57*, 1–27.

- 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)

lavaan (0.6-1.1173) 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                                   1.276
    for the Satorra-Bentler correction

Parameter Estimates:

  Information                                 Expected
  Information saturated (h1) model          Structured
  Standard Errors                           Robust.sem

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  numeric =~
    wordlist          4.346    0.338   12.873    0.000
    cards             4.286    0.303   14.147    0.000
    matrices          4.000    0.262   15.248    0.000
  perception =~
    figures           4.104    0.258   15.881    0.000
    animals           4.446    0.254   17.501    0.000
    occupats          4.298    0.292   14.720    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  numeric ~~
    perception        0.676    0.041   16.405    0.000
```

```
Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .wordlist         29.947    0.449   66.671    0.000
   .cards            29.845    0.448   66.552    0.000
   .matrices         29.734    0.437   68.069    0.000
   .figures          30.075    0.457   65.786    0.000
   .animals          30.108    0.451   66.815    0.000
   .occupats         30.008    0.481   62.438    0.000
    numeric           0.000
    perception        0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .wordlist          7.262    0.898    8.087    0.000
   .cards             6.801    0.812    8.381    0.000
   .matrices          8.382    0.747   11.225    0.000
   .figures           9.260    0.839   11.034    0.000
   .animals           5.433    0.648    8.385    0.000
   .occupats          6.923    0.830    8.343    0.000
    numeric           1.000
    perception        1.000
```

3.3 MUthén ML (MUML) estimator using multiple group SEM

sample statistics

- the total covariance (at the population level) can be decomposed as

$$\text{Cov}(\mathbf{y}) = \mathbf{\Sigma}_T = \mathbf{\Sigma}_W + \mathbf{\Sigma}_B$$

- we can decompose the sample data in a similar way:

$$\mathbf{S}_T = \mathbf{S}_W + \mathbf{S}_B$$

- it is tempting to fit a within-cluster model using $\mathbf{S}_W$, and a between-cluster model using $\mathbf{S}_B$, but unfortunately, we can not use $\mathbf{S}_W$ as an estimator of $\mathbf{\Sigma}_W$ and $\mathbf{S}_B$ as an estimator of $\mathbf{\Sigma}_B$
- if we have balanced data (same number of observations in each cluster), we can estimate $\mathbf{\Sigma}_W$ by the pooled within-clusters covariance matrix:

$$\hat{\mathbf{\Sigma}}_W = \mathbf{S}_{PW} = \frac{\sum_{j=1}^{J} \sum_{i=1}^{N_j} (\mathbf{y}_{ji} - \bar{\mathbf{y}}_j)(\mathbf{y}_{ji} - \bar{\mathbf{y}}_j)'}{N - J}$$

where $N = \sum_{j=1}^{J} N_j$ is the total sample size

- the 'scaled' between-clusters covariance matrix can be estimated by

$$\mathbf{S}_B^\star = \frac{\sum_{j=1}^{J} N_j(\bar{\mathbf{y}} - \bar{\mathbf{y}}_j)(\bar{\mathbf{y}} - \bar{\mathbf{y}}_j)'}{J-1}$$

- note: $\mathbf{S}_B^\star$ is *not* an estimate of $\mathbf{\Sigma}_B$ (even in the balanced case)
- it can be shown that $\mathbf{S}_B^\star$ is an estimate of the sum of $\mathbf{\Sigma}_W$ and $\mathbf{\Sigma}_B$ where the latter term is scaled by the common cluster size $s = N_g$:

$$\mathbf{S}_B^\star = \widehat{\mathbf{\Sigma}_W + s \cdot \mathbf{\Sigma}_B}$$

- Muthén (1989, 1990, 1994) suggested to fit a two-level SEM model by using a conventional multigroup SEM analysis based on $\mathbf{S}_{PW}$ and $\mathbf{S}_B^\star$
- in the balanced case, this results in ML estimates; in the unbalanced case this is called MUML

the MUML setup

- 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 replace s by

$$s^{\star} = [N^2 - \sum_{j=1}^{J} n_j^2]/(N(J-1))$$

sample statistics in R

```
>  library(lavaan)
> Data <- read.table("hoxdata1.dat",header = FALSE)
> ov.names <- c("y1","y2","y3","y4","y5","y6")
> names(Data) <- c(ov.names, "cluster")
> cluster.size <- as.integer(table(Data$cluster))
> nClusters <- length(cluster.size)
> N <- sum(cluster.size)
> # between sample statistics
> DataB <- with(Data, aggregate(Data[,ov.names],
+                        by = list(cluster), 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)
> # within
> cluster.idx <- as.integer(as.factor(Data[,"cluster"]))
> 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)) * 1/(N * (nClusters - 1))
> # 20
>
> # B.cov.star
> B.cov.star <- (B.cov - W.cov)/s
```

setting up the syntax for MUML in lavaan

```
> model <- '
+     ## within model
+
+       # factor loadings
+       f1 =~ c(l1,l1)*y1 + c(l2,l2)*y2 + c(l3,l3)*y3
+       f2 =~ c(l4,l4)*y4 + c(l5,l5)*y5 + c(l6,l6)*y6
+
+       # residual variances
+       y1 ~~ c(r1,r1)*y1; y2 ~~ c(r2,r2)*y2; y3 ~~ c(r3,r3)*y3
+       y4 ~~ c(r4,r4)*y4; y5 ~~ c(r5,r5)*y5; y6 ~~ c(r6,r6)*y6
+
+       # factor variances/covariances
+       f1 ~~ c(1,1)*f1; f2 ~~ c(1,1)*f2; f1 ~~ c(v12,v12)*f2
+
+       # means
+       # y1 + y2 + y3 + y4 + y5 + y6 ~ 0*1
+       # f1 + f2 ~ 0*1
+
+     ## between model
+
+       y1b =~ c(0,4.472136)*y1
+       y2b =~ c(0,4.472136)*y2
+       y3b =~ c(0,4.472136)*y3
+       y4b =~ c(0,4.472136)*y4
+       y5b =~ c(0,4.472136)*y5
+       y6b =~ c(0,4.472136)*y6
```

```
+
+        fb =~ c(0,NA)*y1b + c(0,NA)*y2b + c(0,NA)*y3b +
+              c(0,NA)*y4b + c(0,NA)*y5b + c(0,NA)*y6b
+
+        # residuals
+        y1b ~~ c(0,NA)*y1b; y2b ~~ c(0,NA)*y2b; y3b ~~ c(0,NA)*y3b
+        y4b ~~ c(0,NA)*y4b; y5b ~~ c(0,NA)*y5b; y6b ~~ c(0,NA)*y6b
+
+        # fb variance
+        fb ~~ c(0,1)*fb
+
+        # means
+        y1b ~ c(0,NA)*1
+        y2b ~ c(0,NA)*1
+        y3b ~ c(0,NA)*1
+        y4b ~ c(0,NA)*1
+        y5b ~ c(0,NA)*1
+        y6b ~ c(0,NA)*1
+
+        fb ~ c(0,0)*1
+ '
> fit <- lavaan(model, sample.cov = list(within=W.cov, between=B.cov),
+                sample.cov.rescale = FALSE,
+                sample.mean = list(W.mean, B.mean),
+             sample.nobs = list(N-nClusters, nClusters), std.lv = TRUE)
> summary(fit)

lavaan (0.6-1.1173) converged normally after  69 iterations
```

```
  Number of observations per group
  within                                          1900
  between                                          100

  Estimator                                         ML
  Model Fit Test Statistic                      10.763
  Degrees of freedom                                23
  P-value (Chi-square)                           0.985

Chi-square for each group:

  within                                         2.668
  between                                        8.096

Parameter Estimates:

  Information                                 Expected
  Information saturated (h1) model          Structured
  Standard Errors                             Standard


Group 1 [within]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f1 =~
    y1        (l1)    0.300    0.016   18.507    0.000
```

```
    y2          (l2)     0.397     0.019   20.955     0.000
    y3          (l3)     0.494     0.022   22.267     0.000
  f2 =~
    y4          (l4)     0.305     0.016   19.001     0.000
    y5          (l5)     0.382     0.019   20.394     0.000
    y6          (l6)     0.527     0.022   23.579     0.000
  y1b =~
    y1                   0.000
  y2b =~
    y2                   0.000
  y3b =~
    y3                   0.000
  y4b =~
    y4                   0.000
  y5b =~
    y5                   0.000
  y6b =~
    y6                   0.000
  fb =~
    y1b                  0.000
    y2b                  0.000
    y3b                  0.000
    y4b                  0.000
    y5b                  0.000
    y6b                  0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
```

```
  f1 ~~
    f2        (v12)   -0.013    0.034   -0.384    0.701

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
    y1b               0.000
    y2b               0.000
    y3b               0.000
    y4b               0.000
    y5b               0.000
    y6b               0.000
    fb                0.000
   .y1                0.000
   .y2                0.000
   .y3                0.000
   .y4                0.000
   .y5                0.000
   .y6                0.000
    f1                0.000
    f2                0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1         (r1)   0.251    0.010   24.509    0.000
   .y2         (r2)   0.247    0.014   18.292    0.000
   .y3         (r3)   0.262    0.019   13.884    0.000
   .y4         (r4)   0.243    0.010   24.187    0.000
   .y5         (r5)   0.281    0.013   20.996    0.000
```

```
   .y6        (r6)    0.216    0.020   10.811    0.000
    f1                1.000
    f2                1.000
    y1b               0.000
    y2b               0.000
    y3b               0.000
    y4b               0.000
    y5b               0.000
    y6b               0.000
    fb                0.000


Group 2 [between]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f1 =~
    y1        (l1)    0.300    0.016   18.507    0.000
    y2        (l2)    0.397    0.019   20.955    0.000
    y3        (l3)    0.494    0.022   22.267    0.000
  f2 =~
    y4        (l4)    0.305    0.016   19.001    0.000
    y5        (l5)    0.382    0.019   20.394    0.000
    y6        (l6)    0.527    0.022   23.579    0.000
  y1b =~
    y1                4.472
  y2b =~
    y2                4.472
```

```
  y3b =~
    y3                4.472
  y4b =~
    y4                4.472
  y5b =~
    y5                4.472
  y6b =~
    y6                4.472
  fb =~
    y1b               0.468    0.071    6.603    0.000
    y2b               0.449    0.070    6.384    0.000
    y3b               0.260    0.065    3.986    0.000
    y4b               0.530    0.069    7.671    0.000
    y5b               0.425    0.069    6.123    0.000
    y6b               0.282    0.065    4.368    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  f1 ~~
    f2        (v12)  -0.013    0.034   -0.384    0.701

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
    y1b              -0.003    0.071   -0.038    0.969
    y2b              -0.000    0.070   -0.007    0.995
    y3b               0.014    0.061    0.236    0.813
    y4b              -0.004    0.071   -0.062    0.950
    y5b               0.006    0.068    0.087    0.930
```

```
    y6b               0.002    0.061    0.026    0.979
    fb                0.000
   .y1                0.000
   .y2                0.000
   .y3                0.000
   .y4                0.000
   .y5                0.000
   .y6                0.000
    f1                0.000
    f2                0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1        (r1)    0.251    0.010   24.509    0.000
   .y2        (r2)    0.247    0.014   18.292    0.000
   .y3        (r3)    0.262    0.019   13.884    0.000
   .y4        (r4)    0.243    0.010   24.187    0.000
   .y5        (r5)    0.281    0.013   20.996    0.000
   .y6        (r6)    0.216    0.020   10.811    0.000
    f1                1.000
    f2                1.000
    y1b               0.264    0.050    5.291    0.000
    y2b               0.266    0.049    5.383    0.000
    y3b               0.277    0.045    6.171    0.000
    y4b               0.205    0.047    4.356    0.000
    y5b               0.267    0.048    5.525    0.000
    y6b               0.266    0.044    6.102    0.000
    fb                1.000
```

same model using ML

```
> model.long <- '
+     level: 1
+         f1 =~ y1 + y2 + y3
+         f2 =~ y4 + y5 + y6
+     level: 2
+         f1 =~ y1 + y2 + y3
+         f2 =~ y4 + y5 + y6
+ '
> fit.long <- sem(model.long, data = Data, cluster = "cluster",
+                 std.lv = TRUE)
> summary(fit.long)
```

```
lavaan (0.6-1.1173) converged normally after  47 iterations

  Number of observations                          2000
  Number of clusters [cluster]                     100

  Estimator                                         ML
  Model Fit Test Statistic                      10.413
  Degrees of freedom                                16
  P-value (Chi-square)                           0.844

Parameter Estimates:

  Information                                 Observed
  Observed information based on                Hessian
```

```
  Standard Errors                            Standard


Level 1 [within]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f1 =~
    y1                0.300    0.016   18.501    0.000
    y2                0.397    0.019   20.949    0.000
    y3                0.494    0.022   22.260    0.000
  f2 =~
    y4                0.304    0.016   18.995    0.000
    y5                0.382    0.019   20.382    0.000
    y6                0.527    0.022   23.556    0.000

Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  f1 ~~
    f2               -0.013    0.034   -0.374    0.709

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1                0.000
   .y2                0.000
   .y3                0.000
   .y4                0.000
   .y5                0.000
```

```
   .y6              0.000
    f1              0.000
    f2              0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1              0.251    0.010   24.510    0.000
   .y2              0.247    0.014   18.294    0.000
   .y3              0.262    0.019   13.877    0.000
   .y4              0.243    0.010   24.190    0.000
   .y5              0.281    0.013   20.991    0.000
   .y6              0.216    0.020   10.791    0.000
    f1              1.000
    f2              1.000


Level 2 [cluster]:

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  f1 =~
    y1              0.475    0.074    6.452    0.000
    y2              0.457    0.073    6.300    0.000
    y3              0.260    0.066    3.946    0.000
  f2 =~
    y4              0.539    0.072    7.468    0.000
    y5              0.430    0.070    6.124    0.000
    y6              0.278    0.066    4.226    0.000
```

```
Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  f1 ~~
    f2                0.944    0.094   10.054    0.000

Intercepts:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1               -0.012    0.070   -0.172    0.863
   .y2               -0.002    0.070   -0.029    0.976
   .y3                0.064    0.061    1.062    0.288
   .y4               -0.020    0.071   -0.280    0.779
   .y5                0.027    0.068    0.393    0.695
   .y6                0.007    0.061    0.116    0.908
    f1                0.000
    f2                0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .y1                0.252    0.053    4.763    0.000
   .y2                0.254    0.051    4.941    0.000
   .y3                0.274    0.045    6.123    0.000
   .y4                0.191    0.052    3.657    0.000
   .y5                0.258    0.049    5.315    0.000
   .y6                0.265    0.044    6.028    0.000
    f1                1.000
    f2                1.000
```

4 Last slide

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

  McNeish, D.M., & Stapleton, L.M. (2016). The effect of small sample size on two-level model estimates: A review and illustration. *Educational Psychology Review, 28*, 295–314.

- topics not discussed in this workshop:
  - construct reliability in the multilevel setting
  - mediation and moderation
  - random slopes
  - categorical outcomes
  - missing data
  - the gllamm framework
- when will lavaan 0.6 be officially released?