Introduction to Structural Equation Modeling using lavaan

Intro and regression

R. M. Kuiper (and many others)

Department of Methodology & Statistics
Utrecht University

Outline of this lecture

SEM

SAPI

Lavaan commands

Regression

Path model

Steps

Technical output

Steps Ctd.

Model fit

The end

Table of Contents

SEM

SAPI

Lavaan commands

Regression

Path model

Steps

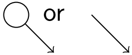
Technical output

Steps Ctd.

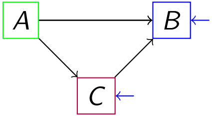
Model fit

The end

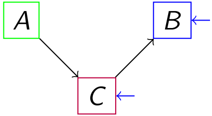
Path Diagram: Graphical representation of SEM

$\boxed{y1}$	Oberved variable
$F1$	Construct = latent (unmeasured) variable / factor
$\longrightarrow$	Direct relationship
$\longleftrightarrow$	Covariance
	Variance
	Residual variance / measurement error

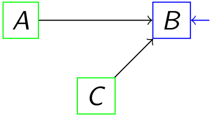
Path models



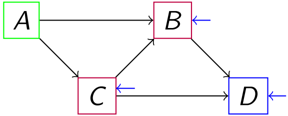
Partial mediation



Full mediation

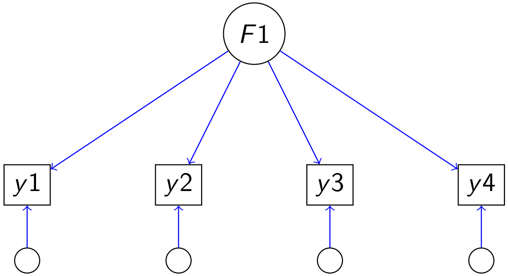


Multiple regression



More complex path models

Measurement models



SEM

- From theory to model (and path diagram)
- Compare

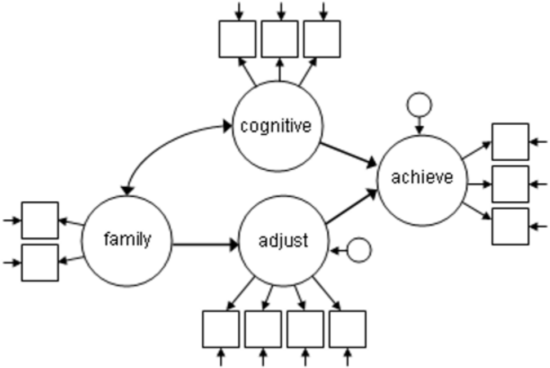


Table of Contents

SEM

SAPI

Lavaan commands

Regression

Path model

Steps

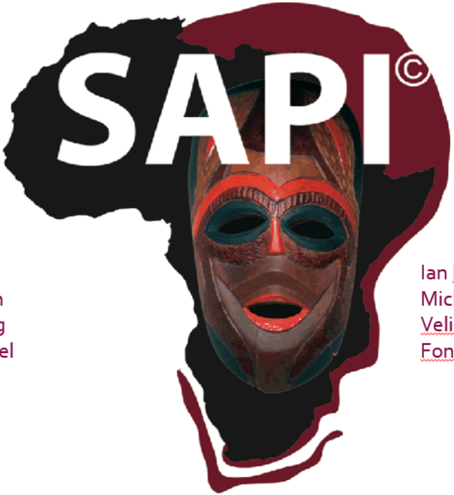
Technical output

Steps Ctd.

Model fit

The end

Example: South African Personality Inventory Project (SAPI)



Carin Hill
Leon Jackson
Deon Meiring
J. Aleweyn Nel

Ian Rothmann
Michael Temane
Velichko H. Valchev
Fons J. R. van de Vijver

Nel, J. A., Valchev, V. H., Rothmann, S., van de Vijver, F. J. R., Meiring, D., & de Bruin, G. P. (2012). Exploring the personality structure in the 11 languages of South Africa. *Journal of Personality, 80*, 915–948.

SAPI details for this lecture

Our data:

- Selection of 1000 participants.
- Selection of 7 out of 262 personality items:
  - Regression:
    - Outcome: Q77, I enjoy telling funny stories
    - Predictor: Age
  - Path model:
    - Outcome: Q196, I make others laugh
    - Mediator: Q77, I enjoy telling funny stories
    - Predictor: Age
  - Later, we use: Age, ReadAb, Q44, Q63, and Q76.

Table of Contents

SEM

SAPI

Lavaan commands

Regression

Path model

Steps

Technical output

Steps Ctd.

Model fit

The end

Steps to take for running a model in lavaan

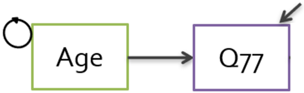
- 1. Loading data into R
- 2. Data screening:
  e.g., check measurement levels, missing data (notation), and correlations.
- 3. Draw your model
- 4. Specify your model in lavaan
- 5. Fit the model with lavaan
- 6. Plot the lavaan model in R
- 7. Ensure model specification is correct using plots and technical output.
  If not, correct and proceed from Step 4 on.
- 8. Check model assumptions:
  e.g., check residuals (histogram or Q-Q plot).
- 9. Acquire the summary from lavaan
- And more steps, discussed later on.

1. Loading data into R

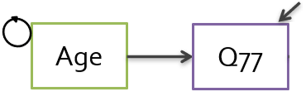
For example, in case of a .txt file:

```
data_sapi <- read.table("Sapi.txt", header = T)

data_sapi[sapply(data_sapi,
    function(x) as.character(x) %in% c("-999") )] <- NA
```

3. Draw your model



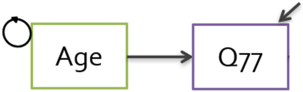
4. Specify your model



- Translate $\rightarrow$ into $\sim$.
- Translate 'residual variance notation' into $\sim\sim$.
  Note = residual variance = a variance = special case covariance.
- Translate $\leftrightarrow$ (= covariance) into $\sim\sim$.
- Do not forget about intercepts: Translate into $\sim$.

- Translate 'circled arrow' (= variance) into $\sim\sim$.
  Note = variance = special case covariance.
- Do not forget about means: Translate into $\sim$.

4. Specify your model: regression SAPI



- Translate $\rightarrow$ into $\sim$.
- Translate 'residual variance notation' into $\sim\sim$.
  Note = residual variance = a variance = special case covariance.
- Translate $\leftrightarrow$ (= covariance) into $\sim\sim$.
- Do not forget about intercepts: Translate into $\sim$.

```
mod.regr <- '
  Q77 ~ Age   # regression
  Q77 ~~ Q77 # residual variance
  Q77 ~ 1     # intercept
'
```

4. Specify your model: regression general

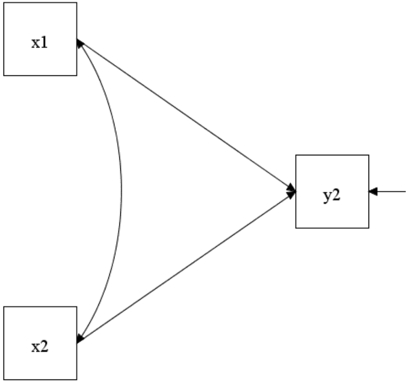
A regression model is specified as follows:
dependent ~ predictor1 + predictor2 + etc.

Additionally, one needs to specify that the error (of the dependent variable) has a variance:
dependent ~~ dependent

If needed, one can specify to include an intercept:
dependent ~ 1

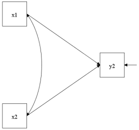
Note that one include intercept directly:
dependent ~ 1 + predictor1 + predictor2 + etc.

4. Specify your model: regression DIY



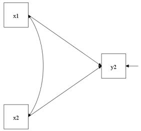
- Do you recognize this model?
- Specify this in lavaan code.
  Do not forget the parameters that are not drawn, but do are there.

4. Specify your model: regression answer




```
mod.regr_DIY <- '
  y2 ~ 1 + x1 + x2  # intercept + regression
  y2 ~~ y2          # residual variance

  # Make it a habit to also specify:
  x1 ~ 1            # mean x1
  x2 ~ 1            # mean x2
  x1 ~~ x1          # variance x1
  x1 ~~ x2          # covariance x1 and x2
  x2 ~~ x2          # variance x2
'
```

4. Specify your model: regression answer 2




```
mod.regr_DIY <- '
  y2 ~ 1 + x1 + x2  # intercept + regression
  y2 ~~ y2          # residual variance

  # Make it a habit to also specify:
  x1 + x2 ~ 1       # means predictors - DO specify this!
  x1 + x2 ~~ x1     # (co)variance predictors
  x2 ~~ x2          # variance x2
'
```

Note: As opposed to what I say in the video, do specify the means for the predictors explicitly; otherwise they will be set to 0.

4. Specify your model: regression answer general

```
y ~ 1 + x_1 + x_2 + ... + x_k  # intercept + regression
y ~~ y                         # residual variance

# (co)variances of exogenous variables:
x_1 + x_2 + ... + x_k ~~ x_1
      x_2 + ... + x_k ~~ x_2
                      ...
                  x_k ~~ x_k
```

5. Fit the regression model

```
fit.regr <- lavaan(model = mod.regr, data = data_sapi)
```

5. Fit the regression model: All code

```
# Data
data_sapi <- read.table("Sapi.txt", header = T)
data_sapi[sapply(data_sapi,
    function(x) as.character(x) %in% c("-999") )] <- NA

# Model
mod.regr <- '
  Q77 ~ Age   # regression
  Q77 ~~ Q77 # residual variance
  Q77 ~ 1     # intercept
'

# Fit model
fit.regr <- lavaan(model = mod.regr, data = data_sapi)
```

6. Plot the lavaan model in R

- lavaanPlot
- graph_sem from tidySEM
- semPaths from semPlot
- probably many more...

Plot with lavaanPlot

```
if (!require("lavaanPlot")) install.packages("lavaanPlot")
library(lavaanPlot)
```

```
lavaanPlot(model = fit.regr,
           graph_options = list(overlap = "true",
                                fontsize = "20"),
           node_options = list(shape = "box",
                               fontname = "Helvetica"),
           edge_options = list(color = "grey"),
           coefs = T, stand = T, covs = T,
           )
```

Plot with tidySEM

```
library(tidySEM)
if (!require("tidySEM")) install.packages("tidySEM")
```

```
graph_sem(fit.regr)
```

Plot with semPlot

```
if (!require("semPlot")) install.packages("semPlot")
library(semPlot)
```

```
semPaths(fit.regr, "par", weighted = FALSE, nCharNodes = 7,
         shapeMan = "rectangle", sizeMan = 8, sizeMan2 = 5)
```

9. Acquiring the summary

```
summary(fit.regr, fit.measures = TRUE,
        ci = TRUE, rsquare = TRUE)
```

```
parameterEstimates(fit.regr)

inspect(fit.regr, 'r2')
# As an example, there are more.

fitMeasures(fit.regr,
            c("npar", "logl", "unrestricted.logl",
              "chisq", "pvalue",
              "cfi", "tli", "rmsea"))
# As an example, there are more.
```

Table of Contents

SEM

SAPI

Lavaan commands

Regression

Path model

Steps

Technical output

Steps Ctd.

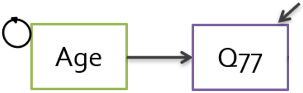
Model fit

The end

Analyzing the SAPI data: Regression

Variables:

- Outcome: Q77, I enjoy telling funny stories
- Predictor: Age



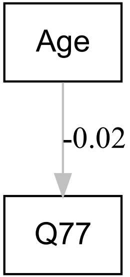
Specifying and fitting regression model in lavaan

```
mod.regr <- '
  Q77 ~  Age # regression
  Q77 ~~ Q77 # residual variance
  Q77 ~  1   # intercept
'
```

```
fit.regr <- lavaan(model = mod.regr, data = data_sapi)
```

Plot with lavaanPlot

```
lavaanPlot(model = fit.regr,
         graph_options = list(overlap = "true",
                              fontsize = "20"),
         node_options = list(shape = "box",
                             fontname = "Helvetica"),
         edge_options = list(color = "grey"),
         coefs = T, stand = F, covs = T,
         )
```



Summary regression model in lavaan

```
summary(fit.regr, fit.measures = TRUE,
        ci = TRUE, rsquare = TRUE)
```

Estimates:

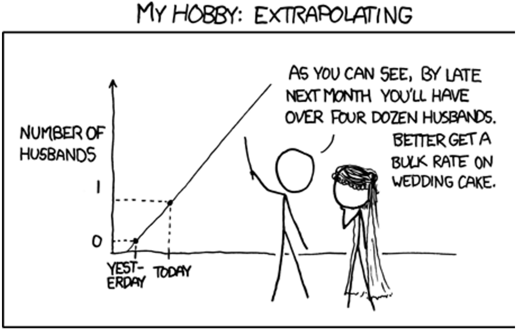
```
parameterEstimates(fit.regr)[c(1,2,3),] # or ...[1:3,]

##   lhs op rhs    est    se      z pvalue ci.lower ci.upper
## 1 Q77  ~ Age -0.022 0.004 -5.275      0   -0.030   -0.014
## 2 Q77 ~~ Q77  1.184 0.068 17.536      0    1.052    1.317
## 3 Q77 ~1      4.275 0.135 31.657      0    4.011    4.540
```

Regression equation

```
parameterEstimates(fit.regr)[c(1,2,3),] # or ...[1:3,]

##    lhs op rhs    est    se      z pvalue ci.lower ci.upper
## 1 Q77  ~  Age -0.022 0.004 -5.275      0   -0.030   -0.014
## 2 Q77 ~~  Q77  1.184 0.068 17.536      0    1.052    1.317
## 3 Q77 ~1       4.275 0.135 31.657      0    4.011    4.540
```

General regression equation (for person i):
$Q77_i = B_0 + B_1 * Age_i + e_i$, with $e_i \sim N(0, \sigma^2)$.
Or: $\hat{Q77}_i = B_0 + B_1 * Age_i$.

Regression equation (for person i):
$Q77_i = 4.275 - 0.022 * Age_i + e_i$, with $e \sim N(0, 1.184)$.


MY HOBBY: EXTRAPOLATING
NUMBER OF HUSBANDS
1
0
YEST-ERDAY
TODAY
AS YOU CAN SEE, BY LATE NEXT MONTH YOU'LL HAVE OVER FOUR DOZEN HUSBANDS.
BETTER GET A BULK RATE ON WEDDING CAKE.

REGRESSION

Summary regression model in lavaan Ctd.

R^2 = R-squared = proportion of shared/explained variance:

```
inspect(fit.regr, 'r2')

##   Q77
## 0.043
```

Intermezzo: Correlation Age and Q77 and R^2 in R

```
corr <- cor(data_sapi[, 2], data_sapi[, 9],
            use = "complete.obs")

round(corr, 3)    # = correlation between Age and Q77

## [1] -0.208
```

```
round(corr^2, 3) # = R^2

## [1] 0.043

# R^2 = proportion of shared/explained variance.
# This is the R^2 in a regression with one predictor.
```

In a regression with one outcome and one predictor,
the R^2 is the square of their correlation.

Intermezzo: Correlation Age and Q77 and R^2 in R Ctd.

Remark:
I would not use lavaan for correlations, but you can:

```
mod.corr <- '
  Q77 ~~ Q77
  Age ~~ Age
  Q77 ~~ Age # covariance
  # If standardized, then correlation
'
fit.corr <- lavaan(model = mod.corr, data = data_sapi)
parameterEstimates(fit.corr, stand=T)[3,'std.all']

## [1] -0.2080665
```

Message intermezzo:
In a regression with one outcome and one predictor,
the R^2 is the square of their correlation.

Table of Contents

SEM

SAPI

Lavaan commands

Regression

Path model

Steps

Technical output

Steps Ctd.

Model fit

The end

Analyzing the SAPI data: Path model (Mediation)

Variables:

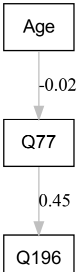
- Outcome: Q196, I make others laugh
- Mediator: Q77, I enjoy telling funny stories
- Predictor: Age

Specification path model in lavaan

```
model.path <- '
  # regressions
  Q77  ~ Age
  Q196 ~ Q77

  # residual variances
  Q77  ~~ Q77
  Q196 ~~ Q196

  # variance of predictor
  Age ~~ Age
  # mean of predictor
  Age ~ 1

  ## intercepts
  Q77  ~ 1
  Q196 ~ 1
'
```

Plot with lavaanPlot

```
lavaanPlot(model = fit_path,
           graph_options = list(overlap = "true",
                                fontsize = "20"),
           node_options = list(shape = "box",
                               fontname = "Helvetica"),
           edge_options = list(color = "grey"),
           coefs = T, stand = F, covs = T,
           ) # Note: stand = F
```



Summary path model in lavaan

```r
summary(fit_path, fit.measures = TRUE,
        ci = TRUE, rsquare = TRUE)
```

Estimates

```r
parameterEstimates(fit_path)[,-7] # i.e., without p-value

##     lhs op  rhs     est    se      z ci.lower ci.upper
## 1   Q77  ~  Age  -0.021 0.004 -4.971   -0.029   -0.013
## 2 Q196   ~  Q77   0.451 0.027 16.724    0.398    0.504
## 3   Q77 ~~  Q77   1.179 0.067 17.479    1.047    1.312
## 4 Q196  ~~ Q196   0.545 0.031 17.479    0.484    0.606
## 5   Age ~~  Age 108.692 6.219 17.479   96.504  120.881
## 6   Age ~1       30.612 0.422 72.580   29.785   31.439
## 7   Q77 ~1        4.248 0.136 31.175    3.981    4.516
## 8 Q196  ~1        2.153 0.102 21.167    1.954    2.352
```

Summary path model in lavaan Ctd.

R^2 = R-squared = proportion of shared/explained variance:

```
inspect(fit_path, 'r2')

##   Q77  Q196
## 0.039 0.314
```

Table of Contents

SEM

SAPI

Lavaan commands

Regression

Path model

Steps

Technical output

Steps Ctd.

Model fit

The end

Step by step (repition): Step 1

Load data into R

Step 2

Data screening
e.g., check measurement levels, missing data (notation), and correlations.

Not part of lavaan,
plenty of information on the internet.

Step 3

Draw your model

Step 4

Specify your model into lavaan syntax
i.e., Translate your drawing into lavaan syntax

Step 5

Fit the model with lavaan

Step 6

Plot the lavaan model

Step 7

Ensure model specification is correct using

- Plots (for example, using lavaanPlot)
- Technical output (using lavInspect), discussed in next intermezzo.

Step 8

Check model assumptions

Not part of lavaan,
plenty of information on the internet.

Step 9

Acquire the summary from lavaan

Table of Contents

SEM

SAPI

Lavaan commands

Regression

Path model

Steps

Technical output

Steps Ctd.

Model fit

The end

Intermezzo: Technical output


PARENTAL
ADVISORY
STATISTICAL

Degrees of freedom

Degrees of freedom (df) =

Number of unique elements - Number of model parameters = knowns - unknowns.

$df = ((p(p+1)/2) + p) - npar$,
with p the number of variables in the model.

Number of model parameters - SAPI regression

Number of model parameters (unknowns):

```
fitMeasures(fit.regr, "npar")

## npar
##    3
```

Which ones can be inspected in the technical output:

```
lavInspect(fit.regr)
```

More explanation later on.

For now: 3 model parameters estimated.

Number of unique elements

Note: Raw data can be summarized by

- a variance matrix (S)
- a mean vector (m)

Number of unique elements (knowns):

- Unique elements in variance matrix $= p(p+1)/2$
- Possibly: Unique elements in mean = p

with p the number of variables in the model.

In SAPI regression ($p = 2$):

- Unique elements S: $p(p+1)/2 = 2 * 3/2 = 3$
- Unique elements m: $p = 2$

Number of model parameters $= 3 + 2 = 5$.

Degrees of freedom - SAPI regression

Degrees of freedom (df) =

Number of unique elements - Number of model parameters =

5 - 3 = 2.

BUT:

```
fitMeasures(fit.regr, "df") # df = 0

## df
##  0
```

Lavaan defaults

```
parameterEstimates(fit.regr)[,1:7] # without ci

##   lhs op rhs     est    se      z pvalue
## 1 Q77  ~ Age  -0.022 0.004 -5.275      0
## 2 Q77 ~~ Q77   1.184 0.068 17.536      0
## 3 Q77 ~1       4.275 0.135 31.657      0
## 4 Age ~~ Age 111.304 0.000     NA     NA
## 5 Age ~1      30.706 0.000     NA     NA
```

By default, lavaan secretly estimates

- the means and variances of exogenous observed variables
- the covariances between exogenous variables

as does Mplus.

SAPI regression: Fully specified

```
mod.regr_full <- '
  Q77 ~  Age # regression
  Q77 ~~ Q77 # residual variance
  Q77 ~  1   # intercept

  # (co)variances and means predictor(s)
  Age ~~ Age # variance predictor
  Age ~ 1    # mean predictor
  #
  # Estimated by lavaan by default,
  # but now also part of lavInspect()

'
fit.regr_full <- lavaan(model = mod.regr_full,
                        data = data_sapi)
```

SAPI regression: Same estimates of course

```
parameterEstimates(fit.regr)[,1:7]

##   lhs op rhs     est    se      z pvalue
## 1 Q77  ~ Age  -0.022 0.004 -5.275      0
## 2 Q77 ~~ Q77   1.184 0.068 17.536      0
## 3 Q77 ~1       4.275 0.135 31.657      0
## 4 Age ~~ Age 111.304 0.000     NA     NA
## 5 Age ~1      30.706 0.000     NA     NA
```

```
parameterEstimates(fit.regr_full)[,1:7]

##   lhs op rhs     est    se      z pvalue
## 1 Q77  ~ Age  -0.022 0.004 -5.275      0
## 2 Q77 ~~ Q77   1.184 0.068 17.536      0
## 3 Q77 ~1       4.275 0.135 31.657      0
## 4 Age ~~ Age 111.304 6.347 17.536      0
## 5 Age ~1      30.706 0.425 72.177      0
```

Number of model parameters - SAPI regression

Now, number of model parameters: 5, see

```
fitMeasures(fit.regr_full, "npar")

## npar
##    5
```

Then, indeed, $df = 5 - 5 = 0$.

This implies that we have a saturated model:

Number of unique elements = Number of model parameters.
$S_{yy}, S_{yx}, S_{xx}, m_y, m_y$ vs $\alpha, \beta, \sigma^2_\epsilon, S_{xx}, m_y$

Technical output

Which model parameters are estimated can be inspected in the technical output:

```
lavInspect(fit.regr_full)
```

Technical output - SAPI regression (Part 1)

```
lavInspect(fit.regr_full)[1:3]

## $lambda
##     Q77 Age
## Q77   0   0
## Age   0   0
##
## $theta
##     Q77 Age
## Q77   0
## Age   0   0
##
## $psi
##     Q77 Age
## Q77   2
## Age   0   4
```

Technical output - SAPI regression (Part 2)

```
lavInspect(fit.regr_full)[4:6]

## $beta
##     Q77 Age
## Q77   0   1
## Age   0   0
##
## $nu
##     intrcp
## Q77      0
## Age      0
##
## $alpha
##     intrcp
## Q77      3
## Age      5
```

Technical output - Model parameters

- lambda
- theta
- psi
- beta
- nu
- alpha

Note: Enumeration/Indication of parameters, not parameter values!

Technical output: General overview

Measurement equation: $y_i = \nu + \Lambda\eta_i + \epsilon_i, \;\; \epsilon_i \sim \mathrm{N}(\mathbf{0}, \Theta)$

Structural equation: $\eta_i = \alpha + B\eta_i + \zeta_i, \;\; \zeta_i \sim \mathrm{N}(\mathbf{0}, \Psi)$

Modeled covariance matrix: $\Sigma = \Theta + \Lambda\Psi\Lambda^T$

- ν (**nu**), vector with means/intercepts of observed variables.
- Λ (**Lambda**), matrix with factor loadings relating observed variables to latent variables.
- Θ (**Theta**), variance-covariance matrix of residuals of observed variables (ϵ_i).
- α (**alpha**), vector with intercepts/means of latent variables.
- B (**Beta**), matrix with structural parameters (β's).
- Ψ (**Psi**), variance-covariance matrix of residuals of latent variables (ζ_i): variances of exogenous variables and residual variances of endogenous variables.

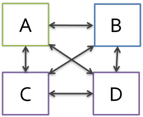
Saturated model (p = 4) - with means

Data

S =

	A	B	C	D
A	σ^2_A			
B	σ_{BA}	σ^2_B		
C	σ_{CA}	σ_{BC}	σ^2_C	
D	σ_{DA}	σ_{DB}	σ_{DC}	σ^2_D

Model



Total unique elements:

- Unique elements S: $p(p+1)/2 = 4 * 5/2 = 10$
- Unique elements m: $p = 4$

$10+4 = 14$

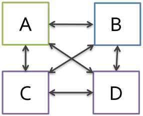
Saturated model (p = 4) - with means

Data

S =

	A	B	C	D
A	σ^2_A			
B	σ_{BA}	σ^2_B		
C	σ_{CA}	σ_{BC}	σ^2_C	
D	σ_{DA}	σ_{DB}	σ_{DC}	σ^2_D

Model



Number of model parameters:

- cross relations / covariances: 6
- variances: 4
- means: 4

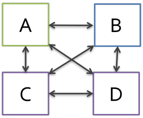
$(6+4)+4 = 14$

Saturated model (p = 4) - with means

Data

$$
\mathbf{S} = \begin{array}{c|cccc}
 & A & B & C & D \\
\hline
A & \sigma^2_A & & & \\
B & \sigma_{BA} & \sigma^2_B & & \\
C & \sigma_{CA} & \sigma_{BC} & \sigma^2_C & \\
D & \sigma_{DA} & \sigma_{DB} & \sigma_{DC} & \sigma^2_D
\end{array}
$$

Model



Total unique elements:
$10+4 = 14$

Number of model parameters:
$(6+4)+4 = 14$

Hence, $df = 14 - 14 = 0$.

Example saturated model (p = 4)

Variables (observed): Age, ReadAb, Q44, and Q63

```
# Variances \& Covariances for residuals of observed variables
lavInspect(fit_sat)$theta

##         Age ReadAb Q44 Q63
## Age       1
## ReadAb    5      2
## Q44       6      8   3
## Q63       7      9  10   4
```

```
# Means / Intercepts of observed variables
lavInspect(fit_sat)$nu

##         intrcp
## Age         11
## ReadAb      12
## Q44         13
## Q63         14
```

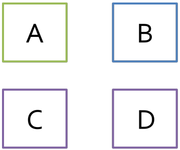
Example saturated model ($p = 4$) Ctd.

```
fitMeasures(fit_sat,
            c("npar", "logl", "unrestricted.logl",
              "chisq", "pvalue",
              "cfi", "tli", "rmsea"))

##               npar               logl unrestricted.logl
##             14.000          -4498.184         -4498.184
##             pvalue                cfi               tli
##                 NA              1.000             1.000
```

Note: At the end, more details about model fit.

Baseline model ($p = 4$)

- Variables: A, B, C, and D
  - How many unique elements?
  - $4(4+1)/2 + 4 = 14$
- Baseline model = Unrelated variables
  - How many model parameters?
  - 4 variances and 4 means = 8 parameters
- How many df?
  - df = 14-8=6



Example baseline model (p = 4)

Variables: Age, ReadAb, Q44, and Q63

```
lavInspect(fit_base4)$theta

##        Age ReadAb Q44 Q63
## Age      1
## ReadAb   0      2
## Q44      0      0   3
## Q63      0      0   0   4
```

```
lavInspect(fit_base4)$nu

##        intrcp
## Age         5
## ReadAb      6
## Q44         7
## Q63         8
```

Example baseline model ($p = 4$) Ctd.

```
fitMeasures(fit_base4,
            c("npar", "logl", "unrestricted.logl",
              "chisq", "pvalue",
              "cfi", "tli", "rmsea"))
```

```
##                npar               logl unrestricted.logl
##               8.000          -4530.256         -4498.184
##              pvalue                cfi               tli
##               0.000              0.000             0.000
```

No deviation from the baseline model,
because we estimated it: CFI $= 0$ and TLI $= 0$.

Note: At the end, more details about model fit.

Saturated model (p = 5): DIY

- Variables: A, B, C, D, and E
  - How many unique elements?
    -
    -
  -
- Saturated model = All variables related
  - How many model parameters?
    -
    -
    -
  -
- How many df?
  -

Saturated model (p = 5): Answer

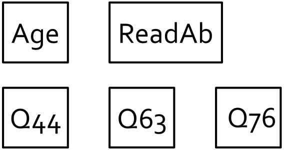
- Variables: A, B, C, D, and E
  - How many unique elements?
    - Unique elements S: $p(p+1)/2 = 5 * 6/2 = 15$
    - Unique elements m: $p = 5$
  - $15 + 5 = 20$

- Saturated model = All variables related
  - How many model parameters?
    - cross relations / covariances: 10
    - variances: 5
    - means: 5
  - $(10 + 5) + 5 = 20$

- How many df?
  - $df = 20 - 20 = 0$

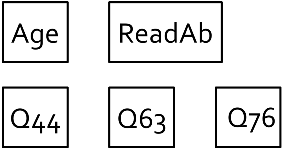
Baseline model ($p = 5$): DIY

- Variables: A, B, C, D, and E
  - How many unique elements?
- Baseline model = Unrelated variables
  - How many model parameters?
- How many df?



Baseline model ($p = 5$): Answer

- Variables: A, B, C, D, and E
  - How many unique elements?
  - $5(5+1)/2 + 5 = 20$
- Baseline model = Unrelated variables
  - How many model parameters?
  - 5 variances and 5 means = 10 parameters
- How many df?
  - df = 20-10=10



Example baseline model (p = 5)

Variables: Age, ReadAb, Q44, Q63, and Q76.

```
lavInspect(fit_base5)$theta

##         Age ReadAb Q44 Q63 Q76
## Age       1
## ReadAb    0      2
## Q44       0      0   3
## Q63       0      0   0   4
## Q76       0      0   0   0   5
```

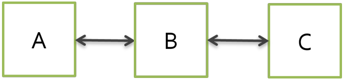
```
lavInspect(fit_base5)$nu

##         intrcp
## Age          6
## ReadAb       7
## Q44          8
## Q63          9
## Q76         10
```

Example baseline model (p = 5) Ctd.

```
fitMeasures(fit_base5,
            c("npar", "logl", "unrestricted.logl",
              "chisq", "pvalue",
              "cfi", "tli", "rmsea"))

##              npar              logl unrestricted.logl
##            10.000         -5440.660         -5360.694
##            pvalue               cfi               tli
##             0.000             0.000             0.000
```

Note: At the end, more details about model fit.

From model to technical output




```
lavInspect(fit_example)$theta

##        Age ReadAb Q44
## Age      3
## ReadAb   1      4
## Q44      2      0   5
```

```
lavInspect(fit_example)$nu

##        intrcp
## Age         6
## ReadAb      7
## Q44         8
```

No relation between A and C. Hence, zero in Theta matrix.

Table of Contents

SEM

SAPI

Lavaan commands

Regression

Path model

Steps

Technical output

Steps Ctd.

Model fit

The end

Step by step - there are more: Step 10

Understand all errors/warnings

Note: Not all warning messages have to be fixed,
but know what the problem is.

Step 11

What to do with missing data?

11. Handling missing data: FIML

Note: Will be discussed in detail in Missing Data lecture.

Model needs to be fully specified (elsewise error):

```
mod.regr_full <- '
  Q77 ~  Age # regression
  Q77 ~~ Q77 # residual variance
  Q77 ~  1   # intercept

  # (co)variances and means predictor(s)
  Age ~~ Age # variance predictor
  Age ~ 1    # mean predictor
'
fit.regr_fiml <- lavaan(model = mod.regr_full, data = data_sapi,
                        missing='fiml', fixed.x=FALSE)

## Warning:  lavaan->lav_data_full():
##    some cases are empty and will be ignored:  453 535.
```

11. Handling missing data: FIML Ctd.

Previous estimates

```
parameterEstimates(fit.regr)[c(1,2,3),]

##   lhs op rhs    est    se      z pvalue ci.lower ci.upper
## 1 Q77  ~ Age -0.022 0.004 -5.275      0   -0.030   -0.014
## 2 Q77 ~~ Q77  1.184 0.068 17.536      0    1.052    1.317
## 3 Q77 ~1      4.275 0.135 31.657      0    4.011    4.540
```

FIML estimates

```
parameterEstimates(fit.regr_fiml)[c(1,2,3),]

##   lhs op rhs    est    se      z pvalue ci.lower ci.upper
## 1 Q77  ~ Age -0.021 0.004 -5.341      0   -0.029   -0.013
## 2 Q77 ~~ Q77  1.133 0.052 21.829      0    1.031    1.234
## 3 Q77 ~1      4.219 0.126 33.524      0    3.972    4.465
```

Step 12

Are you really really sure that everything went well?

Step 13

Interpret results

Using model fit indices, discussed next.

Table of Contents

SEM

SAPI

Lavaan commands

Regression

Path model

Steps

Technical output

Steps Ctd.

Model fit

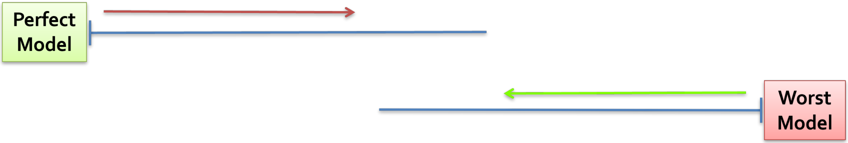
The end

Model Test and Fit Indices

- Log-likelihood ratio (Chi-Square) test
- RMSEA
- CFI/TLI
- Chi-Square Test of Model Fit for the Baseline Model
- SRMR
- Information Criteria: AIC, BIC

A good fit index...

- Not sensitive to sample size
- Sensitive to discrepancies between modeled data and observed data



- Reflects the degree of parsimony of the model

Model-implied cov-matrix vs. perfect fit

- Population: Σ
- Sample: S
- Model-implied: $\hat{\Sigma}$
- Perfect-model to worse: compare S to $\hat{\Sigma}$

Data

$$S = \begin{array}{c|ccc} & A & B & C \\ \hline A & s^2_A & & \\ B & s_{BA} & s^2_B & \\ C & s_{CA} & s_{BC} & s^2_C \end{array}$$

Model-implied

$$\hat{\Sigma} = \begin{array}{c|ccc} & A & B & C \\ \hline A & \sigma^2_A & & \\ B & \sigma_{BA} & \sigma^2_B & \\ C & \sigma_{CA} & \sigma_{BC} & \sigma^2_C \end{array}$$

Log-likelihood ratio test

- $-2log\frac{L_0}{L_a} = -2logL_0 + 2logL_a$
- Statistic is Chi-square distributed
- Model test: unrestricted model is model that led to sample variance-covariance matrix
- $H_0 : \chi^2 = 0$ (perfect fit)

\+ actual test for fit
\- sensitive to sample size
\- inflated with non-normal data

RMSEA (Root Mean Square Error of Approximation)

$RMSEA = \sqrt{\frac{\chi^2 - df}{(n-1)df}}$

= bias-corrected estimate of divergence of model-implied and population var-covar matrix.

Absolute index:

- Index of discrepancy between model and data
  - + sensitive to model complexity, namely: bias correction by '$-df$' (in '$\chi^2 - df$', in formula above))
  - - less suitable with small sample size (with low df)
- Rule of thumb: $<.08$ (mediocre), $<.05$ (close / good)
- Inspect CIs

SRMR (Standardized Root Mean Square Residual)

$$SRMR = \sqrt{\frac{\sum_{i=1}^{p}\sum_{j=1}^{i}\left(\frac{s_{ij}-\hat{\sigma}_{ij}}{s_{ii}s_{jj}}\right)}{p(p+1)/2}} = \sqrt{\frac{\text{sum of squared std residuals}}{\text{size var-covar matrix}}}$$

Absolute index:

- Index of discrepancy between model and sample statistics
  - - sensitive to sample size
  - - insensitive to model complexity
- Rule of thumb: max .08

Relative indices: AIC/BIC

- Information theoretic:
  Combine goodness of fit χ^2 and parsimony (nr. of parameters k).

- AIC (Akaike Information Criterion)
  - $AIC = \chi^2 + 2k$
- BIC (Bayesian Information Criterion)
  - $BIC = \chi^2 + k * ln(n)$
- No rule of thumb, values depend on actual data set.
- Choose the model with the lowest AIC and/or BIC.

Baseline model approach

- Assume unrelated variables (independence)
- Assess improvement of fit by hypothesized model

Independence model

$$\hat{\boldsymbol{\Sigma}}_b = \begin{array}{c|ccc} & A & B & C \\ \hline A & \sigma^2_A & & \\ B & 0 & \sigma^2_B & \\ C & 0 & 0 & \sigma^2_C \end{array}$$

Model-implied

$$\hat{\boldsymbol{\Sigma}} = \begin{array}{c|ccc} & A & B & C \\ \hline A & \sigma^2_A & & \\ B & \sigma_{BA} & \sigma^2_B & \\ C & \sigma_{CA} & \sigma_{BC} & \sigma^2_C \end{array}$$

CFI/TLI

Comparative indices:

TLI (Tucker-Lewis Index)

+ complexity
- somewhat sensitive to small sample
- works only with ML estimator
- >1 = overfitting, more complex than needed

Rule of thumb: $>.90$ (acceptable), $>.95$ (good)

$$TLI = \frac{\frac{\chi^2_b}{df_b} - \frac{\chi^2_m}{df_m}}{\frac{\chi^2_b}{df_b} - 1}$$

CFI (Comparative Fit Index)

+ complexity
+ not sensitive to small sample size
+ consistent over different estimation methods

Rule of thumb: $>.90$ (acceptable), $>.95$ (good)

$$CFI = 1 - \frac{\chi^2_m - df_m}{\chi^2_b - df_b}$$

Which to use?

- Report several indices of different types (CFI, RMSEA)
- All others than chi-square should look good
- Improve the model (exploratory!) by
  - removing n.s. paths and/or
  - adding significant paths,
  - see `https://lavaan.ugent.be/tutorial/modindices.html`,

but only if: it makes theoretical sense, and: you report your modifications

Model Comparison

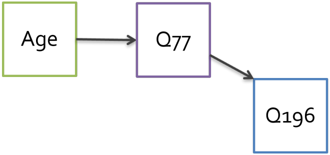
- Nested models (i.e., one model is a special case of the other):
  Chi-square difference test / AIC / BIC
- Non-nested models:
  AIC / BIC

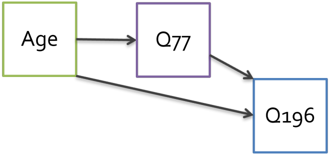
In lavaan: use anova(fit.model1, fit.model2).

If and only if the covariance matrix is equal over models
=
same participants, same variables!

Model Comparisons: Nested models

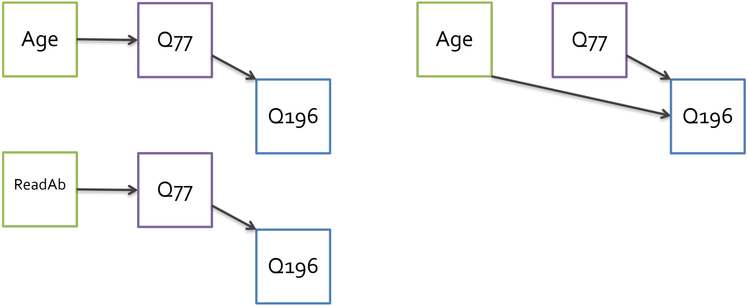
Nested models: one model is a special case of the other.





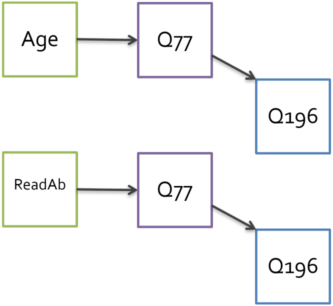
Model Comparisons: Nested models?

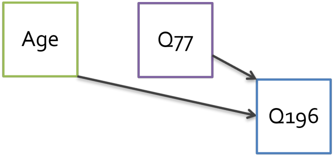
- Nested?
- Chi-square test vs AIC/BIC?



Model Comparisons: Nested models? - Answers

- Nested? No
- Chi-square test vs AIC/BIC? AIC/BIC





Model Comparisons: AIC / BIC

Suitable for comparing

- both nested and non-nested models.
- more than two models.

Choose the model with the lowest AIC and/or BIC.

To be discussed in the Model Selection lecture

- Quantify relative support via IC weights.

```
library(devtools) # Make sure you have Rtools
install_github("rebeccakuiper/ICweights")
library(ICweights) # ?IC.weights
# Or:
library(restriktor) # ?calc_ICweights
```

- Extension AIC for order-restricted hypotheses (GORIC and GORICA).

```
library(restriktor) # ?goric
```

Table of Contents

SEM

SAPI

Lavaan commands

Regression

Path model

Steps

Technical output

Steps Ctd.

Model fit

The end

Summary

- Short SEM overview
- Simple analyses in lavaan
- Step-by-step checklist
- Model fit

Take home message

- Know what's happening:
  - Check whether unique elements - estimated = df
  - Get to know lavaan defaults
  - Read warnings (!)
- Make use of FIML (if possible)

Thanks & How to proceed

Thanks for listening!

Are there any questions?

- Ask fellow participant on course platform.
- Ask teacher during Q&A (or via course platform).
- See if making the lab exercises help.
- Check the lavaan tutorial: e.g., `https://lavaan.ugent.be/tutorial/inspect.html`.
- Do not forget that Google is your best friend :-).

You can start working on the lab exercises.