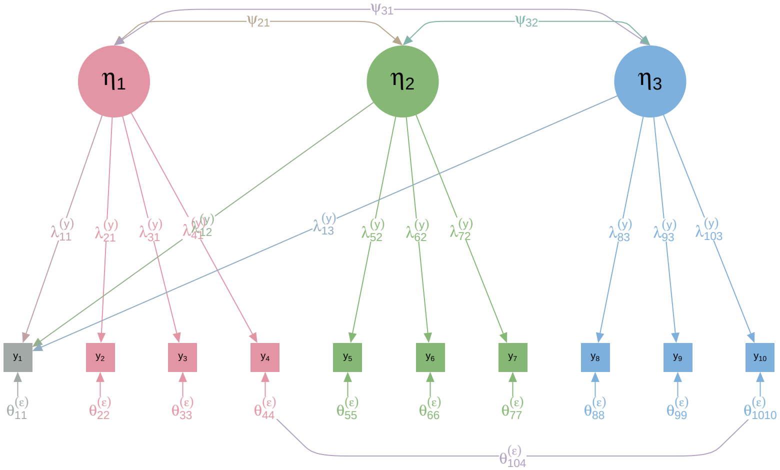
SEM 1: Confirmatory Factor Analysis

Week 2 - Fitting CFA models

Sacha Epskamp

2020



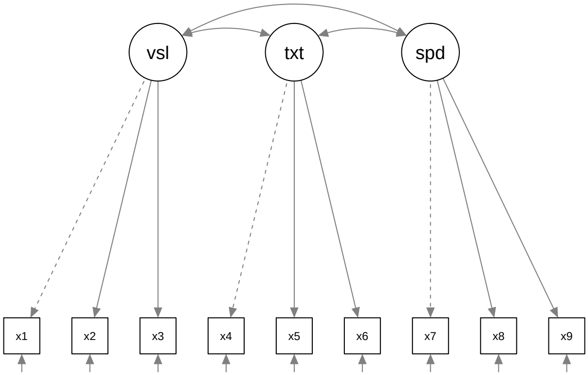
See for examples:

`https://github.com/SachaEpskamp/SEM-code-examples`

and the video lectures!

```r
# Load the package:
library("lavaan")

# Load data:
data("HolzingerSwineford1939")
Data <- HolzingerSwineford1939

# Model:
Model <- '
  visual  =~ x1 + x2 + x3
  textual =~ x4 + x5 + x6
  speed   =~ x7 + x8 + x9
'

# Fit in lavaan:
fit <- cfa(Model, Data)
```

```r
library("semPlot")
semPaths(fit, style = "lisrel")
```



Latent variances not drawn and residuals simplified

Testing for exact fit

Remember the fit function:

$$F_{\mathrm{ML}} = \mathrm{trace}\left(\boldsymbol{S}\boldsymbol{\Sigma}^{-1}\right) - \ln\left|\boldsymbol{S}\boldsymbol{\Sigma}^{-1}\right| - p,$$

Lavaan instead reports `fmin`, which equals $F_{\mathrm{ML}}/2$. If n is the sample size, then we can define:

$$T = nF_{\mathrm{ML}}.$$

If $\mathrm{Var}\left(\boldsymbol{y}\right) = \boldsymbol{\Sigma}$ (the model is true), then T is χ^2 (chi-square) distributed with the same number of degrees of freedom as the model:

$$T \sim \chi^2(\mathrm{DF}) \iff \mathrm{Var}\left(\boldsymbol{y}\right) = \boldsymbol{\Sigma}$$

Often (including in the book), T is simply termed χ^2.

```r
# Model matrices:
n <- nrow(Data)
S <- (n-1)/n * cov(Data[,c("x1","x2","x3","x4","x5",
                           "x6","x7","x8","x9")])
Sigma <- lavInspect(fit, "sigma")

# fmin = F_ml / 2:
F_ml <- sum(diag(S %*% solve(Sigma))) -
  log(det(S %*% solve(Sigma))) - ncol(S)
F_ml
```

```
## [1] 0.283407
```

```r
2 * fitMeasures(fit)['fmin']
```

```
##     fmin
## 0.283407
```

```r
# Chi-square reported by lavaan and computed:
nrow(Data) * F_ml

## [1] 85.30552

fitMeasures(fit)['chisq']

##    chisq
## 85.30552
```

Testing for exact fit

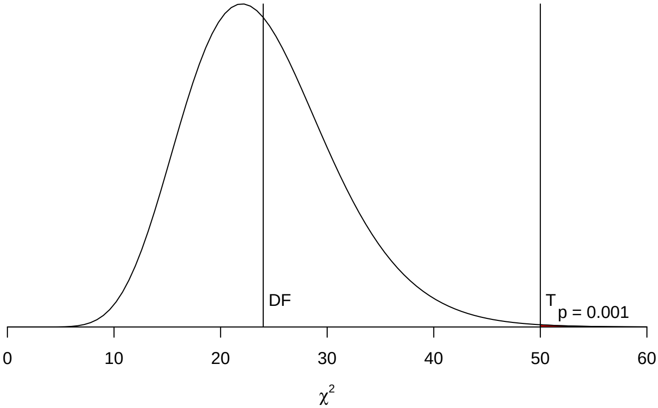
$$T \sim \chi^2(\mathrm{DF}) \iff \mathrm{Var}\left(\boldsymbol{y}\right) = \boldsymbol{\Sigma}$$

Allows for testing the following hypothesis:

$$H_0 : \mathrm{Var}\left(\boldsymbol{y}\right) = \boldsymbol{\Sigma}$$
$$H_1 : \mathrm{Var}\left(\boldsymbol{y}\right) \neq \boldsymbol{\Sigma}$$

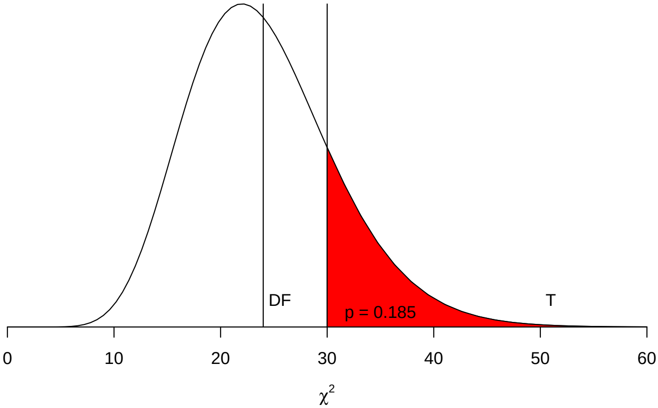
We can reject H_0 if the data is not *likely* under H_0. The $\chi^2(\mathrm{DF})$ distribution computes this likelihood, as an area under the curve right of T. We do not want to reject H_0: p should be above α (typically 0.05).

Degrees of freedom: 24; $T = 50$



$p < 0.05$, model does not fit the data!

Degrees of freedom: 24; $T = 30$



$p > 0.05$, model fits the data!

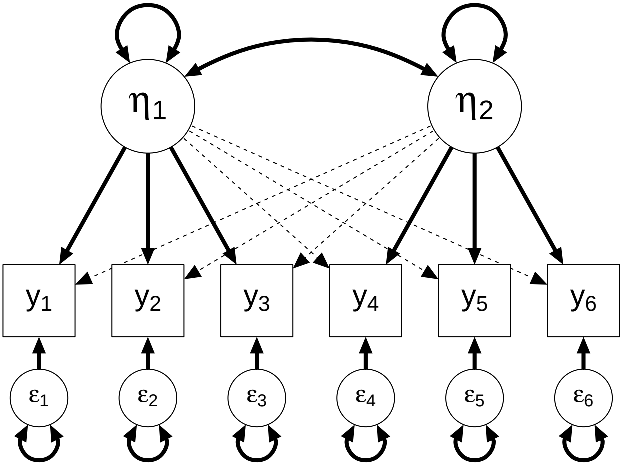
```
fit

## lavaan 0.6-5 ended normally after 35 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of free parameters                         21
##
##   Number of observations                           301
##
## Model Test User Model:
##
##   Test statistic                                85.306
##   Degrees of freedom                                24
##   P-value (Chi-square)                           0.000
```

Model does not fit :(

"All models are wrong but some are useful"

Box, G. E. P. (1979), "Robustness in the strategy of scientific model building", in Launer, R. L.; Wilkinson, G. N., Robustness in Statistics, Academic Press, pp. 201–236.

True model might be not exactly the same, but nearly the same:



As $N \to \infty$, we will always expect to reject H_0.

Testing for exact fit

- The test of exact fit over-rejects in small samples, because T is only chi-square distributed asymptotically (as n becomes large). When n is small, the chi-square distribution approximation can be poor.
- The chi-square test is often underpowered with small samples, leading to under-rejection. In short, the test of exact fit is unreliable when n is small.
- When n is large, the chi-square test has a lot of power, which leads to rejection of models even when the residuals are very small.