SEM 1: Confirmatory Factor Analysis

Week 2 - Fitting CFA models

Sacha Epskamp

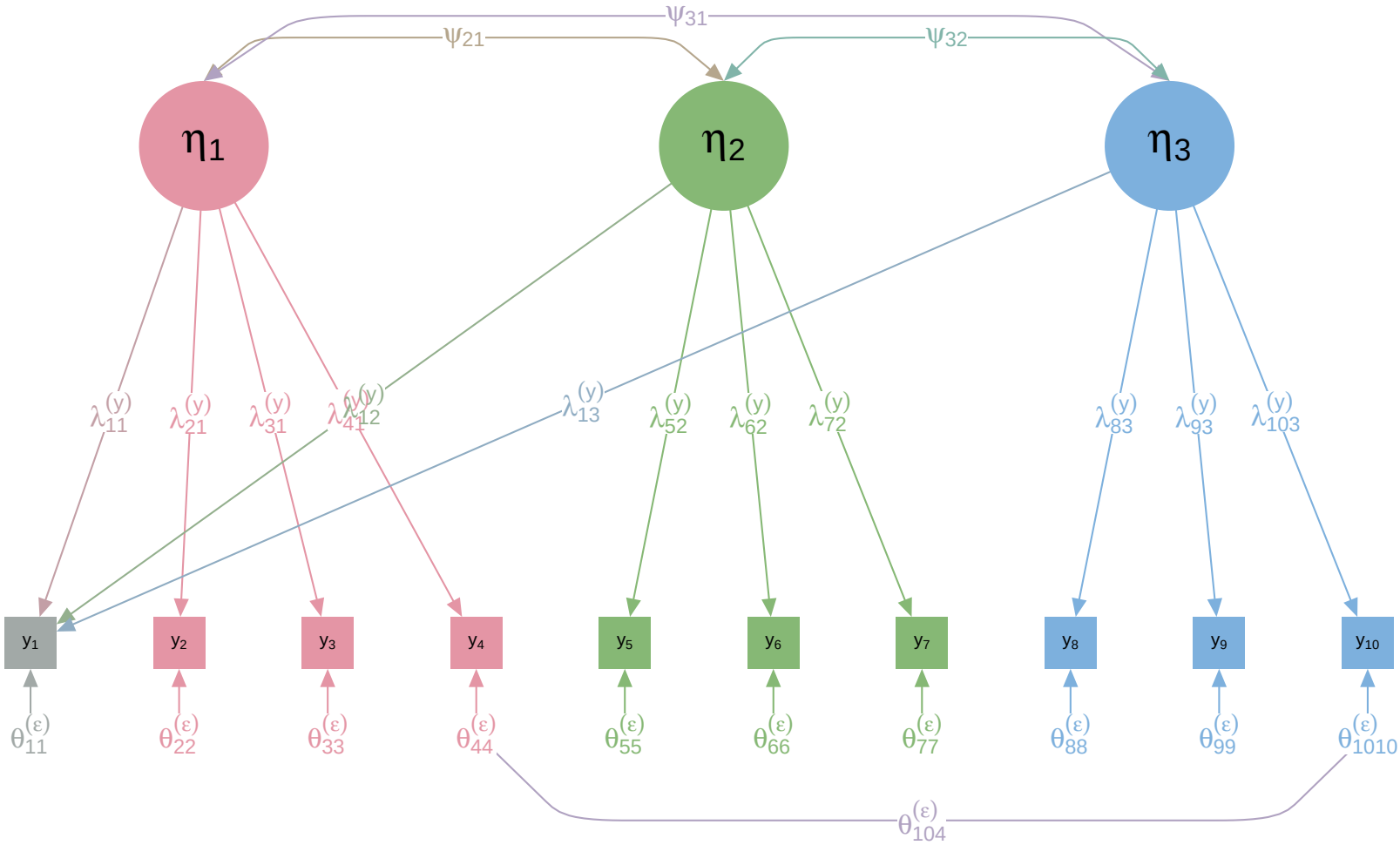
09-04-2019

Name	Pros	Cons
lavaan	Free, extensive, easy to use, path diagrams via semPlot	Still requires code
blavaan	Free, similar to lavaan, Bayesian	Bayesian
Jasp (lavaan)	Free, graphical interface except for model syntax	Some things not trivial, no path diagrams (yet)
Onyx	Free, graphical model specification	Hard to use for larger models, model comparison not easy
OpenMx	Free, flexible matrix specification	Hard to use
Mplus	Very powerful and extensive, can do things other packages can't	Expensive, close-source, dated plain text input
psychonetrics	Totally awesome	Unstable alpha version

See for examples:

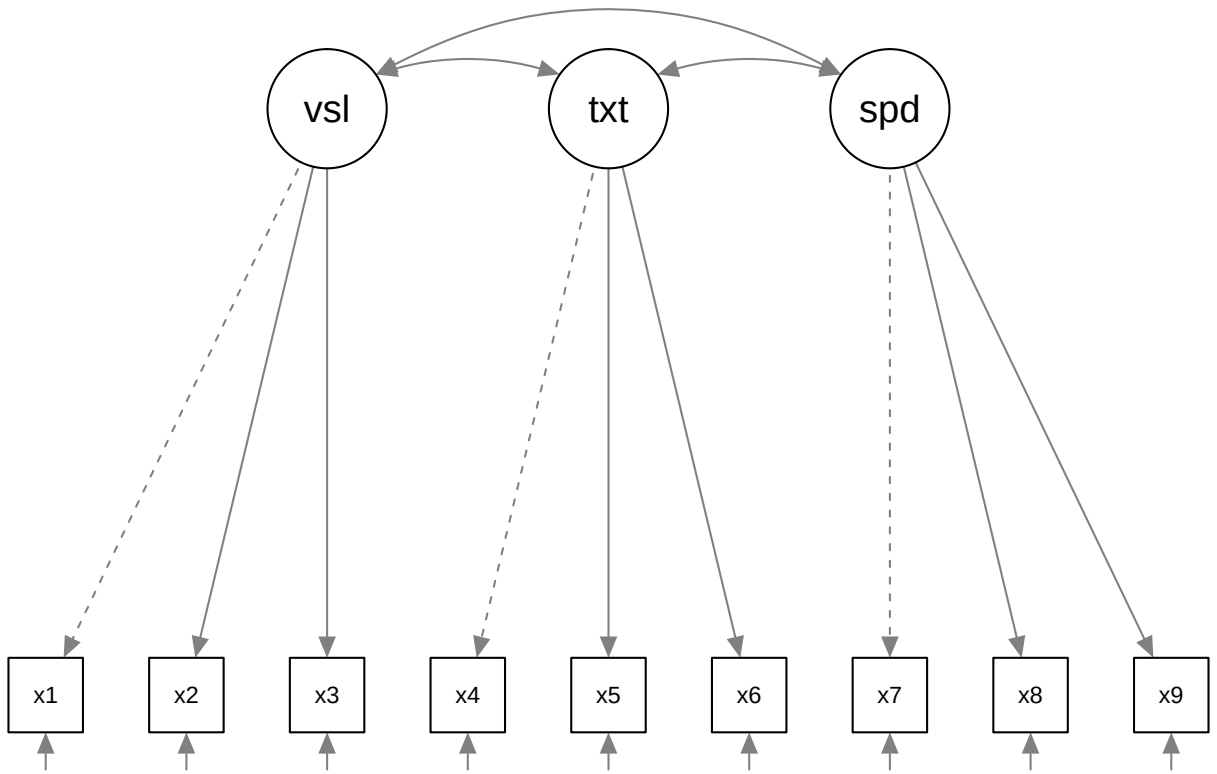
`https://github.com/SachaEpskamp/SEM-code-examples`

and the Youtube video lectures!


ψ_{31}
ψ_{21}
ψ_{32}
η_1
η_2
η_3
$\lambda^{(y)}_{11}$
$\lambda^{(y)}_{21}$
$\lambda^{(y)}_{31}$
$\lambda^{(y)}_{41}$
$\lambda^{(y)}_{12}$
$\lambda^{(y)}_{13}$
$\lambda^{(y)}_{52}$
$\lambda^{(y)}_{62}$
$\lambda^{(y)}_{72}$
$\lambda^{(y)}_{83}$
$\lambda^{(y)}_{93}$
$\lambda^{(y)}_{103}$
y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8 y_9 y_{10}
$\theta^{(\varepsilon)}_{11}$ $\theta^{(\varepsilon)}_{22}$ $\theta^{(\varepsilon)}_{33}$ $\theta^{(\varepsilon)}_{44}$ $\theta^{(\varepsilon)}_{55}$ $\theta^{(\varepsilon)}_{66}$ $\theta^{(\varepsilon)}_{77}$ $\theta^{(\varepsilon)}_{88}$ $\theta^{(\varepsilon)}_{99}$ $\theta^{(\varepsilon)}_{1010}$
$\theta^{(\varepsilon)}_{104}$

```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)
```

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



Latent variances not drawn and residuals simplified

Testing for exact fit

Remember the fit function:

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

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

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

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

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

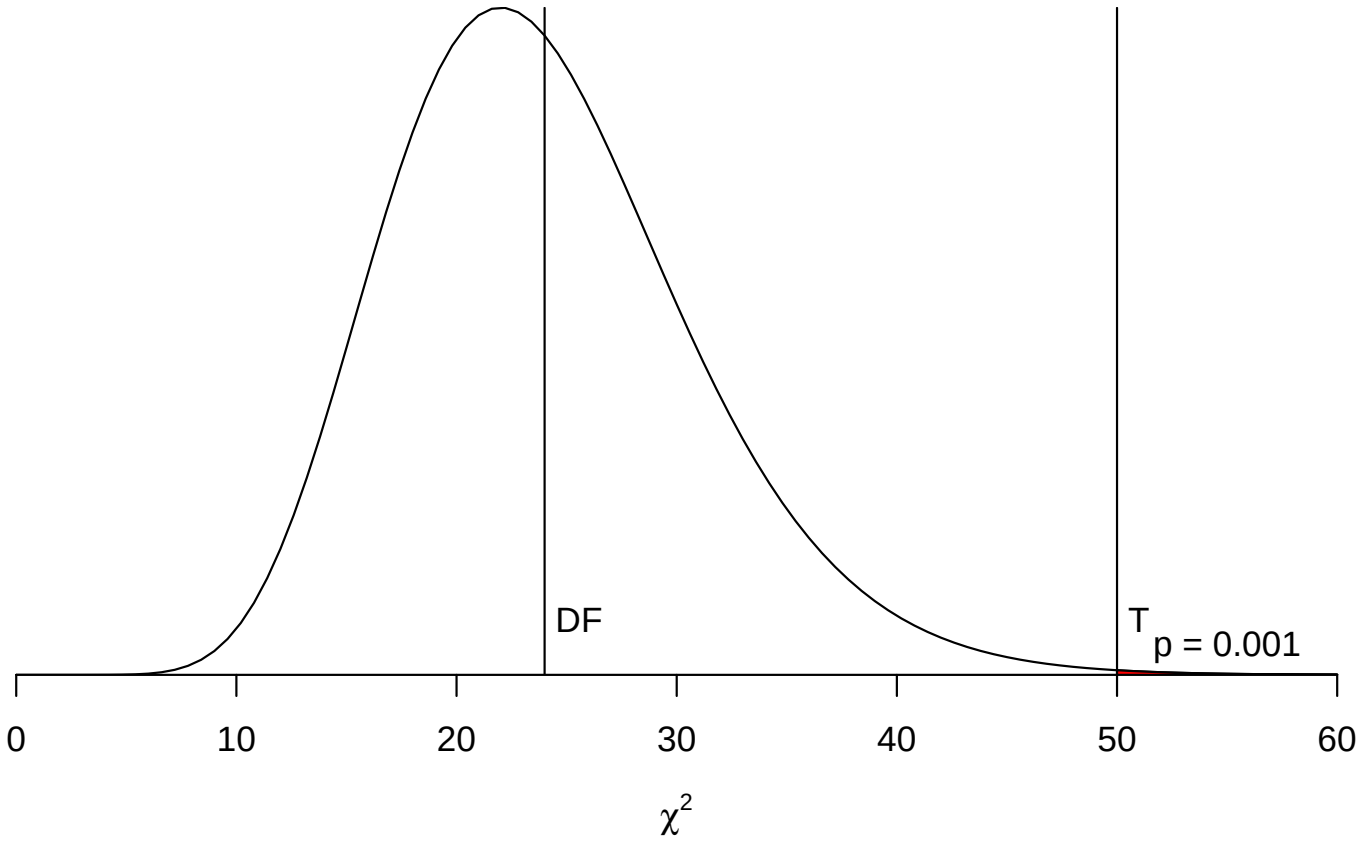
$$T \sim \chi^2(\mathrm{DF}) \iff \mathrm{Var}(\boldsymbol{y}) = \boldsymbol{\Sigma}$$

Allows for testing the following hypothesis:

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

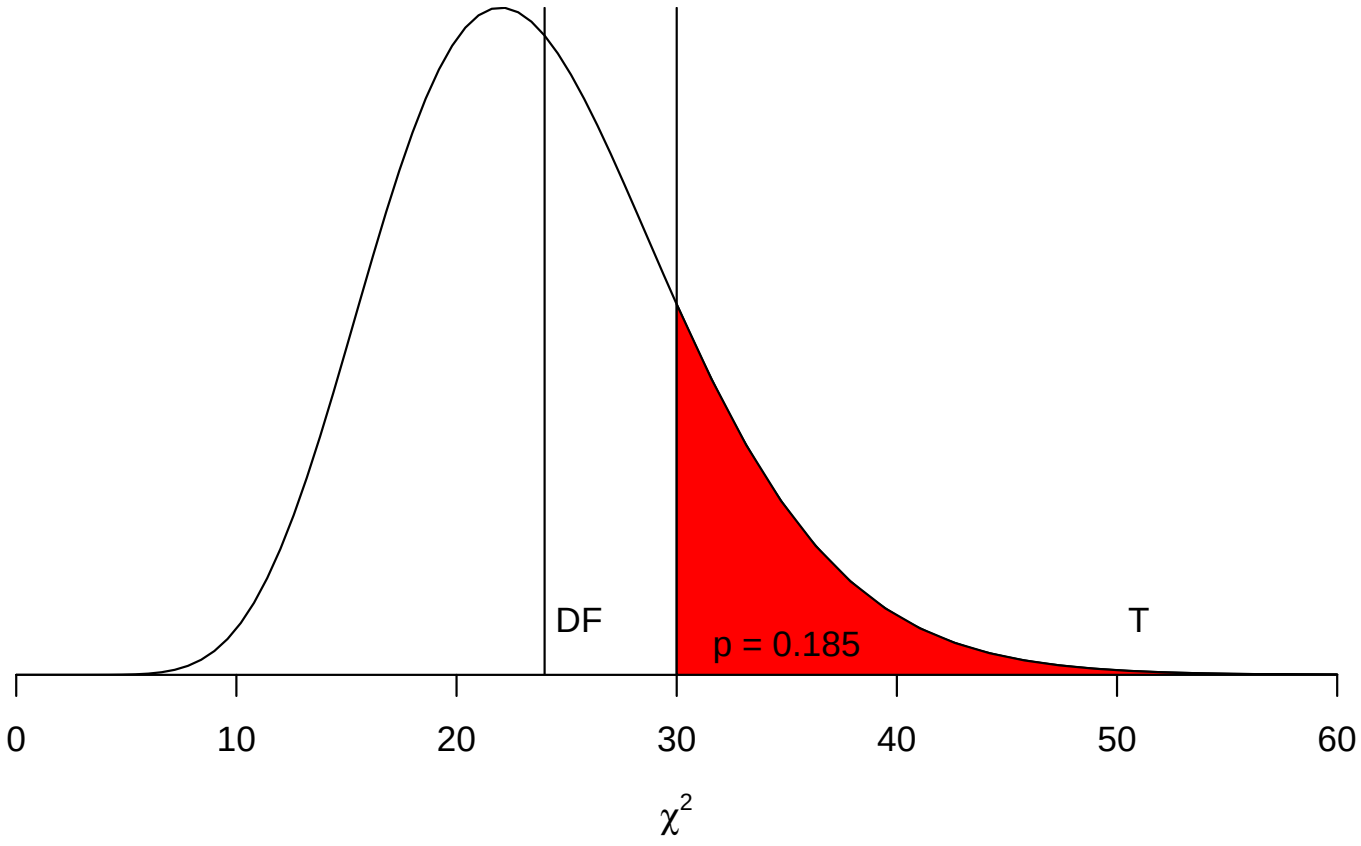
We can reject H_0 if the data is not *likely* under H_0. The $\chi^2(\mathrm{DF})$ distribution computes this likelihood, as an area under the curve right of T. We do not want to reject H_0: p should be above α (typically 0.05).

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!

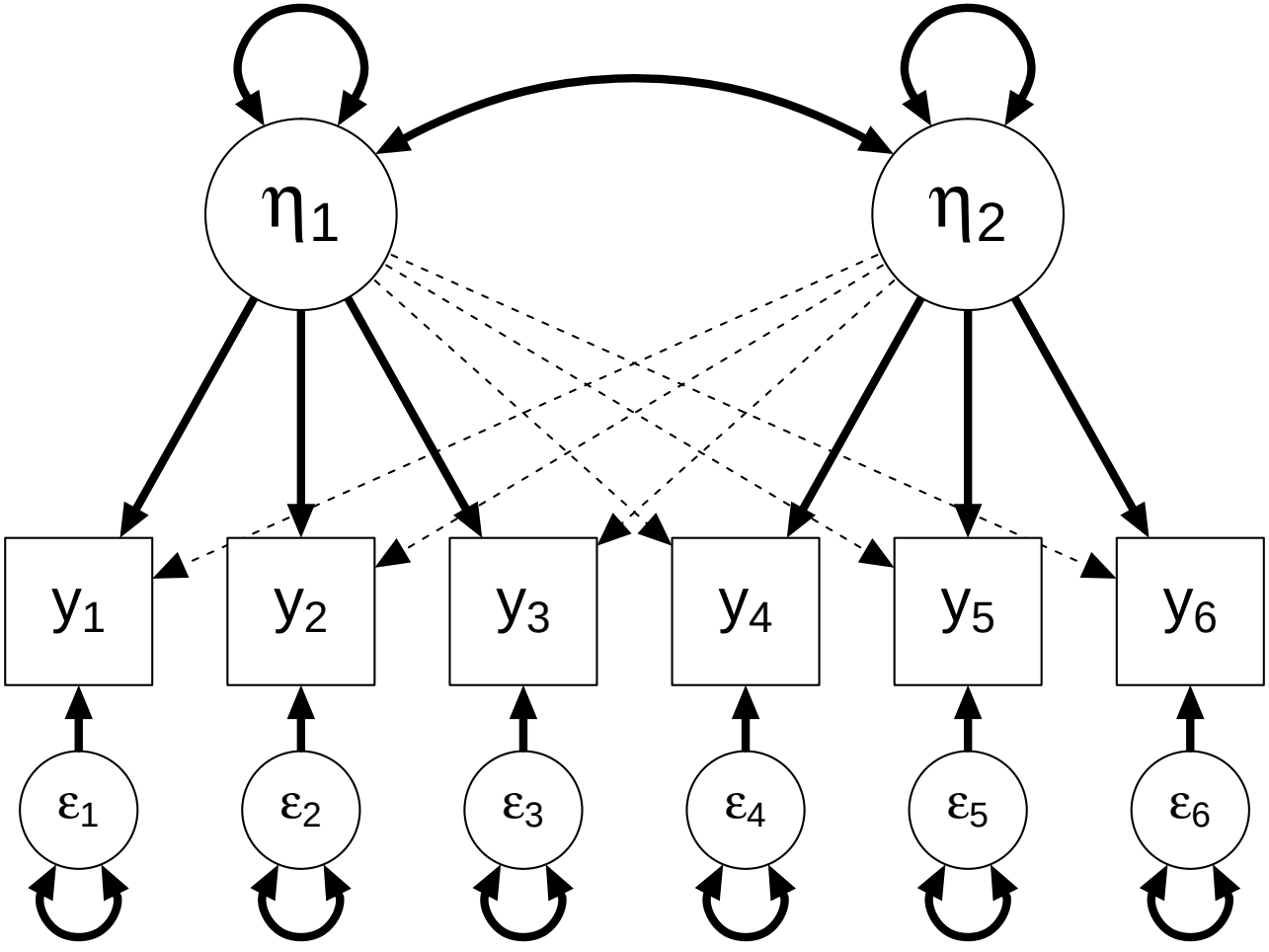
```
fit

## lavaan 0.6-4.1349 ended normally after 35 iterations
##
##   Optimization method                           NLMINB
##   Number of free parameters                         21
##
##   Number of observations                           301
##
##   Estimator                                         ML
##   Model Fit 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:

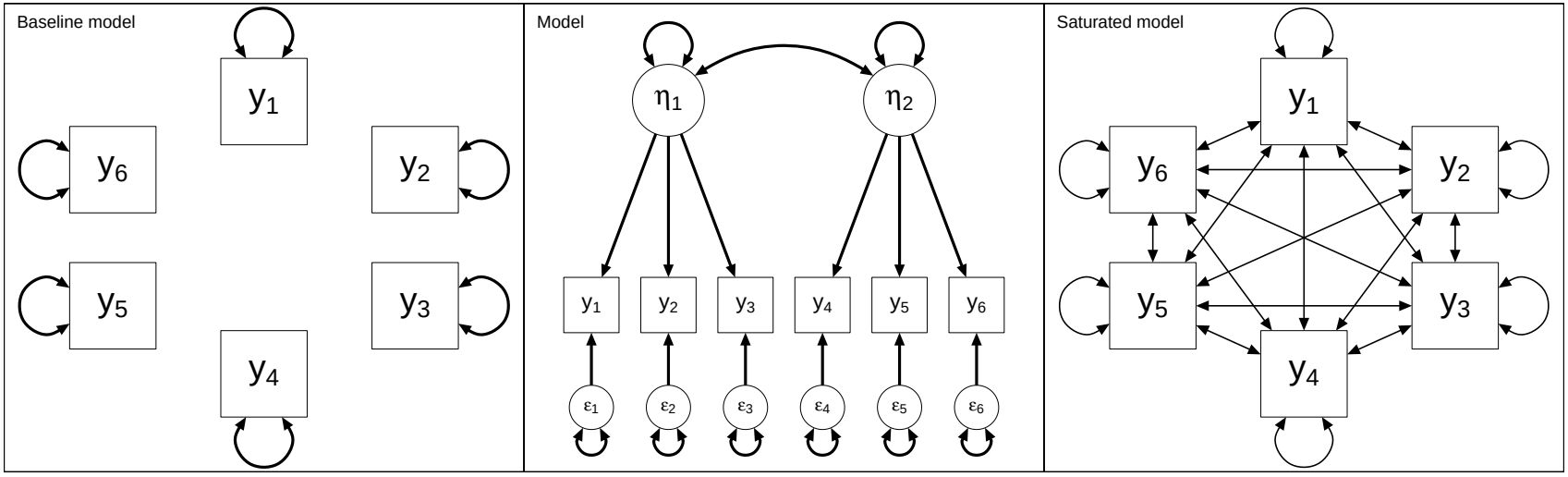


As $N \to \infty$, we will always expect to reject H_0.

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.

Fit indices



To assess fit, model can be compared to *baseline model* or *saturated model*.

- Baseline model: a model in which no items covary
- Saturated model: A perfectly fitting model ($T = 0$; $DF = 0$)

The χ^2 test compares the model (T_M) to the saturated model (should fit about the same). Many fit indices compare the model to the baseline model instead (T_B; should fit much worse than tested model).

RMSEA

Root Mean Square Error of Approximation

$$RMSEA = \sqrt{\frac{T_M - \mathrm{DF}_M}{(n\mathrm{DF}_M)}}$$

```
Tm <- fitMeasures(fit)[['chisq']]
DFm <- fitMeasures(fit)['df']
sqrt((Tm - DFm)/(n * DFm))

##          df
## 0.09212148

fitMeasures(fit)[['rmsea']]

## [1] 0.09212148
```

RMSEA

RMSEA is a measure of absolute fit (no comparison model). It measures the amount of misfit per degrees of freedom. Smaller values indicate better fit.
Proposed benchmarks from a selection of papers:

- $<$ .05 "very good fit" or "close fit"
- .05 – .08 "good fit" or "fair fit"
- .08 – .1 "mediocre fit" or "good"!
- .05 – .08 "good fit" or "fair fit"
- $>$ .10 "poor or unacceptable"

RMSEA

RMSEA is one of the only fit indices for which the asymptotic sampling distribution is known, so we can make confidence intervals and conduct hypothesis tests about its population value.

- Test of Exact Fit
  - H_0: RMSEA $= 0$ in the population
  - Equivalent to the significance test on the chi-square statistic
- Test of Close Fit (MacCallum et al., 1996)
  - H_0: Null hypothesis: $\mathrm{RMSEA} < \mathrm{RMSEA}_{good}$ in the population.
  - RMSEA_{good} is some acceptable value of RMSEA (in lavaan: 0.05)
- Test of Not-Close Fit (MacCallum et al., 1996)
  - H_0: Null hypothesis: $\mathrm{RMSEA} > \mathrm{RMSEA}_{bad}$ in the population.
  - RMSEA_{bad} is some unacceptable value of RMSEA (e.g., 0.08)

RMSEA test of close fit

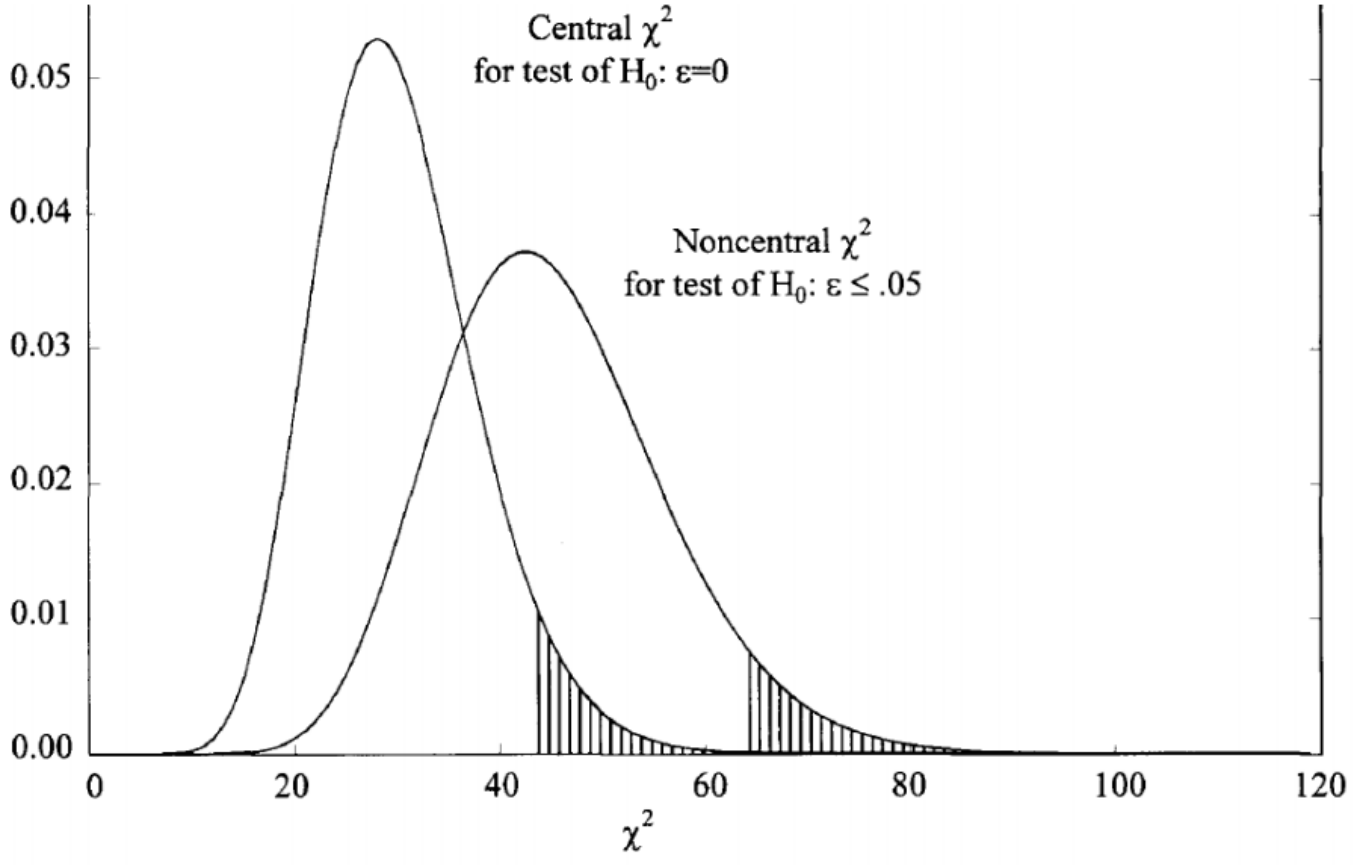


Figure 1. Illustration of difference in critical values between central and noncentral χ^2 distributions.

MacCallum, Browne, & Sugawara (1996)

RMSEA test of close fit

Given some boundary RMSEA_{good}, compute non-centrality parameter:

$$\lambda_c = \mathrm{RMSEA}^2_{good} \times n \times \mathrm{DF}_M$$

Compute a one-tailed test with T, now using non-central distribution $\chi_2(\mathrm{DF}_M, \lambda_c)$.

```
lambda_c <- 0.05^2 * n * DFm
pchisq(Tm, DFm, lambda_c, lower.tail = FALSE)

## [1] 0.0006612368

fitMeasures(fit)['rmsea.pvalue']

## rmsea.pvalue
## 0.0006612368
```

We reject the hypothesis the model fits well (RMSEA smaller than 0.05), this is not good (but also not very bad, as RMSEA around $0.05 - 0.07$ is still acceptable)!

RMSEA test of not-close fit

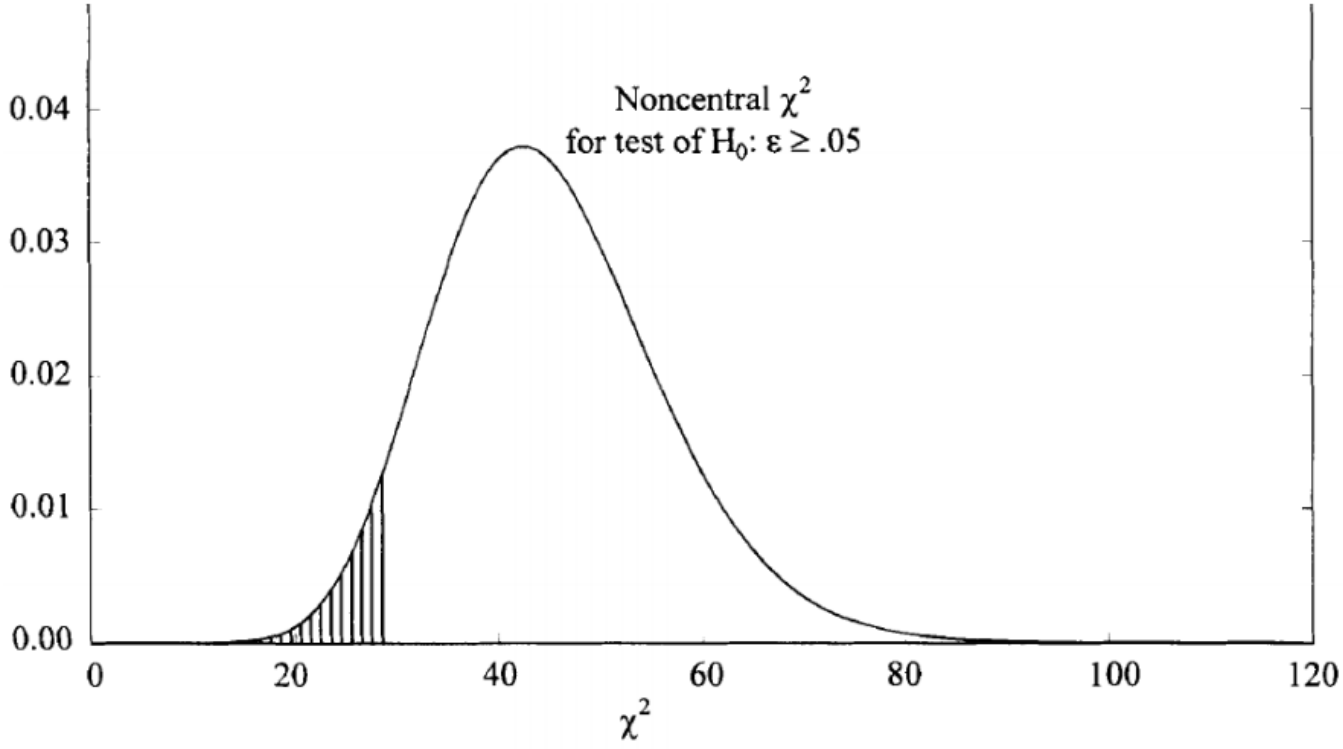


Figure 2. Illustration of critical value of noncentral χ^2 distribution for testing hypothesis of not-close fit.

MacCallum, Browne, & Sugawara (1996)

RMSEA test of not-close fit

Given some boundary RMSEA_{bad}, compute non-centrality parameter:

$$\lambda_c = \mathrm{RMSEA}_{bad}^2 \times n \times \mathrm{DF}_M$$

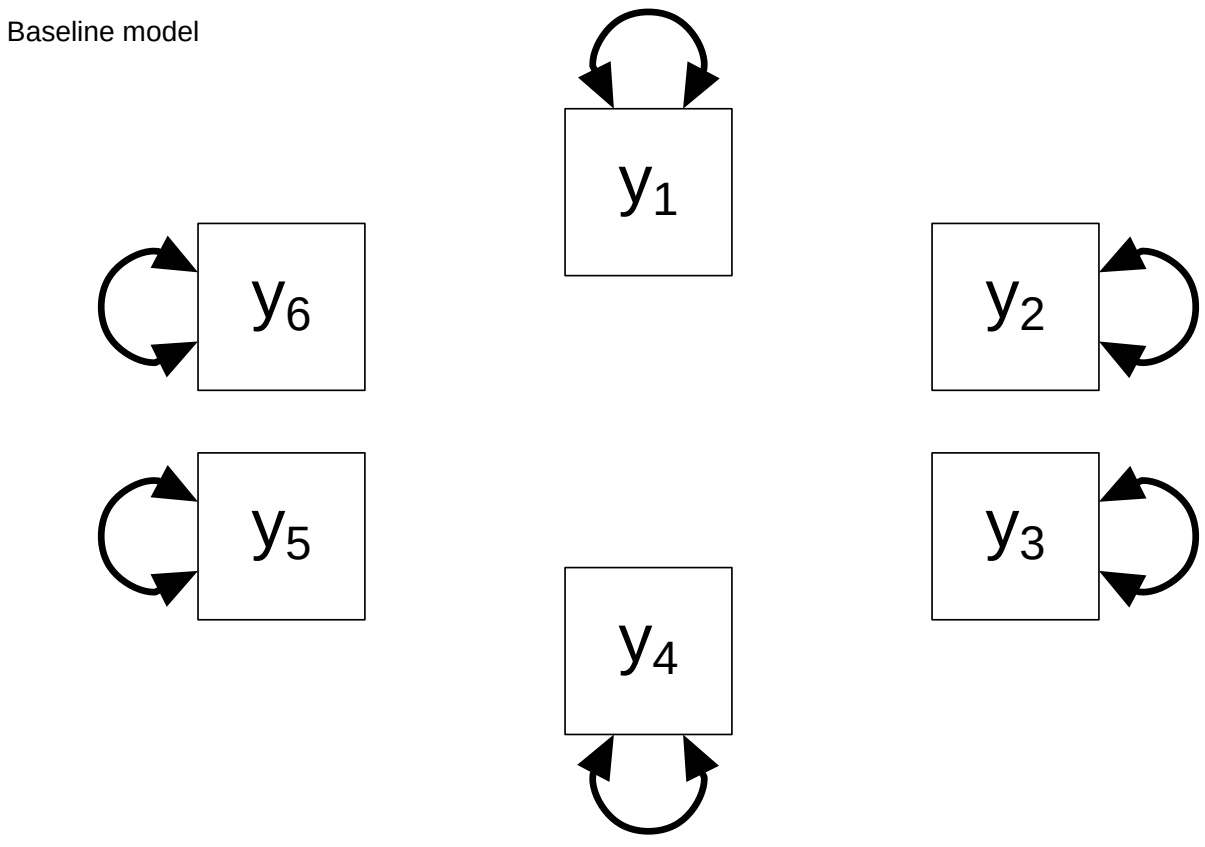
Compute a one-tailed test with T, using the lower tail of non-central distribution $\chi_2(\mathrm{DF}_M, \lambda_c)$.

```
lambda_c <- 0.08^2 * n * DFm
pchisq(Tm, DFm, lambda_c, lower.tail = TRUE)

## [1] 0.8395529
```

We can not reject the hypothesis the model fits poorly (RMSEA larger than 0.08), this is bad!

Incremental fit indices

Incremental fit indices compare the model to the baseline model, which only estimates variances:



Incremental fit indices

```
Tb <- fitMeasures(fit)['baseline.chisq']
Tb

## baseline.chisq
##       918.8516

DFb <- fitMeasures(fit)['baseline.df']
DFb

## baseline.df
##          36
```

How much better does the model fit than the worst possible model?

Incremental fit indices

T_M, df_M: Model test statistic and DF; T_B, df_B: Baseline model test statistic and DF.

$NFI = \frac{T_B - T_M}{T_B}$	% change in the test statistic, ranges from 0-1. Tends to be higher with larger N.	Normed Fit Index, Bentler & Bonnett (1980)
$TLI = \frac{T_B - \frac{df_B}{df_M} T_M}{T_B}$	NFI + reward for parsimonious models. "non-normed" = can take values higher than 1.	Tucker-Lewis Index (TLI), Tucker & Lewis (1973) Non-Normed Fit Index (NNFI), Bentler & Bonnett (1980)
$RFI = \frac{\frac{T_B}{df_B} - \frac{T_M}{df_M}}{T_B / df_B}$	% change in the test statistic relative to its df	Relative Fit Index (Bollen, 1986)
$IFI = \frac{T_B - T_M}{T_B - df_M}$	IFI is less sensitive to sample size than NFI, NNFI, RFI	Incremental Fit Index (Bollen, 1989)

(higher is better)

Incremental fit indices

T_M, df_M: Model test statistic and DF; T_B, df_B: Baseline model test statistic and DF.

$$RNI = \frac{\left(T_B - df_B\right) - \left(T_M - df_M\right)}{T_B - df_B}$$

Nonnormed (can exceed 1)

Relative Noncentrality Index, McDonald & Marsh, 1990

$$CFI = 1 - \frac{T_M - df_M}{T_B - df_B}$$

Constrained to 1 if $df_M > T_M$

Comparative Fit Index, Bentler, 1990

In Practice:
CFI, TLI/NNFI are most commonly reported incremental fit indices. .95 is often used as a cutoff rule-of-thumb for "good fit", and .90 for "acceptable fit" though these cutoffs do not have much empirical support

Goodness of Fit (GFI) and Adjusted GFI

- GFI and AGFI are analogous to R^2 and adjusted R^2 in regression
  - R^2 estimates the proportion of variance in Y that is accounted for by the regression model.
  - GFI estimates the proportion of variance in the sample covariance matrix $\boldsymbol{S}$ that is accounted for by the model structure $\boldsymbol{\Sigma}$
- Both GFI and AGFI can take values from 0 to 1; higher is better.
- Rule of thumb: $>$ .90 is acceptable fit.
- GFI and AGFI tend to be underestimated in small samples.

SRMR

- The largest residual correlation, or list of 5 largest residuals, is very useful for identifying why/how the model does not fit.
- Only look at these if the test of exact fit is significant - if not, the residuals are within the range of sampling error and are most likely noise.
- SRMR is the average of the squared values in the residual correlation matrix. It has been suggested that SRMR should be less than .05 or definitely less than .08. It is less informative than just looking for the biggest residuals!

Residuals

```
residuals(fit)$cov
```

```
##    x1     x2     x3     x4     x5     x6     x7     x8     x9
## x1  0.000
## x2 -0.041  0.000
## x3 -0.010  0.124  0.000
## x4  0.097 -0.017 -0.090  0.000
## x5 -0.014 -0.040 -0.219  0.008  0.000
## x6  0.077  0.038 -0.032 -0.012  0.005  0.000
## x7 -0.177 -0.242 -0.103  0.046 -0.050 -0.017  0.000
## x8 -0.046 -0.062 -0.013 -0.079 -0.047 -0.024  0.082  0.000
## x9  0.175  0.087  0.167  0.056  0.086  0.062 -0.042 -0.032  0.000
```

Strategies for Assessing Fit

- Always report the chi-square test statistic. You may argue that it is too sensitive to minor misspecifications because you have a large sample size, but report it anyway!
- Report several indices (e.g., RMSEA, CFI, RNI, TLI)
- The RMSEA tests of close and not-close fit can be a good index of power (i.e., if neither are significant, you may lack power to detect misspecifications)
- Try to make a holistic judgment based on a set of fit indices
- It goes without saying but... dont cherry pick the indices that make your model look good :)

Sample Size

How big is 'big enough'?

- $n : q$ ratio should be high
  - Theory: to efficiently estimate lots of parameters, a larger sample is needed (5-10 per parameter)
  - There's very little evidence that it matters (Jackson, 2003)
  - This ratio is less important than absolute sample size
- $n \approx 200$ people
  - This is median SEM sample size (Shah & Goldstein, 2006)
  - Appropriate for an average model with ML estimation
  - Other recommendations: 100-200 people minimum
- Use larger n if:
  - Assumptions are violated (e.g., data are nonnormal)
  - Model is complex (e.g., latent interactions, multilevel structure)
  - Indicators have low reliability (factor loadings are low)

Big enough for what?

- Big enough that $\boldsymbol{S}$ is a precise estimate of $\boldsymbol{\Sigma}$
  - No estimation problems (model converges)
  - Parameter estimates have small confidence intervals
- Power to detect model misspecification
  - Chi-square test statistic has sufficient power
  - Fit statistics are accurate

Power to detect non-zero parameters

- G-power cannot help you here: there are too many factors!
- To estimate power, you need to know (or estimate) the model, and all parameter values
- Simulation Method for Estimating Power
  1. Specify a population model with all parameter values
  2. Draw a large number of sample datasets of size n from this hypothetical population (e.g., 1000)
     - `simulateData` in lavaan
  3. Fit the model to each dataset and record whether the parameter value you care about is significant
  4. Count the proportion of significant parameter estimates out of 1000 datasets = power

Power to Detect Misspecification

- Again, Simulation:
  1. Specify a population model
  2. Draw a large number of sample datasets of size n from this hypothetical population (e.g., 1000)
  3. Fit a misspecified model to each dataset and record whether the chi-square test statistic is significant
  4. Count the proportion of significant test statistics out of 1000 datasets = power

Power for Test of (Not-)Close Fit

- RMSEA estimates a population value
  - Its sampling distribution has been worked out
  - So we can put a confidence interval around it
  - This confidence interval allows us to ask whether RMSEA is significantly different from a specified value
- If the population model fit is NOT CLOSE, what is power to reject H_0 by the test of close fit?
- If the population model fit is CLOSE, what is power to reject H_0 by the test of not-close fit?
- Method described in MacCallum et al. (1996) is implemented in online calculators:
  - Power and minimum sample size for RMSEA: `http://quantpsy.org/rmsea/rmsea.htm`
  - Power curves for RMSEA: `http://quantpsy.org/rmsea/rmseaplot.htm`
  - See also `findRMSEAsamplesize` in semTools

Sample size required to reject $RMSEA < 0.05$ is the true $RMSEA = 0.1$ and $DF = 20$:

```
library("semTools")
findRMSEAsamplesize(rmsea0=.05, rmseaA=.1, df=20, power=.80)

## [1] 184
```

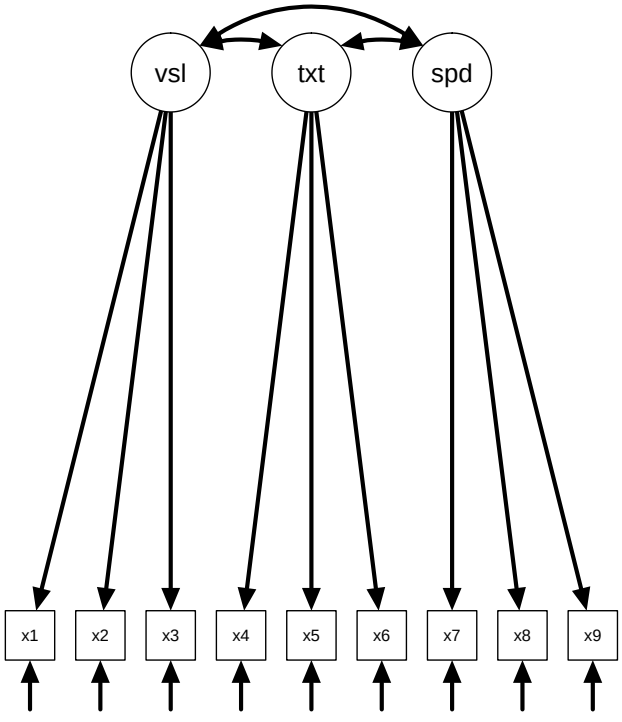
Sample size required to reject $RMSEA > 0.08$ is the true $RMSEA = 0.02$ and $DF = 20$:

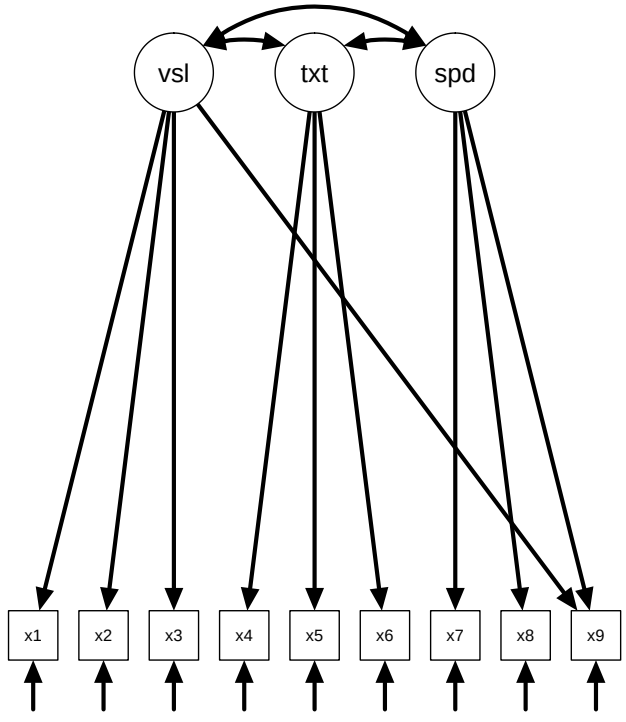
```
findRMSEAsamplesize(rmsea0=.08, rmseaA=.02, df=20, power=.80)

## [1] 194
```

Another model:

```
Model2 <- '
  visual  =~ x1 + x2 + x3 + x9
  textual =~ x4 + x5 + x6
  speed   =~ x7 + x8 + x9
'

# Fit in lavaan:
fit2 <- cfa(Model2, Data)
```

Competing models





A crash course on model selection

Different options:

- Model with a lower Akaike information criterion (AIC) is better
- Model with a lower Bayesian information criterion (BIC) is better
- If models significantly differ, the more complicated model is better
- If models do not significantly differ, the less complicated model is better

Model comparison

Given two models: model A with test statistic T_A and degrees of freedom DF_A, and model B with test statistic T_B and degrees of freedom DF_B. Then we can use the χ^2 distribution to test their difference:

$$T_A - T_B \sim \chi^2(\mathrm{DF}_A - \mathrm{DF}_B)$$

Only if:

- Model B fits the data reasonably well
- Model A is nested in model B
  - Any $\boldsymbol{\Sigma}$ obtained in model A can be reproduced in model B
  - Often: some parameters in model B can be constrained (e.g., fixed to zero) to obtain model A

This test is called a likelihood ratio test

```r
Ta <- fitMeasures(fit)['chisq']
Tb <- fitMeasures(fit2)['chisq']

DFa <- fitMeasures(fit)['df']
DFb <- fitMeasures(fit2)['df']

pchisq(Ta - Tb, DFa - DFb, lower.tail=FALSE)
```

```
##        chisq
## 9.586212e-09
```

```r
# or via lavaan:
anova(fit, fit2)
```

```
## Chi Square Difference Test
##
##      Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## fit2 23 7486.6 7568.1 52.382
## fit  24 7517.5 7595.3 85.305     32.923       1  9.586e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

You can always compare information criteria (lower is better), using Akaike's information criterion (AIC):

$$\mathrm{AIC} = -2\ln(L) + 2q$$

or the Bayesian information criterion (BIC):

$$\mathrm{BIC} = -2\ln(L) + q\ln(n)$$

q is the number of parameters and $\ln(L)$ the log-likelihood.

```
anova(fit, fit2)

## Chi Square Difference Test
##
##      Df    AIC    BIC  Chisq Chisq diff Df diff Pr(>Chisq)
## fit2 23 7486.6 7568.1 52.382
## fit  24 7517.5 7595.3 85.305     32.923       1  9.586e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Modification Indices

- Given a poorly-fitting model, you may want to know what path(s) you could add to make it fit better.
- Modification indices are powerful tools that give the expected reduction in the chi-square test statistic that would result if you added particular parameters.
  - `mi` = "modification index" = expected decrease in test statistic
  - `epc` = "expected parameter change" = the approximate value

Modification indices

```
mod <- modindices(fit)

library("dplyr")
mod %>% arrange(-mi) %>% head(10)
```

```
##         lhs op rhs        mi        epc    sepc.lv   sepc.all    sepc.
## 1    visual =~  x9 36.411031  0.5770215  0.5191001  0.5152491  0.5152
## 2        x7 ~~  x8 34.145089  0.5364440  0.5364440  0.8591510  0.8591
## 3    visual =~  x7 18.630638 -0.4218624 -0.3795158 -0.3489088 -0.3489
## 4        x8 ~~  x9 14.946392 -0.4230959 -0.4230959 -0.8052026 -0.8052
## 5  textual =~  x3  9.150895 -0.2716376 -0.2688377 -0.2380993 -0.2380
## 6        x2 ~~  x7  8.918022 -0.1827254 -0.1827254 -0.1919302 -0.1919
## 7  textual =~  x1  8.902732  0.3503311  0.3467201  0.2974884  0.2974
## 8        x2 ~~  x3  8.531827  0.2182393  0.2182393  0.2230502  0.2230
## 9        x3 ~~  x5  7.858085 -0.1300947 -0.1300947 -0.2119402 -0.2119
## 10   visual =~  x5  7.440646 -0.2098980 -0.1888284 -0.1465688 -0.1465
```

Users should be cautious in their use of modification indices. If a model was created with the aid of MIs, then it should *always* be reported. ***Do not pretend that you have a theoretical reason for part of a model that was put there because it was suggested by a modification index. This is fraud.*** When using modification indices there are two options for best practices. First, you can report the analyses as exploratory. Document all the explorations that you did, and know that your results may or may not generalize. Second, you can use cross-validation. Reserve part of your data for exploration, and use the remaining data to test if the exploratory model generalizes to new data.

From OpenMx documentation

- Fitting CFA models
  - lavaan, Onyx, Jasp, psychonetrics
- Testing for exact fit
  - χ^2 test
- Assessing close fit
  - RMSEA (below 0.05 to 0.08)
  - SRMR (below 0.05)
  - CFI, RNI, NFI, TLI, RFI, IFI (above 0.90 to 0.95)
  - (A) GFI (above 0.90)
- Sample size requirements are complicated, but power can be computed for RMSEA test of (non)close fit
- Model comparison
  - Likelihood ratio test
  - Information criteria
  - Modefication indices

Thank you for your attention!