Psychology 454: Latent Variable Modeling

Further adventures with lavaan

Department of Psychology
Northwestern University
Evanston, Illinois USA

November, 2016
Outline

Introduction to CFA/SEM programs
  What is lavaan?
  lavaan syntax
Confirmatory Factor Analysis
  A simple confirmatory analysis
  compare with EFA
Fixing parameters - starting values and equality constraints
  Providing names to parameters
Means structure
  Multiple groups
  Measurement invariance
Growth Curve analysis
  The STARS model
More statistics
  Modifying the model
More examples
Warnings
Latent Variable Modeling programs

• Commercial programs
  • EQS (Bentler, 1995)

• Open source
  • Mx (Neale, 1994)
  • OpenMx
  • sem (Fox, Nie & Byrnes, 2013)
  • lavaan (Rosseel, 2012)
lavaan 0.5.22 from CRAN. Description from the user’s guide

http://lavaan.ugent.be

- The lavaan package is free open-source software. This means (among other things) that there is no warranty whatsoever.
- The numerical results of the lavaan package are typically very close, if not identical, to the results of the commercial package Mplus. If you wish to compare the results with other SEM packages, you can use the optional argument mimic="EQS" when calling the cfa, sem or growth functions.
- The lavaan package is not finished yet. But it is already very useful for most users, or so we hope. There are a number of known minor issues and some features are simply not implemented yet.
- Some important features that are currently not available in lavaan are:
  - support for hierarchical/multilevel datasets (multilevel cfa, multilevel sem)
  - support for discrete latent variables (mixture models, latent classes)
More about lavaan from the user’s guide

1. We do not expect you to be an expert in R. In fact, the lavaan package is designed to be used by users that would normally never use R. Nevertheless, it may help to familiarize yourself a bit with R, just to be comfortable with it. Perhaps the most important skill that you may need to learn is how to import your own datasets (perhaps in an SPSS format) into R. There are many tutorials on the web to teach you just that. Once you have your data in R, you can start specifying your model. We have tried very hard to make it as easy as possible for users to fit their models. Of course, if you have suggestions on how we can improve things, please let us know.
Data sets available in lavaan

- HolzingerSwineford1939: A data frame with 301 observations of 15 variables.
  - The classic Holzinger and Swineford (1939) dataset consists of mental ability test scores of seventh- and eighth-grade children from two different schools (Pasteur and Grant-White). In the original dataset (available in the MBESS package), there are scores for 26 tests. However, a smaller subset with 9 variables is more widely used in the literature (for example in Joreskog’s 1969 paper, which also uses the 145 subjects from the Grant-White school only).

- PoliticalDemocracy: A data frame of 75 observations of 11 variables.
  - The “famous” Industrialization and Political Democracy dataset. This dataset is used throughout Bollen’s 1989 book (see pages 12, 17, 36 in chapter 2, pages 228 and following in chapter 7, pages 321 and following in chapter 8). The dataset contains various measures of political democracy and industrialization in developing countries.
item syntax

Regression
y ~ f1 + f2 + x1 + x2
f1 ~ f2 + f3
f2 ~ f3 + x1 + x2

Latent variables
f1 =~ y1 + y2 + y3
f2 =~ y4 + y5 + y6
f3 =~ y7 + y8 + y9 + y10

Variances and covariances
y1 ~~ y1
y1 ~~ y2
f1 ~~ f2

Intercepts
y1 ~ 1
f1 ~ 1
Entering the model syntax as a string literal

```r
myModel <- '# regressions
y1 + y2 ~ f1 + f2 + x1 + x2
f1 ~ f2 + f3
f2 ~ f3 + x1 + x2
# latent variable definitions
f1 =~ y1 + y2 + y3
f2 =~ y4 + y5 + y6
f3 =~ y7 + y8 + y9 + y10
# variances and covariances
y1 ~~ y1
y1 ~~ y2
f1 ~~ f2
# intercepts
y1 ~ 1
f1 ~ 1
'
```
The HolzingerSwineford data set

R code

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

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

summary(fit, fit.measures = TRUE)
lavaan.diagram(fit)  # need to use the newer function
```
Holzinger Swineford analysis

Root Mean Square Error of Approximation:

<table>
<thead>
<tr>
<th>RMSEA</th>
<th>0.092</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 Percent Confidence Interval</td>
<td>0.071 0.114</td>
</tr>
</tbody>
</table>

lavaan (0.5-22) converged normally after 35 iterations P-value RMSEA <= 0.05 0.001

Number of observations 301

Estimator ML SRMR 0.065

Minimum Function Test Statistic 85.306

Degrees of freedom 24

P-value (Chi-square) 0.000

Model test baseline model:

Minimum Function Test Statistic 918.852

Degrees of freedom 36

P-value 0.000

User model versus baseline model:

Comparative Fit Index (CFI) 0.931

Tucker-Lewis Index (TLI) 0.896

Loglikelihood and Information Criteria:

Loglikelihood user model (H0) -3737.745

Loglikelihood unrestricted model (H1) -3695.092

Number of free parameters 21

Akaike (AIC) 7517.490

BIC 7595.339

Sample-size adjusted BIC 7528.739

Root Mean Square Error of Approximation:

<table>
<thead>
<tr>
<th>RMSEA</th>
<th>0.092</th>
</tr>
</thead>
<tbody>
<tr>
<td>90 Percent Confidence Interval</td>
<td>0.071 0.114</td>
</tr>
</tbody>
</table>

Standardized Root Mean Square Residual:

User model versus baseline model:

Comparative Fit Index (CFI) 0.931

Tucker-Lewis Index (TLI) 0.896

Loglikelihood and Information Criteria:

Loglikelihood user model (H0) -3737.745

Loglikelihood unrestricted model (H1) -3695.092

Number of free parameters 21

Akaike (AIC) 7517.490

BIC 7595.339

Sample-size adjusted BIC 7528.739

Latent Variables:

<table>
<thead>
<tr>
<th>visual =~</th>
<th>x1 1.000</th>
<th>x2 0.554 0.100 5.554 0.000</th>
<th>x3 0.729 0.109 6.685 0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>textual =~</td>
<td>x4 1.000</td>
<td>x5 1.113 0.065 17.014 0.000</td>
<td>x6 0.926 0.055 16.703 0.000</td>
</tr>
<tr>
<td>speed =~</td>
<td>x7 1.000</td>
<td>x8 1.180 0.165 7.152 0.000</td>
<td>x9 1.082 0.151 7.155 0.000</td>
</tr>
</tbody>
</table>

Parameter Estimates:

<table>
<thead>
<tr>
<th>visual =~</th>
<th>x1 1.000</th>
<th>x2 0.554 0.100 5.554 0.000</th>
<th>x3 0.729 0.109 6.685 0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>textual =~</td>
<td>x4 1.000</td>
<td>x5 1.113 0.065 17.014 0.000</td>
<td>x6 0.926 0.055 16.703 0.000</td>
</tr>
<tr>
<td>speed =~</td>
<td>x7 1.000</td>
<td>x8 1.180 0.165 7.152 0.000</td>
<td>x9 1.082 0.151 7.155 0.000</td>
</tr>
</tbody>
</table>
## more parameters

### Covariances:

|               | Estimate | Std.Err | z-value | P(>|z|) |
|---------------|----------|---------|---------|---------|
| visual ~ textual | 0.408    | 0.074   | 5.552   | 0.000   |
| visual ~ speed | 0.262    | 0.056   | 4.660   | 0.000   |
| textual ~ speed | 0.173    | 0.049   | 3.518   | 0.000   |

### Variances:

|       | Estimate | Std.Err | z-value | P(>|z|) |
|-------|----------|---------|---------|---------|
| .x1   | 0.549    | 0.114   | 4.833   | 0.000   |
| .x2   | 1.134    | 0.102   | 11.146  | 0.000   |
| .x3   | 0.844    | 0.091   | 9.317   | 0.000   |
| .x4   | 0.371    | 0.048   | 7.779   | 0.000   |
| .x5   | 0.446    | 0.058   | 7.642   | 0.000   |
| .x6   | 0.356    | 0.043   | 8.277   | 0.000   |
| .x7   | 0.799    | 0.081   | 9.823   | 0.000   |
| .x8   | 0.488    | 0.074   | 6.573   | 0.000   |
| .x9   | 0.566    | 0.071   | 8.003   | 0.000   |
| visual | 0.809    | 0.145   | 5.564   | 0.000   |
| textual | 0.979    | 0.112   | 8.737   | 0.000   |
| speed  | 0.384    | 0.086   | 4.451   | 0.000   |
Graphic output using (revised) lavaan.diagram

Confirmatory structure
Redo with alternative parameterization

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

fit <- cfa(HS.model, data = HolzingerSwineford1939,std.ov=TRUE,std.lv=TRUE)

summary(fit, fit.measures = TRUE)
lavaan.diagram(fit)  #need to use the newer function
```
lavaan (0.5-22) converged normally after 21 iterations

Number of observations 301

Estimator ML

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

Model test baseline model:

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

User model versus baseline model:

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

Loglikelihood and Information Criteria:

Loglikelihood user model (H0) -3422.624
Loglikelihood unrestricted model (H1) -3379.971

Number of free parameters 21

Akaike (AIC) 6887.248
Bayesian (BIC) 6965.097
Sample-size adjusted Bayesian (BIC) 6898.497

RMSEA
90 Percent Confidence Interval 0.071 0.114
P-value RMSEA <= 0.05 0.001

Standardized Root Mean Square Residual:

SRMR 0.065

Parameter Estimates:

Latent Variables:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| visual =~ x1 | 0.771 | 0.069 | 11.127 | 0.000 |
| x2 | 0.423 | 0.066 | 6.429 | 0.000 |
| x3 | 0.580 | 0.066 | 8.817 | 0.000 |
| textual =~ x4 | 0.850 | 0.049 | 17.474 | 0.000 |
| x5 | 0.854 | 0.049 | 17.576 | 0.000 |
| x6 | 0.837 | 0.049 | 17.082 | 0.000 |
| speed =~ x7 | 0.569 | 0.064 | 8.903 | 0.000 |
| x8 | 0.722 | 0.065 | 11.090 | 0.000 |
| x9 | 0.664 | 0.064 | 10.305 | 0.000 |

Covariances:

<table>
<thead>
<tr>
<th>visual ~</th>
<th>textual</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0.459</td>
</tr>
</tbody>
</table>
### Covariances:

| Covariance | Estimate | Std.Err | z-value | P(>|z|) |
|------------|----------|---------|---------|---------|
| visual ~~ textual | 0.459 | 0.064 | 7.189 | 0.000 |
| visual ~~ speed | 0.471 | 0.073 | 6.461 | 0.000 |
| textual ~~ speed | 0.283 | 0.069 | 4.117 | 0.000 |

### Variances:

| Variable | Estimate | Std.Err | z-value | P(>|z|) |
|----------|----------|---------|---------|---------|
| .x1      | 0.403    | 0.083   | 4.833   | 0.000   |
| .x2      | 0.818    | 0.073   | 11.146  | 0.000   |
| .x3      | 0.660    | 0.071   | 9.317   | 0.000   |
| .x4      | 0.274    | 0.035   | 7.779   | 0.000   |
| .x5      | 0.268    | 0.035   | 7.642   | 0.000   |
| .x6      | 0.297    | 0.036   | 8.277   | 0.000   |
| .x7      | 0.673    | 0.069   | 9.823   | 0.000   |
| .x8      | 0.476    | 0.072   | 6.573   | 0.000   |
| .x9      | 0.556    | 0.069   | 8.003   | 0.000   |
| visual   | 1.000    |         |         |         |
| textual  | 1.000    |         |         |         |
| speed    | 1.000    |         |         |         |
The standardized solution to the Holzinger Swineford 1939 problem

Confirmatory structure
EFA of the Holzinger Swineford 1939 problem

\[ f3 \leftarrow \text{fa(HolzingerSwineford1939[7:15],3)} \]

```
f3
```
```
diagram(f3,cut=.1)
```

Factor Analysis using method = minres
Call: fa(r = HolzingerSwineford1939[7:15], nfactors = 3)

Standardized loadings (pattern matrix)

| x1  | 0.19 | 0.60 | 0.03 | 0.49 | 0.51 | 1.2 |
| x2  | 0.04 | 0.51 | -0.12 | 0.25 | 0.75 | 1.1 |
| x3  | -0.07 | 0.69 | 0.02 | 0.46 | 0.54 | 1.0 |
| x4  | 0.84 | 0.02 | 0.01 | 0.72 | 0.28 | 1.0 |
| x5  | 0.89 | -0.07 | 0.01 | 0.76 | 0.24 | 1.0 |
| x6  | 0.81 | 0.08 | -0.01 | 0.69 | 0.31 | 1.0 |
| x7  | 0.04 | -0.15 | 0.72 | 0.50 | 0.50 | 1.1 |
| x8  | -0.03 | 0.10 | 0.70 | 0.53 | 0.47 | 1.0 |
| x9  | 0.03 | 0.37 | 0.46 | 0.46 | 0.54 | 1.9 |

Mean item complexity = 1.2

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the null model are 36 and the objective function was 3.05 with Chi Square of 904.1

The degrees of freedom for the model are 12 and the objective function was 0.08

The root mean square of the residuals (RMSR) is 0.02

The df corrected root mean square of the residuals is 0.03

The harmonic number of observations is 301 with the empirical chi square 8.03 with prob < 0.78

The total number of observations was 301 with Likelihood Chi Square = 22.38 with prob < 0.034

Tucker Lewis Index of factoring reliability = 0.964

RMSEA index = 0.003 and the 90\% confidence intervals are 0.003 0.088

BIC = -46.11

Fit based upon off diagonal values = 1

Measures of factor score adequacy

| x1  | 0.94 | 0.84 | 0.85 |
| x2  | 0.33 | 0.22 |
| x3  | 0.60 | 0.03 | 0.22 |
| x4  | 0.33 | 1.00 | 0.27 |
| x5  | 0.22 | 0.27 | 1.00 |

Mean item complexity = 1.2

The root mean square of the residuals (RMSR) is 0.02

The df corrected root mean square of the residuals is 0.03

The harmonic number of observations is 301 with the empirical chi square 8.03 with prob < 0.78

The total number of observations was 301 with Likelihood Chi Square = 22.38 with prob < 0.034

Tucker Lewis Index of factoring reliability = 0.964

RMSEA index = 0.003 and the 90\% confidence intervals are 0.003 0.088

BIC = -46.11

Fit based upon off diagonal values = 1

Measures of factor score adequacy

| x1  | 0.94 | 0.84 | 0.85 |
| x2  | 0.33 | 0.22 |
| x3  | 0.60 | 0.03 | 0.22 |
| x4  | 0.33 | 1.00 | 0.27 |
| x5  | 0.22 | 0.27 | 1.00 |
Holzinger Swineford EFA – compare with CFA

Factor Analysis

MR1
0.9
0.8
0.8
MR3
0.7
0.6
0.5
MR2
0.7
0.7
0.5
0.3
0.2
0.3
Holzinger Swineford EFA – simple is FALSE

diagram(f3,cut=.1,simple=FALSE)

Factor Analysis
Fixing parameters to simply models

• Perhaps the greatest power of SEM type programs is the ability to fix parameters or to force equality constraints.
  • EFA allows all parameters to vary
  • CFA allows only certain parameters to vary
• Can fix covariances to be zero
• Can fix different paths to be equal
Two ways of fixing the covariances to be 0

```r
HS.ortho <- ' # three-factor model
visual =~ x1 + x2 + x3
textual =~ x4 + x5 + x6
speed =~ NA*x7 + x8 + x9 # orthogonal factors
visual ~~ 0*speed
textual ~~ 0*speed
speed ~~ 1*speed'
fit.hs.ortho <- cfa(HS.ortho, data=HolzingerSwineford1939, std.ov=TRUE, std.lv=TRUE)
nor (if they are all to be zero)

HS.model <- ' visual =~ x1 + x2 + x3
textual =~ x4 + x5 + x6
speed =~ x7 + x8 + x9'
fit.HS.ortho <- cfa(HS.model, data=HolzingerSwineford1939, orthogonal=TRUE)
lavaan (0.5-22) converged normally after 20 iterations

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of observations</td>
<td>301</td>
</tr>
<tr>
<td>Estimator</td>
<td>ML</td>
</tr>
<tr>
<td>Minimum Function Test Statistic</td>
<td>117.946</td>
</tr>
<tr>
<td>Degrees of freedom</td>
<td>26</td>
</tr>
<tr>
<td>P-value (Chi-square)</td>
<td>0.000</td>
</tr>
</tbody>
</table>
Not as good a fit

> summary(fit.hs.ortho, fit.measures=TRUE)

lavaan (0.5-22) converged normally after 20 iterations

Number of observations 301
Estimator ML
Minimum Function Test Statistic 117.946
Degrees of freedom 26
P-value (Chi-square) 0.000

Model test baseline model:

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

User model versus baseline model:

Comparative Fit Index (CFI) 0.896
Tucker-Lewis Index (TLI) 0.856

Loglikelihood and Information Criteria:

Loglikelihood user model (H0) -3438.944
Loglikelihood unrestricted model (H1) -3379.971
Number of free parameters 19

Akaike (AIC) 6915.889
Bayesian (BIC) 6986.324
Sample-size adjusted Bayesian (BIC) 6926.067

Root Mean Square Error of Approximation:

RMSEA 0.108
90 Percent Confidence Interval 0.089 0.129
P-value RMSEA <= 0.05 0.000

Standardized Root Mean Square Residual:

SRMR 0.125

Parameter Estimates:

Latent Variables:

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

visual =~
  x1  0.777  0.075  10.376  0.000
  x2  0.430  0.067  6.423  0.000
  x3  0.568  0.069  8.270  0.000

textual =~
  x4  0.851  0.049  17.491  0.000
  x5  0.853  0.049  17.545  0.000
  x6  0.837  0.049  17.079  0.000

speed =~
  x7  0.607  0.067  9.040  0.000
  x8  0.800  0.073 10.899  0.000
  x9  0.560  0.066  8.509  0.000
## Covariances:

|        | Estimate | Std.Err | z-value | P(>|z|) |
|--------|----------|---------|---------|---------|
| visual | 0.000    |         |         |         |
| speed  | 0.000    |         |         |         |
| textual| 0.461    | 0.064   | 7.195   | 0.000   |

## Variances:

|        | Estimate | Std.Err | z-value | P(>|z|) |
|--------|----------|---------|---------|---------|
| speed  | 1.000    |         |         |         |
| .x1    | 0.394    | 0.095   | 4.155   | 0.000   |
| .x2    | 0.811    | 0.074   | 10.965  | 0.000   |
| .x3    | 0.675    | 0.074   | 9.085   | 0.000   |
| .x4    | 0.273    | 0.035   | 7.735   | 0.000   |
| .x5    | 0.269    | 0.035   | 7.662   | 0.000   |
| .x6    | 0.297    | 0.036   | 8.263   | 0.000   |
| .x7    | 0.629    | 0.073   | 8.650   | 0.000   |
| .x8    | 0.357    | 0.094   | 3.794   | 0.000   |
| .x9    | 0.683    | 0.071   | 9.640   | 0.000   |
| visual | 1.000    |         |         |         |
| textual| 1.000    |         |         |         |
Fixing starting values

(If you have a problem with a solution, you can help it if you give it a reasonable starting location.)

\[
\text{visual} = \sim x_1 + \text{start}(0.8)\times x_2 + \text{start}(1.2)\times x_3 \\
\text{textual} = \sim x_4 + \text{start}(0.5)\times x_5 + \text{start}(1.0)\times x_6 \\
\text{speed} = \sim x_7 + \text{start}(0.7)\times x_8 + \text{start}(1.8)\times x_9
\]

This technique works for all SEM programs (although details vary). The reason to give good starting values is that the search optimization can get bogged down in the wrong part of the parameter space.
Automatic naming

```r
> model <- ' 
# latent variable definitions
ind60 =~ x1 + x2 + x3 
dem60 =~ y1 + y2 + y3 + y4 
dem65 =~ y5 + y6 + y7 + y8 
# regressions
dem60 ~ ind60 
dem65 ~ ind60 + dem60 
# residual (co)variances
y1 ~~ y5 
y2 ~~ y4 + y6 
y3 ~~ y7 
y4 ~~ y8 
y6 ~~ y8 
'
> fit <- sem(model, data=PoliticalDemocracy)
coef(fit)

ind60=~x2 ind60=~x3 dem60=~y2 dem60=~y3 dem60=~y4 dem65=~y6
 2.180 1.819 1.257 1.058 1.265 1.186
dem65=~y7 dem65=~y8 dem60~ind60 dem65~ind60 dem65~dem60 y1~~y5
 1.280 1.266 1.483 0.572 0.837 0.624
y2~~y4 y2~~y6 y3~~y7 y4~~y8 y6~~y8 y1~~x1
 1.313 2.153 0.795 0.348 1.356 0.082
x2~~x2 x3~~x3 y1~~y1 y2~~y2 y3~~y3 y4~~y4
 0.120 0.467 1.891 7.373 5.068 3.148
y5~~y5 y6~~y6 y7~~y7 y8~~y8 ind60~~ind60 dem60~~dem60
 2.351 4.954 3.431 3.254 0.448 3.956
dem65~~dem65
 0.172
```
Specifying the name

```r
> model <- '# latent variable definitions
    ind60 =~ x1 + x2 + label("myLabel")*x3
dem60 =~ y1 + y2 + y3 + label("anotherLabel")*y4
dem65 =~ y5 + y6 + y7 + y8
    # regressions
dem60 ~ ind60
dem65 ~ ind60 + dem60
    # residual (co)variances
y1 ~~ y5
y2 ~~ y4 + y6
    y3 ~~ y7
y4 ~~ y8
y6 ~~ y8
'

fit <- sem(model, data=PoliticalDemocracy)
coef(fit)
```

<table>
<thead>
<tr>
<th></th>
<th>ind60=~x2</th>
<th>myLabel</th>
<th>dem60=~y2</th>
<th>dem60=~y3</th>
<th>anotherLabel</th>
<th>dem65=~y6</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.180</td>
<td>1.819</td>
<td>1.257</td>
<td>1.058</td>
<td>1.265</td>
<td>1.186</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dem65~ind60</td>
<td>dem65~dem60</td>
<td>y1~~y5</td>
<td>y2~~y4</td>
<td>y2~~y6</td>
<td>y3~~y7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.572</td>
<td>0.837</td>
<td>0.624</td>
<td>1.313</td>
<td>2.153</td>
<td>0.795</td>
<td></td>
</tr>
<tr>
<td></td>
<td>x2~~x2</td>
<td>x3~~x3</td>
<td>y1~~y1</td>
<td>y2~~y2</td>
<td>y3~~y3</td>
<td>y4~~y4</td>
<td></td>
</tr>
</tbody>
</table>
Using names to specify equality constraints

```r
model <- 'visual =~ x1 + x2 + equal("visual =~x2") *x3
textual =~ x4 + x5 + x6
speed =~ x7 + x8 + x9'
fit <- sem(model, data=HolzingerSwineford1939)
coef(fit)
```

lavaan (0.5-22) converged normally after 36 iterations

| Latent Variables: | Estimate | Std.Err | z-value | P(|z|) |
|-------------------|----------|---------|---------|-------|
| visual =~         |          |         |         |       |
| x1                | 1.000    |         |         |       |
| x2 (.p2.)         | 0.649    | 0.088   | 7.355   | 0.000 |
| x3 (v=~2)         | 0.649    | 0.088   | 7.355   | 0.000 |
| textual =~        |          |         |         |       |
| x4                | 1.000    |         |         |       |
| x5                | 1.113    | 0.065   | 17.019  | 0.000 |
| x6                | 0.926    | 0.055   | 16.705  | 0.000 |
Estimate the means

R code

```r
HS.model.means <- ' # three-factor model
visual =~ x1 + x2 + x3
textual =~ x4 + x5 + x6
speed =~ x7 + x8 + x9
# intercepts
x1 ~ 1
x2 ~ 1
x3 ~ 1
x4 ~ 1
x5 ~ 1
x6 ~ 1
x7 ~ 1
x8 ~ 1
x9 ~ 1'

fit.means <- cfa(HS.model.means, data = HolzingerSwineford1939, std.lv = TRUE)
summary(fit.means, fit.measures = TRUE)

or

fit <- cfa(HS.model, data = HolzingerSwineford1939, std.lv = TRUE, meanstructure = TRUE)
summary(fit)
```
Show the variable means (intercepts)

| Intercepts:                          | Estimate | Std.Err | z-value | P(>|z|) |
|--------------------------------------|----------|---------|---------|---------|
| Number of observations               | 301      |         |         |         |
| Estimator                            | ML       |         |         |         |
| Minimum Function Test Statistic      | 85.306   |         |         |         |
| Degrees of freedom                   | 24       |         |         |         |
| P-value (Chi-square)                 | 0.000    |         |         |         |
| Parameter Estimates:                 |          |         |         |         |
| Information                          | Expected |         |         |         |
| Standard Errors                      | Standard |         |         |         |
| Latent Variables:                    |          |         |         |         |
| Visual                               |          |         |         |         |
| x1                                   | 4.936    | 0.067   | 73.473  | 0.000   |
| x2                                   | 6.088    | 0.068   | 89.855  | 0.000   |
| x3                                   | 2.250    | 0.065   | 34.579  | 0.000   |
| x4                                   | 3.061    | 0.067   | 45.694  | 0.000   |
| x5                                   | 4.341    | 0.074   | 58.452  | 0.000   |
| x6                                   | 2.186    | 0.063   | 34.667  | 0.000   |
| x7                                   | 4.186    | 0.063   | 66.766  | 0.000   |
| x8                                   | 5.527    | 0.058   | 94.854  | 0.000   |
| x9                                   | 5.374    | 0.058   | 92.546  | 0.000   |
| Textual                              |          |         |         |         |
| x4                                   |          |         |         |         |
| x5                                   | 1.102    | 0.063   | 17.576  | 0.000   |
| x6                                   | 0.917    | 0.054   | 17.082  | 0.000   |
| x7                                   |          |         |         |         |
| x8                                   |          |         |         |         |
| speed                                |          |         |         |         |
| x7                                   | 0.619    | 0.070   | 8.903   | 0.000   |
| x8                                   | 0.731    | 0.066   | 11.090  | 0.000   |

<table>
<thead>
<tr>
<th>Variances:</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x1</td>
<td>0.549</td>
<td>0.114</td>
<td>4.833</td>
<td>0.000</td>
</tr>
<tr>
<td>x2</td>
<td>1.134</td>
<td>0.102</td>
<td>11.146</td>
<td>0.000</td>
</tr>
<tr>
<td>x3</td>
<td>0.844</td>
<td>0.091</td>
<td>9.317</td>
<td>0.000</td>
</tr>
<tr>
<td>x4</td>
<td>0.371</td>
<td>0.048</td>
<td>7.778</td>
<td>0.000</td>
</tr>
<tr>
<td>x5</td>
<td>0.446</td>
<td>0.058</td>
<td>7.642</td>
<td>0.000</td>
</tr>
<tr>
<td>x6</td>
<td>0.356</td>
<td>0.043</td>
<td>8.277</td>
<td>0.000</td>
</tr>
<tr>
<td>x7</td>
<td>0.488</td>
<td>0.074</td>
<td>6.573</td>
<td>0.000</td>
</tr>
<tr>
<td>x8</td>
<td>0.566</td>
<td>0.071</td>
<td>8.003</td>
<td>0.000</td>
</tr>
<tr>
<td>Textual</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speed</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x7</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x8</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x9</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Show the parameter estimates for the meanstructure=TRUE

### Latent Variables:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| visual =~ |         |         |         |
| x1       | 0.900   | 0.081   | 11.127  | 0.000  |
| x2       | 0.498   | 0.077   | 6.429   | 0.000  |
| x3       | 0.656   | 0.074   | 8.817   | 0.000  |
| textual =~ |       |         |         |
| x4       | 0.990   | 0.057   | 17.474  | 0.000  |
| x5       | 1.102   | 0.063   | 17.576  | 0.000  |
| x6       | 0.917   | 0.054   | 17.082  | 0.000  |
| speed =~ |         |         |         |
| x7       | 0.619   | 0.070   | 8.903   | 0.000  |
| x8       | 0.731   | 0.066   | 11.090  | 0.000  |
| x9       | 0.670   | 0.065   | 10.305  | 0.000  |

### Covariances:

| Estimate | Std.Err | z-value | P(>|z|) |
|----------|---------|---------|---------|
| visual ~~ textual | 0.459   | 0.064   | 7.189   | 0.000  |
| speed    | 0.471   | 0.073   | 6.461   | 0.000  |
| textual ~~ speed | 0.283   | 0.069   | 4.117   | 0.000  |

### Intercepts:

| .x1       | Est. 4.936 | Std.Err 0.067 | z-value 73.473 | P(>|z|) 0.000 |
| .x2       | Est. 6.088 | Std.Err 0.068 | z-value 89.855 | P(>|z|) 0.000 |
| .x3       | Est. 2.250 | Std.Err 0.065 | z-value 34.579 | P(>|z|) 0.000 |
| .x4       | Est. 3.061 | Std.Err 0.067 | z-value 45.694 | P(>|z|) 0.000 |
| .x5       | Est. 4.341 | Std.Err 0.074 | z-value 58.452 | P(>|z|) 0.000 |
| .x6       | Est. 2.186 | Std.Err 0.063 | z-value 34.667 | P(>|z|) 0.000 |
| visual    | Est. 4.186 | Std.Err 0.063 | z-value 66.766 | P(>|z|) 0.000 |
| textual   | Est. 5.527 | Std.Err 0.058 | z-value 94.854 | P(>|z|) 0.000 |
| speed     | Est. 5.374 | Std.Err 0.058 | z-value 92.546 | P(>|z|) 0.000 |

### Variances:

| .x1       | Est. 0.549 | Std.Err 0.114 | z-value 4.833 | P(>|z|) 0.000 |
| .x2       | Est. 1.134 | Std.Err 0.102 | z-value 11.146 | P(>|z|) 0.000 |
| .x3       | Est. 0.844 | Std.Err 0.091 | z-value 9.317 | P(>|z|) 0.000 |
| .x4       | Est. 0.371 | Std.Err 0.048 | z-value 7.778 | P(>|z|) 0.000 |
| .x5       | Est. 0.446 | Std.Err 0.058 | z-value 7.642 | P(>|z|) 0.000 |
| .x6       | Est. 0.356 | Std.Err 0.043 | z-value 8.277 | P(>|z|) 0.000 |
| .x7       | Est. 0.799 | Std.Err 0.081 | z-value 9.823 | P(>|z|) 0.000 |
| .x8       | Est. 0.488 | Std.Err 0.074 | z-value 6.573 | P(>|z|) 0.000 |
| .x9       | Est. 0.566 | Std.Err 0.071 | z-value 8.003 | P(>|z|) 0.000 |

visual = visual

visual = textual

textual = visual

textual = textual

speed = speed

speed = speed
More useful if we want to fix some intercepts to be different from others

R code

```r
# three-factor model
model <- '
visual =~ x1 + x2 + x3
textual =~ x4 + x5 + x6
speed =~ x7 + x8 + x9
# intercepts with fixed values
x1 ~ 0.5*1
x2 ~ 0.5*1
x3 ~ 0.5*1
x4 ~ 0.5*1
'
fit.means <- cfa(model, data = HolzingerSwinford1939, std.lv=TRUE)
summary(fit.means)
```
Analyzing multiple groups

• When studying differences in ages, gender, school, it is useful to be able to model them separately, but to get an overall goodness of fit.
  • Does a basic structure hold in different groups?
• More importantly, we can ask if the parameters in the two groups are the same. That is, we can add equality constraints.
• We can examine equality of the loadings, equality of the covariances, equality of the mean structure.
```r
HS.model <- ' visual =~ x1 + x2 + x3
textual =~ x4 + x5 + x6
speed =~ x7 + x8 + x9 '
fit <- cfa(HS.model, data=HolzingerSwineford1939, group="school")
summary(fit)
```
lavaan (0.5-22) converged normally after 57 iterations

Number of observations per group
Pasteur 156
Grant-White 145

Estimator ML
Minimum Function Test Statistic 115.851
Degrees of freedom 48
P-value (Chi-square) 0.000

Chi-square for each group:
Pasteur 64.309
Grant-White 51.542

Parameter Estimates:

Information Expected
Standard Errors Standard
### Group 1 [Pasteur]:

| Latent variables: | Estimate | Std.err | Z-value | P(>|z|) |
|-------------------|----------|---------|---------|---------|
| visual =~         |          |         |         |         |
| x1                | 1.000    |         |         |         |
| x2                | 0.394    | 0.122   | 3.220   | 0.001   |
| x3                | 0.570    | 0.140   | 4.076   | 0.000   |
| textual =~        |          |         |         |         |
| x4                | 1.000    |         |         |         |
| x5                | 1.183    | 0.102   | 11.613  | 0.000   |
| x6                | 0.875    | 0.077   | 11.421  | 0.000   |
| speed =~          |          |         |         |         |
| x7                | 1.000    |         |         |         |
| x8                | 1.125    | 0.277   | 4.057   | 0.000   |
| x9                | 0.922    | 0.225   | 4.104   | 0.000   |

### Covariances:

| visual ~~ textual | 0.479    | 0.106   | 4.531   | 0.000   |
| visual ~~ speed   | 0.185    | 0.077   | 2.397   | 0.017   |
| textual ~~ speed  | 0.182    | 0.069   | 2.628   | 0.009   |

### Variances:

| x1                | 0.298    | 0.232   | 1.286   | 0.198   |
| x2                | 1.334    | 0.158   | 8.464   | 0.000   |
| x3                | 0.989    | 0.136   | 7.271   | 0.000   |
| x4                | 0.425    | 0.069   | 6.138   | 0.000   |
| x5                | 0.456    | 0.086   | 5.292   | 0.000   |
| x6                | 0.290    | 0.050   | 5.780   | 0.000   |
| x7                | 0.820    | 0.125   | 6.580   | 0.000   |
| x8                | 0.510    | 0.116   | 4.406   | 0.000   |
| x9                | 0.680    | 0.104   | 6.516   | 0.000   |
| visual            | 1.097    | 0.276   | 3.967   | 0.000   |

### Group 2 [Grant-White]:

| Latent variables: | Estimate | Std.err | Z-value | P(>|z|) |
|-------------------|----------|---------|---------|---------|
| visual =~         |          |         |         |         |
| x1                | 1.000    |         |         |         |
| x2                | 0.736    | 0.155   | 4.760   | 0.000   |
| x3                | 0.925    | 0.166   | 5.584   | 0.000   |
| textual =~        |          |         |         |         |
| x4                | 1.000    |         |         |         |
| x5                | 0.990    | 0.087   | 11.418  | 0.000   |
| x6                | 0.963    | 0.085   | 11.377  | 0.000   |
| speed =~          |          |         |         |         |
| x7                | 1.000    |         |         |         |
| x8                | 1.226    | 0.187   | 6.569   | 0.000   |
| x9                | 1.058    | 0.165   | 6.429   | 0.000   |

### Covariances:

| visual ~~ textual | 0.408    | 0.098   | 4.153   | 0.000   |
| visual ~~ speed   | 0.276    | 0.076   | 3.639   | 0.000   |
| textual ~~ speed  | 0.222    | 0.073   | 3.022   | 0.003   |

### Variances:

| x1                | 0.715    | 0.126   | 5.675   | 0.000   |
| x2                | 0.899    | 0.123   | 7.339   | 0.000   |
| x3                | 0.557    | 0.103   | 5.409   | 0.000   |
| x4                | 0.315    | 0.065   | 4.870   | 0.000   |
| x5                | 0.419    | 0.072   | 5.812   | 0.000   |
| x6                | 0.406    | 0.069   | 5.880   | 0.000   |
| x7                | 0.600    | 0.091   | 6.584   | 0.000   |
| x8                | 0.401    | 0.094   | 4.248   | 0.000   |
| x9                | 0.535    | 0.089   | 6.010   | 0.000   |
| visual            | 0.604    | 0.160   | 3.762   | 0.000   |
Multiple groups, multiple constraints

- Can constrain a single parameter to be equal across groups
  - Use the naming convention and the equal command
  - Or, just use the same name for the two parameters

- Can constrain equivalent parameters across groups to be equal (`group.equal`)
Equal loadings across groups

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

fit <- cfa(HS.model, data=HolzingerSwineford1939, group="school",
   group.equal=c("loadings"),std.lv=TRUE)
summary(fit)

llavaan (0.5-22) converged normally after 30 iterations

Number of observations per group
Pasteur 156
Grant-White 145

Estimator ML
Minimum Function Test Statistic 127.834
Degrees of freedom 57
P-value (Chi-square) 0.000

Chi-square for each group:
Pasteur 71.064
Grant-White 56.770

Parameter Estimates:
Group 1 [Pasteur]:

Latent Variables:

| Latent Variable | Estimate | Std.Err | z-value | P(>|z|) |
|-----------------|----------|---------|---------|--------|
| visual =~       |          |         |         |        |
| x1 (.p1.)      | 0.866    | 0.078   | 11.149  | 0.000  |
| x2 (.p2.)      | 0.523    | 0.076   | 6.916   | 0.000  |
| x3 (.p3.)      | 0.683    | 0.071   | 9.689   | 0.000  |
| textual =~     |          |         |         |        |
| x4 (.p4.)      | 0.954    | 0.056   | 17.002  | 0.000  |
| x5 (.p5.)      | 1.033    | 0.061   | 17.012  | 0.000  |
| x6 (.p6.)      | 0.870    | 0.052   | 16.750  | 0.000  |
| speed =~       |          |         |         |        |
| x7 (.p7.)      | 0.630    | 0.066   | 9.500   | 0.000  |
| x8 (.p8.)      | 0.752    | 0.065   | 11.586  | 0.000  |
| x9 (.p9.)      | 0.650    | 0.064   | 10.205  | 0.000  |

Covariances:

| Covariances | Estimate | Std.Err | z-value | P(>|z|) |
|-------------|----------|---------|---------|--------|
| visual ~ textual | 0.485 | 0.087 | 5.555 | 0.000 |
| textual ~ speed | 0.341 | 0.109 | 3.126 | 0.002 |
| textual ~ speed | 0.336 | 0.094 | 3.590 | 0.000 |

Intercepts:

| Intercepts | Estimate | Std.Err | z-value | P(>|z|) |
|------------|----------|---------|---------|--------|
| .x1        | 4.941    | 0.092   | 53.661  | 0.000  |
| .x2        | 5.984    | 0.099   | 60.420  | 0.000  |
| .x3        | 2.487    | 0.093   | 26.734  | 0.000  |
| .x4        | 2.823    | 0.093   | 30.400  | 0.000  |
| .x5        | 3.995    | 0.100   | 39.756  | 0.000  |
| .x6        | 1.922    | 0.081   | 23.732  | 0.000  |
| .x7        | 4.432    | 0.089   | 48.903  | 0.000  |
Equal loadings and means across groups

\[ HS.model <- \text{'visual} = \text{x1} + \text{x2} + \text{x3} \]
\[ \text{textual} = \text{x4} + \text{x5} + \text{x6} \]
\[ \text{speed} = \text{x7} + \text{x8} + \text{x9} \]

\[
fit <- \text{cfa(HS.model, data=HolzingerSwineford1939,}
\text{group=\text{'school'},}
\text{group.equal=c('loadings','means'), std.lv=TRUE)}
\]

\text{summary(fit, fit.measures=TRUE)}

Number of observations per group

- Pasteur: 156
- Grant-White: 145

Estimator

- Minimum Function Test Statistic: 127.834
- Degrees of freedom: 57
- P-value (Chi-square): 0.000

Chi-square for each group:

- Pasteur: 71.064
- Grant-White: 56.770

Model test baseline model:

- Minimum Function Test Statistic: 957.769
- Degrees of freedom: 72
- P-value: 0.000

User model versus baseline model:

- Comparative Fit Index (CFI): 0.920
- Tucker-Lewis Index (TLI): 0.899

Loglikelihood and Information Criteria:

- Loglikelihood user model (H0): -3688.189
- Loglikelihood unrestricted model (H1): -3624.272

Additional metrics:

- Root Mean Square Error of Approximation (RMSEA): 0.091
- 90 Percent Confidence Interval: 0.070 - 0.112
- P-value RMSEA <= 0.05: 0.001
- Standardized Root Mean Square Residual (SRMR): 0.079

Parameter Estimates:

- Expected Standard Errors
- Information
- Standard

38 / 67
Group 1 [Pasteur]:

Latent Variables:

| Latent Variable | Estimate | Std.Err | z-value | P(>|z|) |
|-----------------|----------|---------|---------|---------|
| visual =~       |          |         |         |         |
| x1 (.p1.)      | 0.866    | 0.078   | 11.149  | 0.000   |
| x2 (.p2.)      | 0.523    | 0.076   | 6.916   | 0.000   |
| x3 (.p3.)      | 0.683    | 0.071   | 9.689   | 0.000   |
| textual =~     |          |         |         |         |
| x4 (.p4.)      | 0.954    | 0.056   | 17.002  | 0.000   |
| x5 (.p5.)      | 1.033    | 0.061   | 17.012  | 0.000   |
| x6 (.p6.)      | 0.870    | 0.052   | 16.750  | 0.000   |
| speed =~       |          |         |         |         |
| x7 (.p7.)      | 0.630    | 0.066   | 9.500   | 0.000   |
| x8 (.p8.)      | 0.752    | 0.065   | 11.586  | 0.000   |
| x9 (.p9.)      | 0.650    | 0.064   | 10.205  | 0.000   |

Covariances:

| Covariance     | Estimate | Std.Err | z-value | P(>|z|) |
|----------------|----------|---------|---------|---------|
| visual ~ textual | 0.485    | 0.087   | 5.555   | 0.000   |
| speed ~ textual | 0.341    | 0.109   | 3.126   | 0.002   |
| visual ~ speed  | 0.336    | 0.094   | 3.590   | 0.000   |
| textual ~ speed | 0.341    | 0.093   | 3.670   | 0.000   |

Intercepts:

| Intercept | Estimate | Std.Err | z-value | P(>|z|) |
|-----------|----------|---------|---------|---------|
| .x1       | 4.941    | 0.092   | 53.661  | 0.000   |
| .x2       | 5.984    | 0.099   | 60.420  | 0.000   |
| .x3       | 2.487    | 0.093   | 26.734  | 0.000   |
| .x4       | 2.823    | 0.093   | 30.400  | 0.000   |
| .x5       | 3.995    | 0.100   | 39.756  | 0.000   |
| .x6       | 1.922    | 0.081   | 23.732  | 0.000   |
| .x7       | 4.432    | 0.089   | 49.903  | 0.000   |
Do the measures measure the same construct across groups?

- Is the configuration the same?
  - Most abstract level of invariance – “are the arrows the same”
- Weak invariance – are the loadings the same?
- Strong invariance – equal loadings + intercepts

Use the *semTools* package and the `measurementInvariance` function.
Testing for measurement invariance

measurementInvariance(HS.model, data = HolzingerSwineford1939,
group = "school")

Measurement invariance models:

Model 1 : fit.configural
Model 2 : fit.loadings
Model 3 : fit.intercepts
Model 4 : fit.means

Chi Square Difference Test

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>AIC</th>
<th>BIC</th>
<th>Chisq</th>
<th>Chisq diff</th>
<th>Df diff</th>
<th>Pr(&gt;Chisq)</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit.configural</td>
<td>48</td>
<td>7484.4</td>
<td>7706.8</td>
<td>115.85</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fit.loadings</td>
<td>54</td>
<td>7480.6</td>
<td>7680.8</td>
<td>124.04</td>
<td>8.192</td>
<td>6</td>
<td>0.2244</td>
</tr>
<tr>
<td>fit.intercepts</td>
<td>60</td>
<td>7508.6</td>
<td>7686.6</td>
<td>164.10</td>
<td>40.059</td>
<td>6</td>
<td>4.435e-07  ***</td>
</tr>
<tr>
<td>fit.means</td>
<td>63</td>
<td>7543.1</td>
<td>7710.0</td>
<td>204.61</td>
<td>40.502</td>
<td>3</td>
<td>8.338e-09  ***</td>
</tr>
</tbody>
</table>

---

Signif. codes:  0 ????  0.001 ?***  0.01 ?**  0.05 ?*  0.1 ?  1

Fit measures:

<table>
<thead>
<tr>
<th></th>
<th>cfi</th>
<th>rmsea</th>
<th>cfi.delta</th>
<th>rmsea.delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit.configural</td>
<td>0.923</td>
<td>0.097</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>fit.loadings</td>
<td>0.921</td>
<td>0.093</td>
<td>0.002</td>
<td>0.004</td>
</tr>
<tr>
<td>fit.intercepts</td>
<td>0.882</td>
<td>0.107</td>
<td>0.038</td>
<td>0.015</td>
</tr>
<tr>
<td>fit.means</td>
<td>0.840</td>
<td>0.122</td>
<td>0.042</td>
<td>0.015</td>
</tr>
</tbody>
</table>
Data.growth: A toy data set

1. t1 Measured value at time point 1
2. t2 Measured value at time point 2
3. t3 Measured value at time point 3
4. t4 Measured value at time point 4
5. x1 Predictor 1 influencing intercept and slope
6. x2 Predictor 2 influencing intercept and slope
7. c1 Time-varying covariate time point 1
8. c2 Time-varying covariate time point 2
9. c3 Time-varying covariate time point 3
10. c4 Time-varying covariate time point 4
## Demo.growth : A toy data set

A toy dataset containing measures on 4 time points (t1, t2, t3 and t4), two predictors (x1 and x2) influencing the random intercept and slope, and a time-varying covariate (c1, c2, c3 and c4).

```r
data(Demo.growth)
describe(Demo.growth)
pairs.panels(Demo.growth)
```

<table>
<thead>
<tr>
<th>var</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>400</td>
<td>0.59</td>
<td>1.58</td>
<td>0.67</td>
<td>0.64</td>
<td>1.50</td>
<td>5.21</td>
<td>9.56</td>
<td>-0.18</td>
<td>0.23</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td>t2</td>
<td>400</td>
<td>1.67</td>
<td>2.13</td>
<td>1.88</td>
<td>1.69</td>
<td>2.10</td>
<td>9.95</td>
<td>14.81</td>
<td>-0.02</td>
<td>0.48</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>t3</td>
<td>400</td>
<td>2.59</td>
<td>2.72</td>
<td>2.73</td>
<td>2.62</td>
<td>2.62</td>
<td>11.53</td>
<td>17.55</td>
<td>-0.14</td>
<td>0.46</td>
<td>0.14</td>
<td></td>
</tr>
<tr>
<td>t4</td>
<td>400</td>
<td>3.64</td>
<td>3.38</td>
<td>3.72</td>
<td>3.69</td>
<td>3.14</td>
<td>14.72</td>
<td>22.06</td>
<td>-0.13</td>
<td>0.44</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>x1</td>
<td>400</td>
<td>-0.09</td>
<td>1.03</td>
<td>-0.08</td>
<td>-0.11</td>
<td>1.11</td>
<td>-2.82</td>
<td>2.72</td>
<td>5.54</td>
<td>0.08</td>
<td>-0.33</td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>400</td>
<td>0.14</td>
<td>0.96</td>
<td>0.13</td>
<td>0.15</td>
<td>1.01</td>
<td>-2.83</td>
<td>2.88</td>
<td>5.71</td>
<td>-0.12</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>c1</td>
<td>400</td>
<td>0.01</td>
<td>0.99</td>
<td>-0.03</td>
<td>-0.01</td>
<td>0.98</td>
<td>-2.58</td>
<td>2.57</td>
<td>5.14</td>
<td>0.11</td>
<td>-0.28</td>
<td></td>
</tr>
<tr>
<td>c2</td>
<td>400</td>
<td>0.03</td>
<td>0.95</td>
<td>0.01</td>
<td>0.02</td>
<td>0.87</td>
<td>-2.54</td>
<td>2.71</td>
<td>5.25</td>
<td>0.13</td>
<td>-0.07</td>
<td></td>
</tr>
<tr>
<td>c3</td>
<td>400</td>
<td>0.07</td>
<td>0.93</td>
<td>0.07</td>
<td>0.06</td>
<td>0.93</td>
<td>-3.40</td>
<td>2.61</td>
<td>6.01</td>
<td>-0.02</td>
<td>0.38</td>
<td></td>
</tr>
<tr>
<td>c4</td>
<td>400</td>
<td>-0.02</td>
<td>0.92</td>
<td>-0.01</td>
<td>-0.02</td>
<td>0.95</td>
<td>-2.45</td>
<td>2.65</td>
<td>5.11</td>
<td>0.02</td>
<td>-0.21</td>
<td></td>
</tr>
</tbody>
</table>
Growth: SPLOM
The Demo.growth data set - A cleaner graphic
Fitting a growth model to the toy problem

Intercepts and slopes

```r
model <- ' i =~ 1*t1 + 1*t2 + 1*t3 + 1*t4
 s =~ 0*t1 + 1*t2 + 2*t3 + 3*t4 '

fit <- growth(model, data=Demo.growth)
summary(fit)
```

lavaan (0.5-22) converged normally after 29 iterations

Number of observations 400

Estimator ML
Minimum Function Test Statistic 8.069
Degrees of freedom 5
P-value (Chi-square) 0.152

Parameter Estimates:

<table>
<thead>
<tr>
<th>Information</th>
<th>Expected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Errors</td>
<td>Standard</td>
</tr>
</tbody>
</table>
# Growth model with parameter values

**Latent Variables:**

| Estimate | Std.Err | z-value | P(|z|) |
|----------|---------|---------|-------|
| i =~     |         |         |       |
| t1       | 1.000   | 0.000   | 0.000 |
| t2       | 1.000   | 0.000   | 0.000 |
| t3       | 1.000   | 0.000   | 0.000 |
| t4       | 1.000   | 0.000   | 0.000 |
| s =~     |         |         |       |
| t1       | 0.000   | 0.000   | 0.000 |
| t2       | 1.000   | 0.000   | 0.000 |
| t3       | 2.000   | 0.000   | 0.000 |
| t4       | 3.000   | 0.000   | 0.000 |

**Covariances:**

| Estimate | Std.Err | z-value | P(|z|) |
|----------|---------|---------|-------|
| i ~ s    | 0.618   | 0.071   | 8.686 | 0.000 |

**Intercepts:**

| Estimate | Std.Err | z-value | P(|z|) |
|----------|---------|---------|-------|
| .t1      | 0.000   | 0.000   | 0.000 |
| .t2      | 0.000   | 0.000   | 0.000 |
| .t3      | 0.000   | 0.000   | 0.000 |
| .t4      | 0.000   | 0.000   | 0.000 |

**Variances:**

| Estimate | Std.Err | z-value | P(|z|) |
|----------|---------|---------|-------|
| .t1      | 0.595   | 0.086   | 6.944 | 0.000 |
| .t2      | 0.676   | 0.061   | 11.061| 0.000 |
| .t3      | 0.635   | 0.072   | 8.761 | 0.000 |
| .t4      | 0.508   | 0.124   | 4.090 | 0.000 |

| Estimate | Std.Err | z-value | P(|z|) |
|----------|---------|---------|-------|
| i        | 1.932   | 0.173   | 11.194| 0.000 |
| s        | 0.587   | 0.052   | 11.336| 0.000 |
From the lavaan manual

• “Technically, the growth function is almost identical to the sem function. But a meanstructure is automatically assumed, and the observed intercepts are fixed to zero by default, while the latent variable intercepts and means are freely estimated.

• A slightly more complex model adds two regressors (x1 and x2) that influence the latent growth factors.

• In addition, a time-varying covariate that influences the outcome measure at the four time points has been added to the model.

• A graphical representation of this model together with the corresponding lavaan syntax is presented”.

A growth model
Linear growth with time-varying covariates

# a linear growth model with a time-varying covariate
model <- '  
# intercept and slope with fixed coefficients
  i =~ 1*t1 + 1*t2 + 1*t3 + 1*t4
  s =~ 0*t1 + 1*t2 + 2*t3 + 3*t4
# regressions
  i ~ x1 + x2
  s ~ x1 + x2
# time-varying covariates
  t1 ~ c1
  t2 ~ c2
  t3 ~ c3
  t4 ~ c4
  
fit <- growth(model, data=Demo.growth)
summary(fit,fit.measures=TRUE)

lavaan (0.5-22) converged normally after 31 iterations

Number of observations 400

Estimator ML
Minimum Function Test Statistic 26.059
Degrees of freedom 21
P-value (Chi-square) 0.204
Parameters of the growth model

Latent Variables:

|       | Estimate | Std.Err | z-value | P(>|z|) |
|-------|----------|---------|---------|---------|
| i ~   | 1.000    |         |         |         |
| t1    | 1.000    |         |         |         |
| t2    | 1.000    |         |         |         |
| t3    | 1.000    |         |         |         |
| t4    | 1.000    |         |         |         |
| s ~   |          |         |         |         |
| t1    | 0.000    |         |         |         |
| t2    | 1.000    |         |         |         |
| t3    | 2.000    |         |         |         |
| t4    | 3.000    |         |         |         |

Regressions:

|       | Estimate | Std.Err | z-value | P(>|z|) |
|-------|----------|---------|---------|---------|
| i ~   | 0.608    | 0.060   | 10.134  | 0.000   |
| s ~   | 0.604    | 0.064   | 9.412   | 0.000   |

Covariances:

|       | Estimate | Std.Err | z-value | P(>|z|) |
|-------|----------|---------|---------|---------|
| i ~   | 0.750    | 0.050   | 15.047  | 0.000   |

Intercepts:

|       | Estimate | Std.Err | z-value | P(>|z|) |
|-------|----------|---------|---------|---------|
| t1    | 0.000    |         |         |         |
| t2    | 0.000    |         |         |         |
| t3    | 0.000    |         |         |         |
| t4    | 0.000    |         |         |         |
| i     | 0.580    | 0.062   | 9.368   | 0.000   |
| s     | 0.958    | 0.029   | 32.552  | 0.000   |

Variances:

|       | Estimate | Std.Err | z-value | P(>|z|) |
|-------|----------|---------|---------|---------|
| t1    | 0.580    | 0.080   | 7.230   | 0.000   |
| t2    | 0.590    | 0.054   | 10.969  | 0.000   |
| t3    | 0.481    | 0.055   | 8.745   | 0.000   |
| t4    | 0.535    | 0.098   | 5.466   | 0.000   |
| c1    | 0.580    | 0.080   | 7.230   | 0.000   |
| c2    | 0.590    | 0.054   | 10.969  | 0.000   |
| c3    | 0.481    | 0.055   | 8.745   | 0.000   |
| c4    | 0.535    | 0.098   | 5.466   | 0.000   |
State Trait Auto Regressive Structure

The `sim()` function

- **fx**: The structure of the x factors (defaults to 1 factor with loadings of .8, .7, .6)
- **Phi**: Inter factor correlation matrix
- **fy**: Structure of the y factors (defaults to fx) item[fy]
- **alpha**: The autoregressive correlation over time
- **lambda**: The stability of the traits over time
- **mu**: The means structure
- **n**: Number of simulated cases item [raw] If TRUE and \( n > 0 \), report the data
A (non)simplex factor structure, $\alpha = .0\lambda = 0$

Simulate an uncorrelated factor structure

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
<th>V7</th>
<th>V8</th>
<th>V9</th>
<th>V10</th>
<th>V12</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>1</td>
<td>0.56</td>
<td>0.48</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V2</td>
<td>0.56</td>
<td>1</td>
<td>0.42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V3</td>
<td>0.48</td>
<td>0.42</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.56</td>
<td>0.48</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.56</td>
<td>1</td>
<td>0.42</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.48</td>
<td>0.42</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.56</td>
<td>0.48</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.56</td>
<td>1</td>
<td>0.42</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.48</td>
<td>0.42</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.56</td>
</tr>
<tr>
<td>V12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.56</td>
<td>1</td>
</tr>
</tbody>
</table>

Diagram with factor loadings and correlations.
An autoregressive (simplex) factor structure, $\alpha = .8 \lambda = 0$
A stable factor structure, $\alpha = 0.0 \lambda = 0.5$

Simulate a simplex factor structure with lambda = .5

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
<th>V5</th>
<th>V6</th>
<th>V7</th>
<th>V8</th>
<th>V9</th>
<th>V10</th>
<th>V12</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>1.00</td>
<td>0.56</td>
<td>0.48</td>
<td>0.16</td>
<td>0.14</td>
<td>0.12</td>
<td>0.16</td>
<td>0.14</td>
<td>0.12</td>
<td>0.16</td>
<td>0.12</td>
</tr>
<tr>
<td>V2</td>
<td>0.56</td>
<td>1.00</td>
<td>0.42</td>
<td>0.14</td>
<td>0.12</td>
<td>0.10</td>
<td>0.14</td>
<td>0.12</td>
<td>0.10</td>
<td>0.14</td>
<td>0.10</td>
</tr>
<tr>
<td>V3</td>
<td>0.48</td>
<td>0.42</td>
<td>1.00</td>
<td>0.12</td>
<td>0.10</td>
<td>0.09</td>
<td>0.12</td>
<td>0.10</td>
<td>0.09</td>
<td>0.12</td>
<td>0.09</td>
</tr>
<tr>
<td>V4</td>
<td>0.16</td>
<td>0.14</td>
<td>0.12</td>
<td>1.00</td>
<td>0.56</td>
<td>0.48</td>
<td>0.16</td>
<td>0.14</td>
<td>0.12</td>
<td>0.16</td>
<td>0.14</td>
</tr>
<tr>
<td>V5</td>
<td>0.14</td>
<td>0.12</td>
<td>0.10</td>
<td>0.56</td>
<td>1.00</td>
<td>0.42</td>
<td>0.14</td>
<td>0.12</td>
<td>0.10</td>
<td>0.14</td>
<td>0.12</td>
</tr>
<tr>
<td>V6</td>
<td>0.12</td>
<td>0.10</td>
<td>0.09</td>
<td>0.48</td>
<td>0.42</td>
<td>1.00</td>
<td>0.12</td>
<td>0.10</td>
<td>0.09</td>
<td>0.12</td>
<td>0.09</td>
</tr>
<tr>
<td>V7</td>
<td>0.16</td>
<td>0.14</td>
<td>0.12</td>
<td>0.16</td>
<td>0.14</td>
<td>0.12</td>
<td>1.00</td>
<td>0.56</td>
<td>0.48</td>
<td>0.16</td>
<td>0.14</td>
</tr>
<tr>
<td>V8</td>
<td>0.14</td>
<td>0.12</td>
<td>0.10</td>
<td>0.14</td>
<td>0.12</td>
<td>0.10</td>
<td>0.56</td>
<td>1.00</td>
<td>0.42</td>
<td>0.14</td>
<td>0.12</td>
</tr>
<tr>
<td>V9</td>
<td>0.12</td>
<td>0.10</td>
<td>0.09</td>
<td>0.12</td>
<td>0.10</td>
<td>0.09</td>
<td>0.48</td>
<td>0.42</td>
<td>1.00</td>
<td>0.12</td>
<td>0.09</td>
</tr>
<tr>
<td>V10</td>
<td>0.16</td>
<td>0.14</td>
<td>0.12</td>
<td>0.16</td>
<td>0.14</td>
<td>0.12</td>
<td>0.16</td>
<td>0.14</td>
<td>0.12</td>
<td>1.00</td>
<td>0.56</td>
</tr>
<tr>
<td>V12</td>
<td>0.14</td>
<td>0.12</td>
<td>0.10</td>
<td>0.14</td>
<td>0.12</td>
<td>0.10</td>
<td>0.14</td>
<td>0.12</td>
<td>0.10</td>
<td>0.56</td>
<td>1.00</td>
</tr>
</tbody>
</table>
A simplex factor structure, \( \alpha = .5 \lambda = 5 \)
Testing various STAR models

```
stars <- sim(alpha=.5,lambda=.5)
starsMod <-
F1 =~ a*V1 + b*V2 +c * V3
F2 =~ a*V4 + b* V5 + c * V6
F3 =~ a * V7 + b* V8 + c* V9
F4 =~ a * V10 + b*V11 + c * V12

fitstars <- cfa(starsMod,sample.cov=stars$model,sample.nobs=1000,std.lv=TRUE)
summary(fitstars)
```

Latent Variables:

|   | Estimate | Std.Err | z-value | P(>|z|) |
|---|----------|---------|---------|---------|
| F1 |          |         |         |         |
|   | V1 (a)   | 0.800   | 0.018   | 45.282  | 0.000   |
|   | V2 (b)   | 0.700   | 0.017   | 40.199  | 0.000   |
|   | V3 (c)   | 0.600   | 0.017   | 34.558  | 0.000   |

Covariances:

|   | Estimate | Std.Err | z-value | P(>|z|) |
|---|----------|---------|---------|---------|
| F1 |          |         |         |         |
|   | F2       | 0.750   | 0.025   | 29.605  | 0.000   |
| F2 | F3       | 0.500   | 0.034   | 14.911  | 0.000   |
| F2 | F4       | 0.375   | 0.037   | 10.160  | 0.000   |
| F3 | F4       | 0.750   | 0.025   | 29.616  | 0.000   |
| F3 |          |         |         |         |
|   | F4       | 0.500   | 0.034   | 14.911  | 0.000   |
```
Modifying a model

- There are many reasons a model does not fit.
  - In particular, some paths may be badly fit (usually because they were ignored).
  - How much will a parameter change (Expected Parameter Change) if a parameter is adjusted.
Modification indices for the HS problem

```r
fit <- cfa(HS.model, data = HolzingerSwineford1939)
mi <- modindices(fit)
mi[mi$op == "~", ]
```

<table>
<thead>
<tr>
<th>lhs</th>
<th>op</th>
<th>rhs</th>
<th>mi</th>
<th>epc</th>
<th>sepc.lv</th>
<th>sepc.all</th>
</tr>
</thead>
<tbody>
<tr>
<td>visual</td>
<td>=~</td>
<td>x1</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>visual</td>
<td>=~</td>
<td>x2</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>visual</td>
<td>=~</td>
<td>x3</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>visual</td>
<td>=~</td>
<td>x4</td>
<td>1.211</td>
<td>0.077</td>
<td>0.069</td>
<td>0.059</td>
</tr>
<tr>
<td>visual</td>
<td>=~</td>
<td>x5</td>
<td>7.441</td>
<td>-0.210</td>
<td>-0.189</td>
<td>-0.147</td>
</tr>
<tr>
<td>visual</td>
<td>=~</td>
<td>x6</td>
<td>2.843</td>
<td>0.111</td>
<td>0.100</td>
<td>0.092</td>
</tr>
<tr>
<td>visual</td>
<td>=~</td>
<td>x7</td>
<td>18.631</td>
<td>-0.422</td>
<td>-0.380</td>
<td>-0.349</td>
</tr>
<tr>
<td>visual</td>
<td>=~</td>
<td>x8</td>
<td>4.295</td>
<td>-0.210</td>
<td>-0.189</td>
<td>-0.187</td>
</tr>
<tr>
<td>visual</td>
<td>=~</td>
<td>x9</td>
<td>36.411</td>
<td>0.577</td>
<td>0.519</td>
<td>0.515</td>
</tr>
<tr>
<td>textual</td>
<td>=~</td>
<td>x1</td>
<td>8.903</td>
<td>0.350</td>
<td>0.347</td>
<td>0.297</td>
</tr>
<tr>
<td>textual</td>
<td>=~</td>
<td>x2</td>
<td>0.017</td>
<td>-0.011</td>
<td>-0.011</td>
<td>-0.010</td>
</tr>
<tr>
<td>textual</td>
<td>=~</td>
<td>x3</td>
<td>9.151</td>
<td>-0.272</td>
<td>-0.269</td>
<td>-0.238</td>
</tr>
<tr>
<td>textual</td>
<td>=~</td>
<td>x4</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x7</td>
<td>0.098</td>
<td>-0.021</td>
<td>-0.021</td>
<td>-0.019</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x8</td>
<td>3.359</td>
<td>-0.121</td>
<td>-0.120</td>
<td>-0.118</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x9</td>
<td>4.796</td>
<td>0.138</td>
<td>0.137</td>
<td>0.136</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x1</td>
<td>0.014</td>
<td>0.024</td>
<td>0.015</td>
<td>0.013</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x2</td>
<td>1.580</td>
<td>-0.198</td>
<td>-0.123</td>
<td>-0.105</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x3</td>
<td>0.716</td>
<td>0.136</td>
<td>0.084</td>
<td>0.075</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x4</td>
<td>0.003</td>
<td>-0.005</td>
<td>-0.003</td>
<td>-0.003</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x5</td>
<td>0.201</td>
<td>-0.044</td>
<td>-0.027</td>
<td>-0.021</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x6</td>
<td>0.273</td>
<td>0.044</td>
<td>0.027</td>
<td>0.025</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x7</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x8</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>speed</td>
<td>=~</td>
<td>x9</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
The fitted model

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

```r
$ cov
   x1  x2  x3  x4  x5  x6  x7  x8  x9
x1  1.358
x2  0.448 1.382
x3  0.590 0.327 1.275
x4  0.408 0.226 0.298 1.351
x5  0.454 0.252 0.331 1.090 1.660
x6  0.378 0.209 0.276 0.907 1.010 1.196
x7  0.262 0.145 0.191 0.173 0.193 0.161 1.183
x8  0.309 0.171 0.226 0.205 0.228 0.190 0.453 1.022
x9  0.284 0.157 0.207 0.188 0.209 0.174 0.415 0.490 1.015

$ mean
   x1  x2  x3  x4  x5  x6  x7  x8  x9
 0 0 0 0 0 0 0 0 0 0
```
Examine the raw residuals

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

$cov

<table>
<thead>
<tr>
<th></th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x7</th>
<th>x8</th>
<th>x9</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>-0.041</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td>-0.010</td>
<td>0.124</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td>0.097</td>
<td>-0.017</td>
<td>-0.090</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x5</td>
<td>-0.014</td>
<td>-0.040</td>
<td>-0.219</td>
<td>0.008</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x6</td>
<td>0.077</td>
<td>0.038</td>
<td>-0.032</td>
<td>-0.012</td>
<td>0.005</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x7</td>
<td>-0.177</td>
<td>-0.242</td>
<td>-0.103</td>
<td>0.046</td>
<td>-0.050</td>
<td>-0.017</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x8</td>
<td>-0.046</td>
<td>-0.062</td>
<td>-0.013</td>
<td>-0.079</td>
<td>-0.047</td>
<td>-0.024</td>
<td>0.082</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>x9</td>
<td>0.175</td>
<td>0.087</td>
<td>0.167</td>
<td>0.056</td>
<td>0.086</td>
<td>0.062</td>
<td>-0.042</td>
<td>-0.032</td>
<td>0.000</td>
</tr>
</tbody>
</table>

$mean

<table>
<thead>
<tr>
<th></th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x7</th>
<th>x8</th>
<th>x9</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>x2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

---

Examine the raw residuals

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

$cov

<table>
<thead>
<tr>
<th></th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x7</th>
<th>x8</th>
<th>x9</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>-0.041</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td>-0.010</td>
<td>0.124</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td>0.097</td>
<td>-0.017</td>
<td>-0.090</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x5</td>
<td>-0.014</td>
<td>-0.040</td>
<td>-0.219</td>
<td>0.008</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x6</td>
<td>0.077</td>
<td>0.038</td>
<td>-0.032</td>
<td>-0.012</td>
<td>0.005</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x7</td>
<td>-0.177</td>
<td>-0.242</td>
<td>-0.103</td>
<td>0.046</td>
<td>-0.050</td>
<td>-0.017</td>
<td>0.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x8</td>
<td>-0.046</td>
<td>-0.062</td>
<td>-0.013</td>
<td>-0.079</td>
<td>-0.047</td>
<td>-0.024</td>
<td>0.082</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>x9</td>
<td>0.175</td>
<td>0.087</td>
<td>0.167</td>
<td>0.056</td>
<td>0.086</td>
<td>0.062</td>
<td>-0.042</td>
<td>-0.032</td>
<td>0.000</td>
</tr>
</tbody>
</table>

$mean

<table>
<thead>
<tr>
<th></th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x7</th>
<th>x8</th>
<th>x9</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>x2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```
Examine the standardized residuals

```r
fit <- cfa(HS.model, data = HolzingerSwineford1939)
resid(fit, type = "standardized")
```

$cov

<table>
<thead>
<tr>
<th></th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x7</th>
<th>x8</th>
<th>x9</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x2</td>
<td>-2.196</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x3</td>
<td>-1.199</td>
<td>2.692</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x4</td>
<td>2.465</td>
<td>-0.283</td>
<td>-1.948</td>
<td>NA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x5</td>
<td>-0.362</td>
<td>-0.610</td>
<td>-4.443</td>
<td>0.856</td>
<td>NA</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x6</td>
<td>2.032</td>
<td>0.661</td>
<td>-0.701</td>
<td>NA</td>
<td>0.633</td>
<td>0.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x7</td>
<td>-3.787</td>
<td>-3.800</td>
<td>-1.882</td>
<td>0.839</td>
<td>-0.837</td>
<td>-0.321</td>
<td>NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x8</td>
<td>-1.456</td>
<td>-1.137</td>
<td>-0.305</td>
<td>-2.049</td>
<td>-1.100</td>
<td>-0.635</td>
<td>3.804</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>x9</td>
<td>4.062</td>
<td>1.517</td>
<td>3.328</td>
<td>1.237</td>
<td>1.723</td>
<td>1.436</td>
<td>-2.771</td>
<td>NA</td>
<td>0.000</td>
</tr>
</tbody>
</table>
Many more examples

- The MPlus manual has data sets that may be explored with lavaan code.
  - Chapter 3: Regression and Path Analysis
  - Chapter 5: Confirmatory factor analysis and structural equation modeling
  - Chapter 6: Growth modeling
- LISREL manual also has suitable examples
SEM-AMOS wiki a warning

- It may seem odd to begin with a warning, but the popular misuse and misinterpretation of Structural Equation Modeling is so widespread that users of this wiki should be aware of some of the issues involved before they begin. While this warning is overly brief, you can follow-up these issues and more in the Further Reading section of this article.

- A number of these issues also apply to Confirmatory Factor Analysis. While Structural Equation Modeling has been popular in recent years to test the degree of fit between a proposed structural model and the emergent structure of the data, the perceived superiority of the technique is waning.

- Aside from the fact that the results of Structural Equation Modeling are often poorly reported, the conclusions drawn do not typically grasp the limitations of the technique.
The most obvious, and some ways the most critical issue is that of incorrectly inferring a particular configuration of causal relationships from correlational data. This mistake can be illustrated with the simplest of all structural examples – that of 2 variables (variable A and B). If we ignore the additional complexity of latent structure, the number of possible causal structures is 4. Clearly, the number of possible models grows exponentially as the number of variables grows. In this example, the 4 possible causal models in this example are:

- A causes B;
- B causes A;
- A and B cause each other (a recursive model);
- finally, A and B are unrelated.
If A and B are indeed significantly correlated, it is likely that the first 3 models will be supported by significant fit statistics. If this is the case, what has been proven?

Which of the 3 supported models is the correct model? What makes matters worse is that we have not even conclusively ruled out the last model. It is still possible that the correlation between A and B was spurious.

To reinforce a maxim that most people know, but fail to apply to Structural Equation Modeling – you can not determine causation from correlation.

Yet in most cases, researchers only test one or two models out of all the myriad of potential models, poorly report their results, then proclaim confirmation of their model (implying the exclusion of all other possible models).
So what is the value of Structural Equation Modeling?

If large correlational datasets are already available, and a large range of plausible models are assessed, the results can be valuable in conceiving an experimental study that can test the proposed causal relationships.


