Multivariate Statistics with R

Confirmatory Factor Analysis - Lecture 5

Tom Booth, tom.booth@ed.ac.uk

Friday 25th March 2016

Administration

- Today is the last lecture which has an associated lab.
- The content of today's lab is a continuation/extension of last week.
- All the answers/reporting examples are now available on LEARN to assist if you choose to do the data reduction part of the assessment.
- Speaking of which. . . .

Coursework (1)

- Posted: 24/03/2016 (yesterday)
- Due: 21/04/2016 - 12noon
- Format
  - Much the same as univariate, produce a report to answer **either** the mixed models or data reduction question.
  - You will also need to submit your code.

Coursework (2)

- The report:
  - A maximum of 6 pages (2 of singled sided text, 4 of tables and figures)
  - Times New Roman Font, size 12.
  - Standard page margins (1inch or 2.54cm).
  - Follow APA guidelines for any referencing and tables.
  - Tables and figures **do** count in page allowance.
  - References **do not** count in page allowance.
    - Note we do not expect lots of references!

Coursework (3)

- Submission:
  - Submit your R-code and report electronically to the appropriate hand-in by due date and time.
  - Submit a printed copy of the report to the PG Office by due date and time.
  - Do **not** need to print your R-code.
  - Make sure your exam number (e.g. B000001) appears in the name of both files and also in the header of the report.

Coursework (4)

- The code:
  - **Your code must run and produce the results in your report**
  - This includes reading in data and opening libraries (start of the document).
  - If your code does not run there will be a 10% deduction.

Coursework (5)

- Making sure your code runs:
  1. Save your script and close down RStudio.
  2. Re-open, and type `rm(list=ls())`. This will completely empty your R environment.
  3. Run your script line by line. a. This will mean the first line should read in data. b. Then the code should follow logically your report.
  4. If you hit an error, fix the code, and loop back to (2).
  5. As you move through, delete any script which is not part of the logical sequence in the report.
  6. Repeat until you have a tidy script which runs all the way through.
  7. When you think you are finished, loop to (1) and check 1 more time!

Questions on coursework?

Today's Lecture

- Confirmatory Factor Analysis
  - Model Specification
  - Model Identification
  - Model Estimation
  - Model Evaluation
  - Model Modification
- Example measurement models
  - First order orthogonal model
  - First order oblique model
  - Second order model
  - Bi-factor model

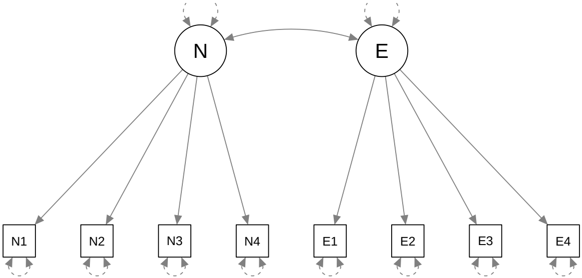
CFA: The specifics

Model Specification: Overview

- In simplistic terms, our specification is simply the model we wish to test.
- Away from any given software, we can think about our model as a diagram, as matrices, as equations, or as all of the above.
- Let's consider the 8 items from the first lecture again.

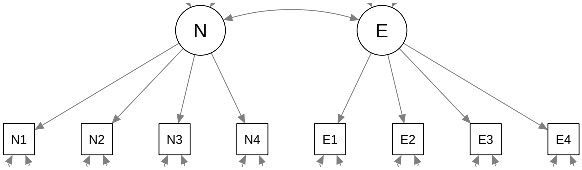
Model Specification: Diagram

```
library(lavaan)
library(semPlot)
model = '
N =~ N1 + N2 + N3 + N4
E =~ E1 + E2 + E3 + E4
N ~~ E
'
semPaths(model)
```



Model Specification: Matrices (1)

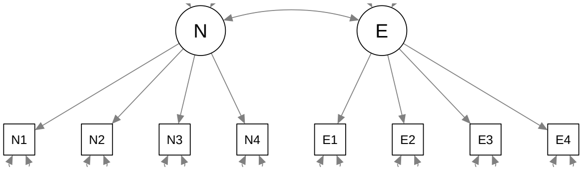
```
semPaths(model)
```



$$\begin{pmatrix} \lambda_{11} & 0 \\ \lambda_{21} & 0 \\ \lambda_{31} & 0 \\ \lambda_{41} & 0 \\ 0 & \lambda_{52} \\ 0 & \lambda_{62} \\ 0 & \lambda_{72} \\ 0 & \lambda_{82} \end{pmatrix}$$

Model Specification: Matrices (2)

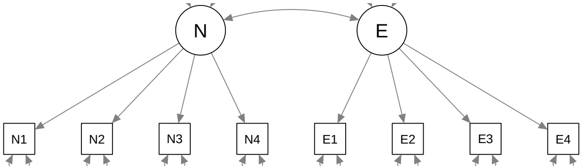
```r
semPaths(model)
```



$$\begin{pmatrix} \varepsilon_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \varepsilon_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \varepsilon_{33} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \varepsilon_{44} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \varepsilon_{55} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \varepsilon_{66} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \varepsilon_{77} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \varepsilon_{88} \end{pmatrix}$$

Model Specification: Matrices (3)

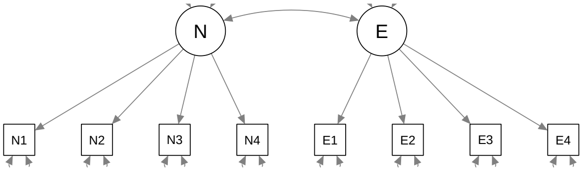
```
semPaths(model)
```



$$\begin{pmatrix} \phi_{11} & 0 \\ \phi_{21} & \phi_{22} \end{pmatrix}$$

Model Specification: Equations

```
semPaths(model)
```



- $\text{N1} = \lambda_{11}{*}\text{N} + \varepsilon_{11}$
- $\text{N2} = \lambda_{21}{*}\text{N} + \varepsilon_{22}$
- $\text{N3} = \lambda_{31}{*}\text{N} + \varepsilon_{33}$
- $\text{N4} = \lambda_{41}{*}\text{N} + \varepsilon_{44}$
- $\text{E1} = \lambda_{52}{*}\text{E} + \varepsilon_{55}$
- $\text{E2} = \lambda_{62}{*}\text{E} + \varepsilon_{66}$
- $\text{E3} = \lambda_{72}{*}\text{E} + \varepsilon_{77}$
- $\text{E4} = \lambda_{82}{*}\text{E} + \varepsilon_{88}$

Model Specification: Scaling the latent variable

- Key step in specification is to provide a scale for the latent variables to ease interpretation.
- Two common ways:
    1. Fix the variance of the latent variable to 1.
    2. Fix an emitted path from the latent variable to an observed variable to 1. This is generally used to identify and scale residuals but can be used in the model more generally.
- 1. conforms to the standard in EFA, and also standardizes the latent variable.
    - But it removes the variance in the latent variable from being freely estimated.
- So when the variance is of interest (2) may be preferable.

Model Specification

- All models are wrong, but some are useful.
- Model specification, in its most basic sense, is defining the parameters of interest for your research question.

  $$Parameters = Hypotheses$$

- With each arrow(path) in a CFA model/diagram, the researcher is making a statement (hypothesis) about the associations between variables.
- Perhaps more importantly, for each arrow(path) **not** included, the researcher is also making a statement or hypothesis (Bollen & Pearl, 2012).

Model Identification: Overview

- Model identification is concerned with the number of known to be identified parameters and unknown parameters.
  - Known to be identified parameters are the variances and covariances of the measured variables.
  - Unknown parameters are those you are looking to estimate.
- A necessary condition for identification is that we have more known to identified than unknown parameters.
  - **Note:** This is a *necessary* not *sufficient* condition.

Model Identification: Overview

- There are three different levels of identification in a model.
  - **Under-identified** models: have < 0 degrees of freedom.
  - **Just Identified** models: have 0 degrees of freedom.
  - **Over-Identified** models: have > 0 degrees of freedom.

Model Identification: An example

- Chou & Bentler (1995) provide a neat example from simultaneous equations:
  - Eq.1: $x + y = 5$
  - Eq.2: $2x + y = 8$
  - Eq.3: $x + 2y = 9$
- Eq.1 is *under-identified*.
- Eq.1 & 2 are *just identified*.
- Eq.1, 2 & 3 are *over identified*.

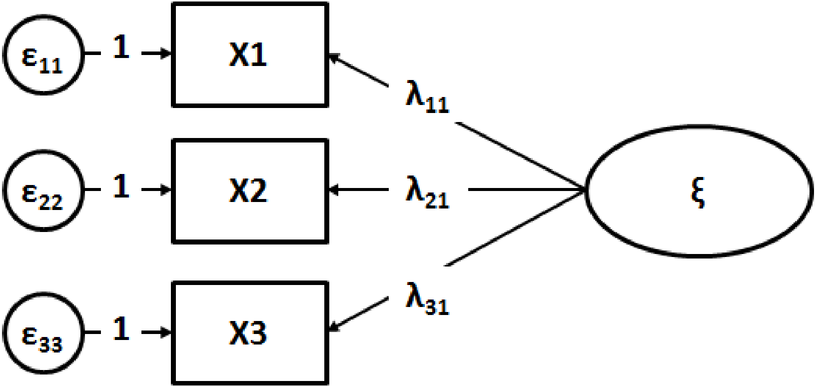
Model Identification: t-rule

- As noted above, identification concerns the degrees of freedom in our model.
- We can calculate the known to be identified parameters by:

$$\frac{(k+1)\,(k)}{2}$$

- where k is the number of observed variables.
- If we subtract the unknown parameters from the quantity above, we get model degrees of freedom.

Model Identification: 3 indicator rule (1)



Model Identification: 3 indicator rule (2)

- For measurement models, the three indicator rule is a simple example of the t-rule in operation.

$$\frac{(k+1)(k)}{2} = \frac{(3+1)3}{2} = \frac{4*3}{2} = \frac{12}{2} = 6$$

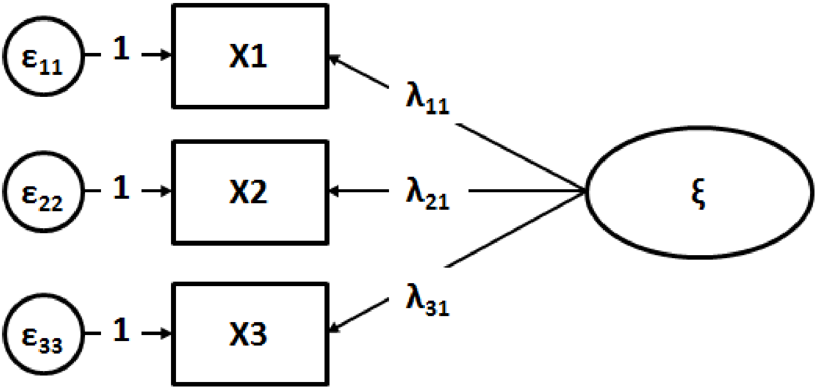
- So there are 6 known to be identified parameters

Model Identification: 3 indicator rule (3)

- This makes sense if we think about what this model is based on. . . covariances (correlations)

	x1	**x2**	**x3**
x1	var(X1)		
x2	cov(x2x1)	var(X2)	
x3	cov(x3x1)	cov(x3x2)	var(X3)

Model Identification: 3 indicator rule (4): Unknowns

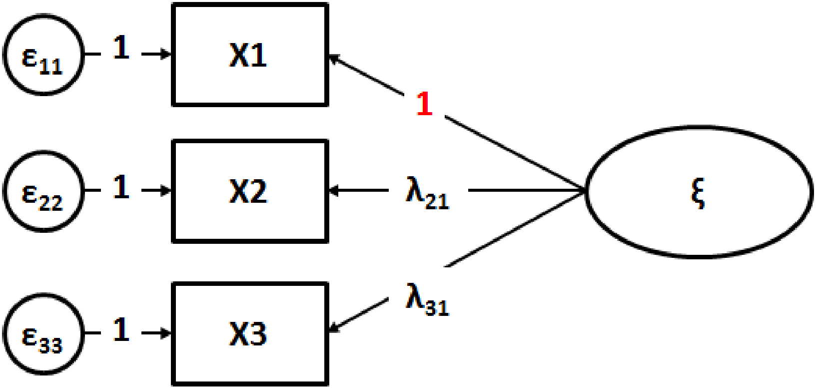


- We have three factor loadings (λ_{11}, λ_{21} & λ_{31})
- Three residual variances (ε_{11}, ε_{22} & ε_{33})
- And 1 factor variance (ϕ_{11})

Model Identification: 3 indicator rule (5)

- So, our model degrees of freedom are 6-7 = -1 degrees of freedom
  - Our model is under-identified.
- **But** remember our scaling constraint...

Model Identification: 3 indicator rule (6)



Model Identification: 3 indicator rule (7)

- When we add this constraint, we only have 6 unknown parameters.
  - 6-6 $=$ 0 degrees of freedom
  - Our model is just identified
- Thus 3 indicators are a minimum for a latent factor.
  - However, as we noted above, zero degrees of freedom means we can not test our model.
  - Just identified models fit perfectly.

Model Identification: 2 indicator rule

- On occasions, you may see models which include factors with only 2 indicators.
- Conceptually one could argue this is problematic.
- If it is the only factor in the model, then statistically it is also problematic.
- However, if there are two correlated factors, the model will be identified if. . .
  1. Each variable only loads on 1 factor.
  2. There are at least 2 variables per latent factor.
  3. There is at least 1 non-zero off-diagonal factor correlation in each row of the matrix.
  4. The error terms are uncorrelated.

Model Identification: In practice

- Always make sure you have a minimum of three indicators per latent variable.
- Always make sure you have set your scaling constraints.
- Within the broader context of SEM, there are additional identification issues with respect to included paths.
    - e.g. recursive paths.

Model Estimation: Overview

- So we have specified our model and checked that it is identified.
- Now we need to provide estimates of the unknown parameters in our model.
- There are many different estimators with different properties, but a good general option is maximum likelihood (ML).
  - Goal of ML is to identify population parameter values with the highest probability of producing the particular sample of data.

Model Estimation: Maximum Likelihood

- Why is maximum likelihood a preferred estimator?
- **Asymptotically unbiased, consistent and efficient**.
  - As samples increase to infinity (asymptotically), ML estimates are unbiased, provide best estimates of the unknown population parameters (efficient) and will converge on the true population estimate (consistent).
- **Scale invariant**.
  - You can rescale your original variables to any linear transformation and this will not impact the estimate of the log-likelihood.
- **Scale free**.
  - Calculations based on a correlation or covariance matrix will yield the same log-likelihood.

Model Estimation: Alternatives to ML

- Asymptotically Distribution Free (ADF).
  - Generally robust to moderate violations of multivariate normality assumptions.
- Weighted Least Squares (WLS).
  - Often recommended for categorical data with means and variance adjustments.
  - Computationally intensive so some prefer to use the related diagonally weight least squares methods (DWLS).
- Full Information Maximum Likelihood (FIML).
  - ML is in the presence of missing data.

Model Evaluation: Overview

- All models are wrong, but some are more wrong than others.
  - But how do we assess how wrong our model is?
- In CFA and SEM, a first step in evaluating our proposed models is to assess model fit.
  - There are a huge number of fit measures, all of which have the general aim of evaluating how well our model fits our data.
- Once we have assessed the fit of our model, we can consider the significance and effect sizes of our parameter estimates.

Model Evaluation: Chi-square (1)

- The χ^2 test was one of the first measures of fit and is used in the equations for many other fit indices.

$$\chi^2 = (N - 1)F_{min}$$

  - Where $(N - 1) =$ sample size - 1
  - and $F_{MIN} =$ is minimum value of a given fit function which seeks to minimize the difference between the observed sample covariance matrix and the model implied covariance matrix.
  - In the context of ML, this would be the estimates of the model parameters which are most likely given the data.
- In large samples (asymptotically) $(N - 1)F_{min}$ is distributed as a χ^2 distribution with df_M

Model Evaluation: Chi-square (2)

- Though regularly reported, the χ^2 is often not a practically useful fit measure as it is dependent on sample size.
  - This is easily seen - $(N-1)F_{min}$
  - So as N increases, the minimum of the fit function is multiplied a bigger value.
  - However, the df_M will not change and as such we end up comparing a larger value against for the same degrees of freedom.
  - This means in large samples, χ^2 will nearly always be significant, even when F_{MIN} is small.

Model Evaluation: Chi-square (3)

- A slight variant is to use the ratio of model χ^2 to degrees of freedom.

$$\frac{\chi^2}{df}$$

- Primary issue here is there are no well established guidelines for this value.

Model Evaluation: Tucker-Lewis Index (TLI)

$$TLI = \frac{(\chi^2/df)_{NM} - (\chi^2/df)_M}{(\chi^2/df)_{NM} - 1}$$

- Sometimes also referred to as the Non-normed fit index (NNFI)
- Values closer to 1.0 = better fit.
- TLI and CFI (see next slide) values are dependent on observed correlations.
  - The lower the correlations, the closer the actual model is to a null model.

Model Evaluation: Comparative Fit Index (CFI)

$$CFI = 1 - \frac{(\chi^2 - df)_M}{(\chi^2 - df)_{NM}}$$

- Compares fit of our model to fit of a null model
  - Null model = no correlation
- Values closer to 1.0 = better fit.
- Frequently reported and "performs well" in siumlations.
- Some argue against the null model as comparison.

Model Evaluation: Root Mean-Square Error of Approximation (RMSEA)

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

- Residual based method that considers the discrepancy between the observed and model implied covariances.
- Often favoured by researchers.
- Values closer to 0 are good.
- Added benefit of being able to compute confidence intervals.
  - See our fit in `psych`

Model Evaluation: Standardized Root Mean-Square Residual (SRMR)

$$SRMR = \sqrt{\left\{2\sum_{i=1}^{p}\sum_{i=1}^{i}[(s_{ij} - \hat{\sigma}_{ij})/(s_{ii}s_{jj})]\right\}/p(p+1)}$$

- Scary looking equation!
  - Take the differences between the sample and model implied covariance matrices (the residuals)
  - Square and sum them
  - And standardize
- Values closer to 0 are good.
- Will tend to be small in large samples and in models with lots of parameters.

Model Evaluation: Information Criteria

$$AIC = -2ln(L) + 2k$$

$$BIC = -2ln(L) + log(N)k$$

- Where ln(L) = log-likelihood of the model, and k = no. of estimated parameters.
  - Values of both the AIC and BIC are not informative in their own right, but are informative when comparing alternative models.
- Important observation is that the parsimony penalty for the BIC is bigger.

Model Evaluation: Fit Indices Summary

Fit Index	Suggested Cut-offs	Parsimony Correction	Compare non-nested models?
X^2	$p < 0.05$	No	No
X^2/df ratio	3:1 (ish!)	X^2/df ratio	No
CFI	>0.90 to 0.95	1 per estimated param	No
TLI	>0.90 to 0.95	X^2/df ratio	No
RMSEA	<0.05 to 0.08	X^2/df ratio	No
SRMR	<0.05 to 0.08	None.	No
AIC	Smaller the better	2*k	Yes
BIC	Smaller the better	Log(N)*k	Yes
saBIC	Smaller the better	Log((N+2)/24)*k	Yes

- See Schermelleh-Engel et al. (2003) and Schreiber et al. (2006) for reviews of fit criteria.

Model Evaluation: Parameters

- And after this we can consider our parameters of interest.
- Our model should show reasonable levels of fit prior to interpretation.
- Once our model fits, we can consider:
    1. Magnitude of factor loadings
    2. Magnitude of factor correlations
    3. Significance of the estimates
    4. Reliability of the factors (see reading list)

Model Modification (1)

- But what if our model does not fit?
- There are various ways to explore the location of model misfit.
- The most common is to just modification indices.
- Recall:
  - All model parameters not estimated are fixed to 0.
  - Thus, deviation from 0 will lead to misfit.
  - We can calculate this difference in fit if this parameter is freely estimated.
  - This is an MI.

Model Modification (2)

- But if we modify, are we strictly working with a confirmatory model?
- CFA/SEM when used to test specific models is confirmatory.
- But model modification is common especially when a model does not meet conventions for model fit.
  - Modification is often based on quantitative model derived measures such as modification indices.
  - After modification the model is no longer confirmatory, it is exploratory.

Model Modification (3)

- This poses problems:
  - Are the additional parameters capitalizing on chance (sample specific)?
  - Are the additional parameters theoretically justified?
- Modified models require replication in a new sample.
  - Rarely seen in published research.

Example measurement models

Examples and lavaan code

- Now we have spoken theoretically about the stages in CFA, lets look at some models.
- We will use four different measurement models to emphasize the `lavaan` code for CFA.
- We will also discuss how different model modifications have different theoretical interpretations.

Examples: Cognitive data

- To demonstrate the different models we will use a subset of 10 of the 14 cognitive tests you analyzed in lab 3.
- We can hypothesize them to measure the following:
  - **Speed**: Symbol Search, Digit Symbol, Choice RT
  - **Verbal Memory**: Logical Memory, Verbal Paired Associates, NART
  - **Processing**: Spatial Span, Letter-Number Sequencing, Digit Span Backward, Block Design

Example: Data

```r
cog <- read.csv("m:/Lab3_data1.csv", header=T)
cog <- cog[,c(4,5,7,1,3,13,2,11,12,14)]
head(cog)[1:4]
```

```
##         SS       DS       CRT       LM
## 1 23.13305 40.70213 0.7741314 42.58540
## 2 30.99057 71.73216 0.6290338 61.51030
## 3 25.31002 53.32898 0.6600838 93.70052
## 4 29.34980 47.79678 0.7839042 58.32237
## 5 36.49594 72.29500 0.5338873 91.38010
## 6 26.19773 63.06650 0.7293803 34.07832
```

Arguments to lavaan models (1)

- Here we use the function `cfa()` within lavaan.

```
cfa(model=, data=, std.lv=, std.ov=, missing=,
    estimator=)
```

- `model=` : Here we add the name of our model object (examples below)
- `data=` : As always we need to give our data set, where variable names match the model object
- `std.lv=` : a TRUE or FALSE statement, scale the latent variable by first loading (F) or latent variance (T)

Arguments to lavaan models (2)

```
cfa(model=, data=, std.lv=, std.ov=, missing=,
    estimator=)
```

- `std.ov=` : Standardize the observed variables - TRUE or FALSE
- `missing=` : How missing data is treated. Big advantage of CFA/SEM is use of full information maximum likelihood (FIML) with ML estimation.
- `estimator=` : Many options by ML and DWLS are likely most useful.

Specification

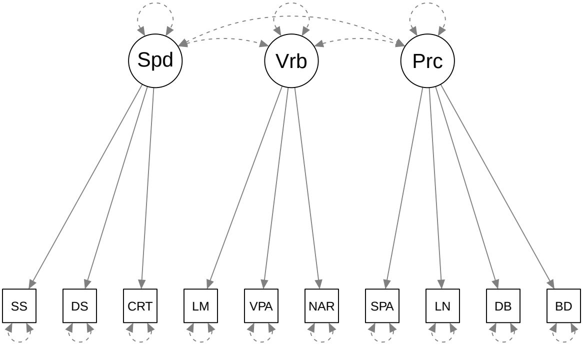
First order orthogonal model: Code

```
orthog = '
Speed =~ SS + DS + CRT
Verbal =~ LM + VPA + NART
Proc =~ SPAN + LN + DB + BD

# * used to fix paths to a value
Speed ~~ 0*Verbal
Speed ~~ 0*Proc
Verbal ~~ 0*Proc
'
```

First order orthogonal model: Diagram

```r
semPaths(orthog)
```



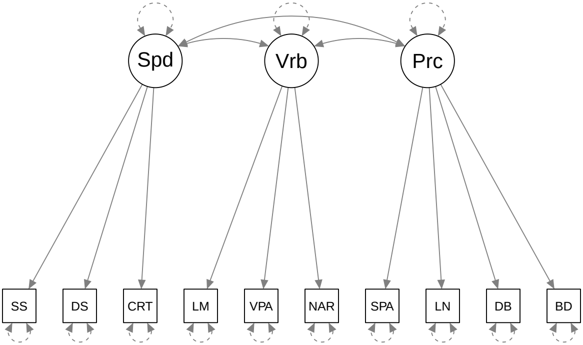
First order oblique model: Code

```
oblique = '
Speed =~ SS + DS + CRT
Verbal =~ LM + VPA + NART
Proc =~ SPAN + LN + DB + BD

Speed ~~ Verbal
Speed ~~ Proc
Verbal ~~ Proc
'
```

First order oblique model: Diagram

```r
semPaths(oblique)
```



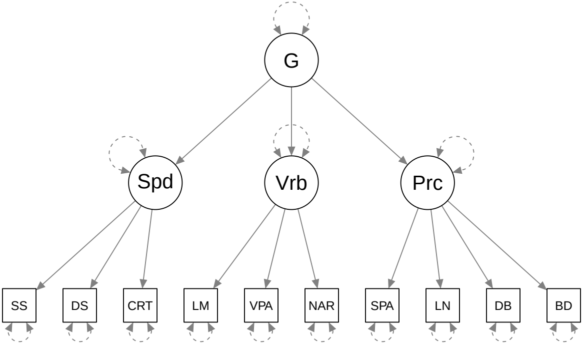
Second order model: Code

```
HO = '
Speed =~ SS + DS + CRT
Verbal =~ LM + VPA + NART
Proc =~ SPAN + LN + DB + BD

G =~ Speed + Verbal + Proc
'
```

Second order model: Diagram

```r
semPaths(HO)
```



Bi-factor model: Code

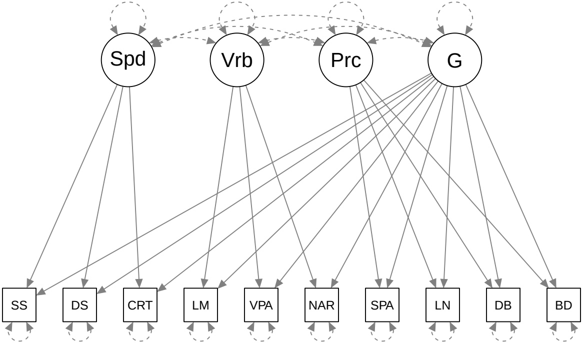
```
bifact = '
Speed =~ SS + DS + CRT
Verbal =~ LM + VPA + NART
Proc =~ SPAN + LN + DB + BD

G =~ SS + DS + CRT +
      LM + VPA + NART +
      SPAN + LN + DB + BD

G ~~ 0*Speed
G ~~ 0*Verbal
G ~~ 0*Proc
Speed ~~ 0*Verbal
Speed ~~ 0*Proc
Verbal ~~ 0*Proc
'
```

Bi-factor model: Diagram

```r
semPaths(bifact)
```



Estimation

Running the models

```
orthres <- cfa(orthog, cog, std.lv=T, std.ov=T,
             missing="fiml", estimator="ML")

obres <- cfa(oblique, cog, std.lv=T, std.ov=T,
             missing="fiml", estimator="ML")

HOres <- cfa(HO, cog, std.lv=T, std.ov=T,
             missing="fiml", estimator="ML")

bifres <- cfa(bifact, cog, std.lv=T, std.ov=T,
             missing="fiml", estimator="ML")
```

Coding short-cuts

- We actually don't need to specify factor correlations when we use `cfa()` as it defaults to the oblique model.
- We can also run an orthogonal model by adding an extra command rather then setting a new model. So;

```
oblique2 = '
Speed =~ SS + DS + CRT
Verbal =~ LM + VPA + NART
Proc =~ SPAN + LN + DB + BD
'
# Models run with this code will
# reproduce the orthogonal and
# oblique above
cfa(oblique2, cog)
cfa(oblique2, cog, orthogonal=T)
```

Evaluation

Model Fit Comparison (1)

- We can use a neat function in `semtools()` to compare model fits.

```r
compareFit(orthres, obres, HOres, bifres)
```

```
################### Nested Model Comparison #########################
                    chi df     p delta.cfi
bifres - HOres    84.39  7 <.001    0.0329
HOres - obres      0.00  0 <.001    0.0000
obres - orthres  547.58  3 <.001    0.2315

#################### Fit Indices Summaries ##########################
          chisq df pvalue   cfi   tli        aic        bic rmsea  srmr
orthres 748.433 35  .000  .697  .610  23743.155  23886.992  .151  .206
obres   200.851 32  .000  .928  .899  23201.572  23359.794  .077  .042
HOres   200.851 32  .000  .928  .899  23201.572  23359.794  .077  .042
bifres  116.463 25  .000† .961† .930† 23131.185† 23322.968† .064† .029†
```

Paremeter Estimates (1)

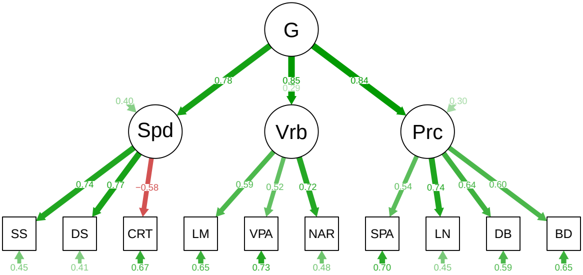
- So our bi-factor model gives us the best fit.
  - It is also the model with the most parameters so we may want to think about over-fitting.
- As such, we now want to evaluate the parameters.
- We can have a look at the full output by calling our results object `bifres`.

```
# Look at standardized parameter values
summary(bifres, standardized=T)
```

Parameter Estimates (2)

- But we can also add the parameters to our diagram.

```r
# Example using the HO model
# It is clearer!
semPaths(HOres, intercepts=F, thresholds=F,
         style="lisrel", "std")
```



Factor Saturation (1)

- When we hypothesized a single variable as the higher-order or general (bi-factor) common cause, we can estimate the general factor saturation.
  - The amount of variance in the set of observed variables which is related to the general factor.
- Though there are many alternative estimates, recent studies (McDonald, 1999; Revelle & Zinbarg, 2009; Zinbarg, Revelle, Yovel & Li, 2005) suggested the omega-hierarchical (ω_h) provides the best estimate.
- $\omega_h =$ ratio of the sum of the correlations accounted for by the general factor over the sum of the total correlations (Revelle & Wilt, 2013).

Factor Saturation (2)

```
library(semTools)
reliability(bifres)[1:5,1:3]
```

```
##              Speed    Verbal      Proc
## alpha   -0.3621378 0.6544825 0.7210962
## omega    0.2278598 0.4721120 0.4178757
## omega2   0.1765168 0.2870413 0.1866434
## omega3   0.1765168 0.2870413 0.1857940
## avevar   0.5181698 0.4306100 0.4363372
```

Modification

For information

- Our model does not need modification to fit our data.
- So lets look at our oblique model which did not fit as well and demonstrate making modifications.

```
summary(obres, modindices=T)
```

- Our biggest MI value is for the correlated residual for logical memory and verbal paired associates (MI=36.312)
- So lets add this to our model

Oblique model modification (1)

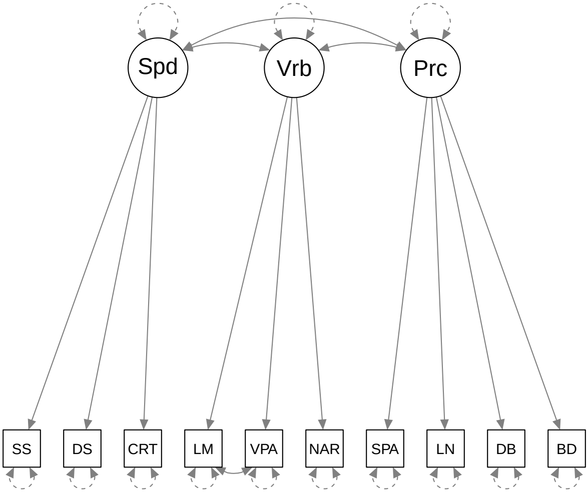
```
oblique2 = '
Speed =~ SS + DS + CRT
Verbal =~ LM + VPA + NART
Proc =~ SPAN + LN + DB + BD

Speed ~~ Verbal
Speed ~~ Proc
Verbal ~~ Proc

! modification
LM ~~ VPA
'
```

Oblique model modification (2)

```
semPaths(oblique2)
```



Oblique model modification (3)

```r
obres2 <- cfa(oblique2, cog, std.lv=T, std.ov=T,
             missing="fiml", estimator="ML")
```

```r
compareFit(obres, obres2)
```

```
################### Nested Model Comparison #########################
                  chi  df      p delta.cfi
obres2 - obres 34.89   1  <.001    0.0144

#################### Fit Indices Summaries ##########################
        chisq df pvalue   cfi   tli        aic        bic rmsea  srmr
obres  200.851 32  .000† .928  .899  23201.572  23359.794  .077  .042
obres2 165.965 31  .000† .943† .917† 23168.686† 23331.702† .070† .037†
```

Reporting CFA (1)

- When reporting on CFA, we need to provide information on all aspects just described.
- **Specification**: Model to be tested, and where this model derived from.
- **Identification**: Constraints placed on the model for identification.
- **Estimation**: The method and software used to run the model: e.g. maximum likelihood in lavaan.
- **Modification**
  - Prior to fitting the model, a discussion of how modification information will be used.
  - After estimation, discussion of which paths were added, why, and the resultant change in model fit.

Reporting CFA (2)

- **Evaluation**:
  - Prior to estimating the model, a discussion of which fit indices and what cut-offs will be used.
  - After estimation, a discussion of overall fit, then discussion of parameters of interest.
  - Interpretation of the model estimates versus any theory or hypotheses.
- **NOTE**: As is always the case in reporting a methods and results section, you need to provide enough information to reproduce the model.