Using the psych package to generate and test structural models

William Revelle

August 22, 2020

Contents

1 The psych package 2
1.1 Preface ................................................. 2
1.2 Creating and modeling structural relations .................... 2

2 Functions for generating correlational matrices with a particular structure 3
2.1 sim.congeneric .......................................... 4
2.2 sim.hierarchical ...................................... 6
2.3 sim.item and sim.circ .................................. 7
2.4 sim.structure ........................................... 7
   2.4.1 \textbf{f} is a vector implies a congeneric model .............. 7
   2.4.2 \textbf{f} is a matrix implies an independent factors model: .... 10
   2.4.3 \textbf{f} is a matrix and Phi \neq I is a correlated factors model . 10
   2.4.4 \textbf{f} and \textbf{f} are matrices, and Phi \neq I represents their correlations . 15
   2.4.5 A hierarchical structure among the latent predictors ......... 15

3 Exploratory functions for analyzing structure 15
3.1 Exploratory simple structure models .......................... 18
3.2 Exploratory hierarchical models ............................... 22
   3.2.1 A bifactor solution .................................... 22
   3.2.2 A hierarchical solution ................................. 22

4 Exploratory Structural Equation Modeling (ESEM) 26

5 Confirmatory models 28
   5.1 Using psych as a front end for the sem package ............... 30
   5.2 Testing a congeneric model versus a tau equivalent model .... 30
5.3 Testing the dimensionality of a hierarchical data set by creating the model . 32
5.4 Testing the dimensionality based upon an exploratory analysis . . . . . . . 33
5.5 Specifying a three factor model . . . . . . . . . . . . . . . . . . . . . . . . 34
5.6 Allowing for an oblique solution . . . . . . . . . . . . . . . . . . . . . . . . 35
5.7 Extract a bifactor solution using \textit{omega} and then test that model using \textit{sem} . 37
5.7.1 \textit{sem} of Thurstone 9 variable problem . . . . . . . . . . . . . . . 37
5.8 Examining a hierarchical solution . . . . . . . . . . . . . . . . . . . . . . . . 40
5.9 Estimating Omega using EFA followed by CFA . . . . . . . . . . . . . . . 43

6 Summary and conclusion 45
1 The psych package

1.1 Preface

The psych package (Revelle, 2020) has been developed to include those functions most useful for teaching and learning basic psychometrics and personality theory. Functions have been developed for many parts of the analysis of test data, including basic descriptive statistics (describe and pairs.panels), dimensionality analysis (ICLUST, VSS, principal, factor.pa), reliability analysis (omega, guttman) and eventual scale construction (cluster.cor, score.items). The use of these and other functions is described in more detail in the accompanying vignette (overview.pdf) as well as in the complete user’s manual and the relevant help pages. (These vignettes are also available at https://personality-project.org/r/overview.pdf and https://personality-project.org/r/psych_for_sem.pdf).

This vignette is concerned with the problem of modeling structural data and using the psych package as a front end for the much more powerful sem package of John Fox (Fox, 2006, 2009; Fox et al., 2013). Future releases of this vignette will include examples for using the lavaan package of Yves Rosseel (Rosseel, 2012).

The first section discusses how to simulate particular latent variable structures. The second considers several Exploratory Factor Analysis (EFA) solutions to these problems. The third section considers how to do confirmatory factor analysis and structural equation modeling using the sem package but with the input prepared using functions in the psych package.

1.2 Creating and modeling structural relations

One common application of psych is the creation of simulated data matrices with particular structures to use as examples for principal components analysis, factor analysis, cluster analysis, and structural equation modeling. This vignette describes some of the functions used for creating, analyzing, and displaying such data sets. The examples use two other packages: Rgraphviz and sem. Although not required to use the psych package, sem is required for these examples. Although Rgraphviz had been used for the graphical displays, it has now been replaced with graphical functions within psych. The analyses themselves require only the sem package to do the structural modeling.

Since writing this vignette, the power of the lavaan has allowed me to replace many of the sem commands with calls to lavaan. lavaan is loaded along with psych.
2 Functions for generating correlational matrices with a particular structure

The `sim` family of functions create data sets with particular structure. Most of these functions have default values that will produce useful examples. Although graphical summaries of these structures will be shown here, some of the options of the graphical displays will be discussed in a later section.

The `sim` functions include:

- **sim.structure** A function to combine a measurement and structural model into one data matrix. Useful for understanding structural equation models. Combined with `structure.diagram` to see the proposed structure.

- **sim.congeneric** A function to create congeneric items/tests for demonstrating classical test theory. This is just a special case of `sim.structure`.

- **sim.hierarchical** A function to create data with a hierarchical (bifactor) structure.

- **sim.general** A function to simulate a general factor and multiple group factors. This is done in a somewhat more obvious, although less general, method than `sim.hierarchical`.

- **simCor** A generalized function to simulate data from a specified correlation matrix.

- **sim.item** A function to create items that either have a simple structure or a circumplex structure.

- **sim.circ** Create data with a circumplex structure.

- **sim.dichot** Create dichotomous item data with a simple or circumplex structure.

- **sim.minor** Create a factor structure for `nvar` variables defined by `nfact` major factors and $\frac{nvar}{2}$ "minor" factors for `n` observations.

- **sim.parallel** Create a number of simulated data sets using `sim.minor` to show how parallel analysis works.

- **sim.rasch** Create IRT data following a Rasch model.

- **sim.irt** Create a two parameter IRT logistic (2PL) model.

- **sim.anova** Simulate a 3 way balanced ANOVA or linear model, with or without repeated measures. Useful for teaching courses in research methods.

To make these examples replicable for readers, all simulations are prefaced by setting the random seed to a fixed (and for some, memorable) number (Adams, 1980). For normal use of the simulations, this is not necessary.
2.1 sim.congeneric

Classical test theory considers tests to be \( \tau \) equivalent if they have the same covariance with a vector of latent true scores, but perhaps different error variances. Tests are considered congeneric if they each have the same true score component (perhaps to a different degree) and independent error components. The sim.congeneric function may be used to generate either structure.

The first example considers four tests with equal loadings on a latent factor (that is, a \( \tau \) equivalent model). If the number of subjects is not specified, a population correlation matrix will be generated. If N is specified, then the sample correlation matrix is returned. If the “short” option is FALSE, then the population matrix, sample matrix, and sample data are all returned as elements of a list.

```r
> library(psych)
> library(psychTools)
> set.seed(42)
> tau <- sim.congeneric(loads=c(.8,.8,.8,.8)) #population values
> tau.samp <- sim.congeneric(loads=c(.8,.8,.8,.8),N=100) # sample correlation matrix for 100 cases
> round(tau.samp,2)
V1  V2  V3  V4
V1 1.00 0.68 0.72 0.66
V2 0.68 1.00 0.65 0.67
V3 0.72 0.65 1.00 0.76
V4 0.66 0.67 0.76 1.00

> tau.samp <- sim.congeneric(loads=c(.8,.8,.8,.8),N=100, short=FALSE)
> tau.samp

Call: NULL

$model (Population correlation matrix)
V1  V2  V3  V4
V1 1.00 0.64 0.64 0.64
V2 0.64 1.00 0.64 0.64
V3 0.64 0.64 1.00 0.64
V4 0.64 0.64 0.64 1.00

$r (Sample correlation matrix for sample size = 100 )
V1  V2  V3  V4
V1 1.00 0.70 0.62 0.58
V2 0.70 1.00 0.65 0.64
V3 0.62 0.65 1.00 0.59
V4 0.58 0.64 0.59 1.00

> dim(tau.samp$observed)
[1] 100  4

In this last case, the generated data are retrieved from tau.samp$observed. Congeneric data are created by specifying unequal loading values. The default values are loadings of c(.8,.7,.6,.5). As seen in Figure 1, \( \tau \) equivalence is the special case where all paths are equal.
> cong <- sim.congeneric(N=100)
> round(cong,2)

<table>
<thead>
<tr>
<th></th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
<th>V4</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>1.00</td>
<td>0.57</td>
<td>0.53</td>
<td>0.46</td>
</tr>
<tr>
<td>V2</td>
<td>0.57</td>
<td>1.00</td>
<td>0.35</td>
<td>0.41</td>
</tr>
<tr>
<td>V3</td>
<td>0.53</td>
<td>0.35</td>
<td>1.00</td>
<td>0.43</td>
</tr>
<tr>
<td>V4</td>
<td>0.46</td>
<td>0.41</td>
<td>0.43</td>
<td>1.00</td>
</tr>
</tbody>
</table>

> #plot.new()
> m1 <- structure.diagram(c("a","b","c","d"))

Figure 1: Tau equivalent tests are special cases of congeneric tests. Tau equivalence assumes \( a=b=c=d \)
2.2 sim.hierarchical

The previous function, sim.congeneric, is used when one factor accounts for the pattern of correlations. A slightly more complicated model is when one broad factor and several narrower factors are observed. An example of this structure might be the structure of mental abilities, where there is a broad factor of general ability and several narrower factors (e.g., spatial ability, verbal ability, working memory capacity). Another example is in the measure of psychopathology where a broad general factor of neuroticism is seen along with more specific anxiety, depression, and aggression factors. This kind of structure may be simulated with sim.hierarchical specifying the loadings of each sub factor on a general factor (the g-loadings) as well as the loadings of individual items on the lower order factors (the f-loadings). An early paper describing a bifactor structure was by Holzinger and Swineford (1937). A helpful description of what makes a good general factor is that of Jensen and Weng (1994).

For those who prefer real data to simulated data, six data sets are included in the bifactor data set. One is the original 14 variable problem of Holzinger and Swineford (1937) (holzinger), a second is a nine variable problem adapted by Bechtoldt (1961) from Thurstone and Thurstone (1941) (the data set is used as an example in the SAS manual and discussed in great detail by McDonald (1999)), a third is from a recent paper by Reise et al. (2007) with 16 measures of patient reports of interactions with their health care provider.

```r
> set.seed(42)
> gload=matrix(c(.9,.8,.7),nrow=3)
> fload <- matrix(c(.8,.7,.6,rep(0,9),.7,.6,.5,
+ rep(0,9),.7,.6,.4), ncol=3)
> fload #echo it to see the structure
     [,1] [,2] [,3]  
[1,]  0.8  0.0  0.0  
[2,]  0.7  0.0  0.0  
[3,]  0.6  0.0  0.0  
[4,]  0.0  0.7  0.0  
[5,]  0.0  0.6  0.0  
[6,]  0.0  0.5  0.0  
[7,]  0.0  0.0  0.7  
[8,]  0.0  0.0  0.6  
[9,]  0.0  0.0  0.4
> bifact <- sim.hierarchical(gload=gload,fload=fload)
> round(bifact,2)
     V1  V2  V3  V4  V5  V6  V7  V8  V9
V1 1.00 0.56 0.48 0.40 0.35 0.29 0.35 0.30 0.20
V2 0.56 1.00 0.42 0.35 0.30 0.25 0.31 0.26 0.18
V3 0.48 0.42 1.00 0.30 0.26 0.22 0.26 0.23 0.15
V4 0.40 0.35 0.30 1.00 0.42 0.35 0.27 0.24 0.16
V5 0.35 0.30 0.26 0.42 1.00 0.30 0.24 0.20 0.13
V6 0.29 0.25 0.22 0.35 0.30 1.00 0.20 0.17 0.11
V7 0.35 0.31 0.26 0.27 0.24 0.20 1.00 0.42 0.28
```
These data can be represented as either a *bifactor* (Figure 2 panel A) or *hierarchical* (Figure 2 Panel B) factor solution. The analysis was done with the *omega* function.

### 2.3 sim.item and sim.circ

Many personality questionnaires are thought to represent multiple, independent factors. A particularly interesting case is when there are two factors and the items either have *simple structure* or *circumplex structure*. Examples of such items with a circumplex structure are measures of emotion (Rafaeli and Revelle, 2006) where many different emotion terms can be arranged in a two dimensional space, but where there is no obvious clustering of items. Typical personality scales are constructed to have simple structure, where items load on one and only one factor.

An additional challenge to measurement with emotion or personality items is that the items can be highly skewed and are assessed with a small number of discrete categories (do not agree, somewhat agree, strongly agree).

The more general *sim.item* function, and the more specific, *sim.circ* functions simulate items with a two dimensional structure, with or without skew, and varying the number of categories for the items. An example of a circumplex structure is shown in Figure 3.

### 2.4 sim.structure

A more general case is to consider three matrices, \( f_x, \phi_{xy}, f_y \) which describe, in turn, a measurement model of x variables, \( f_x \), a measurement model of y variables, \( f_y \), and a covariance matrix between and within the two sets of factors. If \( f_x \) is a vector and \( f_y \) and \( \phi_{xy} \) are NULL, then this is just the congeneric model. If \( f_x \) is a matrix of loadings with \( n \) rows and \( c \) columns, then this is a measurement model for \( n \) variables across \( c \) factors.

If \( \phi_{xy} \) is not null, but \( f_y \) is NULL, then the factors in \( f_x \) are correlated. Finally, if all three matrices are not NULL, then the data show the standard linear structural relations (LISREL) structure.

Consider the following examples:

#### 2.4.1 \( f_x \) is a vector implies a congeneric model

```r
> set.seed(42)
> fx <- c(.9,.8,.7,.6)
> cong1 <- sim.structure(fx)
> cong1

Call: sim.structure(fx = fx)

$model (Population correlation matrix)
```
Figure 2: (Left panel) A bifactor solution represents each test in terms of a general factor and a residualized group factor. (Right Panel) A hierarchical factor solution has g as a second order factor accounting for the correlations between the first order factors
> circ <- sim.circ(16)
> f2 <- fa(circ,2)
> plot(f2,title="16 simulated variables in a circumplex pattern")

Figure 3: Emotion items or interpersonal items frequently show a circumplex structure. Data generated by sim.circ and factor loadings found by the principal axis algorithm using factor.pa.
2.4.2 $f_x$ is a matrix implies an independent factors model:

```r
> set.seed(42)
> fx <- matrix(c(.9,.8,.7,rep(0,9),.7,.6,.5,rep(0,9),.6,.5,.4), ncol=3)
> three.fact <- sim.structure(fx)
> three.fact

Call: sim.structure(fx = fx)

$model (Population correlation matrix)

V1  V2  V3  V4  V5  V6  V7  V8  V9
V1 1.00 0.72 0.63 0.00 0.00 0.00 0.00 0.00 0.00
V2 0.72 1.00 0.56 0.00 0.00 0.00 0.00 0.00 0.00
V3 0.63 0.56 1.00 0.42 0.35 0.00 0.00 0.00 0.00
V4 0.00 0.00 0.42 1.00 0.30 0.00 0.00 0.00 0.00
V5 0.00 0.00 0.35 0.30 1.00 0.00 0.00 0.00 0.00
V6 0.00 0.00 0.30 0.30 1.00 0.00 0.00 0.00 0.00
V7 0.00 0.00 0.00 0.00 0.00 1.00 0.3 0.24 0.00
V8 0.00 0.00 0.00 0.00 0.00 0.00 1.0 0.20 0.00
V9 0.00 0.00 0.00 0.00 0.00 0.00 0.24 0.2 1.00

$reliability (population reliability)

[1] 0.81 0.64 0.49 0.36
```

2.4.3 $f_x$ is a matrix and $\Phi \neq I$ is a correlated factors model

```r
> Phi = matrix(c(1,.5,.3,.5,1,.2,.3,.2,1), ncol=3)
> cor.f3 <- sim.structure(fx,Phi)
> fx

[,1] [,2] [,3]
[1,] 0.9 0.0 0.0
[2,] 0.8 0.0 0.0
[3,] 0.7 0.0 0.0
[4,] 0.0 0.7 0.0
[5,] 0.0 0.6 0.0
[6,] 0.0 0.5 0.0
[7,] 0.0 0.0 0.6
[8,] 0.0 0.0 0.5
[9,] 0.0 0.0 0.4

> Phi

[,1] [,2] [,3]
[1,] 1.0 0.5 0.3
[2,] 0.5 1.0 0.2
[3,] 0.3 0.2 1.0

> cor.f3
```
Figure 4: Three uncorrelated factors generated using the sim.structure function and drawn using structure.diagram.
Call: sim.structure(fx = fx, Phi = Phi)

$model (Population correlation matrix)

<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>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>1.00</td>
<td>0.72</td>
<td>0.63</td>
<td>0.315</td>
<td>0.270</td>
<td>0.23</td>
<td>0.162</td>
<td>0.14</td>
<td>0.108</td>
</tr>
<tr>
<td>V2</td>
<td>0.72</td>
<td>1.00</td>
<td>0.560</td>
<td>0.280</td>
<td>0.240</td>
<td>0.20</td>
<td>0.144</td>
<td>0.12</td>
<td>0.096</td>
</tr>
<tr>
<td>V3</td>
<td>0.63</td>
<td>0.560</td>
<td>1.00</td>
<td>0.265</td>
<td>0.210</td>
<td>0.17</td>
<td>0.126</td>
<td>0.10</td>
<td>0.084</td>
</tr>
<tr>
<td>V4</td>
<td>0.32</td>
<td>0.280</td>
<td>1.00</td>
<td>0.425</td>
<td>0.350</td>
<td>0.35</td>
<td>0.084</td>
<td>0.07</td>
<td>0.056</td>
</tr>
<tr>
<td>V5</td>
<td>0.27</td>
<td>0.240</td>
<td>0.210</td>
<td>1.000</td>
<td>0.420</td>
<td>0.42</td>
<td>0.350</td>
<td>0.30</td>
<td>0.280</td>
</tr>
<tr>
<td>V6</td>
<td>0.23</td>
<td>0.200</td>
<td>0.175</td>
<td>0.350</td>
<td>0.300</td>
<td>1.00</td>
<td>0.060</td>
<td>0.05</td>
<td>0.040</td>
</tr>
<tr>
<td>V7</td>
<td>0.16</td>
<td>0.144</td>
<td>0.126</td>
<td>0.084</td>
<td>0.072</td>
<td>0.06</td>
<td>1.000</td>
<td>0.30</td>
<td>0.240</td>
</tr>
<tr>
<td>V8</td>
<td>0.14</td>
<td>0.120</td>
<td>0.105</td>
<td>0.070</td>
<td>0.060</td>
<td>0.05</td>
<td>0.300</td>
<td>1.00</td>
<td>0.200</td>
</tr>
<tr>
<td>V9</td>
<td>0.11</td>
<td>0.096</td>
<td>0.084</td>
<td>0.056</td>
<td>0.048</td>
<td>0.04</td>
<td>0.240</td>
<td>0.20</td>
<td>1.000</td>
</tr>
</tbody>
</table>

$reliability (population reliability)

[1] 0.81 0.64 0.49 0.49 0.36 0.25 0.36 0.25 0.16

Using symbolic loadings and path coefficients  For some purposes, it is helpful not to specify particular values for the paths, but rather to think of them symbolically. This can be shown with symbolic loadings and path coefficients by using the structure.list and phi.list functions to create the fx and Phi matrices (Figure 5).

> fxs <- structure.list(9,list(F1=c(1,2,3),F2=c(4,5,6),F3=c(7,8,9)))
> Phis <- phi.list(3,list(F1=c(2,3),F2=c(1,3),F3=c(1,2)))
> fxs  #show the matrix

F1   F2   F3
[1,] "a1" "0"  "0"
[2,] "a2" "0"  "0"
[3,] "a3" "0"  "0"
[4,] "0"  "b4" "0"
[5,] "0"  "b5" "0"
[6,] "0"  "b6" "0"
[7,] "0"  "0"  "c7"
[8,] "0"  "0"  "c8"
[9,] "0"  "0"  "c9"

> Phis #show this one as well

F1   F2   F3
F1 "rba" "rca"
F2 "rab" "1"  "rcb"
F3 "rac" "rbc" "1"

The structure.list and phi.list functions allow for creation of fx, Phi, and fy matrices in a very compact form, just by specifying the relevant variables.
> #plot.new()
> corf3.mod <- structure.diagram(fxs,Phis)

Structural model

![Structural model diagram](image)

Figure 5: Three correlated factors with symbolic paths. Created using structure.diagram and structure.list and phi.list for ease of input.
Drawing path models from Exploratory Factor Analysis solutions

Alternatively, this result can represent the estimated factor loadings and oblique correlations found using `factanal` (Maximum Likelihood factoring) or `fa` (Principal axis or minimum residual (minres) factoring) followed by a promax rotation using the `Promax` function (Figure 6.

Comparing this figure with the previous one (Figure 5), it will be seen that one path was dropped because it was less than the arbitrary “cut” value of .2.

```r
> f3.p <- Promax(fa(cor.f3$model,3))
> #plot.new()
> mod.f3p <- structure.diagram(f3.p,cut=.2)
```

![Structural model](image)

Figure 6: The empirically fitted structural model. Paths less than cut (.2 in this case, the default is .3) are not shown.
2.4.4 $f_x$ and $f_y$ are matrices, and Phi $\neq I$ represents their correlations

A more complicated model is when there is a $f_y$ vector or matrix representing a set of Y latent variables that are associated with the a set of y variables. In this case, the Phi matrix is a set of correlations within the X set and between the X and Y set.

```r
> set.seed(42)
> fx <- matrix(c(.9,.8,.7,rep(0,9),.7,.6,.5,rep(0,9),.6,.5,.4), ncol=3)
> fy <- c(.6,.5,.4)
> Phi <- matrix(c(1,.48,.32,.4,.48,1,.32,.3,.3,.32,1,.2,.4,.3,.2,1), ncol=4)
> twelveV <- sim.structure(fx,Phi, fy)$model
> colnames(twelveV) <-rownames(twelveV) <- c(paste("x",1:9,sep=""),paste("y",1:3,sep=""))
> round(twelveV,2)
```

Data with this structure may be created using the `sim.structure` function, and shown either with the numeric values or symbolically using the `structure.diagram` function (Figure 7).

```r
> fxs <- structure.list(9,list(X1=c(1,2,3), X2 =c(4,5,6),X3 = c(7,8,9)))
> phi <- phi.list(4,list(F1=c(4),F2=c(4),F3=c(4),F4=c(1,2,3)))
> fyx <- structure.list(3,list(Y=c(1,2,3)),"y")
```

2.4.5 A hierarchical structure among the latent predictors.

Measures of intelligence and psychopathology frequently have a general factor as well as multiple group factors. The general factor then is thought to predict some dependent latent variable. Compare this with the previous model (see Figure 7).

These two models can be compared using structural modeling procedures (see below).

3 Exploratory functions for analyzing structure

Given correlation matrices such as those seen above for congeneric or bifactor models, the question becomes how best to estimate the underlying structure. Because these data sets were generated from a known model, the question becomes how well does a particular model recover the underlying structure.
Figure 7: A symbolic structural model. Three independent latent variables are regressed on a latent $Y$. 

```
> #plot.new()
> sg3 <- structure.diagram(fxs,phi,fyx)
```
> fxh <- structure.list(9, list(X1=c(1:3), X2=c(4:6), X3=c(7:9), g=NULL))
> fy <- structure.list(3, list(Y=c(1, 2, 3)))
> Phi <- diag(1, 5, 5)
> Phi[4, c(1:3)] <- letters[1:3]
> Phi[5, 4] <- "r"
> #plot.new()
> hi.mod <- structure.diagram(fxh, Phi, fy)

Structural model

Figure 8: A symbolic structural model with a general factor and three group factors. The general factor is regressed on the latent Y variable.
3.1 Exploratory simple structure models

The technique of principal components provides a set of weighted linear composites that best approximates a particular correlation or covariance matrix. If these are then rotated to provide a more interpretable solution, the components are no longer the principal components. The principal function will extract the first n principal components (default value is 1) and if n > 1, rotate to simple structure using a varimax, quartimin, or Promax criterion.

> principal(cong1$model)

Principal Components Analysis
Call: principal(r = cong1$model)
Standardized loadings (pattern matrix) based upon correlation matrix
   PC1 h2 u2 com
V1 0.89 0.80 0.20 1
V2 0.85 0.73 0.27 1
V3 0.80 0.64 0.36 1
V4 0.73 0.53 0.47 1

PC1
SS loadings 2.69
Proportion Var 0.67
Mean item complexity = 1
Test of the hypothesis that 1 component is sufficient.
The root mean square of the residuals (RMSR) is 0.11
Fit based upon off diagonal values = 0.96

> fa(cong1$model)

Factor Analysis using method = minres
Call: fa(r = cong1$model)
Standardized loadings (pattern matrix) based upon correlation matrix
   MR1 h2 u2 com
V1 0.9 0.81 0.19 1
V2 0.8 0.64 0.36 1
V3 0.7 0.49 0.51 1
V4 0.6 0.36 0.64 1

MR1
SS loadings 2.30
Proportion Var 0.57
Mean item complexity = 1
Test of the hypothesis that 1 factor is sufficient.
The degrees of freedom for the null model are 6 and the objective function was 1.65
The degrees of freedom for the model are 2 and the objective function was 0
The root mean square of the residuals (RMSR) is 0
The df corrected root mean square of the residuals is 0
Fit based upon off diagonal values = 1
Measures of factor score adequacy
Correlation of (regression) scores with factors 0.94
Multiple R square of scores with factors 0.88
Minimum correlation of possible factor scores 0.77

It is important to note that although the principal components function does not exactly reproduce the model parameters, the factor.pa function, implementing principal axes or minimum residual (minres) factor analysis, does.

Consider the case of three underlying factors as seen in the bifact example above. Because the number of observations is not specified, there is no associated $\chi^2$ value. The factor.congruence function reports the cosine of the angle between the factors.

```r
> pc3 <- principal(bifact,3)
> pa3 <- fa(bifact,3,fm="pa")
> ml3 <- fa(bifact,3,fm="ml")
> pc3
```

Principal Components Analysis
Call: principal(r = bifact, nfactors = 3)
Standardized loadings (pattern matrix) based upon correlation matrix

<table>
<thead>
<tr>
<th></th>
<th>RC1</th>
<th>RC3</th>
<th>RC2</th>
<th>h2</th>
<th>u2</th>
<th>com</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>0.75</td>
<td>0.27</td>
<td>0.21</td>
<td>0.69</td>
<td>0.31</td>
<td>1.4</td>
</tr>
<tr>
<td>V2</td>
<td>0.76</td>
<td>0.21</td>
<td>0.16</td>
<td>0.64</td>
<td>0.36</td>
<td>1.2</td>
</tr>
<tr>
<td>V3</td>
<td>0.78</td>
<td>0.11</td>
<td>0.10</td>
<td>0.63</td>
<td>0.37</td>
<td>1.1</td>
</tr>
<tr>
<td>V4</td>
<td>0.29</td>
<td>0.69</td>
<td>0.15</td>
<td>0.59</td>
<td>0.41</td>
<td>1.5</td>
</tr>
<tr>
<td>V5</td>
<td>0.20</td>
<td>0.71</td>
<td>0.11</td>
<td>0.56</td>
<td>0.44</td>
<td>1.2</td>
</tr>
<tr>
<td>V6</td>
<td>0.07</td>
<td>0.76</td>
<td>0.08</td>
<td>0.59</td>
<td>0.41</td>
<td>1.0</td>
</tr>
<tr>
<td>V7</td>
<td>0.26</td>
<td>0.16</td>
<td>0.70</td>
<td>0.58</td>
<td>0.42</td>
<td>1.4</td>
</tr>
<tr>
<td>V8</td>
<td>0.20</td>
<td>0.11</td>
<td>0.71</td>
<td>0.55</td>
<td>0.45</td>
<td>1.2</td>
</tr>
<tr>
<td>V9</td>
<td>0.00</td>
<td>0.06</td>
<td>0.73</td>
<td>0.53</td>
<td>0.47</td>
<td>1.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>RC1</th>
<th>RC3</th>
<th>RC2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS loadings</td>
<td>1.99</td>
<td>1.73</td>
<td>1.64</td>
</tr>
<tr>
<td>Proportion Var</td>
<td>0.22</td>
<td>0.19</td>
<td>0.18</td>
</tr>
<tr>
<td>Cumulative Var</td>
<td>0.22</td>
<td>0.41</td>
<td>0.60</td>
</tr>
<tr>
<td>Proportion Explained</td>
<td>0.37</td>
<td>0.32</td>
<td>0.31</td>
</tr>
<tr>
<td>Cumulative Proportion</td>
<td>0.37</td>
<td>0.69</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Mean item complexity = 1.2
Test of the hypothesis that 3 components are sufficient.
The root mean square of the residuals (RMSR) is 0.1

Fit based upon off diagonal values = 0.88

```r
> pa3
```

Factor Analysis using method = pa
Call: fa(r = bifact, nfactors = 3, fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix

<table>
<thead>
<tr>
<th></th>
<th>PA1</th>
<th>PA3</th>
<th>PA2</th>
<th>h2</th>
<th>u2</th>
<th>com</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>0.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.64</td>
<td>0.36</td>
<td>1</td>
</tr>
<tr>
<td>V2</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.49</td>
<td>0.51</td>
<td>1</td>
</tr>
<tr>
<td>V3</td>
<td>0.6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.36</td>
<td>0.64</td>
<td>1</td>
</tr>
<tr>
<td>V4</td>
<td>0.0</td>
<td>0.7</td>
<td>0.0</td>
<td>0.49</td>
<td>0.51</td>
<td>1</td>
</tr>
<tr>
<td>V5</td>
<td>0.0</td>
<td>0.6</td>
<td>0.0</td>
<td>0.36</td>
<td>0.64</td>
<td>1</td>
</tr>
</tbody>
</table>
With factor correlations of

PA1  PA3  PA2
PA1  1.00  0.72  0.63
PA3  0.72  1.00  0.56
PA2  0.63  0.56  1.00

Mean item complexity = 1
Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the null model are 36 and the objective function was 1.88
The degrees of freedom for the model are 12 and the objective function was 0

The root mean square of the residuals (RMSR) is 0
The df corrected root mean square of the residuals is 0

Fit based upon off diagonal values = 1
Measures of factor score adequacy

> m3

Factor Analysis using method = ml
Call: fa(r = bifact, nfactors = 3, fm = "ml")

Standardized loadings (pattern matrix) based upon correlation matrix

ML1  ML3  ML2  h²  u² com
V1  0.8  0.0  0.0  0.64  0.36  1
V2  0.7  0.0  0.0  0.49  0.51  1
V3  0.6  0.0  0.0  0.36  0.64  1
V4  0.0  0.7  0.0  0.49  0.51  1
V5  0.0  0.6  0.0  0.36  0.64  1
V6  0.0  0.5  0.0  0.25  0.75  1
V7  0.0  0.0  0.7  0.49  0.51  1
V8  0.0  0.0  0.6  0.36  0.64  1
V9  0.0  0.0  0.4  0.16  0.84  1
Mean item complexity = 1
Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the null model are 36 and the objective function was 1.88
The degrees of freedom for the model are 12 and the objective function was 0

The root mean square of the residuals (RMSR) is 0
The df corrected root mean square of the residuals is 0

Fit based upon off diagonal values = 1
Measures of factor score adequacy

<table>
<thead>
<tr>
<th></th>
<th>ML1</th>
<th>ML3</th>
<th>ML2</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML1</td>
<td>0.90</td>
<td>0.85</td>
<td>0.83</td>
</tr>
<tr>
<td>ML3</td>
<td>0.80</td>
<td>0.72</td>
<td>0.69</td>
</tr>
<tr>
<td>ML2</td>
<td>0.61</td>
<td>0.45</td>
<td>0.38</td>
</tr>
</tbody>
</table>

By default, all three of these procedures use the varimax rotation criterion. Perhaps it is useful to apply an oblique transformation such as Promax or oblimin to the results. The Promax function in psych differs slightly from the standard promax in that it reports the factor intercorrelations.

> ml3p <- Promax(ml3)
> ml3p

Call: NULL
Standardized loadings (pattern matrix) based upon correlation matrix

<table>
<thead>
<tr>
<th></th>
<th>ML1</th>
<th>ML3</th>
<th>ML2</th>
<th>h2</th>
<th>u2</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>0.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.64</td>
<td>0.36</td>
</tr>
<tr>
<td>V2</td>
<td>0.7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.49</td>
<td>0.51</td>
</tr>
<tr>
<td>V3</td>
<td>0.6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.36</td>
<td>0.64</td>
</tr>
<tr>
<td>V4</td>
<td>0.0</td>
<td>0.7</td>
<td>0.0</td>
<td>0.49</td>
<td>0.51</td>
</tr>
<tr>
<td>V5</td>
<td>0.0</td>
<td>0.6</td>
<td>0.0</td>
<td>0.36</td>
<td>0.64</td>
</tr>
<tr>
<td>V6</td>
<td>0.0</td>
<td>0.5</td>
<td>0.0</td>
<td>0.25</td>
<td>0.75</td>
</tr>
<tr>
<td>V7</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>0.49</td>
<td>0.51</td>
</tr>
<tr>
<td>V8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.6</td>
<td>0.36</td>
<td>0.64</td>
</tr>
<tr>
<td>V9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.4</td>
<td>0.16</td>
<td>0.84</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>ML1</th>
<th>ML3</th>
<th>ML2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS loadings</td>
<td>1.49</td>
<td>1.10</td>
<td>1.01</td>
</tr>
<tr>
<td>Proportion Var</td>
<td>0.17</td>
<td>0.12</td>
<td>0.11</td>
</tr>
<tr>
<td>Cumulative Var</td>
<td>0.17</td>
<td>0.29</td>
<td>0.40</td>
</tr>
</tbody>
</table>
3.2 Exploratory hierarchical models

In addition to the conventional oblique factor model, an alternative model is to consider the correlations between the factors to represent a higher order factor. This can be shown either as a bifactor solution Holzinger and Swineford (1937); Schmid and Leiman (1957) with a general factor for all variables and a set of residualized group factors, or as a hierarchical structure. An exploratory hierarchical model can be applied to this kind of data structure using the omega function. Graphic options include drawing a Schmid - Leiman bifactor solution (Figure 9) or drawing a hierarchical factor solution (Figure 10).

3.2.1 A bifactor solution

The bifactor solution has a general factor loading for each variable as well as a set of residual group factors. This approach has been used extensively in the measurement of ability and has more recently been used in the measure of psychopathology (Reise et al., 2007). Data sets included in the bifactor data include the original (Holzinger and Swineford, 1937) data set (holzinger) as well as a set from Reise et al. (2007) (reise) and a nine variable problem from Thurstone.

3.2.2 A hierarchical solution

Both of these graphical representations are reflected in the output of the omega function. The first was done using a Schmid-Leiman transformation, the second was not. As will be seen later, the objects returned from these two analyses may be used as models for a sem analysis. It is also useful to examine the estimates of reliability reported by omega.

> om.bi

Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
   digits = digits, title = title, sl = sl, labels = labels,
   plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
   covar = covar)
  Alpha: 0.78
  G.6: 0.78
  Omega Hierarchical: 0.7
  Omega H asymptotic: 0.85
> om.bi <- omega(bifact)

Figure 9: An exploratory bifactor solution to the nine variable problem
> om.hi <- omega(bifact, sl=FALSE)

Omega

Figure 10: An exploratory hierarchical solution to the nine variable problem.
Omega Total 0.82

Schmid Leiman Factor loadings greater than 0.2

<table>
<thead>
<tr>
<th></th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
<th>h2</th>
<th>u2</th>
<th>p2</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>0.72</td>
<td>0.35</td>
<td></td>
<td>0.64</td>
<td>0.36</td>
<td>0.81</td>
</tr>
<tr>
<td>V2</td>
<td>0.63</td>
<td>0.31</td>
<td></td>
<td>0.49</td>
<td>0.51</td>
<td>0.81</td>
</tr>
<tr>
<td>V3</td>
<td>0.54</td>
<td>0.26</td>
<td></td>
<td>0.36</td>
<td>0.64</td>
<td>0.81</td>
</tr>
<tr>
<td>V4</td>
<td>0.56</td>
<td>0.42</td>
<td></td>
<td>0.49</td>
<td>0.51</td>
<td>0.64</td>
</tr>
<tr>
<td>V5</td>
<td>0.48</td>
<td>0.36</td>
<td></td>
<td>0.36</td>
<td>0.64</td>
<td>0.64</td>
</tr>
<tr>
<td>V6</td>
<td>0.40</td>
<td>0.30</td>
<td></td>
<td>0.25</td>
<td>0.75</td>
<td>0.64</td>
</tr>
<tr>
<td>V7</td>
<td>0.49</td>
<td></td>
<td></td>
<td>0.50</td>
<td>0.49</td>
<td>0.51</td>
</tr>
<tr>
<td>V8</td>
<td>0.42</td>
<td></td>
<td></td>
<td>0.43</td>
<td>0.36</td>
<td>0.64</td>
</tr>
<tr>
<td>V9</td>
<td>0.28</td>
<td></td>
<td></td>
<td>0.29</td>
<td>0.16</td>
<td>0.84</td>
</tr>
</tbody>
</table>

With eigenvalues of:

<table>
<thead>
<tr>
<th></th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
<td>2.41</td>
<td>0.28</td>
<td>0.40</td>
</tr>
<tr>
<td>max</td>
<td>0.40</td>
<td>0.52</td>
<td></td>
</tr>
</tbody>
</table>

general/max 4.67 max/min = 1.82
mean percent general = 0.65 with sd = 0.14 and cv of 0.21
Explained Common Variance of the general factor = 0.67

The degrees of freedom are 12 and the fit is 0
The root mean square of the residuals is 0
The df corrected root mean square of the residuals is 0

Compare this with the adequacy of just a general factor and no group factors
The degrees of freedom for just the general factor are 27 and the fit is 0.23
The root mean square of the residuals is 0.07
The df corrected root mean square of the residuals is 0.08

Measures of factor score adequacy

<table>
<thead>
<tr>
<th></th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation of scores with factors</td>
<td>0.86</td>
<td>0.47</td>
<td>0.57</td>
</tr>
<tr>
<td>Multiple R square of scores with factors</td>
<td>0.74</td>
<td>0.22</td>
<td>0.33</td>
</tr>
<tr>
<td>Minimum correlation of factor score estimates</td>
<td>0.47</td>
<td>-0.56</td>
<td>-0.35</td>
</tr>
</tbody>
</table>

Total, General and Subset omega for each subset

<table>
<thead>
<tr>
<th></th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Yet one more way to treat the hierarchical structure of a data set is to consider hierarchical cluster analysis using the ICLUST algorithm (Figure 11). ICLUST is most appropriate for forming item composites.

Hierarchical cluster analysis of bifact data

![Hierarchical cluster analysis of bifact data](image)

Figure 11: A hierarchical cluster analysis of the bifact data set using ICLUST

4 Exploratory Structural Equation Modeling (ESEM)

Traditional Exploratory Factor Analysis (EFA) examines how latent variables can account for the correlations within a data set. All loadings and cross loadings are found and
rotation is done to some approximation of simple structure. Traditional Confirmatory Factor Analysis (CFA) tests such models by fitting just a limited number of loadings and typically does not allow any (or many) cross loadings. Structural Equation Modeling then applies two such measurement models, one to a set of X variables, another to a set of Y variables, and then tries to estimate the correlation between these two sets of latent variables. (Some SEM procedures estimate all the parameters from the same model, thus making the loadings in set Y affect those in set X.) It is possible to do a similar, exploratory modeling (ESEM) by conducting two Exploratory Factor Analyses, one in set X, one in set Y, and then finding the correlations of the X factors with the Y factors, as well as the correlations of the Y variables with the X factors and the X variables with the Y factors.

Consider the simulated data set of three ability variables, two motivational variables, and three outcome variables:

```
Call: sim.structural(fx = fx, Phi = Phi, fy = fy)

$model (Population correlation matrix)

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>Q</th>
<th>A</th>
<th>nach</th>
<th>Anx</th>
<th>gpa</th>
<th>Pre</th>
<th>MA</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>1.00</td>
<td>0.72</td>
<td>0.54</td>
<td>0.00</td>
<td>0.38</td>
<td>0.32</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>Q</td>
<td>0.72</td>
<td>1.00</td>
<td>0.48</td>
<td>0.00</td>
<td>0.34</td>
<td>0.28</td>
<td>0.22</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>0.54</td>
<td>0.48</td>
<td>1.00</td>
<td>0.48</td>
<td>-0.42</td>
<td>0.50</td>
<td>0.42</td>
<td>0.34</td>
</tr>
<tr>
<td>nach</td>
<td>0.00</td>
<td>0.00</td>
<td>0.48</td>
<td>1.00</td>
<td>-0.56</td>
<td>0.34</td>
<td>0.28</td>
<td>0.22</td>
</tr>
<tr>
<td>Anx</td>
<td>0.00</td>
<td>0.00</td>
<td>-0.42</td>
<td>-0.56</td>
<td>1.00</td>
<td>-0.29</td>
<td>-0.24</td>
<td>-0.20</td>
</tr>
<tr>
<td>gpa</td>
<td>0.38</td>
<td>0.34</td>
<td>0.50</td>
<td>0.34</td>
<td>-0.29</td>
<td>1.00</td>
<td>0.30</td>
<td>0.24</td>
</tr>
<tr>
<td>Pre</td>
<td>0.32</td>
<td>0.28</td>
<td>0.42</td>
<td>0.28</td>
<td>-0.24</td>
<td>0.30</td>
<td>1.00</td>
<td>0.20</td>
</tr>
<tr>
<td>MA</td>
<td>0.25</td>
<td>0.22</td>
<td>0.34</td>
<td>0.22</td>
<td>-0.20</td>
<td>0.24</td>
<td>0.20</td>
<td>1.00</td>
</tr>
</tbody>
</table>

$reliability (population reliability)

<table>
<thead>
<tr>
<th></th>
<th>V</th>
<th>Q</th>
<th>A</th>
<th>nach</th>
<th>Anx</th>
<th>gpa</th>
<th>Pre</th>
<th>MA</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>0.81</td>
<td>0.64</td>
<td>0.72</td>
<td>0.64</td>
<td>0.49</td>
<td>0.36</td>
<td>0.25</td>
<td>0.16</td>
</tr>
</tbody>
</table>

We can fit this by using the `esem` function and then draw the solution (see Figure 12) using the `esem.diagram` function (which is normally called automatically by `esem`.

Exploratory Structural Equation Modeling Analysis using method = minres

```
Call: esem(r = gre.gpa$gm, varsX = 1:5, varsY = 6:8, nfX = 2, nfY = 1, n.obs = 1000, plot = FALSE)

For the 'X' set:

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>0.91</td>
<td>-0.06</td>
</tr>
<tr>
<td>Q</td>
<td>0.81</td>
<td>-0.05</td>
</tr>
</tbody>
</table>
```

28
A  0.53  0.57  
nach -0.10  0.81  
Anx  0.08 -0.71  

For the 'Y' set:  
  Y1  
gpa  0.6  
Pre  0.5  
MA  0.4  

Correlations between the X and Y sets.  
 X1    X2    Y1  
 X1  1.00  0.19  0.68  
 X2  0.19  1.00  0.67  
 Y1  0.68  0.67  1.00  

The degrees of freedom for the null model are 56 and the empirical chi square function was 6930.29 with prob < 0.0027  
The degrees of freedom for the model are 7 and the empirical chi square function was 21.83 with prob < 0.0027  

The root mean square of the residuals (RMSR) is 0.02  
The df corrected root mean square of the residuals is 0.04 with the empirical chi square 21.83 with prob < 0.0027  
The total number of observations was 1000 with fitted Chi Square = 2175.06 with prob < 0  

Empirical BIC = -26.53  
ESABIC = -4.29  
Fit based upon off diagonal values = 1  
To see the item loadings for the X and Y sets combined, and the associated fa output, print with short=FALSE.

5 Confirmatory models  

Although the exploratory models shown above do estimate the goodness of fit of the model and compare the residual matrix to a zero matrix using a $\chi^2$ statistic, they estimate more parameters than are necessary if there is indeed a simple structure, and they do not allow for tests of competing models. The sem function in the sem package by John Fox allows for confirmatory tests. The interested reader is referred to the sem manual for more detail (Fox et al., 2013).
Figure 12: An example of a Exploratory Structure Equation Model.
5.1 Using psych as a front end for the sem package

Because preparation of the sem commands is a bit tedious, several of the psych package functions have been designed to provide the appropriate commands. That is, the functions structure.list, phi.list, structure.diagram, structure.sem, and omega.graph may be used as a front end to sem. Usually with no modification, but sometimes with just slight modification, the model output from the structure.diagram, structure.sem, and omega.graph functions is meant to provide the appropriate commands for sem.

5.2 Testing a congeneric model versus a tau equivalent model

The congeneric model is a one factor model with possibly unequal factor loadings. The tau equivalent model model is one with equal factor loadings. Tests for these may be done by creating the appropriate structures. The structure.graph function which requires Rgraphviz, or structure.diagram or the structure.sem functions which do not may be used.

The following example tests the hypothesis (which is actually false) that the correlations found in the cong data set (see 2.1) are tau equivalent. Because the variable labels in that data set were V1 ... V4, we specify the labels to match those.

```r
> library(sem)
> mod.tau <- structure.sem(c("a","a","a","a"),labels=paste("V",1:4,sep=""))
> mod.tau  #show it
Path  Parameter Value
[1.] "X1->V1" "a" NA
[2.] "X1->V2" "a" NA
[3.] "X1->V3" "a" NA
[4.] "X1->V4" "a" NA
[5.] "V1<->V1" "x1e" NA
[6.] "V2<->V2" "x2e" NA
[7.] "V3<->V3" "x3e" NA
[8.] "V4<->V4" "x4e" NA
[9.] "X1<->X1" NA "1"
attr("class")
[1] "mod"
> sem.tau <- sem(mod.tau,cong,100)
> summary(sem.tau,digits=2)
Model Chisquare = 6.593496 Df = 5 Pr(>Chisq) = 0.2526696
AIC = 16.5935
BIC = -16.43236

Normalized Residuals
Min. 1st Qu. Median Mean 3rd Qu. Max.
-1.03157 -0.44199 -0.25025 -0.07905 0.52702 0.88767

R-square for Endogenous Variables
V1 V2 V3 V4
0.5245 0.4592 0.4500 0.4432

Parameter Estimates
```
Test whether the data are congeneric. That is, whether a one factor model fits. Compare this to the prior model using the `anova` function.

```r
> mod.cong <- structure.sem(c("a","b","c","d"),labels=paste("V",1:4,sep=""))
> mod.cong #show the model
```

```r
Path Parameter Value
[1,] "X1->V1" "a" NA
[2,] "X1->V2" "b" NA
[3,] "X1->V3" "c" NA
[4,] "X1->V4" "d" NA
[5,] "V1<->V1" "x1e" NA
[6,] "V2<->V2" "x2e" NA
[7,] "V3<->V3" "x3e" NA
[8,] "V4<->V4" "x4e" NA
[9,] "X1<->X1" NA "1"
```

```r
attr(,"class")
[1] "mod"
```

```r
> sem.cong <- sem(mod.cong,cong,100)
> summary(sem.cong,digits=2)
```

```r
Model Chisquare = 2.941678 Df = 2 Pr(>Chisq) = 0.2297327
AIC = 18.94168
BIC = -6.268663

Normalized Residuals
            Min.  1st Qu.   Median     Mean  3rd Qu.   Max.
V1  0.688000 0.438400 0.394200 0.352400 0.360500  0.541200

R-square for Endogenous Variables
    V1    V2    V3    V4
0.6880 0.4384 0.3942 0.3524
```

```r
Parameter Estimates
Estimate Std Error z value Pr(>|z|)
a  0.8294562 0.0978677 8.475279 2.345174e-17 V1 <--- X1
b  0.6621164 0.1006678 6.577243 4.792500e-11 V2 <--- X1
c  0.6278767 0.1014686 6.187891 6.097433e-10 V3 <--- X1
d  0.5936695 0.1023882 5.798224 6.702094e-09 V4 <--- X1
x1e 0.3120026 0.1004487 3.106089 1.895798e-03 V1 <--> V1
x2e 0.5616018 0.1015489 5.530356 3.195810e-08 V2 <--> V2
x3e 0.6057707 0.1042129 5.812822 6.142832e-09 V3 <--> V3
x4e 0.6475566 0.1073299 6.033326 1.606191e-09 V4 <--> V4
```

```r
Iterations = 12
```

```r
> anova(sem.cong,sem.tau) #test the difference between the two models
```

LR Test for Difference Between Models
The **anova** comparison of the congeneric versus tau equivalent model shows that the change in $\chi^2$ is significant given the change in degrees of freedom.

### 5.3 Testing the dimensionality of a hierarchical data set by creating the model

The bifact correlation matrix was created to represent a hierarchical structure. Various confirmatory models can be applied to this matrix.

The first example creates the model directly, the next several create models based upon exploratory factor analyses. mod.one is a congeneric model of one factor accounting for the relationships between the nine variables. Although not correct, with 100 subjects, this model can not be rejected. However, an examination of the residuals suggests serious problems with the model.

```r
> mod.one <- structure.sem(letters[1:9],labels=paste("V",1:9,sep=""))
> mod.one #show the model

Path Parameter Value
[1,] "X1->V1" "a" NA
[2,] "X1->V2" "b" NA
[3,] "X1->V3" "c" NA
[4,] "X1->V4" "d" NA
[5,] "X1->V5" "e" NA
[6,] "X1->V6" "f" NA
[7,] "X1->V7" "g" NA
[8,] "X1->V8" "h" NA
[9,] "X1->V9" "i" NA
[10,] "V1<->V1" "x1e" NA
[11,] "V2<->V2" "x2e" NA
[12,] "V3<->V3" "x3e" NA
[13,] "V4<->V4" "x4e" NA
[14,] "V5<->V5" "x5e" NA
[15,] "V6<->V6" "x6e" NA
[16,] "V7<->V7" "x7e" NA
[17,] "V8<->V8" "x8e" NA
[18,] "V9<->V9" "x9e" NA
[19,] "X1<->X1" NA "1"
attr(,"class")
[1] "mod"
```

```r
> sem.one <- sem(mod.one,bifact,100)
> summary(sem.one,digits=2)

Model Chisquare = 21.16848 Df = 27 Pr(>Chisq) = 0.778334
AIC = 57.16848
BIC = -103.1711

Normalized Residuals

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
</table>
```

33
R-square for Endogenous Variables

V1   V2   V3   V4   V5   V6   V7   V8   V9
0.5636 0.4524 0.3377 0.3292 0.2522 0.1798 0.2568 0.1980 0.0932

Parameter Estimates

| Parameter | Estimate   | Std Error | z value | Pr(>|z|) |
|-----------|------------|-----------|---------|----------|
| a         | 0.7507129  | 0.0951287 | 7.891549| 2.984584e-15 |
| b         | 0.6726412  | 0.0980715 | 6.858682| 6.433059e-05 |
| c         | 0.5812000  | 0.1013793 | 5.732145| 6.433059e-05 |
| d         | 0.5737425  | 0.1016317 | 5.645309| 6.433059e-05 |
| e         | 0.5021915  | 0.1039278 | 4.832116| 6.433059e-05 |
| f         | 0.4239908  | 0.1000468 | 4.216227| 6.433059e-05 |
| g         | 0.5067957  | 0.1086717 | 4.216227| 6.433059e-05 |
| h         | 0.4450171  | 0.1086717 | 4.216227| 6.433059e-05 |
| i         | 0.3052415  | 0.1086717 | 4.216227| 6.433059e-05 |

5.4 Testing the dimensionality based upon an exploratory analysis

Alternatively, the output from an exploratory factor analysis can be used as input to the structure.sem function.

```r
> f1 <- fa(bifact)
> mod.f1 <- structure.sem(f1)
> sem.f1 <- sem(mod.f1,bifact,100)
> sem.f1

Model Chisquare = 21.16848 Df = 27

V1   V2   V3   V4   V5   V6   V7   V8   V9
0.7507129 0.6726412 0.5812000 0.5737425 0.5021915 0.4239908 0.5067957 0.4450171 0.3052415
x1e x2e x3e x4e x5e x6e x7e x8e x9e
0.4364302 0.5475539 0.6622982 0.6708197 0.7478036 0.8202314 0.7431581 0.8019593 0.9068284
```
The answers are, of course, identical.

5.5 Specifying a three factor model

An alternative model is to extract three factors and try this solution. The \texttt{fa} factor analysis function (using the \textit{minimum residual} algorithm) is used to detect the structure. Alternatively, the \texttt{factanal} could have been used. Rather than use the default rotation of \texttt{oblimin}, we force an orthogonal solution (even though we know it will be a poor solution).

\begin{verbatim}
> f3 <-fa(bifact,3,rotate="varimax")
> mod.f3 <- structure.sem(f3)
> sem.f3 <- sem(mod.f3,bifact,100)
> summary(sem.f3,digits=2)

Model Chisquare = 53.86635 Df = 27 Pr(>Chisq) = 0.001579738
AIC = 89.86635
BIC = -70.47325

Normalized Residuals
Min. 1st Qu. Median Mean 3rd Qu. Max.
-0.000003 0.000000 1.950175 1.642171 2.632737 4.011789

R-square for Endogenous Variables
V1 V2 V3 V4 V5 V6 V7 V8 V9
0.64 0.49 0.36 0.49 0.36 0.25 0.49 0.36 0.16

Parameter Estimates
                     Estimate Std Error z value Pr(>|z|)
F1V1 0.8000000 0.1114517 7.177994 7.074151e-13 V1 <--- MR1
F1V2 0.7000001 0.1089845 6.422931 1.336754e-10 V2 <--- MR1
F1V3 0.6500000 0.1068002 5.579685 1.193216e-08 V3 <--- MR1
F2V4 0.6999999 0.1427544 4.903527 9.413091e-07 V4 <--- MR3
F2V5 0.6000001 0.1328610 4.515998 6.301927e-06 V5 <--- MR3
F2V6 0.4999995 0.1680059 4.166522 3.092827e-05 V6 <--- MR3
F3V7 0.7000000 0.1680059 4.166522 3.092827e-05 V7 <--- MR2
F3V8 0.6000000 0.1530271 3.920873 8.822871e-05 V8 <--- MR2
F3V9 0.4000000 0.1265777 3.160368 1.575701e-03 V9 <--- MR2
x1e 0.3600000 0.1294344 2.777407 0.5205146e-03 V1 <--- V1
x2e 0.5099999 0.1165643 4.375286 1.212834e-05 V2 <--- V2
x3e 0.6399999 0.1130156 5.662936 1.488043e-09 V3 <--- V3
x4e 0.5100000 0.1739239 2.932316 3.364404e-03 V4 <--- V4
x5e 0.6399998 0.1475345 4.337967 1.438068e-05 V5 <--- V5
x6e 0.7500007 0.1336788 5.610466 2.017821e-08 V6 <--- V6
x7e 0.5100000 0.2136118 2.387509 1.696298e-02 V7 <--- V7
x8e 0.6400001 0.1734024 3.690837 2.235172e-04 V8 <--- V8
x9e 0.8400000 0.1362332 6.165889 7.008420e-10 V9 <--- V9

Iterations = 24
> round(residuals(sem.f3),2)
\end{verbatim}
The residuals show serious problems with this model. Although the residuals within each of the three factors are zero, the residuals between groups are much too large.

5.6 Allowing for an oblique solution

The previous solution is clearly very bad. What would happen if the exploratory solution were allowed to have correlated (oblique) factors?

```r
> f3 <- fa(bifact,3) # extract three factors and do an oblique rotation
> mod.f3 <- structure.sem(f3) # create the sem model
> mod.f3 # show it
```

```
Path Parameter Value
[1,] "MR1->V1" "F1V1" NA
[2,] "MR1->V2" "F1V2" NA
[3,] "MR1->V3" "F1V3" NA
[4,] "MR3->V4" "F2V4" NA
[5,] "MR3->V5" "F2V5" NA
[6,] "MR3->V6" "F2V6" NA
[7,] "MR2->V7" "F3V7" NA
[8,] "MR2->V8" "F3V8" NA
[9,] "MR2->V9" "F3V9" NA
[10,] "V1<->V1" "x1e" NA
[11,] "V2<->V2" "x2e" NA
[12,] "V3<->V3" "x3e" NA
[13,] "V4<->V4" "x4e" NA
[14,] "V5<->V5" "x5e" NA
[15,] "V6<->V6" "x6e" NA
[16,] "V7<->V7" "x7e" NA
[17,] "V8<->V8" "x8e" NA
[18,] "V9<->V9" "x9e" NA
[19,] "MR3<->MR1" "rF2F1" NA
[20,] "MR2<->MR1" "rF3F1" NA
[21,] "MR2<->MR3" "rF3F2" NA
[22,] "MR1<->MR1" NA "1"
[23,] "MR3<->MR3" NA "1"
[24,] "MR2<->MR2" NA "1"
attr(,"class")
[1] "mod"
```

The structure being tested may be seen using `structure.graph`

This makes much better sense, and in fact (as hoped) recovers the original structure.
Figure 13: A three factor, oblique solution.
5.7 Extract a bifactor solution using omega and then test that model using sem

A bifactor solution has previously been shown (Figure 9). The output from the omega function includes the sem commands for the analysis. As an example of doing this with real rather than simulated data, consider 9 variables from Thurstone. For completeness, the stdCoef from sem is used as well as the summary function.

5.7.1 sem of Thurstone 9 variable problem

The sem manual includes an example of a hierarchical solution to 9 mental abilities originally reported by Thurstone and used in the SAS manual for PROC CALIS and discussed in detail by McDonald (1999). The data matrix, as reported by Fox may be found in the Thurstone data set (which is “lazy loaded”). Using the commands just shown, it is possible to analyze this data set using a bifactor solution (Figure 14).

```r
> sem.bi <- sem(om.th.bi$model$sem,Thurstone,213) #use the model created by omega
> summary(sem.bi,digits=2)

Model Chisquare =  24.2163  Df  = 18  Pr(>Chisq) = 0.1480685
AIC = 78.2163
BIC = -72.28696

Normalized Residuals
Min. 1st Qu. Median Mean 3rd Qu. Max.
-0.8211748 -0.3341044 -0.0000009 0.0281693 0.1561983 1.7968088

R-square for Endogenous Variables

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R-square</td>
<td>0.8276</td>
<td>0.8302</td>
<td>0.7315</td>
<td>0.7472</td>
<td>0.6126</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.4824</td>
<td>0.8503</td>
<td>0.4996</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Parameter Estimates

|               | Estimate | Std Error | z value | Pr(|z|) |
|---------------|----------|-----------|---------|--------|
| Sentences     | 0.7678671| 0.0705939| 10.87723| 1.479833e-27 |
| Vocabulary    | 0.7909248| 0.0696923| 11.34880| 7.518003e-30 |
| Sent.Completion| 0.7536211 | 0.0711322 | 10.59403 | 3.164903e-26 |
| First.Letters | 0.6083814| 0.0706384| 8.612613 | 7.141338e-18 |
| Four.Letter.Words | 0.5973949 | 0.0709294 | 8.412546 | 3.715499e-17 |
| Suffixes      | 0.5717903| 0.0715775| 7.988405 | 1.366950e-15 |
| Letter.Series | 0.5668949| 0.0724934| 7.819952 | 5.284337e-15 |
| Pedigrees     | 0.6623314| 0.0700303| 9.457775 | 3.145633e-21 |
| Letter.Group  | 0.5299524| 0.0732494| 7.227450 | 4.921470e-13 |
| F1*Sentences  | 0.4878698| 0.0814109| 5.992680 | 2.064107e-09 |
| F1*Vocabulary | 0.4523234| 0.0853995| 5.414452 | 6.147524e-08 |
| F1*Sent.Completion | 0.4044507 | 0.0872734 | 4.634298 | 3.581494e-06 |
| F2*First.Letters | 0.6140531 | 0.0847114 | 7.245723 | 4.205973e-13 |
| F2*Four.Letter.Words | 0.5058663 | 0.0814588 | 6.209650 | 5.310276e-10 |
| F2*Suffixes    | 0.3943208| 0.0780538| 5.051907 | 4.374195e-07 |
| F3*Letter.Series | 0.7272955 | 0.1584486 | 4.590101 | 4.430304e-06 |
| F3*Pedigrees   | 0.2468417| 0.0867756| 2.844605 | 4.446649e-03 |
| F3*Letter.Group | 0.4091495 | 0.1135238 | 3.608854 | 3.132541e-04 |
| e1            | 0.1723633| 0.0340564| 5.061104 | 4.168346e-07 |
```
Figure 14: A bifactor solution to the Thurstone 9 variable problem. All items load on a general factor of ability, the residual factors account for the correlations between items within groups.
Sentences 0.1698419 0.03001233 5.6590697 1.521958e-08
Vocabulary 0.2684749 0.03316228 8.0957909 5.689350e-16
Sent.Completion 0.2528108 0.07942835 3.1828791 1.458185e-03
First.Letters 0.3873510 0.05955079 8.6912013 3.586269e-18
Four.Letter.Words 0.1496709 0.21861502 0.6846325 4.935759e-01
Suffixes 0.5003855 0.05956551 8.4005902 4.442400e-17
Letter.Series 0.5517474 0.08455914 6.5249884 6.800680e-11

Sentences Sentences <--- g
Vocabulary Vocabulary <--- g
Sent.Completion Sent.Completion <--- g
First.Letters First.Letters <--- g
Four.Letter.Words Four.Letter.Words <--- g
Suffixes Suffixes <--- g
Letter.Series Letter.Series <--- g
Pedigrees Pedigrees <--- g
Letter.Group Letter.Group <--- g
F1*Sentences Sentences <--- F1*
F1*Vocabulary Vocabulary <--- F1*
F1*Sent.Completion Sent.Completion <--- F1*
F2*First.Letters First.Letters <--- F2*
F2*Four.Letter.Words Four.Letter.Words <--- F2*
F2*Suffixes Suffixes <--- F2*
F3*Letter.Series Letter.Series <--- F3*
F3*Pedigrees Pedigrees <--- F3*
F3*Letter.Group Letter.Group <--- F3*
e1 Sentences <--- Sentences
e2 Vocabulary <--- Vocabulary
e3 Sent.Completion <--- Sent.Completion
e4 First.Letters <--- First.Letters
e5 Four.Letter.Words <--- Four.Letter.Words
e6 Suffixes <--- Suffixes
e7 Letter.Series <--- Letter.Series
e8 Pedigrees <--- Pedigrees
e9 Letter.Group <--- Letter.Group

Iterations = 72
> stdCoef(sem.bi,digits=2)

<table>
<thead>
<tr>
<th>Std. Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Sentences 0.7678671</td>
</tr>
<tr>
<td>2 Vocabulary 0.7909246</td>
</tr>
<tr>
<td>3 Sent.Completion 0.7536211</td>
</tr>
<tr>
<td>4 First.Letters 0.6083814</td>
</tr>
<tr>
<td>5 Four.Letter.Words 0.5973349</td>
</tr>
<tr>
<td>6 Suffixes 0.5717900</td>
</tr>
<tr>
<td>7 Letter.Series 0.5668950</td>
</tr>
<tr>
<td>8 Pedigrees 0.6623317</td>
</tr>
<tr>
<td>9 Letter.Group 0.5299523</td>
</tr>
<tr>
<td>10 F1*Sentences 0.4878697</td>
</tr>
<tr>
<td>11 F1*Vocabulary 0.4523233</td>
</tr>
<tr>
<td>12 F1*Sent.Completion 0.4044507</td>
</tr>
<tr>
<td>13 F2*First.Letters 0.6105831</td>
</tr>
<tr>
<td>14 F2*Four.Letter.Words 0.5058063</td>
</tr>
<tr>
<td>15 F2*Suffixes 0.3943206</td>
</tr>
<tr>
<td>16 F3*Letter.Series 0.7272967</td>
</tr>
</tbody>
</table>
5.8 Examining a hierarchical solution

A hierarchical solution to this data set was previously found by the omega function (Figure 10). The output of that analysis can be used as a model for a sem analysis. Once again, the stdCoef function helps see the structure. Alternatively, using the omega function on the Thurstone data will create the model for this particular data set.

\[ \text{> sem.hi <- sem(om.hi$model$sem, Thurstone, 213)} \]
\[ \text{> summary(sem.hi, digits=2)} \]

Model Chisquare = 38.1963  Df = 24  Pr(>Chisq) = 0.03310059
AIC = 80.1963
BIC = -90.47471

Normalized Residuals

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized</td>
<td>-0.9724643</td>
<td>-0.4164673</td>
<td>-0.0000001</td>
<td>0.0401007</td>
<td>0.0938588</td>
<td>1.6274666</td>
</tr>
</tbody>
</table>

R-square for Endogenous Variables

<table>
<thead>
<tr>
<th>Variables</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>Sentences</th>
<th>Vocabulary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pedigrees</td>
<td>0.6758</td>
<td>0.6112</td>
<td>0.6642</td>
<td>0.8185</td>
<td>0.8351</td>
</tr>
<tr>
<td>F1Sentences</td>
<td>0.7329</td>
<td></td>
<td></td>
<td>0.6355</td>
<td>0.4936</td>
</tr>
<tr>
<td>F1Vocabulary</td>
<td></td>
<td>0.5186</td>
<td></td>
<td>0.6097</td>
<td></td>
</tr>
</tbody>
</table>

Parameter Estimates

| Variables      | Estimate | Std Error | z value | Pr(>|z|) |
|----------------|----------|-----------|---------|---------|
| gF1            | 1.4438115 | 0.25653564 | 5.628113 | 1.821922e-08 |
| gF2            | 1.2538296 | 0.21136562 | 5.932041 | 2.991910e-09 |
| gF3            | 1.4065517 | 0.28990804 | 5.230605 | 1.689563e-07 |
| F1Sentences    | 0.5151232 | 0.06292248 | 8.186632 | 2.686376e-16 |
| F1Vocabulary   | 0.5203104 | 0.06338431 | 8.208820 | 2.233734e-16 |
| F1Sent.Completion | 0.4874316 | 0.06081528 | 8.104954 | 1.101786e-15 |
| F2First.Letters | 0.5211221 | 0.06106205 | 8.534304 | 4.10015e-17 |
| F2First.Letters | 0.4907664 | 0.05902388 | 8.414446 | 3.71866e-17 |
| F2Suffixes     | 0.4380644 | 0.05595794 | 7.828458 | 4.983915e-15 |
| F3Letter.Series | 0.4524352 | 0.06956903 | 6.858297 | 6.98649e-12 |

41
Figure 15: Hierarchical analysis of the Thurstone 9 variable problem using an exploratory algorithm can provide the appropriate sem code for analysis using the sem package.
| Item          | Estimate | Std. Error | t value | Pr(>|t|) |
|--------------|----------|------------|---------|---------|
| F3Pedigrees  | 0.4172903| 0.06215816 | 6.713363| 1.901887e-11 |
| F3Letter.Group| 0.4076312| 0.06131399 | 6.648258| 2.965820e-11 |
| e1           | 0.1814979| 0.02847741 | 6.373397| 1.848862e-10 |
| e2           | 0.1649304| 0.02776938 | 5.939292| 2.862558e-09 |
| e3           | 0.2671331| 0.03363630 | 8.006771| 1.177597e-15 |
| e4           | 0.3015024| 0.05102191 | 5.909274| 2.862558e-09 |
| e5           | 0.3645010| 0.05263547 | 6.925008| 4.359513e-12 |
| e6           | 0.5064150| 0.05933649 | 8.493180| 2.010593e-17 |
| e7           | 0.4813697| 0.06224844 | 7.733041| 1.513055e-14 |
| e8           | 0.5051017| 0.06332869 | 7.975875| 1.513055e-15 |

---

```r
> stdCoef(sem.hi, digits=2)
```

<table>
<thead>
<tr>
<th>Std. Estimate</th>
<th>Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 gF1 0.8220754 F1</td>
<td>Sentences &lt;---- gF1 Sentences</td>
<td></td>
</tr>
<tr>
<td>2 gF2 0.7817998 F2</td>
<td>Vocabulary &lt;---- gF2 Vocabulary</td>
<td></td>
</tr>
<tr>
<td>3 gF3 0.8150140 F3</td>
<td>Sentences &lt;---- gF3 Sentences</td>
<td></td>
</tr>
<tr>
<td>4 F1Sentences 0.9047111</td>
<td>Sent.Completion &lt;---- Sent.Completion</td>
<td></td>
</tr>
<tr>
<td>5 F1Vocabulary 0.9138214</td>
<td>First.Letters &lt;---- First.Letters</td>
<td></td>
</tr>
<tr>
<td>6 F1Sent.Completion 0.8560764</td>
<td>Four.Letter.Words &lt;---- Four.Letter.Words</td>
<td></td>
</tr>
<tr>
<td>7 F2First.Letters 0.8357617</td>
<td>First.Letters &lt;---- First.Letters</td>
<td></td>
</tr>
<tr>
<td>8 F2Four.Letter.Words 0.7971819</td>
<td>Suffixes &lt;---- Suffixes</td>
<td></td>
</tr>
<tr>
<td>9 F2Suffixes 0.7025560</td>
<td>Letter.Series &lt;---- Letter.Series</td>
<td></td>
</tr>
<tr>
<td>10 F3Letter.Series 0.7808129</td>
<td>Pedigrees &lt;---- F3 F3Pedigrees</td>
<td></td>
</tr>
<tr>
<td>11 F3Pedigrees 0.7201599</td>
<td>Pedigrees &lt;---- F3 F3Pedigrees</td>
<td></td>
</tr>
<tr>
<td>12 F3Letter.Group 0.7034902</td>
<td>Letter.Group &lt;---- F3 F3Letter.Group</td>
<td></td>
</tr>
<tr>
<td>13 e1 0.1814979</td>
<td>Sentences &lt;---- Sentences</td>
<td></td>
</tr>
<tr>
<td>14 e2 0.1649304</td>
<td>Vocabulary &lt;---- Vocabulary</td>
<td></td>
</tr>
<tr>
<td>15 e3 0.2671331</td>
<td>Sent.Completion &lt;---- Sent.Completion</td>
<td></td>
</tr>
<tr>
<td>16 e4 0.3015024</td>
<td>First.Letters &lt;---- First.Letters</td>
<td></td>
</tr>
<tr>
<td>17 e5 0.3645010</td>
<td>Four.Letter.Words &lt;---- Four.Letter.Words</td>
<td></td>
</tr>
<tr>
<td>18 e6 0.5064151</td>
<td>Suffixes &lt;---- Suffixes</td>
<td></td>
</tr>
<tr>
<td>19 e7 0.3903313</td>
<td>Letter.Series &lt;---- Letter.Series</td>
<td></td>
</tr>
</tbody>
</table>
Using the Thurstone data set, we see what happens when a hierarchical model is applied to real data. The exploratory structure derived from the omega function (Figure 15) provides estimates in close approximation to those found using sem. The model definition created by using omega is the same hierarchical model discussed in the sem help page. The bifactor model, with 6 more parameters does provide a better fit to the data than the hierarchical model.

Similar analyses can be done with other data that are organized hierarchically. Examples of these analyses are analyzing the 14 variables of holzinger and the 16 variables of reise. The output from the following analyses has been limited to just the comparison between the bifactor and hierarchical solutions.

**5.9 Estimating Omega using EFA followed by CFA**

The function omegaSem combines both an exploratory factor analysis using omega, then calls the appropriate sem functions and organizes the results as in a standard omega analysis.

An example is found from the Thurstone data set of 9 cognitive variables:

```r
> om_sem <- omegaSem(Thurstone,n.obs=213)
```
Call: omegaSem(m = Thurstone, n.obs = 213)

Omega
Call: omegah(m = m, nfactors = nfactors, fm = fm, key = key, flip = flip,
digits = digits, title = title, sl = sl, labels = labels,
plot = plot, n.obs = n.obs, rotate = rotate, Phi = Phi, option = option,
covar = covar)

Alpha: 0.89
G.6: 0.91
Omega Hierarchical: 0.74
Omega H asymptotic: 0.79
Omega Total 0.93

Schmid Leiman Factor loadings greater than 0.2

g            F1* F2* F3* h2  u2  p2
Sentences       0.71  0.56 0.82 0.18 0.61
Vocabulary      0.73  0.55 0.84 0.16 0.63
Sent.Completion 0.68  0.52 0.74 0.26 0.63
First.Leters    0.65  0.56 0.73 0.27 0.57
Four.Letter.Words 0.62  0.49 0.63 0.37 0.61
Suffixes        0.56  0.41 0.50 0.50 0.63
Letter.Series   0.59  0.62 0.73 0.27 0.48
Pedigrees       0.58  0.24 0.34 0.51 0.49 0.66
Letter.Group    0.54  0.46 0.52 0.48 0.56

With eigenvalues of:
g         F1* F2* F3*
3.58  0.96 0.74 0.72

general/max 3.73 max/min = 1.34
mean percent general = 0.6 with sd = 0.05 and cv of 0.09

Explained Common Variance of the general factor = 0.6

The degrees of freedom are 12 and the fit is 0.01
The number of observations was 213 with Chi Square = 2.98 with prob < 1
The root mean square of the residuals is 0.01
The df corrected root mean square of the residuals is 0.01
RMSEA index = 0 and the 10 % confidence intervals are 0 0
BIC = -61.36

Compare this with the adequacy of just a general factor and no group factors
The degrees of freedom for just the general factor are 27 and the fit is 1.48
The number of observations was 213 with Chi Square = 306.71 with prob < 3.3e-49
The root mean square of the residuals is 0.14
The df corrected root mean square of the residuals is 0.16
RMSEA index = 0.22 and the 10 % confidence intervals are 0.199 0.244
BIC = 161.96

Measures of factor score adequacy

g            F1* F2* F3*
Correlation of scores with factors 0.86 0.73 0.72 0.75
Multiple R square of scores with factors 0.74 0.54 0.51 0.57
Minimum correlation of factor score estimates 0.49 0.07 0.03 0.13

Total, General and Subset omega for each subset

g            F1* F2* F3*
Omega total for total scores and subscales 0.93 0.92 0.83 0.79
Omega general for total scores and subscales 0.74 0.58 0.50 0.47
Omega group for total scores and subscales 0.16 0.34 0.32 0.32

The following analyses were done using the lavaan package

Omega Hierarchical from a confirmatory model using sem = 0.79
Omega Total from a confirmatory model using sem = 0.93

With loadings of:

<table>
<thead>
<tr>
<th></th>
<th>g</th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
<th>h2</th>
<th>u2</th>
<th>p2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentences</td>
<td>0.77</td>
<td>0.49</td>
<td></td>
<td></td>
<td>0.82</td>
<td>0.18</td>
<td>0.72</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>0.79</td>
<td>0.45</td>
<td></td>
<td></td>
<td>0.83</td>
<td>0.17</td>
<td>0.75</td>
</tr>
<tr>
<td>Sent.Completion</td>
<td>0.75</td>
<td>0.40</td>
<td></td>
<td></td>
<td>0.73</td>
<td>0.27</td>
<td>0.77</td>
</tr>
<tr>
<td>First.Letters</td>
<td>0.61</td>
<td>0.61</td>
<td></td>
<td></td>
<td>0.74</td>
<td>0.26</td>
<td>0.50</td>
</tr>
<tr>
<td>Four.Letter.Words</td>
<td>0.60</td>
<td>0.50</td>
<td></td>
<td></td>
<td>0.61</td>
<td>0.39</td>
<td>0.59</td>
</tr>
<tr>
<td>Suffixes</td>
<td>0.57</td>
<td>0.39</td>
<td></td>
<td></td>
<td>0.48</td>
<td>0.52</td>
<td>0.68</td>
</tr>
<tr>
<td>Letter.Series</td>
<td>0.57</td>
<td>0.39</td>
<td></td>
<td></td>
<td>0.73</td>
<td>0.85</td>
<td>0.15</td>
</tr>
<tr>
<td>Pedigrees</td>
<td>0.66</td>
<td></td>
<td></td>
<td></td>
<td>0.25</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>Letter.Group</td>
<td>0.53</td>
<td></td>
<td></td>
<td></td>
<td>0.41</td>
<td>0.45</td>
<td>0.55</td>
</tr>
</tbody>
</table>

With eigenvalues of:

<table>
<thead>
<tr>
<th></th>
<th>g</th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3.86</td>
<td>0.60</td>
<td>0.78</td>
<td>0.75</td>
</tr>
</tbody>
</table>

The degrees of freedom of the confirmatory model are 18 and the fit is 24.33052 with p = 0.1444947
general/max = 4.92 max/min = 1.3
mean percent general = 0.65 with sd = 0.15 and cv of 0.23
Explained Common Variance of the general factor = 0.64

Measures of factor score adequacy

<table>
<thead>
<tr>
<th></th>
<th>g</th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation of scores with factors</td>
<td>0.90</td>
<td>0.68</td>
<td>0.80</td>
<td>0.85</td>
</tr>
<tr>
<td>Multiple R square of scores with factors</td>
<td>0.81</td>
<td>0.46</td>
<td>0.63</td>
<td>0.73</td>
</tr>
<tr>
<td>Minimum correlation of factor score estimates</td>
<td>0.61</td>
<td>-0.08</td>
<td>0.27</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Total, General and Subset omega for each subset

<table>
<thead>
<tr>
<th></th>
<th>g</th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Omega total for total scores and subscales</td>
<td>0.93</td>
<td>0.92</td>
<td>0.82</td>
<td>0.80</td>
</tr>
<tr>
<td>Omega general for total scores and subscales</td>
<td>0.79</td>
<td>0.69</td>
<td>0.48</td>
<td>0.50</td>
</tr>
<tr>
<td>Omega group for total scores and subscales</td>
<td>0.14</td>
<td>0.23</td>
<td>0.35</td>
<td>0.31</td>
</tr>
</tbody>
</table>

To get the standard sem fit statistics, ask for summary on the fitted object

Comparing the two models graphically (Figure 16 with Figure 14 shows that while not identical, they are very similar. The sem version is basically a forced simple structure. Notice that the values of $\omega_h$ are not identical from the EFA and CFA models. The CFA solution yields higher values of $\omega_h$ because, by forcing a pure cluster solution (no cross loadings), the correlations between the factors is forced to be through the $g$ factor.

## 6 Summary and conclusion

The use of exploratory and confirmatory models for understanding real data structures is an important advance in psychological research. To understand these approaches it is helpful to try them first on “baby” data sets. To the extent that the models we use can be tested on simple, artificial examples, it is perhaps easier to practice their application.
Figure 16: Confirmatory Omega structure using omegaSem
The *psych* tools for simulating structural models and for specifying models are a useful supplement to the power of packages such as *sem*. The techniques that can be used on simulated data set can also be applied to real data sets.
> sessionInfo()

R version 4.0.2 (2020-06-22)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Catalina 10.15.6

Matrix products: default
BLAS: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib

Random number generation:
  RNG: Mersenne-Twister
  Normal: Inversion
  Sample: Rounding

locale:

attached base packages:
[1] stats graphics grDevices utils datasets methods base

other attached packages:
[1] GPArotation_2014.11-1 sem_3.1-11 psychTools_2.0.8 psych_2.0.8

loaded via a namespace (and not attached):
[1] Rcpp_1.0.5 splines_4.0.2 MASS_7.3-51.6 mnormt_2.0.1 statmod_1.4.34
[6] pbivnorm_0.6.0 arm_1.11-1 lattice_0.20-41 mi_1.0 minqa_1.2.4
[11] tools_4.0.2 matrixcalc_1.0-3 parallel_4.0.2 grid_4.0.2 tmvnsim_1.0-2
[16] nlme_3.1-148 coda_0.19-3 abind_1.4-5 lme4_1.1-23 lavaan_0.6-6
[21] Matrix_1.2-18 nloptr_1.2.2.2 compiler_4.0.2 boot_1.3-25 stats4_4.0.2
[26] foreign_0.8-80

References


