Using the psych package to generate and test structural models

William Revelle

February 8, 2011

Contents

1 The psych package 3
  1.1 Preface .......................................................... 3
  1.2 Creating and modeling structural relations ....................... 3

2 Functions for generating correlational matrices with a particular structure 3
  2.1 sim.congeneric .................................................. 4
  2.2 sim.hierarchical ................................................ 7
  2.3 sim.item and sim.circ .......................................... 8
  2.4 sim.structure .................................................. 8
    2.4.1 $\mathbf{f}_x$ is a vector implies a congeneric model ............... 11
    2.4.2 $\mathbf{f}_x$ is a matrix implies an independent factors model: .... 11
    2.4.3 $\mathbf{f}_x$ is a matrix and $\Phi \neq I$ is a correlated factors model . . 13
    2.4.4 $\mathbf{f}_x$ and $\mathbf{f}_y$ are matrices, and $\Phi \neq I$ represents their correlations . . 17
    2.4.5 A hierarchical structure among the latent predictors .......... 19

3 Exploratory functions for analyzing structure ........................................ 19
  3.1 Exploratory simple structure models .......................... 19
  3.2 Exploratory hierarchical models .................................. 25
    3.2.1 A bifactor solution ....................................... 26
    3.2.2 A hierarchical solution ................................... 27

4 Confirmatory models ......................................................... 29
  4.1 Using psych as a front end for the sem package ..................... 30
  4.2 Testing a congeneric model versus a tau equivalent model ......... 30
  4.3 Testing the dimensionality of a hierarchical data set by creating the model . 33
  4.4 Testing the dimensionality based upon an exploratory analysis ....... 35
4.5 Specifying a three factor model ............................................. 35
4.6 Allowing for an oblique solution ............................................ 37
4.7 Extract a bifactor solution using omega and then test that model using sem 39
  4.7.1 sem of Thurstone 9 variable problem ................................. 39
4.8 Examining a hierarchical solution ......................................... 44
4.9 Estimating Omega using EFA followed by CFA .......................... 49

5 Summary and conclusion .......................... 51
1 The psych package

1.1 Preface

The psych package (Revelle, 2010) 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 http://personality-project.org/r/overview.pdf and http://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). 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.

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.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 \( \sqrt{nvar - nfact} \) “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.

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)
> set.seed(42)
> tau <- sim.congeneric(loads = c(0.8, 0.8, 0.8, 0.8))
> tau.samp <- sim.congeneric(loads = c(0.8, 0.8, 0.8, 0.8), N = 100)
> 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(0.8, 0.8, 0.8, 0.8), N = 100,
+  short = FALSE)
> tau.samp

$\text{model (Population correlation matrix)}$

\begin{verbatim}
   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
\end{verbatim}

$r \text{ (Sample correlation matrix for sample size = 100 )}$

\begin{verbatim}
   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
\end{verbatim}

> 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.

```r
> cong <- sim.congeneric(N = 100)
> round(cong, 2)
   V1  V2  V3  V4
V1 1.00 0.57 0.53 0.46
V2 0.57 1.00 0.35 0.41
V3 0.53 0.35 1.00 0.43
V4 0.46 0.41 0.43 1.00
```

```r
> 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(0.9, 0.8, 0.7), nrow = 3)
> fload <- matrix(c(0.9, 0.8, 0.7, rep(0, 9), 0.7, 0.6, 0.5, rep(0,
+ 9), 0.6, 0.5, 0.4), ncol = 3)
> fload
     [,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
> bifact <- sim.hierarchical(gload = gload, fload = fload)
> round(bifact, 2)
```

7
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, $\{x, y\}$, $\{x, y\}$, $\{x, y\}$ which describe, in turn, a measurement model of x variables, $\{x, y\}$, a measurement model of y variables, $\{x, y\}$, and a covariance matrix between and within the two sets of factors. If $\{x, y\}$ is a vector and $\{x, y\}$ and $\{x, y\}$ are NULL, then this is just the congeneric model. If $\{x, y\}$ is a matrix of loadings with n rows and c columns, then this is a measurement model for n variables across c factors. If $\{x, y\}$ is not null, but $\{x, y\}$ is NULL, then the factors in $\{x, y\}$ are correlated. Finally, if all
> op <- par(mfrow = c(1, 2))
> m.bi <- omega(bifact, title = "A bifactor model")
> m.hi <- omega(bifact, sl = FALSE, title = "A hierarchical model")
> op <- par(mfrow = c(1, 1))

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.
three matrices are not NULL, then the data show the standard linear structural relations (LISREL) structure.

Consider the following examples:

2.4.1 $\vec{f}_X$ is a vector implies a congeneric model

```r
> set.seed(42)
> fx <- c(0.9, 0.8, 0.7, 0.6)
> cong1 <- sim.structure(fx)
> cong1
Call: sim.structure(fx = fx)

$model (Population correlation matrix)
  V1  V2  V3  V4
V1 1.00 0.72 0.63 0.54
V2 0.72 1.00 0.56 0.48
V3 0.63 0.56 1.00 0.42
V4 0.54 0.48 0.42 1.00

$reliability (population reliability)
[1] 0.81 0.64 0.49 0.36
```

2.4.2 $\vec{f}_X$ is a matrix implies an independent factors model:

```r
> set.seed(42)
> fx <- matrix(c(0.9, 0.8, 0.7, rep(0, 9), 0.7, 0.6, 0.5, rep(0, + 9), 0.6, 0.5, 0.4), 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 0.0 0.00
V2 0.72 1.00 0.56 0.00 0.00 0.00 0.00 0.0 0.00
V3 0.63 0.56 1.00 0.00 0.00 0.00 0.00 0.0 0.00
V4 0.00 0.00 0.00 1.00 0.42 0.35 0.00 0.0 0.00
V5 0.00 0.00 0.00 0.42 1.00 0.30 0.00 0.0 0.00
```

11
$\text{reliability (population reliability)}$

\begin{align*}
[1] & \quad 0.81 \quad 0.64 \quad 0.49 \quad 0.49 \quad 0.36 \quad 0.25 \quad 0.36 \quad 0.25 \quad 0.16 \\
\end{align*}

Figure 4: Three uncorrelated factors generated using the sim.structure function and drawn using structure.diagram.
2.4.3 $\vec{f}_x$ is a matrix and $\Phi \neq I$ is a correlated factors model

```r
> Phi = matrix(c(1, 0.5, 0.3, 0.5, 1, 0.2, 0.3, 0.2, 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
Call: sim.structure(fx = fx, Phi = Phi)

$model (Population correlation matrix)
   V1 V2 V3 V4 V5 V6 V7 V8 V9
V1 1.00 0.720 0.315 0.270 0.23 0.162 0.14 0.108
V2 0.72 1.000 0.280 0.240 0.20 0.144 0.12 0.096
V3 0.31 0.280 1.000 0.245 0.210 0.17 0.126 0.10 0.084
V4 0.27 0.210 0.245 1.000 0.420 0.35 0.072 0.06 0.048
V5 0.23 0.350 0.175 0.350 1.000 0.060 0.05 0.040
V6 0.16 0.144 0.126 0.084 0.072 0.06 1.000 0.30 0.240
V7 0.14 0.120 0.105 0.070 0.060 0.05 0.30 1.00 0.200
V8 0.11 0.096 0.084 0.056 0.048 0.04 0.240 0.20 1.000

$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).

```r
> 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
   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
   F1  F2  F3
F1 "1" "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.

Drawing path models from Exploratory Factor Analysis solutions  Alternatively,
this result can represent the estimated factor loadings and oblique correlations found us-
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))
```
> corf3.mod <- structure.diagram(fxs, Phis)

Structural model

Figure 5: Three correlated factors with symbolic paths. Created using structure.diagram and structure.list and phi.list for ease of input.
> mod.f3p <- structure.diagram(f3.p, cut = 0.2)

```
Structural model

V1
V2
V3
V4
V5
V6
V7
V8
V9
MR3
0.7
0.6
0.5
MR1
1
0.4
MR2
0.9
```

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  \( \vec{f}_x \) and \( \vec{f}_y \) are matrices, and \( \Phi \neq I \) represents their correlations

A more complicated model is when there is a \( \vec{f}_y \) 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(0.9, 0.8, 0.7, rep(0, 9), 0.7, 0.6, 0.5, rep(0,
+   9), 0.6, 0.5, 0.4), ncol = 3)
> fy <- c(0.6, 0.5, 0.4)
> Phi <- matrix(c(1, 0.48, 0.32, 0.4, 0.48, 1, 0.32, 0.3, 0.32,
+   0.32, 1, 0.2, 0.4, 0.3, 0.2, 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")
```
Figure 7: A symbolic structural model. Three independent latent variables are regressed on a latent $Y$. 
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.

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 based upon correlation matrix
   PC1     h2     u2
V1 0.89 0.80 0.20
V2 0.85 0.73 0.27
V3 0.80 0.64 0.36
V4 0.73 0.53 0.47

   PC1
SS loadings 2.69
Proportion Var 0.67

Test of the hypothesis that 1 factor is sufficient.
> 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"
> hi.mod <- structure.diagram(fxh, Phi, fy)

Structural model

![Structural model diagram]

Figure 8: A symbolic structural model with a general factor and three group factors. The general factor is regressed on the latent Y variable.
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 0.14

Fit based upon off diagonal values = 0.96
> fa(cong1$model)

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

   MR1
SS loadings  2.30
Proportion Var 0.57

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 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>MR1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation of scores with factors</td>
<td>0.94</td>
</tr>
<tr>
<td>Multiple R square of scores with factors</td>
<td>0.88</td>
</tr>
<tr>
<td>Minimum correlation of possible factor scores</td>
<td>0.77</td>
</tr>
</tbody>
</table>

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.
> 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 based upon correlation matrix

<table>
<thead>
<tr>
<th></th>
<th>RC1</th>
<th>RC3</th>
<th>RC2</th>
<th>h2</th>
<th>u2</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>0.82</td>
<td>0.29</td>
<td>0.22</td>
<td>0.80</td>
<td>0.20</td>
</tr>
<tr>
<td>V2</td>
<td>0.82</td>
<td>0.23</td>
<td>0.18</td>
<td>0.75</td>
<td>0.25</td>
</tr>
<tr>
<td>V3</td>
<td>0.82</td>
<td>0.15</td>
<td>0.12</td>
<td>0.71</td>
<td>0.29</td>
</tr>
<tr>
<td>V4</td>
<td>0.32</td>
<td>0.68</td>
<td>0.13</td>
<td>0.58</td>
<td>0.42</td>
</tr>
<tr>
<td>V5</td>
<td>0.22</td>
<td>0.70</td>
<td>0.10</td>
<td>0.56</td>
<td>0.44</td>
</tr>
<tr>
<td>V6</td>
<td>0.08</td>
<td>0.77</td>
<td>0.08</td>
<td>0.60</td>
<td>0.40</td>
</tr>
<tr>
<td>V7</td>
<td>0.24</td>
<td>0.12</td>
<td>0.66</td>
<td>0.51</td>
<td>0.49</td>
</tr>
<tr>
<td>V8</td>
<td>0.16</td>
<td>0.08</td>
<td>0.68</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>V9</td>
<td>0.03</td>
<td>0.08</td>
<td>0.71</td>
<td>0.51</td>
<td>0.49</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>2.25</td>
<td>1.73</td>
<td>1.53</td>
</tr>
<tr>
<td>Proportion Var</td>
<td>0.25</td>
<td>0.19</td>
<td>0.17</td>
</tr>
<tr>
<td>Cumulative Var</td>
<td>0.25</td>
<td>0.44</td>
<td>0.61</td>
</tr>
</tbody>
</table>

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the null model are 36 and the objective function was 2.37
The degrees of freedom for the model are 12 and the objective function was 0.71

Fit based upon off diagonal values = 0.9

> pa3

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

<table>
<thead>
<tr>
<th></th>
<th>PA1</th>
<th>PA3</th>
<th>PA2</th>
<th>h2</th>
<th>u2</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>0.9</td>
<td>0.00</td>
<td>0.00</td>
<td>0.81</td>
<td>0.19</td>
</tr>
<tr>
<td>V2</td>
<td>0.8</td>
<td>0.00</td>
<td>0.00</td>
<td>0.64</td>
<td>0.36</td>
</tr>
<tr>
<td>V3</td>
<td>0.7</td>
<td>0.00</td>
<td>0.00</td>
<td>0.49</td>
<td>0.51</td>
</tr>
<tr>
<td>V4</td>
<td>0.0</td>
<td>0.7</td>
<td>0.00</td>
<td>0.49</td>
<td>0.51</td>
</tr>
<tr>
<td>V5</td>
<td>0.0</td>
<td>0.6</td>
<td>0.00</td>
<td>0.36</td>
<td>0.64</td>
</tr>
</tbody>
</table>
V6 0.0 0.5 0.00 0.25 0.75
V7 0.0 0.0 0.59 0.36 0.64
V8 0.0 0.0 0.50 0.25 0.75
V9 0.0 0.0 0.40 0.16 0.84

PA1  PA3  PA2
SS loadings  1.94 1.10 0.77
Proportion Var 0.22 0.12 0.09
Cumulative Var 0.22 0.34 0.42

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

Test of the hypothesis that 3 factors are sufficient.
The degrees of freedom for the null model are 36 and the objective function was 2.37
The degrees of freedom for the model are 12 and the objective function was 0
The root mean square of the residuals 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
   PA1  PA3  PA2
Correlation of scores with factors  0.94 0.86 0.79
Multiple R square of scores with factors  0.88 0.73 0.63
Minimum correlation of possible factor scores  0.77 0.47 0.25

> ml3
Factor Analysis using method = ml
Call: fa(r = bifact, nfactors = 3, fm = "ml")
Standardized loadings based upon correlation matrix
   ML1  ML2  ML3  h2  u2
V1  0.9  0.0  0.0  0.81 0.19
V2  0.8  0.0  0.0  0.64 0.36
V3  0.7  0.0  0.0  0.49 0.51
V4  0.0  0.7  0.0  0.49 0.51
V5  0.0  0.6  0.0  0.36 0.64

23
V6 0.0 0.5 0.0 0.25 0.75
V7 0.0 0.0 0.6 0.36 0.64
V8 0.0 0.0 0.5 0.25 0.75
V9 0.0 0.0 0.4 0.16 0.84

ML1  ML2  ML3
SS loadings 1.94 1.10 0.77
Proportion Var 0.22 0.12 0.09
Cumulative Var 0.22 0.34 0.42

With factor correlations of
ML1  ML2  ML3
ML1 1.00 0.72 0.63
ML2 0.72 1.00 0.56
ML3 0.63 0.56 1.00

Test of the hypothesis that 3 factors are sufficient.
The degrees of freedom for the null model are 36 and the objective function was 2.37
The degrees of freedom for the model are 12 and the objective function was 0
The root mean square of the residuals 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 scores with factors 0.94 0.86 0.79
Multiple R square of scores with factors 0.89 0.73 0.63
Minimum correlation of possible factor scores 0.77 0.47 0.25

> factor.congruence(list(pc3, pa3, ml3))

RC1  RC3  RC2  PA1  PA3  PA2  ML1  ML2  ML3
RC1 1.00 0.52 0.42 0.94 0.25 0.18 0.94 0.25 0.18
RC3 0.52 1.00 0.33 0.30 0.93 0.12 0.30 0.93 0.12
RC2 0.42 0.33 1.00 0.24 0.15 0.94 0.24 0.15 0.94
PA1 0.94 0.30 0.24 1.00 0.00 0.00 1.00 0.00 0.00
PA3 0.25 0.93 0.15 0.00 1.00 0.00 0.00 1.00 0.00
PA2 0.18 0.12 0.94 0.00 0.00 1.00 0.00 0.00 1.00
ML1 0.94 0.30 0.24 1.00 0.00 0.00 1.00 0.00 0.00
ML2 0.25 0.93 0.15 0.00 1.00 0.00 0.00 1.00 0.00

24
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.

```r
> ml3p <- Promax(ml3)
> ml3p

Call: NULL
Standardized loadings based upon correlation matrix
         ML1   ML2   ML3  h2   u2
V1   0.90  0.00  0.00 0.81 0.19
V2   0.80  0.00  0.00 0.64 0.36
V3   0.70  0.00  0.00 0.49 0.51
V4   0.00  0.70  0.00 0.49 0.51
V5   0.00  0.60  0.00 0.36 0.64
V6   0.00  0.50  0.00 0.25 0.75
V7   0.00  0.00  0.60 0.36 0.64
V8   0.00  0.00  0.50 0.25 0.75
V9   0.00  0.00  0.40 0.16 0.84

ML1   ML2   ML3
SS loadings 1.94 1.10 0.77
Proportion Var 0.22 0.12 0.09
Cumulative Var 0.22 0.34 0.42

With factor correlations of
         ML1   ML2   ML3
ML1  1.00  0.00  0.00
ML2  0.00  1.00  0.00
ML3  0.00  0.00  1.00

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 \texttt{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

```r
> om.bi <- omega(bifact)
```

![Diagram of bifactor solution](image)

**Figure 9: An exploratory bifactor solution to the nine variable problem**

The \textit{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 \texttt{bifactor} data include the original (Holzinger and Swineford, 1937) data set (\texttt{holzinger}) as well as a set from Reise et al. (2007) (\texttt{reise}) and a nine variable problem from Thurstone.
3.2.2 A hierarchical solution

> om.hi <- omega(bifact, sl = FALSE)

Both of these graphical representations are reflected in the output of the \texttt{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 \texttt{sem} analysis. It is also useful to examine the estimates of reliability reported by \texttt{omega}.

> om.bi

Omega

Call: omega(m = bifact)

Alpha: 0.79
G.6: 0.79
Omega Hierarchical: 0.69  
Omega H asymptotic: 0.84  
Omega Total: 0.82

Schmid Leiman Factor loadings greater than 0.2

\[ \begin{array}{cccccc}
g & F1^* & F2^* & F3^* & h2 & u2 & p2 \\
V1 & 0.90 & 0.40 & 0.98 & 0.02 & 0.84 \\
V2 & 0.80 & 0.32 & 0.73 & 0.27 & 0.86 \\
V3 & 0.65 & & 0.46 & 0.54 & 0.93 \\
V4 & 0.50 & 0.47 & 0.47 & 0.53 & 0.53 \\
V5 & 0.43 & 0.42 & 0.36 & 0.64 & 0.51 \\
V6 & 0.36 & 0.35 & 0.25 & 0.75 & 0.51 \\
V7 & 0.37 & & 0.17 & 0.83 & 0.82 \\
V8 & 0.32 & & 0.12 & 0.88 & 0.81 \\
V9 & 0.25 & & & 0.08 & 0.92 & 0.80 \\
\end{array} \]

With eigenvalues of:

\[ \begin{array}{ccc}
g & F1^* & F2^* \\
2.74 & 0.13 & 0.58 \\
\end{array} \]

general/max 4.73 max/min = 4.32
mean percent general = 0.73 with sd = 0.17 and cv of 0.23

The degrees of freedom are 12 and the fit is 0.07
The root mean square of the residuals is 0.03
The df corrected root mean square of the residuals is 0.07

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 0.21
The root mean square of the residuals is 0.05
The df corrected root mean square of the residuals is 0.08

Measures of factor score adequacy

\[ \begin{array}{cccc}
g & F1^* & F2^* & F3^* \\
\text{Correlation of scores with factors} & 0.94 & 0.47 & 0.67 \\
\text{Multiple R square of scores with factors} & 0.88 & 0.22 & 0.45 \\
\text{Minimum correlation of factor score estimates} & 0.75 & -0.55 & -0.11 \\
\end{array} \]

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 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, 2009).
4.1 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.

4.2 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 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
Path Parameter StartValue
1 X1->V1 a
2 X1->V2 a
3 X1->V3 a
4 X1->V4 a
5 V1<->V1 x1e
6 V2<->V2 x2e
7 V3<->V3 x3e
8 V4<->V4 x4e
9 X1<->X1 <fixed> 1
```

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

Model Chisquare =  6.6  Df =  5  Pr(>Chisq) = 0.25
Chisquare (null model) =  105  Df =  6

30
Goodness-of-fit index = 0.97  
Adjusted goodness-of-fit index = 0.94  
RMSEA index = 0.057  90% CI: (NA, 0.16)  
Bentler-Bonnett NFI = 0.94  
Tucker-Lewis NNFI = 0.98  
Bentler CFI = 0.98  
SRMR = 0.07  
BIC = -16

Normalized Residuals  
Min. 1st Qu. Median Mean 3rd Qu. Max.  
-1.03 -0.44 -0.25 -0.08 0.53 0.89

Parameter Estimates  
Estimate Std Error z value Pr(>|z|)  
a 0.69 0.064 10.8 0.0e+00 V1 <--- X1  
x1e 0.43 0.082 5.2 1.8e-07 V1 <--> V1  
x2e 0.56 0.098 5.7 1.5e-08 V2 <--> V2  
x3e 0.58 0.101 5.7 1.1e-08 V3 <--> V3  
x4e 0.59 0.103 5.8 8.3e-09 V4 <--> V4

Iterations = 10

Test whether the data are congeneric. That is, whether a one factor model fits. Compare this to the prior model using the \texttt{anova} function.

\begin{verbatim}
> mod.cong <- structure.sem(c("a", "b", "c", "d"), labels = paste("V", + 1:4, sep = ""))
> mod.cong

Path Parameter StartValue
1 X1->V1 a
2 X1->V2 b
3 X1->V3 c
4 X1->V4 d
5 V1<->V1 x1e
6 V2<->V2 x2e
7 V3<->V3 x3e
8 V4<->V4 x4e
9 X1<->X1 <fixed> 1

> sem.cong <- sem(mod.cong, cong, 100)
> summary(sem.cong, digits = 2)
\end{verbatim}
Model Chisquare = 2.9  Df = 2  Pr(>Chisq) = 0.23  
Chisquare (null model) = 105  Df = 6  
Goodness-of-fit index = 0.99  
Adjusted goodness-of-fit index = 0.93  
RMSEA index = 0.069  90% CI: (NA, 0.22)  
Bentler-Bonnett NFI = 0.97  
Tucker-Lewis NNFI = 0.97  
Bentler CFI = 1  
SRMR = 0.03  
BIC = -6.3  

Normalized Residuals  
Min. 1st Qu. Median Mean 3rd Qu. Max.  
-0.57 -0.07  0.03  0.01  0.16  0.54  

Parameter Estimates  
Estimate Std Error z value Pr(>|z|)  
a  0.83  0.098  8.4  0.0e+00  V1 <--- X1  
b  0.66  0.100  6.6  3.4e-11  V2 <--- X1  
c  0.63  0.102  6.2  6.4e-10  V3 <--- X1  
d  0.59  0.105  5.7  1.5e-08  V4 <--- X1  
x1e  0.31  0.101  3.1  2.1e-03  V1 <--> V1  
x2e  0.56  0.100  5.6  2.1e-08  V2 <--> V2  
x3e  0.61  0.104  5.8  6.5e-09  V3 <--> V3  
x4e  0.65  0.111  5.9  4.7e-09  V4 <--> V4  

Iterations = 12  
> anova(sem.cong, sem.tau)  
LR Test for Difference Between Models  

<table>
<thead>
<tr>
<th>Model</th>
<th>Df</th>
<th>Model Chisq</th>
<th>Df</th>
<th>LR Chisq</th>
<th>Pr(&gt;Chisq)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>2</td>
<td>2.9417</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model 2</td>
<td>5</td>
<td>6.5935</td>
<td>3</td>
<td>3.6518</td>
<td>0.3016</td>
</tr>
</tbody>
</table>

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.
4.3 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

Path  Parameter  StartValue
1  X1->V1 a
2  X1->V2 b
3  X1->V3 c
4  X1->V4 d
5  X1->V5 e
6  X1->V6 f
7  X1->V7 g
8  X1->V8 h
9  X1->V9 i
10 V1<->V1 x1e
11 V2<->V2 x2e
12 V3<->V3 x3e
13 V4<->V4 x4e
14 V5<->V5 x5e
15 V6<->V6 x6e
16 V7<->V7 x7e
17 V8<->V8 x8e
18 V9<->V9 x9e
19 X1<->X1 <fixed> 1

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

Model Chisquare = 19  Df = 27 Pr(>Chisq) = 0.88
Chisquare (null model) = 235  Df = 36
Goodness-of-fit index = 0.96

33
Adjusted goodness-of-fit index = 0.93
RMSEA index = 0 90% CI: (NA, 0.040)
Bentler-Bonnett NFI = 0.92
Tucker-Lewis NNFI = 1.1
Bentler CFI = 1
SRMR = 0.053
BIC = -106

Normalized Residuals
  Min. 1st Qu. Median Mean 3rd Qu. Max.
-0.27 -0.18  0.00  0.14  0.12  1.61

Parameter Estimates
    Estimate Std Error  z value Pr(>|z|)  
a  0.88     0.084   10.5   0.0e+00  V1 <--- X1
b  0.80     0.088    9.1   0.0e+00  V2 <--- X1
c  0.70     0.092    7.6  3.8e-14  V3 <--- X1
d  0.54     0.099    5.5  4.9e-08  V4 <--- X1
e  0.47     0.101    4.6  3.5e-06  V5 <--- X1
f  0.39     0.103    3.8  1.3e-04  V6 <--- X1
g  0.40     0.103    3.9  8.3e-05  V7 <--- X1
h  0.34     0.104    3.3  3.8e-05  V8 <--- X1
i  0.27     0.105    2.6  9.1e-03  V9 <--- X1
x1e 0.23    0.061    3.7  2.4e-04  V1 <--> V1
x2e 0.36    0.069    5.3  1.1e-07  V2 <--> V2
x3e 0.51    0.084    6.1  1.0e-09  V3 <--> V3
x4e 0.71    0.107    6.6  4.1e-11  V4 <--> V4
x5e 0.78    0.116    6.7  4.9e-11  V5 <--> V5
x6e 0.84    0.123    6.8  2.8e-12  V6 <--> V6
x7e 0.84    0.122    6.8  7.9e-12  V7 <--> V7
x8e 0.88    0.128    6.9  5.0e-12  V8 <--> V8
x9e 0.92    0.133    7.0  3.5e-12  V9 <--> V9

Iterations = 14

> round(residuals(sem.one), 2)

     V1  V2  V3  V4  V5  V6  V7  V8  V9
V1  0.00 0.02 0.02 -0.02 -0.02 -0.02 -0.02 -0.02 -0.01
V2  0.02 0.00 0.00 -0.03 -0.03 -0.03 -0.02 -0.02 -0.02
V3  0.02 0.00 0.00 -0.02 -0.03 -0.02 -0.02 -0.02 -0.02
V4 -0.02 -0.03 -0.02  0.00  0.17  0.14  0.02  0.01  0.01
4.4 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 <- factanal(covmat = bifact, factors = 1)
> mod.f1 <- structure.sem(f1)
> sem.f1 <- sem(mod.f1, bifact, 100)
> sem.f1

Model Chisquare = 18.72871 Df = 27

  V1   V2   V3   V4   V5   V6   V7   V8
0.8801449 0.7978613 0.6986695 0.5401625 0.4691098 0.3944311 0.4036073 0.3400459
  V9 x1e x2e x3e x4e x5e x6e x7e
0.2742160 0.2253461 0.3634188 0.5118600 0.7082243 0.7799344 0.8444243 0.8371012
     x8e x9e
0.8843691 0.9248059

Iterations = 14

The answers are, of course, identical.

4.5 Specifying a three factor model

An alternative model is to extract three factors and try this solution. The fa factor analysis function (using the minimum residual algorithm) is used to detect the structure. Alternatively, the factanal could have been used.

```r
> f3 <- fa(bifact, 3)
> mod.f3 <- structure.sem(f3)
> sem.f3 <- sem(mod.f3, bifact, 100)
> summary(sem.f3, digits = 2)

Model Chisquare = 38 Df = 27 Pr(>Chisq) = 0.071
Chisquare (null model) = 235 Df = 36
Goodness-of-fit index = 0.9
Adjusted goodness-of-fit index = 0.84
RMSEA index = 0.065  90% CI: (NA, 0.11)
Bentler-Bonnett NFI = 0.84
Tucker-Lewis NNFI = 0.92
Bentler CFI = 0.94
SRMR = 0.15
BIC = -86

Normalized Residuals

<table>
<thead>
<tr>
<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>0.0</td>
<td>0.0</td>
<td>1.1</td>
<td>1.1</td>
<td>2.0</td>
<td>3.4</td>
</tr>
</tbody>
</table>

Parameter Estimates

| Estimate | Std Error | z value | Pr(>|z|) |
|----------|-----------|---------|----------|
| F2V1     | 1.02      | NaN     | NaN      |
| F3V2     | 0.80      | 0.091   | 8.8      | 0.0e+00 |
| F3V3     | 0.70      | 0.094   | 7.5      | 7.1e-14 |
| F3V4     | 0.70      | 0.110   | 6.4      | 1.7e-10 |
| F3V5     | 0.60      | 0.110   | 5.5      | 4.3e-08 |
| F3V6     | 0.50      | 0.111   | 4.5      | 7.2e-06 |
| x1e      | -0.04     | NaN     | NaN      |
| x2e      | 0.36      | 0.080   | 4.5      | 7.1e-06 |
| x3e      | 0.51      | 0.087   | 5.9      | 3.8e-09 |
| x4e      | 0.51      | 0.118   | 4.3      | 1.5e-05 |
| x5e      | 0.64      | 0.117   | 5.5      | 4.3e-08 |
| x6e      | 0.75      | 0.122   | 6.1      | 8.6e-10 |
| x7e      | 1.00      | 0.142   | 7.0      | 2.0e-12 |
| x8e      | 1.00      | 0.142   | 7.0      | 2.0e-12 |
| x9e      | 1.00      | 0.142   | 7.0      | 2.0e-12 |
| rF2F1    | 0.64      | NaN     | NaN      |
| rF3F1    | 0.72      | 0.104   | 6.9      | 5.1e-12 |
| rF3F2    | 0.88      | NaN     | NaN      |

Iterations = 18

Aliased parameters: F2V1 x1e rF2F1 rF3F2

> round(residuals(sem.f3), 2)

<table>
<thead>
<tr>
<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>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.34</td>
<td>0.28</td>
<td>0.23</td>
<td></td>
</tr>
</tbody>
</table>

36
V2 0.00 0.00 0.00 0.00 0.00 0.00 0.30 0.25 0.20
V3 0.00 0.00 0.00 0.00 0.00 0.00 0.26 0.22 0.18
V4 0.00 0.00 0.00 0.00 0.00 0.00 0.24 0.20 0.16
V5 0.00 0.00 0.00 0.00 0.00 0.00 0.20 0.17 0.13
V6 0.00 0.00 0.00 0.00 0.00 0.00 0.17 0.14 0.11
V7 0.34 0.30 0.26 0.24 0.20 0.17 0.00 0.30 0.24
V8 0.28 0.25 0.22 0.20 0.17 0.14 0.30 0.00 0.20
V9 0.23 0.20 0.18 0.16 0.13 0.11 0.24 0.20 0.00

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.

4.6 Allowing for an oblique solution

That solution is clearly very bad. What would happen if the exploratory solution were allowed to have correlated (oblique) factors? This analysis is done on a sample of size 100 with the bifactor structure created by sim.hierarchical.

> set.seed(42)
> bifact.s <- sim.hierarchical()
> f3 <- fa(bifact.s, 3)
> mod.f3p <- structure.sem(f3.p)
> mod.f3p

<table>
<thead>
<tr>
<th>Path</th>
<th>Parameter</th>
<th>StartValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MR1-&gt;V1</td>
<td>F2V1</td>
</tr>
<tr>
<td>2</td>
<td>MR2-&gt;V2</td>
<td>F3V2</td>
</tr>
<tr>
<td>3</td>
<td>MR1-&gt;V3</td>
<td>F2V3</td>
</tr>
<tr>
<td>4</td>
<td>MR3-&gt;V4</td>
<td>F1V4</td>
</tr>
<tr>
<td>5</td>
<td>MR3-&gt;V5</td>
<td>F1V5</td>
</tr>
<tr>
<td>6</td>
<td>MR3-&gt;V6</td>
<td>F1V6</td>
</tr>
<tr>
<td>7</td>
<td>V1&lt;-&gt;V1</td>
<td>x1e</td>
</tr>
<tr>
<td>8</td>
<td>V2&lt;-&gt;V2</td>
<td>x2e</td>
</tr>
<tr>
<td>9</td>
<td>V3&lt;-&gt;V3</td>
<td>x3e</td>
</tr>
<tr>
<td>10</td>
<td>V4&lt;-&gt;V4</td>
<td>x4e</td>
</tr>
<tr>
<td>11</td>
<td>V5&lt;-&gt;V5</td>
<td>x5e</td>
</tr>
<tr>
<td>12</td>
<td>V6&lt;-&gt;V6</td>
<td>x6e</td>
</tr>
<tr>
<td>13</td>
<td>V7&lt;-&gt;V7</td>
<td>x7e</td>
</tr>
<tr>
<td>14</td>
<td>V8&lt;-&gt;V8</td>
<td>x8e</td>
</tr>
<tr>
<td>15</td>
<td>V9&lt;-&gt;V9</td>
<td>x9e</td>
</tr>
<tr>
<td>16</td>
<td>MR3&lt;-&gt;MR3</td>
<td>&lt;fixed&gt; 1</td>
</tr>
</tbody>
</table>
Unfortunately, the model as created automatically by `structure.sem` is not identified and would fail to converge if run. The problem is that the covariances between items on different factors is a product of the factor loadings and the between factor covariance. Multiplying the factor loadings by a constant can be compensated for by dividing the between factor covariances by the same constant. Thus, one of these paths must be fixed to provide a scale for the solution. That is, it is necessary to fix some of the paths to set values in order to properly identify the model. This can be done using the `edit` function and hand modification of particular paths. Set one path for each latent variable to be fixed.

e.g.,

```r
mod.adjusted <- edit(mod.f3p)
```

Alternatively, the model can be adjusted by specifying the changes directly.

When this is done

```r
> mod.f3p.adjusted <- mod.f3p
> mod.f3p.adjusted[c(1, 4), 2] <- NA
> mod.f3p.adjusted[c(1, 4), 3] <- "1"
> sem.f3p.adjusted <- sem(mod.f3p.adjusted, bifact.s, 100)
> summary(sem.f3p.adjusted, digits = 2)
```

```r
Model Chisquare = 111 Df = 32 Pr(>Chisq) = 1.2e-10
Chisquare (null model) = 169 Df = 36
Goodness-of-fit index = 0.75
Adjusted goodness-of-fit index = 0.65
RMSEA index = 0.16  90% CI: (NA, NA)
Bentler-Bonnett NFI = 0.34
Tucker-Lewis NNFI = 0.33
Bentler CFI = 0.41
SRMR = 0.22
BIC = -36

Normalized Residuals
  Min. 1st Qu.  Median  Mean 3rd Qu.  Max.
  -0.5    1.3     2.0    1.9    2.6    5.6

Parameter Estimates
  Estimate Std Error z value Pr(>|z|)
F3V2 1.0e+00 5.03 2.0e-01 8.4e-01 V2 <--- MR2
```
4.7 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 std.coef from sem is used as well as the summary function.

4.7.1 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 bifactor data set. Using the commands just shown, it is possible to analyze this data set using a bifactor solution (Figure 13).

> sem.bi <- sem(om.th.bi$model, Thurstone, 213)
> summary(sem.bi, digits = 2)

Model Chisquare =  24   Df =  18 Pr(>Chisq) = 0.15
Chisquare (null model) =  1102   Df =  36
Goodness-of-fit index =  0.98
Adjusted goodness-of-fit index =  0.94
RMSEA index =  0.04   90% CI: (NA, 0.078)
Figure 12: A three factor, oblique solution.
Figure 13: 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.
Bentler-Bonnett NFI = 0.98  
Tucker-Lewis NNFI = 0.99  
Bentler CFI = 1  
SRMR = 0.035  
BIC = -72  

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></td>
<td>-0.82</td>
<td>-0.33</td>
<td>0.00</td>
<td>0.03</td>
<td>0.16</td>
<td>1.80</td>
</tr>
</tbody>
</table>

Parameter Estimates

| Parameter       | Estimate | Std Error | z value | Pr(>|z|) |
|-----------------|----------|-----------|---------|---------|
| Sentences       | 0.77     | 0.073     | 10.57   | 0.0e+00 |
| Vocabulary      | 0.79     | 0.072     | 10.92   | 0.0e+00 |
| Sent.Completion | 0.75     | 0.073     | 10.27   | 0.0e+00 |
| First.Letters   | 0.61     | 0.072     | 8.43    | 0.0e+00 |
| 4.Letter.Words  | 0.60     | 0.074     | 8.09    | 6.7e-16 |
| Suffixes        | 0.57     | 0.071     | 8.00    | 1.3e-15 |
| Letter.Series   | 0.57     | 0.074     | 7.63    | 2.3e-14 |
| Pedigrees       | 0.66     | 0.069     | 9.55    | 0.0e+00 |
| Letter.Group    | 0.53     | 0.079     | 6.71    | 2.0e-11 |
| F1*Sentences    | 0.49     | 0.085     | 5.71    | 1.1e-08 |
| F1*Vocabulary   | 0.45     | 0.090     | 5.00    | 5.7e-07 |
| F1*Sent.Completion | 0.40 | 0.093     | 4.33    | 1.5e-05 |
| F2*First.Letters | 0.61   | 0.086     | 7.16    | 8.2e-13 |
| F2*4.Letter.Words | 0.51  | 0.085     | 5.96    | 2.5e-09 |
| F2*Suffixes     | 0.39     | 0.078     | 5.04    | 4.7e-07 |
| F3*Letter.Series | 0.73  | 0.159     | 4.56    | 5.1e-06 |
| F3*Pedigrees    | 0.25     | 0.089     | 2.77    | 5.6e-03 |
| F3*Letter.Group | 0.41     | 0.122     | 3.35    | 8.1e-04 |
| e1              | 0.17     | 0.034     | 5.05    | 4.4e-07 |
| e2              | 0.17     | 0.030     | 5.65    | 1.6e-08 |
| e3              | 0.27     | 0.033     | 8.09    | 6.7e-16 |
| e4              | 0.25     | 0.079     | 3.18    | 1.5e-03 |
| e5              | 0.39     | 0.063     | 6.13    | 8.8e-10 |
| e6              | 0.52     | 0.060     | 8.68    | 0.0e+00 |
| e7              | 0.15     | 0.223     | 0.67    | 5.0e-01 |
| e8              | 0.50     | 0.060     | 8.39    | 0.0e+00 |
| e9              | 0.55     | 0.085     | 6.51    | 7.4e-11 |

Sentences <- g
Vocabulary ← Vocabulary
Sent.Completion ← Sent.Completion
First.Letters ← First.Letters
Suffixes ← Suffixes
Letter.Series ← Letter.Series
Pedigrees ← Pedigrees
Letter.Group ← Letter.Group
F1*Sentences ← Sentences ← F1*
F1*Vocabulary ← Vocabulary ← F1*
F2*First.Letters ← First.Letters ← 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 4.Letter.Words ← 4.Letter.Words
e6 Suffixes ← Suffixes
e7 Letter.Series ← Letter.Series
e8 Pedigrees ← Pedigrees
e9 Letter.Group ← Letter.Group

Iterations = 72

> std.coef(sem.bi, digits = 2)

<table>
<thead>
<tr>
<th></th>
<th>Std. Estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentences</td>
<td>0.7678671</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>0.7909246</td>
</tr>
<tr>
<td>Sent.Completion</td>
<td>0.7536211</td>
</tr>
<tr>
<td>First.Letters</td>
<td>0.6083814</td>
</tr>
<tr>
<td>4.Letter.Words</td>
<td>0.5973349</td>
</tr>
<tr>
<td>Suffixes</td>
<td>0.5717900</td>
</tr>
<tr>
<td>Letter.Series</td>
<td>0.5668950</td>
</tr>
<tr>
<td>Pedigrees</td>
<td>0.6623317</td>
</tr>
<tr>
<td>Letter.Group</td>
<td>0.5299523</td>
</tr>
<tr>
<td>F1*Sentences</td>
<td>0.4878697</td>
</tr>
</tbody>
</table>
4.8 Examining a hierarchical solution

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

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

Model Chisquare = 38  Df = 24 Pr(>Chisq) = 0.033
Chisquare (null model) = 1102  Df = 36
Goodness-of-fit index = 0.96
Adjusted goodness-of-fit index = 0.92
RMSEA index = 0.053  90% CI: (0.015, 0.083)
Bentler-Bonnett NFI = 0.97
\end{verbatim}
Figure 14: Hierarchical analysis of the Thurstone 9 variable problem using an exploratory algorithm can provide the appropriate sem code for analysis using the sem package.
Tucker-Lewis NNFI = 0.98
Bentler CFI = 0.99
SRMR = 0.044
BIC = -90

Normalized Residuals
Min. 1st Qu. Median Mean 3rd Qu. Max.
-0.97  -0.42    0.00   0.04   0.09   1.63

Parameter Estimates

|          | Estimate | Std Error | z value | Pr(>|z|) |
|----------|----------|-----------|---------|----------|
gF1       | 1.44     | 0.264     | 5.5     | 4.6e-08  |
gF2       | 1.25     | 0.217     | 5.8     | 7.1e-09  |
gF3       | 1.41     | 0.279     | 5.0     | 4.8e-07  |
F1Sentences | 0.52     | 0.065     | 7.9     | 2.2e-15  |
F1Vocabulary | 0.52     | 0.065     | 8.0     | 1.3e-15  |
F1Sent.Completion | 0.49    | 0.062     | 7.8     | 5.8e-15  |
F2First.Letters | 0.52    | 0.063     | 8.3     | 2.2e-16  |
F24.Letter.Words | 0.50    | 0.060     | 8.3     | 0.0e+00  |
F2Suffixes | 0.44     | 0.056     | 7.8     | 8.7e-15  |
F3Letter.Series | 0.45    | 0.071     | 6.3     | 2.3e-10  |
F3Pedigrees | 0.42     | 0.061     | 6.8     | 8.1e-12  |
F3Letter.Group | 0.41     | 0.065     | 6.3     | 2.7e-10  |
e1        | 0.18     | 0.028     | 6.4     | 1.7e-10  |
e2        | 0.16     | 0.028     | 5.9     | 3.0e-09  |
e3        | 0.27     | 0.033     | 8.0     | 1.6e-15  |
e4        | 0.30     | 0.051     | 5.9     | 2.7e-09  |
e5        | 0.36     | 0.052     | 7.0     | 3.4e-12  |
e6        | 0.51     | 0.060     | 8.4     | 0.0e+00  |
e7        | 0.39     | 0.062     | 6.3     | 2.3e-10  |
e8        | 0.48     | 0.065     | 7.4     | 1.8e-13  |
e9        | 0.51     | 0.065     | 7.7     | 9.5e-15  |
gF1       | F1 <--- g |
gF2       | F2 <--- g |
gF3       | F3 <--- g |
F1Sentences | Sentences <--- F1 |
F1Vocabulary | Vocabulary <--- F1 |
F1Sent.Completion | Sent.Completion <--- F1 |
F2First.Letters | First.Letters <--- F2 |

46
F2Suffixes Suffixes <--- F2
F3Letter.Series Letter.Series <--- F3
F3Pedigrees Pedigrees <--- F3
e1 Sentences <-> Sentences
e2 Vocabulary <-> Vocabulary
e3 Sent.Completion <-> Sent.Completion
e4 First.Letters <-> First.Letters
e5 4.Letter.Words <-> 4.Letter.Words
e6 Suffixes <-> Suffixes
e7 Letter.Series <-> Letter.Series
e8 Pedigrees <-> Pedigrees
e9 Letter.Group <-> Letter.Group

Iterations = 53

> std.coef(sem.hi, digits = 2)

Std. Estimate
1  gF1  0.8220756  F1 <--- g
2  gF2  0.7817998  F2 <--- g
3  gF3  0.8150140  F3 <--- g
4  F1Sentences 0.9047110  Sentences <--- F1
5   F1Vocabulary 0.9138215  Vocabulary <--- F1
6  F1Sent.Completion 0.8560765  Sent.Completion <--- F1
7  F2First.Letters 0.8357616  First.Letters <--- F2
8  F24.Letter.Words 0.7971820  4.Letter.Words <--- F2
9  F2Suffixes 0.7025559  Suffixes <--- F2
10 F3Letter.Series 0.7808129  Letter.Series <--- F3
11  F3Pedigrees 0.7201599  Pedigrees <--- F3
12 F3Letter.Group 0.7034902  Letter.Group <--- F3
13   e1  0.1814980  Sentences <-> Sentences
14   e2  0.1649303  Vocabulary <-> Vocabulary
15   e3  0.2671331  Sent.Completion <-> Sent.Completion
16   e4  0.3015025  First.Letters <-> First.Letters
17   e5  0.3645009  4.Letter.Words <-> 4.Letter.Words
18   e6  0.5064152  Suffixes <-> Suffixes
19   e7  0.3903313  Letter.Series <-> Letter.Series
20   e8  0.4813697  Pedigrees <-> Pedigrees
21   e9  0.5051015  Letter.Group <-> Letter.Group
22  0.3241917  F1 <-> F1
23  0.3887891  F2 <-> F2
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 14) 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.

```r
> data(bifactor)
> om.holz.bi <- omega(Holzinger, 4)
> sem.holz.bi <- sem(om.holz.bi$model, Holzinger, 355)
> om.holz.hi <- omega(Holzinger, 4, sl = FALSE)
> sem.holz.hi <- sem(om.holz.hi$model, Holzinger, 355)
> anova(sem.holz.bi, sem.holz.hi)
```

LR Test for Difference Between Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Df</th>
<th>Model Chisq</th>
<th>Df</th>
<th>LR Chisq</th>
<th>Pr(&gt;Chisq)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 1</td>
<td>24</td>
<td>38.196</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model 2</td>
<td>18</td>
<td>24.216</td>
<td>6</td>
<td>13.98</td>
<td>0.02986 *</td>
</tr>
</tbody>
</table>
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```r
> data(bifactor)
> om.holz.bi <- omega(Holzinger, 4)
> sem.holz.bi <- sem(om.holz.bi$model, Holzinger, 355)
> om.holz.hi <- omega(Holzinger, 4, sl = FALSE)
> sem.holz.hi <- sem(om.holz.hi$model, Holzinger, 355)
> anova(sem.holz.bi, sem.holz.hi)
```
4.9 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
> data(bifactor)
> om.sem <- omegaSem(Thurstone, n.obs = 213)
```

Call: `omegaSem(m = Thurstone, n.obs = 213)`

Omega

Call: `omega(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)`

<table>
<thead>
<tr>
<th></th>
<th>Alpha</th>
<th>G.6</th>
<th>Omega Hierarchical</th>
<th>Omega H asymptotic</th>
<th>Omega Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.89</td>
<td>0.91</td>
<td>0.74</td>
<td>0.79</td>
<td>0.93</td>
</tr>
</tbody>
</table>

Schmid Leiman Factor loadings greater than 0.2

<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>0.71</td>
<td>0.57</td>
<td></td>
<td></td>
<td>0.82</td>
<td>0.18</td>
<td>0.61</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>0.73</td>
<td>0.55</td>
<td></td>
<td></td>
<td>0.84</td>
<td>0.16</td>
<td>0.63</td>
</tr>
<tr>
<td>Sent.Completion</td>
<td>0.68</td>
<td>0.52</td>
<td></td>
<td></td>
<td>0.73</td>
<td>0.27</td>
<td>0.63</td>
</tr>
<tr>
<td>First.Letters</td>
<td>0.65</td>
<td>0.56</td>
<td>0.56</td>
<td></td>
<td>0.73</td>
<td>0.27</td>
<td>0.57</td>
</tr>
<tr>
<td>4.Letter.Words</td>
<td>0.62</td>
<td>0.49</td>
<td>0.49</td>
<td>0.49</td>
<td>0.63</td>
<td>0.37</td>
<td>0.61</td>
</tr>
<tr>
<td>Suffixes</td>
<td>0.56</td>
<td>0.41</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.63</td>
</tr>
<tr>
<td>Letter.Series</td>
<td>0.59</td>
<td>0.61</td>
<td>0.72</td>
<td>0.28</td>
<td>0.48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pedigrees</td>
<td>0.58</td>
<td>0.23</td>
<td>0.34</td>
<td>0.50</td>
<td>0.50</td>
<td>0.66</td>
<td></td>
</tr>
<tr>
<td>Letter.Group</td>
<td>0.54</td>
<td>0.46</td>
<td>0.53</td>
<td>0.47</td>
<td>0.56</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

With eigenvalues of:

<table>
<thead>
<tr>
<th>g</th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.58</td>
<td>0.96</td>
<td>0.74</td>
<td>0.71</td>
</tr>
</tbody>
</table>

general/max 3.71 max/min = 1.35
mean percent general = 0.6 with sd = 0.05 and cv of 0.09

49
The degrees of freedom are 12 and the fit is 0.01
The number of observations was 213 with Chi Square = 2.82 with prob < 1
The root mean square of the residuals is 0
The df corrected root mean square of the residuals is 0.01
RMSEA index = 0 and the 90 % confidence intervals are 0 0.023
BIC = -61.51

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 = 307.1 with prob < 2.8e-49
The root mean square of the residuals is 0.1
The df corrected root mean square of the residuals is 0.16
RMSEA index = 0.224 and the 90 % confidence intervals are 0.223 0.226
BIC = 162.35

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></td>
<td>0.86</td>
<td>0.73</td>
<td>0.72</td>
</tr>
<tr>
<td>Multiple R square of scores with factors</td>
<td></td>
<td>0.74</td>
<td>0.54</td>
<td>0.52</td>
</tr>
<tr>
<td>Minimum correlation of factor score estimates</td>
<td></td>
<td>0.49</td>
<td>0.08</td>
<td>0.03</td>
</tr>
</tbody>
</table>

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>
</tr>
</thead>
<tbody>
<tr>
<td>Sentences</td>
<td>0.77</td>
<td>0.49</td>
<td>0.83</td>
<td>0.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vocabulary</td>
<td>0.79</td>
<td>0.45</td>
<td>0.83</td>
<td>0.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sent.Completion</td>
<td>0.75</td>
<td>0.40</td>
<td>0.73</td>
<td>0.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>First.Leters</td>
<td>0.61</td>
<td>0.61</td>
<td>0.75</td>
<td>0.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.Letter.Words</td>
<td>0.60</td>
<td>0.51</td>
<td>0.61</td>
<td>0.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Suffixes</td>
<td>0.57</td>
<td>0.39</td>
<td>0.48</td>
<td>0.52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Letter.Series</td>
<td>0.57</td>
<td>0.73</td>
<td>0.85</td>
<td>0.15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pedigrees</td>
<td>0.66</td>
<td>0.25</td>
<td>0.50</td>
<td>0.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Letter.Group</td>
<td>0.53</td>
<td>0.41</td>
<td>0.45</td>
<td>0.55</td>
<td></td>
<td></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>3.88</td>
<td>0.61</td>
<td>0.79</td>
<td>0.76</td>
<td></td>
</tr>
</tbody>
</table>
Comparing the two models graphically (Figure 15 with Figure 13 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.

**Omega from SEM**

![Diagram of Omega from SEM](image)

Figure 15: Confirmatory Omega structure using omegaSem

5 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. The psych tools for simulating structural models and for specifying models are a useful
supplement to the power of packages such as \textit{sem}. The techniques that can be used on simulated data set can also be applied to real data sets.
References


53
Index

anova, 31, 32
bifactor, 7, 8, 25, 26, 39, 44, 48

circumplex structure, 8
cluster.cor, 3
congeneric, 4
describe, 3
edit, 38
fa, 14, 35
factanal, 14, 35
factor.congruence, 21
factor.pa, 3, 21
guttman, 3
hierarchical, 8
holzinger, 26, 48
ICLUST, 3, 29
minimum residual, 35
oblimin, 25
omega, 3, 8, 26, 27, 39, 44, 48, 49
omega.graph, 30
omegaSem, 49, 51
pairs.panels, 3
phi.list, 14, 30
principal, 3, 19, 21
principal components, 19
Promax, 14, 19, 25
promax, 25
psych, 3, 25, 30, 51
quartimin, 19
R function

anova, 31, 32
bifactor, 7, 26, 39, 44
cluster.cor, 3
describe, 3
edit, 38
fa, 14, 35
factanal, 14, 35
factor.congruence, 21
factor.pa, 3, 21
guttman, 3
holzinger, 26, 48
ICLUST, 3, 29
oblimin, 25
omega, 3, 8, 26, 27, 39, 44, 48, 49
omega.graph, 30
omegaSem, 49, 51
pairs.panels, 3
phi.list, 14, 30
principal, 3, 19, 21
Promax, 14, 19, 25
promax, 25
psych, 3
psych package
bifactor, 7, 26, 39, 44
cluster.cor, 3
describe, 3
fa, 14, 35
factor.congruence, 21
factor.pa, 3, 21
guttman, 3
holzinger, 26, 48
ICLUST, 3, 29
omega, 3, 8, 26, 27, 39, 44, 48, 49
omega.graph, 30
omegaSem, 49, 51
pairs.panels, 3
phi.list, 14, 30
principal, 3, 19, 21
Promax, 14, 19, 25
psych, 3
reise, 26, 48
score.items, 3
sim, 3, 4
sim.circ, 8
sim.congeneric, 4, 7
sim.hierarchical, 7, 37
sim.item, 8
sim.structure, 8, 17
structure.diagram, 17, 30
structure.graph, 30, 39
structure.list, 14, 30
structure.sem, 30, 38
Thurstone, 44
VSS, 3
quartimin, 19
reise, 26, 48
Rgraphviz, 30
score.items, 3
sem, 27, 29, 30, 39, 44, 48, 49
sim, 3, 4
sim.circ, 8
sim.congeneric, 4, 7
sim.hierarchical, 7, 37
sim.item, 8
sim.structure, 8, 17
std.coef, 39, 44
structure.diagram, 17, 30
structure.graph, 30, 39
structure.list, 14, 30
structure.sem, 30, 38
summary, 39
Thurstone, 44
varimax, 19
VSS, 3

R package
psych, 3, 25, 30, 51
Rgraphviz, 3
sem, 3, 29, 39, 52
reise, 26, 48
Rgraphviz, 3, 30
rotated, 19
score.items, 3
sem, 3, 27, 29, 30, 39, 44, 48, 49, 52
sim, 3, 4
sim.circ, 8
sim.congeneric, 4, 7
sim.hierarchical, 7, 37
sim.item, 8
sim.structure, 8, 17
simple structure, 8, 19
std.coef, 39, 44
structure.diagram, 8, 17
structure.graph, 30, 39
structure.list, 14, 30
structure.sem, 30, 38
summary, 39
tau, 4
Thurstone, 44
varimax, 19
VSS, 3