Using the psych package to generate and test structural models

William Revelle

December 20, 2009

Contents

1 The psych package

1.1 Preface

The *psych* package (Revelle, 2009) 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.

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 τ 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.

```
> 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
```

```
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)
```

```
     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
```

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).

Figure 1: Tau equivalent tests are special cases of congeneric tests. Tau equivalence assumes a=b=c=d

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.

```
> 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)
```

```
     V1   V2   V3   V4   V5   V6   V7   V8   V9
V1 1.00 0.72 0.63 0.45 0.39 0.32 0.34 0.28 0.23
V2 0.72 1.00 0.56 0.40 0.35 0.29 0.30 0.25 0.20
V3 0.63 0.56 1.00 0.35 0.30 0.25 0.26 0.22 0.18
V4 0.45 0.40 0.35 1.00 0.42 0.35 0.24 0.20 0.16
V5 0.39 0.35 0.30 0.42 1.00 0.30 0.20 0.17 0.13
V6 0.32 0.29 0.25 0.35 0.30 1.00 0.17 0.14 0.11
V7 0.34 0.30 0.26 0.24 0.20 0.17 1.00 0.30 0.24
V8 0.28 0.25 0.22 0.20 0.17 0.14 0.30 1.00 0.20
V9 0.23 0.20 0.18 0.16 0.13 0.11 0.24 0.20 1.00
```

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.

```
> 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))
```

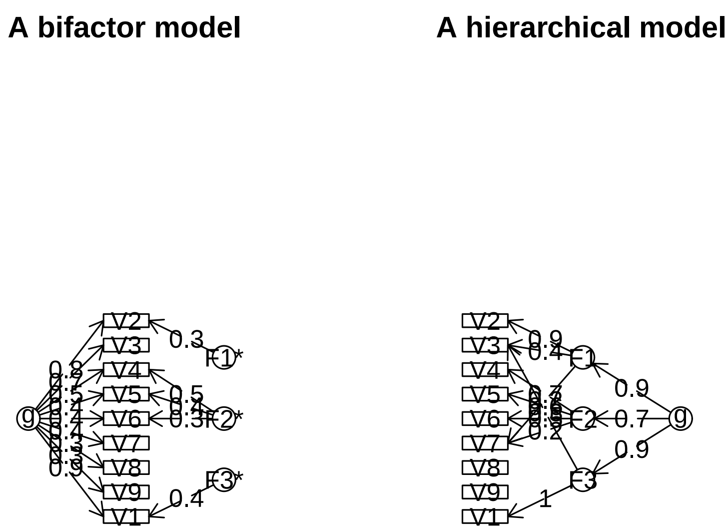

Figure 2: (Left panel) A bifactor solution represents each test in terms of a general factor and a residualized group factor. (Right Panel) A hierarchical factor solution has g as a second order factor accounting for the correlations between the first order factors

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, $\vec{f}_x, \vec{\phi}_{xy}, \vec{f}_y$ which describe, in turn, a measurement model of x variables, $\vec{f}_x$, a measurement model of y variables, $\vec{f}_x$, and a covariance matrix between and within the two sets of factors. If $\vec{f}_x$ is a vector and $\vec{f}_y$ and $\vec{phi}_{xy}$ are NULL, then this is just the congeneric model. If $\vec{f}_x$ is a matrix of loadings with n rows and c columns, then this is a measurement model for n variables across c factors. If $\vec{phi}_{xy}$ is not null, but $\vec{f}_y$ is NULL, then the factors in $\vec{f}_x$ 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 $\vec{f}_x$ is a vector implies a congeneric model

```
> 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)
```

```
> circ <- sim.circ(16)
> f2 <- factor.pa(circ, 2)
> plot(f2, main = "16 simulated variables in a circumplex pattern")
Use ICLUST.graph to see the  hierarchical structure
```

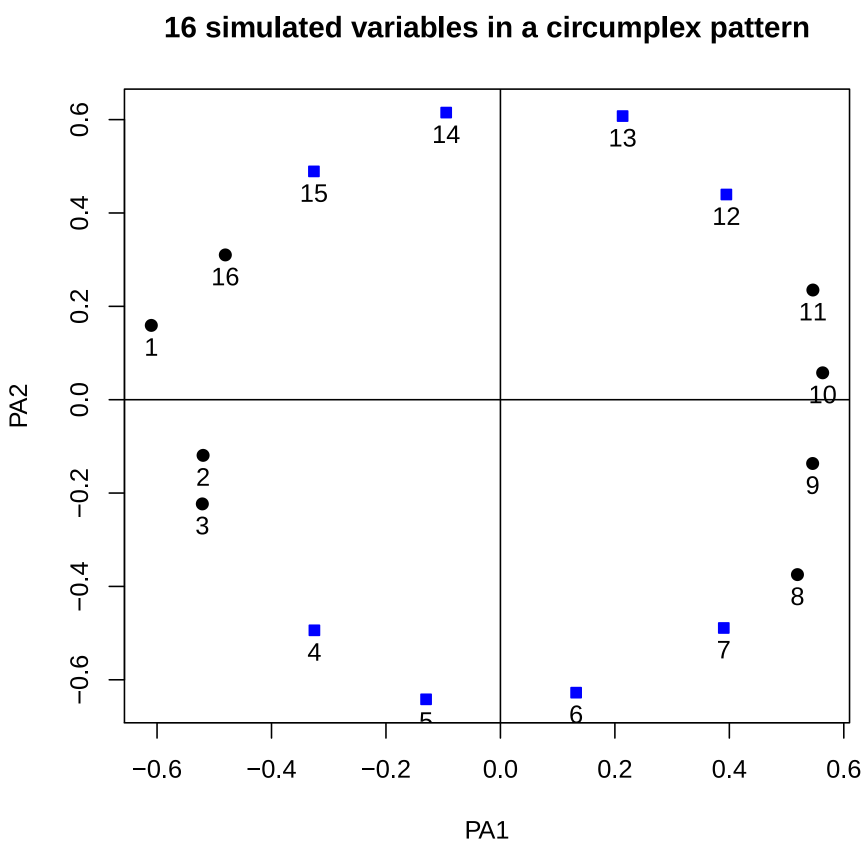

Figure 3: Emotion items or interpersonal items frequently show a circumplex structure. Data generated by sim.circ and factor loadings found by the principal axis algorithm using factor.pa.

```
     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:

```
> 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
V6 0.00 0.00 0.00 0.35 0.30 1.00 0.00 0.0 0.00
V7 0.00 0.00 0.00 0.00 0.00 0.00 1.00 0.3 0.24
V8 0.00 0.00 0.00 0.00 0.00 0.00 0.30 1.0 0.20
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 0.49 0.36 0.25 0.36 0.25 0.16
```

2.4.3 $\vec{f}_x$ is a matrix and Phi $\neq I$ is a correlated factors model

```
> 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
```

Figure 4: Three uncorrelated factors generated using the sim.structure function and drawn using structure.diagram.

```
      [,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 0.630 0.315 0.270 0.23 0.162 0.14 0.108
V2 0.72 1.000 0.560 0.280 0.240 0.20 0.144 0.12 0.096
V3 0.63 0.560 1.000 0.245 0.210 0.17 0.126 0.10 0.084
V4 0.32 0.280 0.245 1.000 0.420 0.35 0.084 0.07 0.056
V5 0.27 0.240 0.210 0.420 1.000 0.30 0.072 0.06 0.048
V6 0.23 0.200 0.175 0.350 0.300 1.00 0.060 0.05 0.040
V7 0.16 0.144 0.126 0.084 0.072 0.06 1.000 0.30 0.240
V8 0.14 0.120 0.105 0.070 0.060 0.05 0.300 1.00 0.200
V9 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).

```
> 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 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.

```
> f3.p <- Promax(fa(cor.f3$model, 3))
```

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.

```
> plot.new()
> corf3.mod <- structure.diagram(fxs, Phis)
```

Figure 5: Three correlated factors with symbolic paths. Created using structure.diagram and structure.list and phi.list for ease of input.

```
> plot.new()
> mod.f3p <- structure.diagram(f3.p, cut = 0.2)
```

Figure 6: The empirically fitted structural model. Paths less than cut (.2 in this case, the default is .3) are not shown.

```
> 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)
```

```
     x1   x2   x3   x4   x5   x6   x7   x8   x9   y1   y2   y3
x1 1.00 0.72 0.63 0.30 0.26 0.22 0.17 0.14 0.12 0.22 0.18 0.14
x2 0.72 1.00 0.56 0.27 0.23 0.19 0.15 0.13 0.10 0.19 0.16 0.13
x3 0.63 0.56 1.00 0.24 0.20 0.17 0.13 0.11 0.09 0.17 0.14 0.11
x4 0.30 0.27 0.24 1.00 0.42 0.35 0.13 0.11 0.09 0.13 0.10 0.08
x5 0.26 0.23 0.20 0.42 1.00 0.30 0.12 0.10 0.08 0.11 0.09 0.07
x6 0.22 0.19 0.17 0.35 0.30 1.00 0.10 0.08 0.06 0.09 0.08 0.06
x7 0.17 0.15 0.13 0.13 0.12 0.10 1.00 0.30 0.24 0.07 0.06 0.05
x8 0.14 0.13 0.11 0.11 0.10 0.08 0.30 1.00 0.20 0.06 0.05 0.04
x9 0.12 0.10 0.09 0.09 0.08 0.06 0.24 0.20 1.00 0.05 0.04 0.03
y1 0.22 0.19 0.17 0.13 0.11 0.09 0.07 0.06 0.05 1.00 0.30 0.24
y2 0.18 0.16 0.14 0.10 0.09 0.08 0.06 0.05 0.04 0.30 1.00 0.20
y3 0.14 0.13 0.11 0.08 0.07 0.06 0.05 0.04 0.03 0.24 0.20 1.00
```

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).

```
> 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).

```
> plot.new()
> sg3 <- structure.diagram(fxs, phi, fyx)
```

Figure 7: A symbolic structural model. Three independent latent variables are regressed on a latent Y.

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 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 aproximates 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)
   V  PC1   h2   u2
V1 1 0.89 0.80 0.20
V2 2 0.85 0.73 0.27
V3 3 0.80 0.64 0.36
V4 4 0.73 0.53 0.47

                PC1
SS loadings    2.69
Proportion Var 0.67

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  0.14

Fit based upon off diagonal values = 0.96
```

```
> fa(cong1$model)
```

```
Factor Analysis using method =  minres
Call: fa(r = cong1$model)
   V MR1   h2   u2
```

```
V1 1 0.9 0.81 0.19
V2 2 0.8 0.64 0.36
V3 3 0.7 0.49 0.51
V4 4 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

Fit based upon off diagonal values = 1
Measures of factor score adequacy
                                               MR1
Correlation of scores with factors            0.94
Multiple R square of scores with factors      0.88
Minimum correlation of factor score estimates 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 χ^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)
   item  RC1  RC3  RC2   h2   u2
V1    1 0.82           0.80 0.20
V2    2 0.82           0.75 0.25
V3    3 0.82           0.71 0.29
V4    4 0.32 0.68      0.58 0.42
V5    5      0.70      0.56 0.44
V6    6      0.77      0.60 0.40
```

```
V7    7           0.66 0.51 0.49
V8    8           0.68 0.50 0.50
V9    9           0.71 0.51 0.49

                RC1  RC3  RC2
SS loadings    2.25 1.73 1.53
Proportion Var 0.25 0.19 0.17
Cumulative Var 0.25 0.44 0.61

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")
   item  PA1  PA3  PA2   h2   u2
V1    1 0.78 0.34      0.81 0.19
V2    2 0.70           0.64 0.36
V3    3 0.61           0.49 0.51
V4    4      0.63      0.49 0.51
V5    5      0.54      0.36 0.64
V6    6      0.45      0.25 0.75
V7    7           0.55 0.36 0.64
V8    8           0.47 0.25 0.75
V9    9           0.37 0.16 0.84

                PA1  PA3  PA2
SS loadings    1.67 1.21 0.93
Proportion Var 0.19 0.13 0.10
Cumulative Var 0.19 0.32 0.42

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

Fit based upon off diagonal values = 1
```

```
Measures of factor score adequacy
                                                PA1  PA3   PA2
Correlation of scores with factors             0.85 0.73  0.67
Multiple R square of scores with factors       0.72 0.53  0.45
Minimum correlation of factor score estimates 0.44 0.06 -0.11
```

```
> ml3
```

```
Factor Analysis using method =  ml
Call: fa(r = bifact, nfactors = 3, fm = "ml")
   item  ML1  ML2  ML3   h2   u2
V1    1 0.78 0.34      0.81 0.19
V2    2 0.70           0.64 0.36
V3    3 0.61           0.49 0.51
V4    4      0.63      0.49 0.51
V5    5      0.54      0.36 0.64
V6    6      0.45      0.25 0.75
V7    7           0.56 0.36 0.64
V8    8           0.46 0.25 0.75
V9    9           0.37 0.16 0.84

                ML1  ML2  ML3
SS loadings    1.67 1.21 0.93
Proportion Var 0.19 0.13 0.10
Cumulative Var 0.19 0.32 0.42

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

Fit based upon off diagonal values = 1
Measures of factor score adequacy
                                                ML1  ML2   ML3
Correlation of scores with factors             0.85 0.73  0.67
Multiple R square of scores with factors       0.72 0.53  0.45
Minimum correlation of factor score estimates 0.44 0.07 -0.10
```

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

```
     RC1  RC3  RC2  PA1  PA3  PA2  ML1  ML2  ML3
RC1 1.00 0.52 0.42 0.99 0.70 0.65 0.99 0.70 0.65
RC3 0.52 1.00 0.33 0.57 0.96 0.50 0.57 0.96 0.50
```

```
RC2 0.42 0.33 1.00 0.45 0.43 0.95 0.45 0.43 0.95
PA1 0.99 0.57 0.45 1.00 0.72 0.67 1.00 0.72 0.67
PA3 0.70 0.96 0.43 0.72 1.00 0.63 0.72 1.00 0.62
PA2 0.65 0.50 0.95 0.67 0.63 1.00 0.67 0.62 1.00
ML1 0.99 0.57 0.45 1.00 0.72 0.67 1.00 0.72 0.67
ML2 0.70 0.96 0.43 0.72 1.00 0.62 0.72 1.00 0.62
ML3 0.65 0.50 0.95 0.67 0.62 1.00 0.67 0.62 1.00
```

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
   item  ML1  ML2  ML3   h2   u2
V1    1 0.83           0.81 0.19
V2    2 0.74           0.64 0.36
V3    3 0.65           0.49 0.51
V4    4      0.69      0.49 0.51
V5    5      0.59      0.36 0.64
V6    6      0.49      0.25 0.75
V7    7           0.60 0.36 0.64
V8    8           0.50 0.25 0.75
V9    9           0.40 0.16 0.84

                ML1  ML2  ML3
SS loadings    1.8 1.18 0.83
Proportion Var 0.2 0.13 0.09
Cumulative Var 0.2 0.33 0.42

 With factor correlations of
     ML1  ML2  ML3
ML1 1.00 0.67 0.59
ML2 0.67 1.00 0.55
ML3 0.59 0.55 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 `omega` function. Graphic options include drawing a Schmid - Leiman bifactor solution (Figure 9) or drawing a hierarchical factor solution f(Figure 10).

3.2.1 A bifactor solution

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

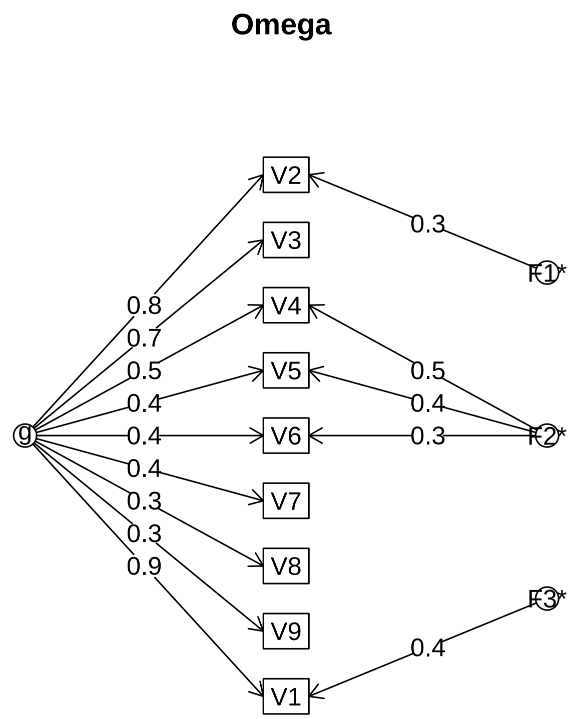

Figure 9: An exploratory bifactor solution to the nine variable problem

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

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

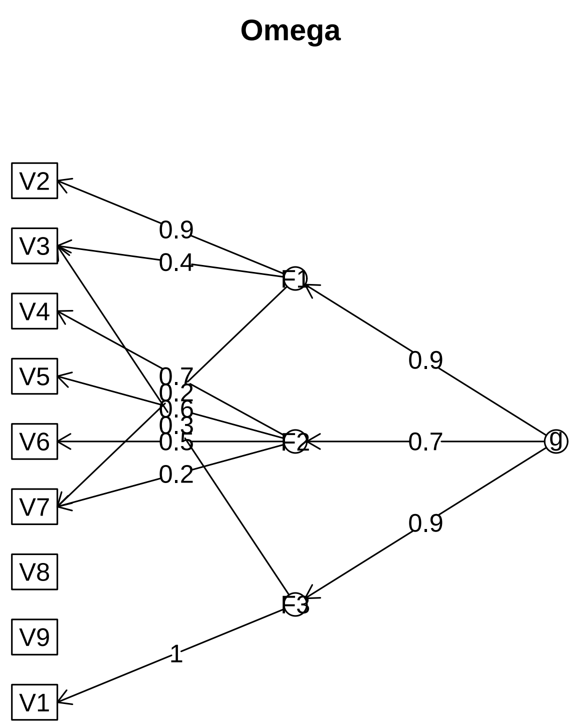

Figure 10: An exploratory hierarchical solution to the nine variable problem.

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: 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
      g  F1*  F2*  F3*   h2   u2
V1 0.90           0.40 0.98 0.02
V2 0.80 0.32           0.73 0.27
V3 0.65                0.46 0.54
V4 0.50      0.47      0.47 0.53
V5 0.43      0.42      0.36 0.64
V6 0.36      0.35      0.25 0.75
V7 0.37                0.17 0.83
V8 0.32                0.12 0.88
V9 0.25                0.08 0.92

With eigenvalues of:
   g  F1*  F2*  F3*
2.74 0.13 0.58 0.17

general/max  4.73   max/min =   4.32
The degrees of freedom for the model is 12  and the fit was  0.07

Measures of factor score adequacy
                                                 g   F1*   F2*   F3*
Correlation of scores with factors            0.94  0.47  0.67  0.62
Multiple R square of scores with factors      0.88  0.22  0.45  0.39
Minimum correlation of factor score estimates 0.75 -0.55 -0.11 -0.23
```

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.

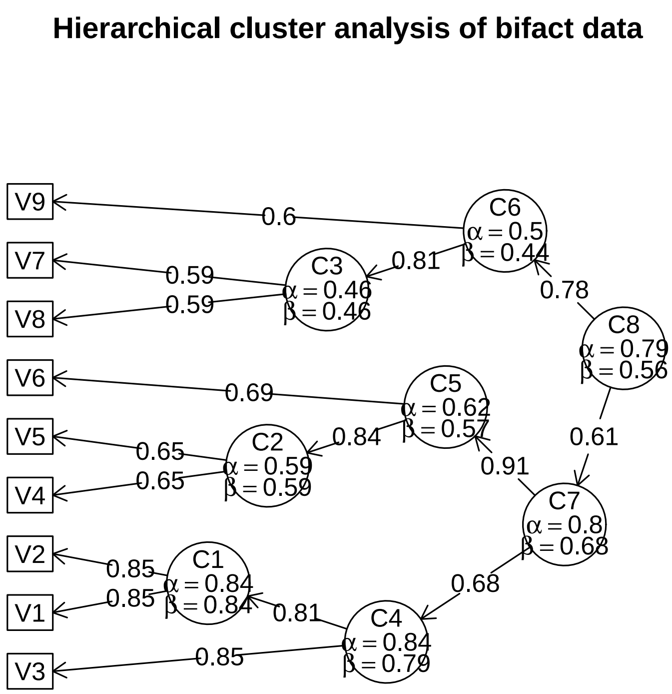

Figure 11: A hierarchical cluster analysis of the bifact data set using ICLUST

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 χ^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 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.

```
> 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
```

```
> 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
 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.
 -1.030  -0.442  -0.250  -0.079   0.527   0.888

 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 `anova` function.

```
> 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)
```

```
 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.
 -0.574  -0.070   0.034   0.011   0.160   0.541

 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
```

```
        Model Df Model Chisq Df LR Chisq Pr(>Chisq)
Model 1        2      2.9417
Model 2        5      6.5935  3   3.6518     0.3016
```

The `anova` comparison of the congeneric versus tau equivalent model shows that the change in χ^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.

```
> 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
 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.
-2.7e-01 -1.8e-01 -1.4e-06  1.4e-01  1.2e-01  1.6e+00

 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    1.1e-03  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    1.6e-11  V5 <--> V5
x6e 0.84     0.123      6.8    7.5e-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
V5 -0.02 -0.03 -0.03  0.17  0.00  0.11  0.01  0.01  0.01
V6 -0.02 -0.03 -0.02  0.14  0.11  0.00  0.01  0.01  0.00
V7 -0.02 -0.02 -0.02  0.02  0.01  0.01  0.00  0.16  0.13
V8 -0.02 -0.02 -0.02  0.01  0.01  0.01  0.16  0.00  0.11
V9 -0.01 -0.02 -0.02  0.01  0.01  0.00  0.13  0.11  0.00
```

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.

```
> 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.

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

```
 Model Chisquare =  66   Df =  28 Pr(>Chisq) = 5.7e-05
 Chisquare (null model) =  235   Df =  36
 Goodness-of-fit index =  0.84
 Adjusted goodness-of-fit index =  0.75
 RMSEA index =  0.12   90% CI: (0.081, 0.15)
 Bentler-Bonnett NFI =  0.72
 Tucker-Lewis NNFI =  0.75
 Bentler CFI =  0.8
 SRMR =  0.21
 BIC =  -62

 Normalized Residuals
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-5.5e-05  1.0e+00  2.0e+00  1.9e+00  2.6e+00  4.0e+00

 Parameter Estimates
     Estimate Std Error z value Pr(>|z|)
F1V1  0.79    0.094      8.431  0.0e+00  V1 <--- MR1
F2V1  0.23    0.089      2.574  1.0e-02  V1 <--- MR2
F3V1  0.60    2.484      0.240  8.1e-01  V1 <--- MR3
F1V2  0.80    0.093      8.579  0.0e+00  V2 <--- MR1
F1V3  0.70    0.095      7.368  1.7e-13  V3 <--- MR1
F2V4  0.70    0.129      5.416  6.1e-08  V4 <--- MR2
F2V5  0.60    0.124      4.850  1.2e-06  V5 <--- MR2
F2V6  0.50    0.120      4.166  3.1e-05  V6 <--- MR2
x1e  -0.16    2.958     -0.054  9.6e-01  V1 <--> V1
x2e   0.36    0.085      4.215  2.5e-05  V2 <--> V2
x3e   0.51    0.089      5.703  1.2e-08  V3 <--> V3
x4e   0.51    0.152      3.359  7.8e-04  V4 <--> V4
x5e   0.64    0.136      4.719  2.4e-06  V5 <--> V5
x6e   0.75    0.130      5.763  8.3e-09  V6 <--> V6
x7e   1.00    0.142      7.034  2.0e-12  V7 <--> V7
x8e   1.00    0.142      7.034  2.0e-12  V8 <--> V8
x9e   1.00    0.142      7.034  2.0e-12  V9 <--> V9
```

```
 Iterations =  18
```

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

```
     V1   V2   V3   V4   V5   V6   V7   V8   V9
V1 0.12 0.09 0.08 0.29 0.25 0.21 0.34 0.28 0.23
V2 0.09 0.00 0.00 0.40 0.35 0.29 0.30 0.25 0.20
V3 0.08 0.00 0.00 0.35 0.30 0.25 0.26 0.22 0.18
V4 0.29 0.40 0.35 0.00 0.00 0.00 0.24 0.20 0.16
V5 0.25 0.35 0.30 0.00 0.00 0.00 0.20 0.17 0.13
V6 0.21 0.29 0.25 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)
> f3.p <- Promax(f3)
> mod.f3p <- structure.sem(f3.p)
> mod.f3p
```

```
   Path       Parameter StartValue
1  MR1->V1    F1V1
2  MR1->V2    F1V2
3  MR1->V3    F1V3
4  MR2->V4    F2V4
5  MR2->V5    F2V5
6  MR2->V6    F2V6
7  MR3->V7    F3V7
8  MR3->V8    F3V8
9  MR3->V9    F3V9
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 MR2<->MR1 rF2F1
20 MR3<->MR1 rF3F1
21 MR3<->MR2 rF3F2
22 MR1<->MR1 <fixed>   1
23 MR2<->MR2 <fixed>   1
24 MR3<->MR3 <fixed>   1
```

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.,

mod.adjusted <- edit(mod.f3p)

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

When this is done

```
> 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)
```

```
 Model Chisquare =  8.7   Df =  26 Pr(>Chisq) = 1
 Chisquare (null model) =  169   Df =  36
 Goodness-of-fit index =  0.98
 Adjusted goodness-of-fit index =  0.97
 RMSEA index =  0   90% CI: (NA, NA)
 Bentler-Bonnett NFI =  0.95
```

```
 Tucker-Lewis NNFI =  1.2
 Bentler CFI =  1
 SRMR =  0.14
 BIC =  -111

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -2.19   -1.05   -0.64   -0.81   -0.41   -0.16

 Parameter Estimates
      Estimate Std Error z value Pr(>|z|)
F1V2  0.78     0.112     7.0     3.1e-12  V2 <--- MR1
F1V3  0.67     0.115     5.9     4.6e-09  V3 <--- MR1
F2V5  0.66     0.129     5.1     3.5e-07  V5 <--- MR2
F2V6  0.55     0.129     4.3     2.0e-05  V6 <--- MR2
F3V7  0.64     0.138     4.6     3.6e-06  V7 <--- MR3
F3V8  0.53     0.135     4.0     7.8e-05  V8 <--- MR3
F3V9  0.43     0.135     3.2     1.5e-03  V9 <--- MR3
x1e   0.31     0.094     3.3     1.1e-03  V1 <--> V1
x2e   0.53     0.098     5.4     6.3e-08  V2 <--> V2
x3e   0.65     0.107     6.1     9.9e-10  V3 <--> V3
x4e   0.39     0.127     3.1     2.0e-03  V4 <--> V4
x5e   0.68     0.118     5.7     9.3e-09  V5 <--> V5
x6e   0.77     0.123     6.3     3.3e-10  V6 <--> V6
x7e   0.64     0.146     4.4     1.2e-05  V7 <--> V7
x8e   0.75     0.137     5.5     4.0e-08  V8 <--> V8
x9e   0.84     0.136     6.2     6.6e-10  V9 <--> V9
rF2F1 0.74     0.090     8.3     2.2e-16  MR1 <--> MR2
rF3F1 0.68     0.121     5.6     2.4e-08  MR1 <--> MR3
rF3F2 0.60     0.140     4.3     1.8e-05  MR2 <--> MR3

 Iterations =  18
```

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

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

Figure 12: A three factor, oblique solution.

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).

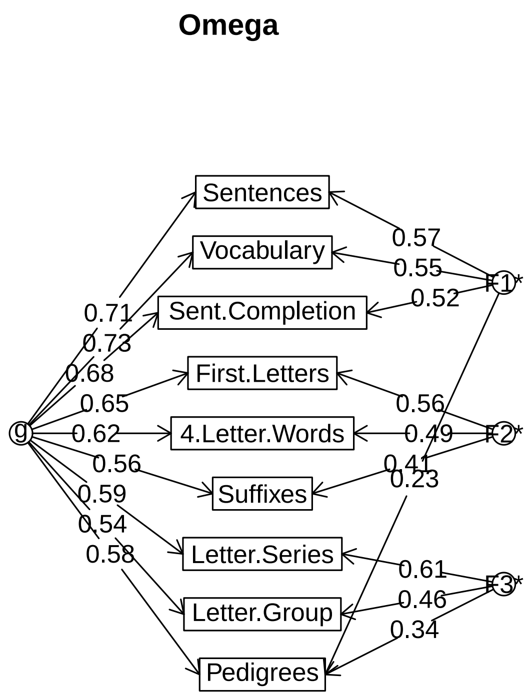

Figure 13: A bifactor solution to the Thurstone 9 variable problem. All items load on a general factor of ability, the residual factors account for the correlations between items within groups.

```
> 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)
 Bentler-Bonnett NFI =  0.98
 Tucker-Lewis NNFI =  0.99
 Bentler CFI =  1
 SRMR =  0.035
 BIC =  -72

 Normalized Residuals
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-8.2e-01 -3.3e-01 -8.9e-07  2.8e-02  1.6e-01  1.8e+00

 Parameter Estimates
                  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           Sentences <--- g
Vocabulary          Vocabulary <--- g
Sent.Completion     Sent.Completion <--- g
First.Letters       First.Letters <--- g
4.Letter.Words      4.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*4.Letter.Words   4.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                  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)
```

```
                       Std. Estimate
1           Sentences     0.7678671                          Sentences <--- g
2          Vocabulary     0.7909246                         Vocabulary <--- g
```

```
3     Sent.Completion     0.7536211             Sent.Completion <--- g
4       First.Letters     0.6083814               First.Letters <--- g
5      4.Letter.Words     0.5973349              4.Letter.Words <--- g
6            Suffixes     0.5717900                    Suffixes <--- g
7       Letter.Series     0.5668950               Letter.Series <--- g
8           Pedigrees     0.6623317                   Pedigrees <--- g
9        Letter.Group     0.5299523                Letter.Group <--- g
10       F1*Sentences     0.4878697                 Sentences <--- F1*
11      F1*Vocabulary     0.4523233                Vocabulary <--- F1*
12 F1*Sent.Completion     0.4044507           Sent.Completion <--- F1*
13   F2*First.Letters     0.6140531             First.Letters <--- F2*
14  F2*4.Letter.Words     0.5058063            4.Letter.Words <--- F2*
15       F2*Suffixes      0.3943206                  Suffixes <--- F2*
16   F3*Letter.Series     0.7272957             Letter.Series <--- F3*
17      F3*Pedigrees      0.2468418                 Pedigrees <--- F3*
18    F3*Letter.Group     0.4091494              Letter.Group <--- F3*
19                 e1     0.1723633           Sentences <--> Sentences
20                 e2     0.1698418         Vocabulary <--> Vocabulary
21                 e3     0.2684748 Sent.Completion <--> Sent.Completion
22                 e4     0.2528108     First.Letters <--> First.Letters
23                 e5     0.3873510   4.Letter.Words <--> 4.Letter.Words
24                 e6     0.5175675             Suffixes <--> Suffixes
25                 e7     0.1496710     Letter.Series <--> Letter.Series
26                 e8     0.5003859           Pedigrees <--> Pedigrees
27                 e9     0.5517473       Letter.Group <--> Letter.Group
28                        1.0000000                       F1* <--> F1*
29                        1.0000000                       F2* <--> F2*
30                        1.0000000                       F3* <--> F3*
31                        1.0000000                           g <--> g
```

Compare this solution to the one reported below, and to the sem manual.

4.8 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 `std.coef` function helps see the structure. Alternatively, using the `omega` function on the `Thurstone` data (in the `bifactor` data set) will create the model for this particular data set.

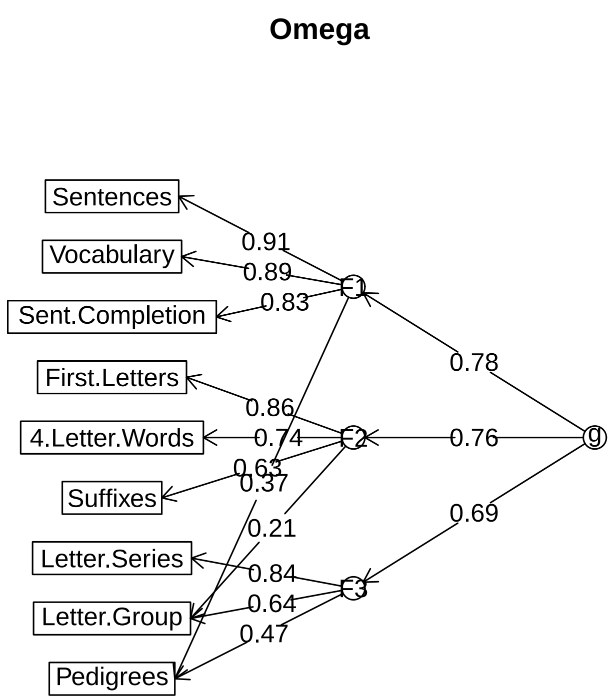

Figure 14: Hierarchical analysis of the Thurstone 9 variable problem using an exploratory algorithm can provide the appropriate sem code for analysis using the sem package.

```
> 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
 Tucker-Lewis NNFI =  0.98
 Bentler CFI =  0.99
 SRMR =  0.044
 BIC =  -90

 Normalized Residuals
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-9.7e-01 -4.2e-01 -1.3e-06  4.0e-02  9.4e-02  1.6e+00

 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
F24.Letter.Words  4.Letter.Words <--- F2
F2Suffixes        Suffixes <--- F2
F3Letter.Series   Letter.Series <--- F3
F3Pedigrees       Pedigrees <--- F3
F3Letter.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 =  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
24                         0.3357522                           F3 <--> F3
25                         1.0000000                             g <--> g
```

```
> anova(sem.hi, sem.bi)
```

```
LR Test for Difference Between Models

        Model Df Model Chisq Df LR Chisq Pr(>Chisq)
Model 1       24      38.196
Model 2       18      24.216  6    13.98    0.02986 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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.

```
> 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

        Model Df Model Chisq Df LR Chisq Pr(>Chisq)
```

```
Model 1       63      147.66
Model 2       73      178.79 10   31.129  0.0005587 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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 *sem*. The techniques that can be used on simulated data set can also be applied to real data sets.

References

Adams, D. (1980). *The hitchhiker's guide to the galaxy.* Harmony Books, New York, 1st American edition.

Bechtoldt, H. (1961). An empirical study of the factor analysis stability hypothesis. *Psychometrika*, 26(4):405–432.

Fox, J. (2006). Structural equation modeling with the sem package in R. *Structural Equation Modeling*, 13:465–486.

Fox, J. (2009). *sem: Structural Equation Models.* R package version 0.9-15.

Holzinger, K. and Swineford, F. (1937). The bi-factor method. *Psychometrika*, 2(1):41–54.

Jensen, A. R. and Weng, L.-J. (1994). What is a good g? *Intelligence*, 18(3):231–258.

McDonald, R. P. (1999). *Test theory: A unified treatment.* L. Erlbaum Associates, Mahwah, N.J.

Rafaeli, E. and Revelle, W. (2006). A premature consensus: Are happiness and sadness truly opposite affects? *Motivation and Emotion*, 30(1):1–12.

Reise, S., Morizot, J., and Hays, R. (2007). The role of the bifactor model in resolving dimensionality issues in health outcomes measures. *Quality of Life Research*, 16(0):19–31.

Revelle, W. (2009). *psych: Procedures for Personality and Psychological Research.* R package version 1.0-75.

Schmid, J. J. and Leiman, J. M. (1957). The development of hierarchical factor solutions. *Psychometrika*, 22(1):83–90.

Thurstone, L. L. and Thurstone, T. G. (1941). *Factorial studies of intelligence.* The University of Chicago press, Chicago, Ill.

Index