Using the psych package to generate and test structural models

William Revelle

August 8, 2012

Contents

1 The psych package

1.1 Preface

The *psych* package (Revelle, 2011) 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; Fox et al., 2011). Future releases of this vignette will include examples for using the *lavaan* package of Yves Rosseel (Rosseel, 2011).

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

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

sim.circ Create data with a circumplex structure.

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

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

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

sim.rasch Create IRT data following a Rasch model.

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

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

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

2.1 sim.congeneric

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

The first example considers four tests with equal loadings on a latent factor (that is, a 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(.8,.8,.8,.8)) #population values
> tau.samp <- sim.congeneric(loads=c(.8,.8,.8,.8),N=100) # sample correlation matrix for 100 cases
> round(tau.samp,2)
```

```
     V1   V2   V3   V4
V1 1.00 0.68 0.72 0.66
V2 0.68 1.00 0.65 0.67
V3 0.72 0.65 1.00 0.76
V4 0.66 0.67 0.76 1.00
```

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

```
Call: NULL

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

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

```
> dim(tau.samp$observed)
```

```
[1] 100   4
```

In this last case, the generated data are retrieved from tau.samp$observed. Congeneric data are created by specifying unequal loading values. The default values are loadings of c(.8,.7,.6,.5). As seen in Figure 1, tau equivalence is the special case where all paths are equal.

```
> cong <- sim.congeneric(N=100)
> round(cong,2)
```

```
     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
```

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

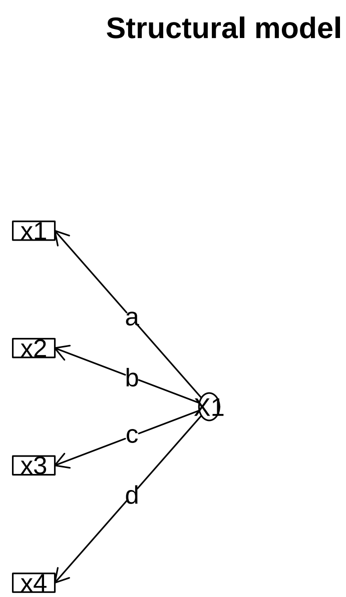

Figure 1: Tau equivalent tests are special cases of congeneric tests. Tau equivalence assumes a=b=c=d

2.2 sim.hierarchical

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

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

```
> set.seed(42)
> gload=matrix(c(.9,.8,.7),nrow=3)
> fload <- matrix(c(.8,.7,.6,rep(0,9),.7,.6,.5,
+ rep(0,9),.7,.6,.4),   ncol=3)
> fload #echo it to see the structureSw

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

> bifact <- sim.hierarchical(gload=gload,fload=fload)
> round(bifact,2)

     V1   V2   V3   V4   V5   V6   V7   V8   V9
V1 1.00 0.56 0.48 0.40 0.35 0.29 0.35 0.30 0.20
V2 0.56 1.00 0.42 0.35 0.30 0.25 0.31 0.26 0.18
V3 0.48 0.42 1.00 0.30 0.26 0.22 0.26 0.23 0.15
V4 0.40 0.35 0.30 1.00 0.42 0.35 0.27 0.24 0.16
V5 0.35 0.30 0.26 0.42 1.00 0.30 0.24 0.20 0.13
V6 0.29 0.25 0.22 0.35 0.30 1.00 0.20 0.17 0.11
V7 0.35 0.31 0.26 0.27 0.24 0.20 1.00 0.42 0.28
```

```
V8 0.30 0.26 0.23 0.24 0.20 0.17 0.42 1.00 0.24
V9 0.20 0.18 0.15 0.16 0.13 0.11 0.28 0.24 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.

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(.9,.8,.7,.6)
```

```
> 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))
```

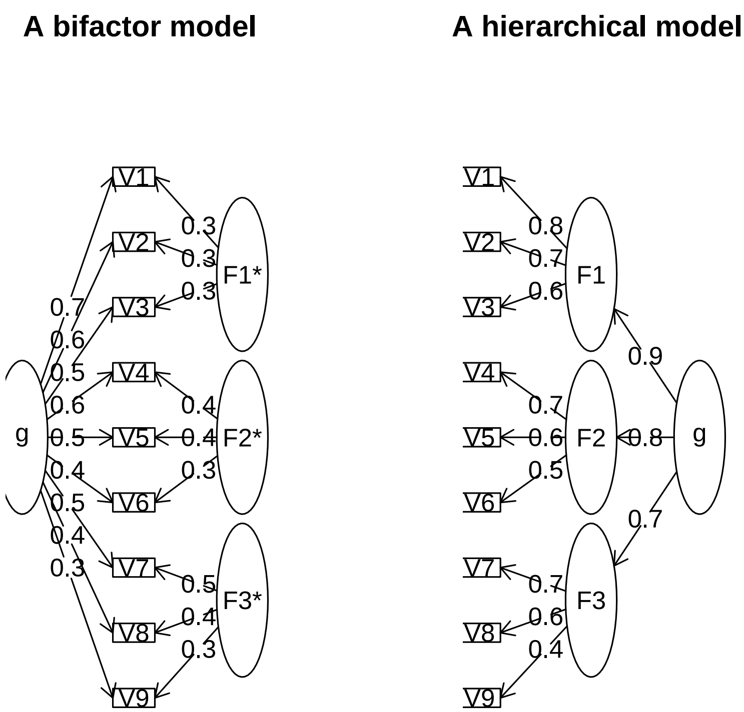

Figure 2: (Left panel) A bifactor solution represents each test in terms of a general factor and a residualized group factor. (Right Panel) A hierarchical factor solution has g as a second order factor accounting for the correlations between the first order factors

```
> circ <- sim.circ(16)
> f2 <- fa(circ,2)
> plot(f2,title="16 simulated variables in a circumplex pattern")
```

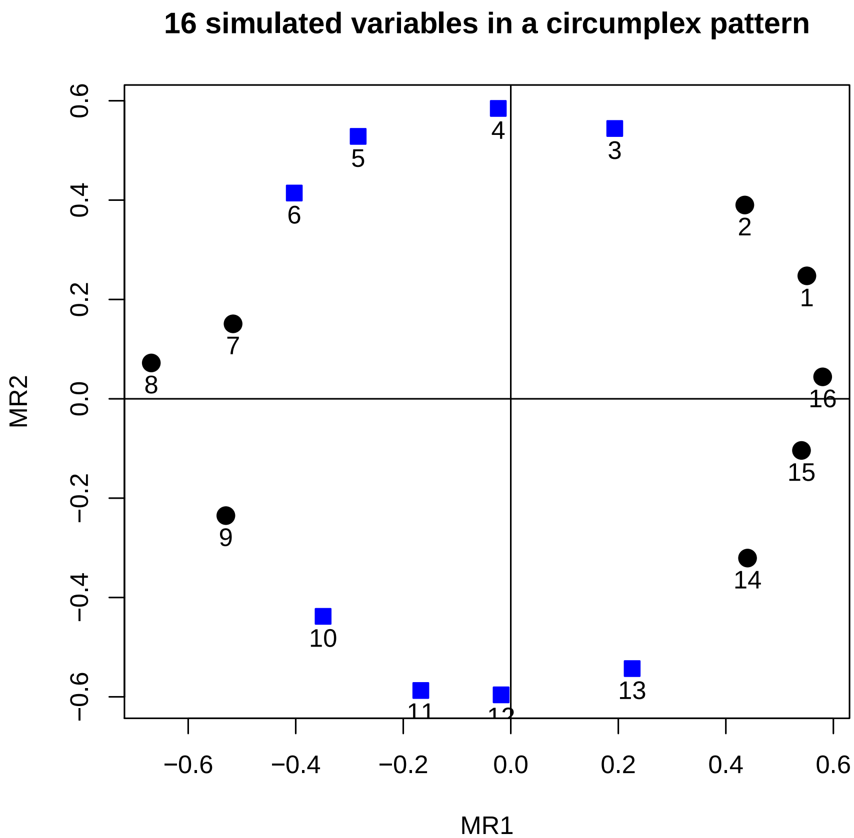

Figure 3: Emotion items or interpersonal items frequently show a circumplex structure. Data generated by sim.circ and factor loadings found by the principal axis algorithm using factor.pa.

```
> 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:

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

```
Call: sim.structure(fx = fx)

 $model (Population correlation matrix) 
     V1   V2   V3   V4   V5   V6   V7  V8   V9
V1 1.00 0.72 0.63 0.00 0.00 0.00 0.00 0.0 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,.5,.3,.5,1,.2,.3,.2,1), ncol=3)
> cor.f3 <- sim.structure(fx,Phi)
> fx
```

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

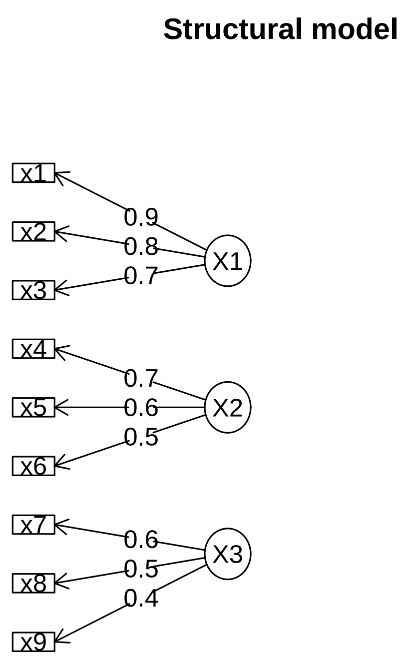

Figure 4: Three uncorrelated factors generated using the sim.structure function and drawn using structure.diagram.

```
> 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  #show the matrix

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

> Phis #show this one as well

   F1    F2    F3
F1 "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.

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

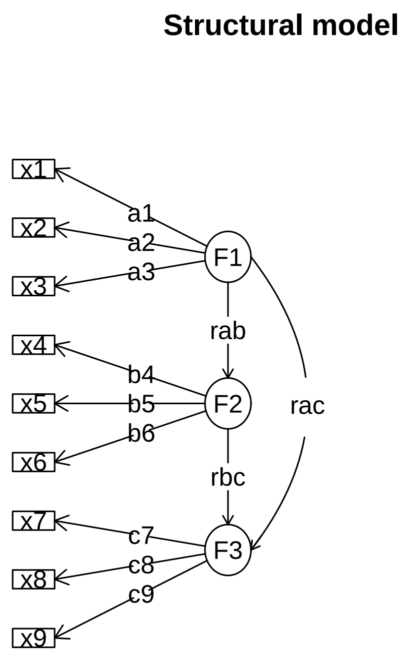

Figure 5: Three correlated factors with symbolic paths. Created using structure.diagram and structure.list and phi.list for ease of input.

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))

> #plot.new()
> mod.f3p <- structure.diagram(f3.p,cut=.2)
```

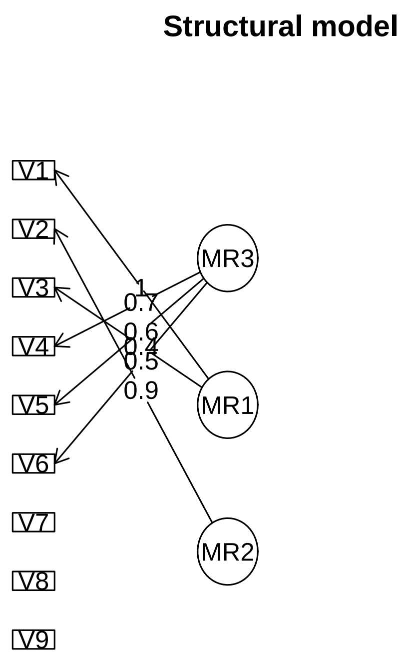

Figure 6: The empirically fitted structural model. Paths less than cut (.2 in this case, the default is .3) are not shown.

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.

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

```
     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).

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.

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

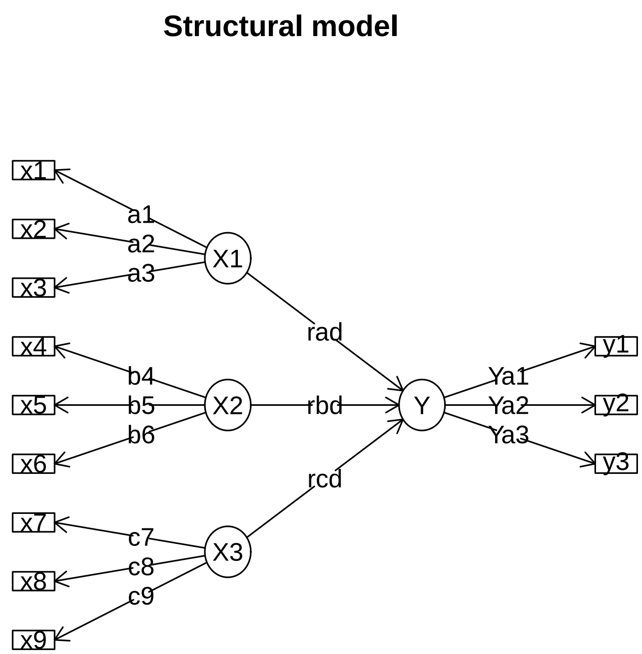

Figure 7: A symbolic structural model. Three independent latent variables are regressed on a latent Y.

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

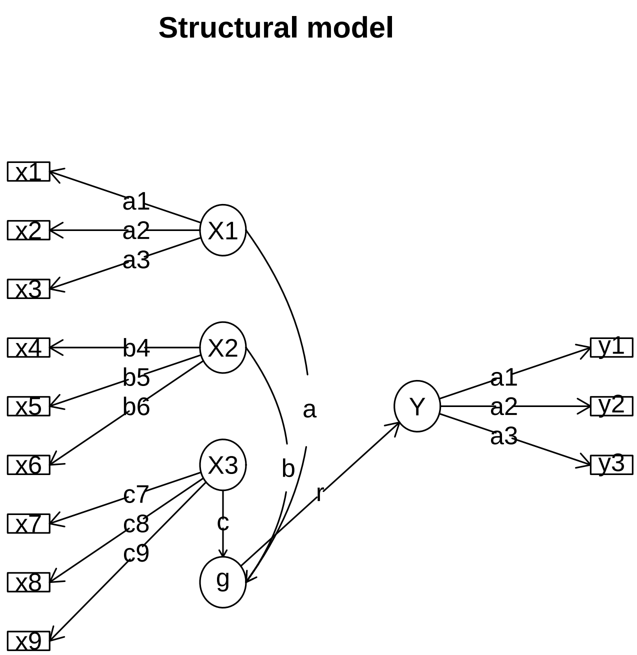

Figure 8: A symbolic structural model with a general factor and three group factors. The general factor is regressed on the latent Y variable.

3.1 Exploratory simple structure models

The technique of *principal components* provides a set of weighted linear composites that best 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)
Standardized loadings (pattern matrix) 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 component 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)
Standardized loadings (pattern matrix) 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 (RMSR) is  0
The df corrected root mean square of the residuals is  0

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

```
                                              MR1
Correlation of scores with factors            0.94
Multiple R square of scores with factors      0.88
Minimum correlation of possible factor scores  0.77
```

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

Consider the case of three underlying factors as seen in the bifact example above. Because the number of observations is not specified, there is no associated 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)
Standardized loadings (pattern matrix) based upon correlation matrix
    PC1  PC3  PC2   h2   u2
V1 0.75 0.27 0.21 0.69 0.31
V2 0.76 0.21 0.16 0.64 0.36
V3 0.78 0.11 0.10 0.63 0.37
V4 0.29 0.69 0.15 0.59 0.41
V5 0.20 0.71 0.11 0.56 0.44
V6 0.07 0.76 0.08 0.59 0.41
V7 0.26 0.16 0.70 0.58 0.42
V8 0.20 0.11 0.71 0.55 0.45
V9 0.00 0.06 0.73 0.53 0.47

                PC1  PC3  PC2
SS loadings    1.99 1.73 1.64
Proportion Var 0.22 0.19 0.18
Cumulative Var 0.22 0.41 0.60

Test of the hypothesis that 3 components are sufficient.

The degrees of freedom for the null model are  36  and the objective function was  1.88
The degrees of freedom for the model are 12  and the objective function was  0.72

Fit based upon off diagonal values = 0.88

> pa3

Factor Analysis using method =  pa
Call: fa(r = bifact, nfactors = 3, fm = "pa")
Standardized loadings (pattern matrix) based upon correlation matrix
   PA1 PA3  PA2   h2   u2
V1 0.8 0.0 0.00 0.64 0.36
V2 0.7 0.0 0.00 0.49 0.51
V3 0.6 0.0 0.00 0.36 0.64
V4 0.0 0.7 0.00 0.49 0.51
V5 0.0 0.6 0.00 0.36 0.64
V6 0.0 0.5 0.00 0.25 0.75
V7 0.0 0.0 0.69 0.48 0.52
```

```
V8 0.0 0.0 0.61 0.36 0.64
V9 0.0 0.0 0.40 0.16 0.84

                PA1  PA3  PA2
SS loadings    1.49 1.10 1.01
Proportion Var 0.17 0.12 0.11
Cumulative Var 0.17 0.29 0.40

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

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

Fit based upon off diagonal values = 1
Measures of factor score adequacy             
                                              PA1  PA3  PA2
Correlation of scores with factors            0.9 0.85 0.83
Multiple R square of scores with factors      0.8 0.72 0.69
Minimum correlation of possible factor scores  0.6 0.45 0.38
```

```
> ml3
```

```
Factor Analysis using method =  ml
Call: fa(r = bifact, nfactors = 3, fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
   ML1 ML3 ML2   h2   u2
V1 0.8 0.0 0.0 0.64 0.36
V2 0.7 0.0 0.0 0.49 0.51
V3 0.6 0.0 0.0 0.36 0.64
V4 0.0 0.7 0.0 0.49 0.51
V5 0.0 0.6 0.0 0.36 0.64
V6 0.0 0.5 0.0 0.25 0.75
V7 0.0 0.0 0.7 0.49 0.51
V8 0.0 0.0 0.6 0.36 0.64
V9 0.0 0.0 0.4 0.16 0.84

                ML1  ML3  ML2
SS loadings    1.49 1.10 1.01
Proportion Var 0.17 0.12 0.11
Cumulative Var 0.17 0.29 0.40

 With factor correlations of 
     ML1  ML3  ML2
ML1 1.00 0.72 0.63
ML3 0.72 1.00 0.56
ML2 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  1.88
The degrees of freedom for the model are 12  and the objective function was  0

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

Fit based upon off diagonal values = 1
Measures of factor score adequacy
                                               ML1  ML3  ML2
Correlation of scores with factors            0.90 0.85 0.83
Multiple R square of scores with factors      0.80 0.72 0.69
Minimum correlation of possible factor scores  0.61 0.45 0.38
```

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

```
     PC1  PC3  PC2  PA1  PA3  PA2  ML1  ML3  ML2
PC1 1.00 0.49 0.42 0.93 0.24 0.21 0.93 0.24 0.21
PC3 0.49 1.00 0.35 0.27 0.94 0.15 0.27 0.93 0.15
PC2 0.42 0.35 1.00 0.22 0.16 0.94 0.22 0.16 0.94
PA1 0.93 0.27 0.22 1.00 0.00 0.00 1.00 0.00 0.00
PA3 0.24 0.94 0.16 0.00 1.00 0.00 0.00 1.00 0.00
PA2 0.21 0.15 0.94 0.00 0.00 1.00 0.00 0.00 1.00
ML1 0.93 0.27 0.22 1.00 0.00 0.00 1.00 0.00 0.00
ML3 0.24 0.93 0.16 0.00 1.00 0.00 0.00 1.00 0.00
ML2 0.21 0.15 0.94 0.00 0.00 1.00 0.00 0.00 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
Standardized loadings (pattern matrix) based upon correlation matrix
   ML1 ML3 ML2   h2   u2
V1 0.8 0.0 0.0 0.64 0.36
V2 0.7 0.0 0.0 0.49 0.51
V3 0.6 0.0 0.0 0.36 0.64
V4 0.0 0.7 0.0 0.49 0.51
V5 0.0 0.6 0.0 0.36 0.64
V6 0.0 0.5 0.0 0.25 0.75
V7 0.0 0.0 0.7 0.49 0.51
V8 0.0 0.0 0.6 0.36 0.64
V9 0.0 0.0 0.4 0.16 0.84

                ML1  ML3  ML2
SS loadings    1.49 1.10 1.01
Proportion Var 0.17 0.12 0.11
Cumulative Var 0.17 0.29 0.40
    ML1 ML3 ML2
ML1   1   0   0
ML3   0   1   0
ML2   0   0   1
```

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)
```

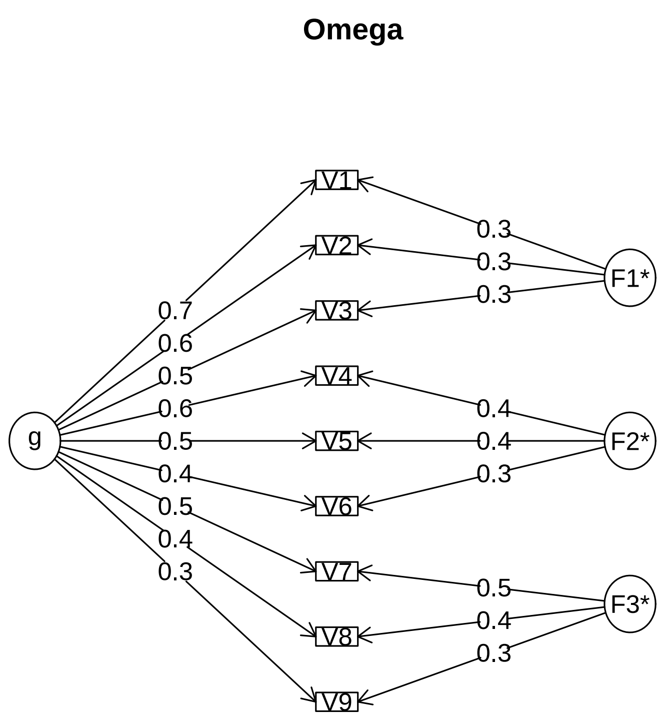

Figure 9: An exploratory bifactor solution to the nine variable problem

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)
```

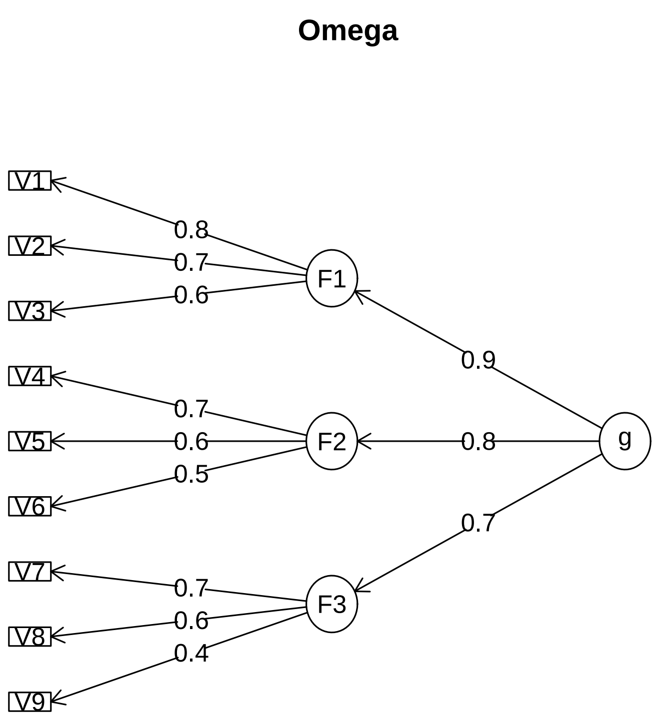

Figure 10: An exploratory hierarchical solution to the nine variable problem.

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:                 0.78
G.6:                   0.78
Omega Hierarchical:    0.7
Omega H asymptotic:    0.85
Omega Total            0.82

Schmid Leiman Factor loadings greater than  0.2
      g  F1*  F2*  F3*   h2   u2   p2
V1 0.72 0.35           0.64 0.36 0.81
V2 0.63 0.31           0.49 0.51 0.81
V3 0.54 0.26           0.36 0.64 0.81
V4 0.56      0.42      0.49 0.51 0.64
V5 0.48      0.36      0.36 0.64 0.64
V6 0.40      0.30      0.25 0.75 0.64
V7 0.49           0.50 0.49 0.51 0.49
V8 0.42           0.43 0.36 0.64 0.49
V9 0.28           0.29 0.16 0.84 0.49

With eigenvalues of:
   g  F1*  F2*  F3*
2.41 0.28 0.40 0.52

general/max  4.67   max/min =   1.82
mean percent general =  0.65    with sd =  0.14 and cv of  0.21

The degrees of freedom are 12  and the fit is  0

The root mean square of the residuals is  0
The df corrected root mean square of the residuals is  0

Compare this with the adequacy of just a general factor and no group factors
The degrees of freedom for just the general factor are 27  and the fit is  0.23

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

```
Measures of factor score adequacy
                                                  g   F1*   F2*   F3*
Correlation of scores with factors             0.86  0.47  0.57  0.64
Multiple R square of scores with factors       0.74  0.22  0.33  0.41
Minimum correlation of factor score estimates 0.47 -0.56 -0.35 -0.18
```

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.

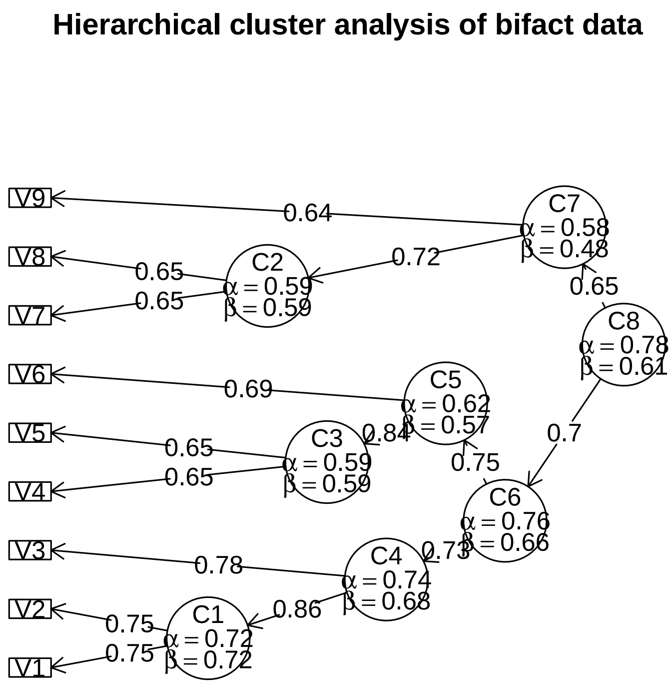

Figure 11: A hierarchical cluster analysis of the bifact data set using ICLUST

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 et al., 2011).

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   #show it

      Path      Parameter Value
 [1,] "X1->V1"  "a"       NA
 [2,] "X1->V2"  "a"       NA
 [3,] "X1->V3"  "a"       NA
 [4,] "X1->V4"  "a"       NA
 [5,] "V1<->V1" "x1e"     NA
 [6,] "V2<->V2" "x2e"     NA
 [7,] "V3<->V3" "x3e"     NA
 [8,] "V4<->V4" "x4e"     NA
 [9,] "X1<->X1" NA        "1"
attr(,"class")
[1] "mod"
```

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

 Model Chisquare =  6.6   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
 AIC =  17
 AICc =  7.2
 BIC =  30
 CAIC =  -21

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

 R-square for Endogenous Variables
  V1   V2   V3   V4
0.52 0.46 0.45 0.44

 Parameter Estimates
    Estimate Std Error z value Pr(>|z|)
a   0.69     0.063     10.9    1.2e-27  V1 <--- X1
x1e 0.43     0.081      5.3    1.3e-07  V1 <--> V1
x2e 0.56     0.098      5.7    1.2e-08  V2 <--> V2
x3e 0.58     0.100      5.7    9.3e-09  V3 <--> V3
x4e 0.59     0.102      5.8    7.5e-09  V4 <--> V4

 Iterations =  11
```

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  #show the model

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

> sem.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 =  0.99
 SRMR =  0.03
 AIC =  19
 AICc =  4.5
 BIC =  40
 CAIC =  -8.3

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

 R-square for Endogenous Variables
  V1   V2   V3   V4
0.69 0.44 0.39 0.35

 Parameter Estimates
    Estimate Std Error z value Pr(>|z|)
a   0.83     0.098     8.5     2.3e-17  V1 <--- X1
b   0.66     0.101     6.6     4.8e-11  V2 <--- X1
c   0.63     0.101     6.2     6.1e-10  V3 <--- X1
d   0.59     0.102     5.8     6.7e-09  V4 <--- X1
x1e 0.31     0.100     3.1     1.9e-03  V1 <--> V1
x2e 0.56     0.102     5.5     3.2e-08  V2 <--> V2
x3e 0.61     0.104     5.8     6.1e-09  V3 <--> V3
x4e 0.65     0.107     6.0     1.6e-09  V4 <--> V4

 Iterations =  12

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

LR Test for Difference Between Models

         Model Df Model Chisq Df LR Chisq Pr(>Chisq)
sem.cong        2      2.9417
sem.tau         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  #show the model

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

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

 Model Chisquare =  21   Df =  27 Pr(>Chisq) = 0.78
 Chisquare (null model) =  186   Df =  36
 Goodness-of-fit index =  0.95
 Adjusted goodness-of-fit index =  0.92
 RMSEA index =  0   90% CI: (NA, 0.054)
 Bentler-Bonnett NFI =  0.89
 Tucker-Lewis NNFI =  1.1
 Bentler CFI =  1
 SRMR =  0.053
 AIC =  57
 AICc =  30
 BIC =  104
 CAIC =  -130

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -0.33   -0.29   -0.19    0.04    0.00    1.89

 R-square for Endogenous Variables
   V1    V2    V3    V4    V5    V6    V7    V8    V9
0.564 0.452 0.338 0.329 0.252 0.180 0.257 0.198 0.093

 Parameter Estimates
    Estimate Std Error z value Pr(>|z|)
a   0.75     0.095     7.9     3.0e-15  V1 <--- X1
b   0.67     0.098     6.9     6.9e-12  V2 <--- X1
c   0.58     0.101     5.7     9.9e-09  V3 <--- X1
```

```
d   0.57     0.102     5.6     1.6e-08  V4 <--- X1
e   0.50     0.104     4.8     1.4e-06  V5 <--- X1
f   0.42     0.106     4.0     6.4e-05  V6 <--- X1
g   0.51     0.104     4.9     1.0e-06  V7 <--- X1
h   0.45     0.106     4.2     2.5e-05  V8 <--- X1
i   0.31     0.109     2.8     5.0e-03  V9 <--- X1
x1e 0.44     0.089     4.9     9.0e-07  V1 <--> V1
x2e 0.55     0.097     5.7     1.4e-08  V2 <--> V2
x3e 0.66     0.107     6.2     5.6e-10  V3 <--> V3
x4e 0.67     0.108     6.2     4.6e-10  V4 <--> V4
x5e 0.75     0.115     6.5     8.7e-11  V5 <--> V5
x6e 0.82     0.123     6.7     2.4e-11  V6 <--> V6
x7e 0.74     0.115     6.5     9.6e-11  V7 <--> V7
x8e 0.80     0.121     6.6     3.2e-11  V8 <--> V8
x9e 0.91     0.132     6.9     6.4e-12  V9 <--> V9

 Iterations =  11
```

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

```
      V1    V2    V3    V4    V5    V6    V7    V8    V9
V1  0.00  0.06  0.04 -0.03 -0.03 -0.03 -0.03 -0.03 -0.03
V2  0.06  0.00  0.03 -0.03 -0.04 -0.03 -0.03 -0.03 -0.03
V3  0.04  0.03  0.00 -0.03 -0.03 -0.03 -0.03 -0.03 -0.03
V4 -0.03 -0.03 -0.03  0.00  0.13  0.11 -0.02 -0.02 -0.02
V5 -0.03 -0.04 -0.03  0.13  0.00  0.09 -0.02 -0.02 -0.02
V6 -0.03 -0.03 -0.03  0.11  0.09  0.00 -0.02 -0.02 -0.02
V7 -0.03 -0.03 -0.03 -0.02 -0.02 -0.02  0.00  0.19  0.13
V8 -0.03 -0.03 -0.03 -0.02 -0.02 -0.02  0.19  0.00  0.10
V9 -0.03 -0.03 -0.03 -0.02 -0.02 -0.02  0.13  0.10  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 =  21.16848   Df =  27

       V1        V2        V3        V4        V5        V6        V7
0.7507129 0.6726412 0.5811209 0.5737425 0.5021915 0.4239908 0.5067957
       V8        V9       x1e       x2e       x3e       x4e       x5e
0.4450171 0.3052415 0.4364302 0.5475539 0.6622982 0.6708197 0.7478036
      x6e       x7e       x8e       x9e
0.8202314 0.7431581 0.8019593 0.9068284

 Iterations =  11
```

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. Rather than use the default rotation of `oblimin`, we force an orthogonal solution (even though we know it will be a poor solution).

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

```
 Model Chisquare =  54   Df =  27 Pr(>Chisq) = 0.0016
 Chisquare (null model) =  186   Df =  36
 Goodness-of-fit index =  0.89
 Adjusted goodness-of-fit index =  0.81
 RMSEA index =  0.1   90% CI: (0.06, 0.14)
 Bentler-Bonnett NFI =  0.71
 Tucker-Lewis NNFI =  0.76
 Bentler CFI =  0.82
 SRMR =  0.2
 AIC =  90
 AICc =  62
 BIC =  137
 CAIC =  -97

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0     0.0     2.0     1.6     2.6     4.0

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

 Parameter Estimates
     Estimate Std Error z value Pr(>|z|)
F1V1 0.80     0.11      7.2     7.1e-13  V1 <--- MR1
F1V2 0.70     0.11      6.4     1.3e-10  V2 <--- MR1
F1V3 0.60     0.11      5.6     1.9e-08  V3 <--- MR1
F2V4 0.70     0.14      4.9     9.4e-07  V4 <--- MR3
F2V5 0.60     0.13      4.5     6.3e-06  V5 <--- MR3
F2V6 0.50     0.12      4.0     5.4e-05  V6 <--- MR3
F3V7 0.70     0.17      4.2     3.1e-05  V7 <--- MR2
F3V8 0.60     0.15      3.9     8.8e-05  V8 <--- MR2
F3V9 0.40     0.13      3.2     1.6e-03  V9 <--- MR2
x1e  0.36     0.13      2.8     5.5e-03  V1 <--> V1
x2e  0.51     0.12      4.4     1.2e-05  V2 <--> V2
x3e  0.64     0.11      5.7     1.5e-08  V3 <--> V3
x4e  0.51     0.17      2.9     3.4e-03  V4 <--> V4
x5e  0.64     0.15      4.3     1.4e-05  V5 <--> V5
x6e  0.75     0.13      5.6     2.0e-08  V6 <--> V6
x7e  0.51     0.21      2.4     1.7e-02  V7 <--> V7
x8e  0.64     0.17      3.7     2.2e-04  V8 <--> V8
x9e  0.84     0.14      6.2     7.0e-10  V9 <--> V9
```

```
 Iterations =  24
```

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

```
     V1   V2   V3   V4   V5   V6   V7   V8   V9
V1 0.00 0.00 0.00 0.40 0.35 0.29 0.35 0.30 0.20
V2 0.00 0.00 0.00 0.35 0.30 0.25 0.31 0.26 0.18
V3 0.00 0.00 0.00 0.30 0.26 0.22 0.26 0.23 0.15
V4 0.40 0.35 0.30 0.00 0.00 0.00 0.27 0.24 0.16
V5 0.35 0.30 0.26 0.00 0.00 0.00 0.24 0.20 0.13
V6 0.29 0.25 0.22 0.00 0.00 0.00 0.20 0.17 0.11
V7 0.35 0.31 0.26 0.27 0.24 0.20 0.00 0.00 0.00
V8 0.30 0.26 0.23 0.24 0.20 0.17 0.00 0.00 0.00
V9 0.20 0.18 0.15 0.16 0.13 0.11 0.00 0.00 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

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

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

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

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

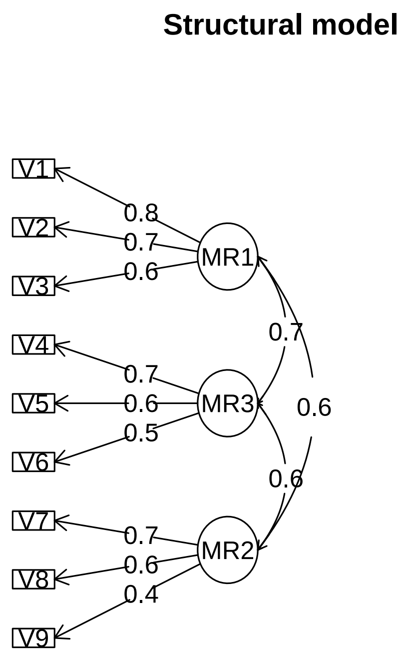

Figure 12: A three factor, oblique solution.

This makes much better sense, and in fact (as hoped) recovers the original structure.

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 `stdCoef` 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 `Thurstone` data set (which is "lazy loaded"). Using the commands just shown, it is possible to analyze this data set using a bifactor solution (Figure 13).

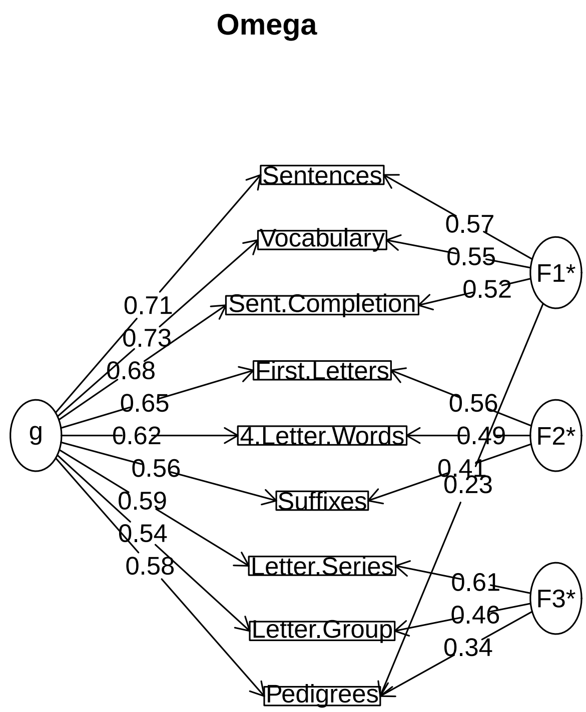

Figure 13: A bifactor solution to the Thurstone 9 variable problem. All items load on a general factor of ability, the residual factors account for the correlations between items within groups.

```
> sem.bi <- sem(om.th.bi$model,Thurstone,213) #use the model created by omega
> 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 =  0.99
 SRMR =  0.035
 AIC =  78
 AICc =  32
 BIC =  169
 CAIC =  -90

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -0.82   -0.33    0.00    0.03    0.16    1.80

 R-square for Endogenous Variables
      Sentences      Vocabulary Sent.Completion   First.Letters
           0.83            0.83            0.73            0.75
 4.Letter.Words        Suffixes   Letter.Series       Pedigrees
           0.61            0.48            0.85            0.50
   Letter.Group
           0.45

 Parameter Estimates
                   Estimate Std Error z value Pr(>|z|)
Sentences          0.77     0.071     10.88   1.5e-27
Vocabulary         0.79     0.070     11.35   7.5e-30
Sent.Completion    0.75     0.071     10.59   3.2e-26
First.Letters      0.61     0.071      8.61   7.1e-18
4.Letter.Words     0.60     0.071      8.42   3.7e-17
Suffixes           0.57     0.072      7.99   1.4e-15
Letter.Series      0.57     0.072      7.82   5.3e-15
Pedigrees          0.66     0.070      9.46   3.1e-21
Letter.Group       0.53     0.073      7.23   4.9e-13
F1*Sentences       0.49     0.081      5.99   2.1e-09
F1*Vocabulary      0.45     0.084      5.41   6.1e-08
F1*Sent.Completion 0.40     0.087      4.63   3.6e-06
F2*First.Letters   0.61     0.085      7.25   4.2e-13
F2*4.Letter.Words  0.51     0.081      6.21   5.3e-10
F2*Suffixes        0.39     0.078      5.05   4.4e-07
F3*Letter.Series   0.73     0.158      4.59   4.4e-06
F3*Pedigrees       0.25     0.087      2.84   4.4e-03
F3*Letter.Group    0.41     0.114      3.60   3.1e-04
e1                 0.17     0.034      5.06   4.2e-07
e2                 0.17     0.030      5.66   1.5e-08
e3                 0.27     0.033      8.10   5.7e-16
e4                 0.25     0.079      3.18   1.5e-03
e5                 0.39     0.063      6.13   8.7e-10
e6                 0.52     0.060      8.69   3.6e-18
e7                 0.15     0.219      0.68   4.9e-01
e8                 0.50     0.060      8.40   4.4e-17
e9                 0.55     0.085      6.52   6.8e-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
```

> stdCoef(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 `stdCoef` function helps see the structure. Alternatively, using the `omega` function on the `Thurstone` data will create the model for this particular data set.

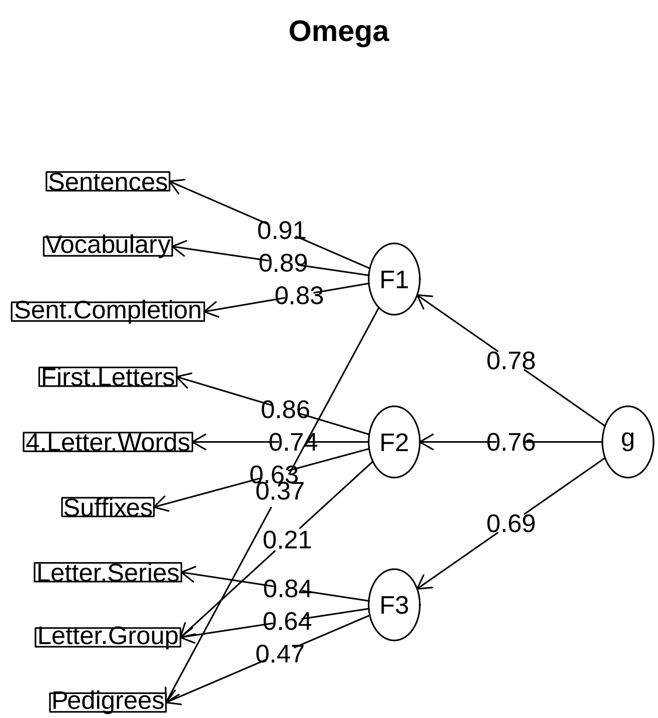

Figure 14: Hierarchical analysis of the Thurstone 9 variable problem using an exploratory algorithm can provide the appropriate sem code for analysis using the sem package.

```
> 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
 AIC =  80
 AICc =  43
 BIC =  151
 CAIC =  -114

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

 R-square for Endogenous Variables
            F1              F2              F3       Sentences
          0.68            0.61            0.66            0.82
    Vocabulary Sent.Completion   First.Letters  4.Letter.Words
          0.84            0.73            0.70            0.64
      Suffixes   Letter.Series       Pedigrees    Letter.Group
          0.49            0.61            0.52            0.49

 Parameter Estimates
                 Estimate Std Error z value Pr(>|z|)
gF1              1.44     0.257     5.6     1.8e-08
gF2              1.25     0.211     5.9     3.0e-09
gF3              1.41     0.269     5.2     1.7e-07
F1Sentences      0.52     0.063     8.2     2.7e-16
F1Vocabulary     0.52     0.063     8.2     2.2e-16
F1Sent.Completion 0.49    0.061     8.0     1.1e-15
F2First.Letters  0.52     0.061     8.5     1.4e-17
F24.Letter.Words 0.50     0.059     8.4     3.7e-17
F2Suffixes       0.44     0.056     7.8     4.9e-15
F3Letter.Series  0.45     0.066     6.9     7.0e-12
F3Pedigrees      0.42     0.062     6.7     1.9e-11
F3Letter.Group   0.41     0.061     6.6     3.0e-11
e1               0.18     0.028     6.4     1.8e-10
e2               0.16     0.028     5.9     2.9e-09
e3               0.27     0.033     8.0     1.2e-15
e4               0.30     0.051     5.9     3.4e-09
e5               0.36     0.053     6.9     4.4e-12
e6               0.51     0.060     8.5     2.0e-17
e7               0.39     0.059     6.6     4.8e-11
e8               0.48     0.062     7.7     1.1e-14
e9               0.51     0.063     8.0     1.5e-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 =  54

> stdCoef(sem.hi,digits=2)

                     Std. Estimate
1                gF1     0.8220754                            F1 <--- g
2                gF2     0.7817998                            F2 <--- g
3                gF3     0.8150140                            F3 <--- g
4        F1Sentences     0.9047111                    Sentences <--- F1
5       F1Vocabulary     0.9138214                   Vocabulary <--- F1
6  F1Sent.Completion     0.8560764              Sent.Completion <--- F1
7    F2First.Letters     0.8357617                First.Letters <--- F2
8   F24.Letter.Words     0.7971819               4.Letter.Words <--- F2
9         F2Suffixes     0.7025560                     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     0.1814979           Sentences <--> Sentences
14                e2     0.1649304         Vocabulary <--> Vocabulary
15                e3     0.2671331 Sent.Completion <--> Sent.Completion
16                e4     0.3015024     First.Letters <--> First.Letters
17                e5     0.3645010   4.Letter.Words <--> 4.Letter.Words
18                e6     0.5064151             Suffixes <--> Suffixes
19                e7     0.3903313     Letter.Series <--> Letter.Series
20                e8     0.4813697           Pedigrees <--> Pedigrees
21                e9     0.5051016       Letter.Group <--> Letter.Group
22                       0.3241920                          F1 <--> F1
23                       0.3887891                          F2 <--> F2
24                       0.3357521                          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)
sem.hi       24      38.196
sem.bi       18      24.216  6    13.98    0.02986 *
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 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.

```
> 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)
sem.holz.bi       63      147.66
sem.holz.hi       73      178.79 10   31.129  0.0005587 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

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:

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

Schmid Leiman Factor loadings greater than  0.2
                   g   F1*   F2*   F3*   h2   u2   p2
Sentences       0.71  0.57             0.82 0.18 0.61
Vocabulary      0.73  0.55             0.84 0.16 0.63
Sent.Completion 0.68  0.52             0.73 0.27 0.63
```

```
First.Letters   0.65        0.56       0.73 0.27 0.57
4.Letter.Words  0.62        0.49       0.63 0.37 0.61
Suffixes        0.56        0.41       0.50 0.50 0.63
Letter.Series   0.59             0.61 0.72 0.28 0.48
Pedigrees       0.58  0.23       0.34 0.50 0.50 0.66
Letter.Group    0.54             0.46 0.53 0.47 0.56

With eigenvalues of:
   g  F1*  F2*  F3*
3.58 0.96 0.74 0.71

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

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  NA NA
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  0.199 0.243
BIC =  162.35

Measures of factor score adequacy
                                                 g  F1*  F2*  F3*
Correlation of scores with factors            0.86 0.73 0.72 0.75
Multiple R square of scores with factors      0.74 0.54 0.52 0.56
Minimum correlation of factor score estimates 0.49 0.08 0.03 0.11

 Omega Hierarchical from a confirmatory model using sem =  0.79
 Omega Total  from a confirmatory model using sem =  0.93
With loadings of
                   g  F1*  F2*  F3*   h2   u2
Sentences       0.77 0.49           0.83 0.17
Vocabulary      0.79 0.45           0.83 0.17
Sent.Completion 0.75 0.40           0.73 0.27
First.Letters   0.61      0.61      0.75 0.25
4.Letter.Words  0.60      0.51      0.61 0.39
Suffixes        0.57      0.39      0.48 0.52
Letter.Series   0.57           0.73 0.85 0.15
Pedigrees       0.66           0.25 0.50 0.50
Letter.Group    0.53           0.41 0.45 0.55

With eigenvalues of:
   g  F1*  F2*  F3*
3.88 0.61 0.79 0.76
```

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 ω_h are not identical from the EFA and CFA models. The CFA solution yields higher values of ω_h because, by forcing a pure cluster solution (no cross loadings), the correlations between the factors is forced to be through the g factor.

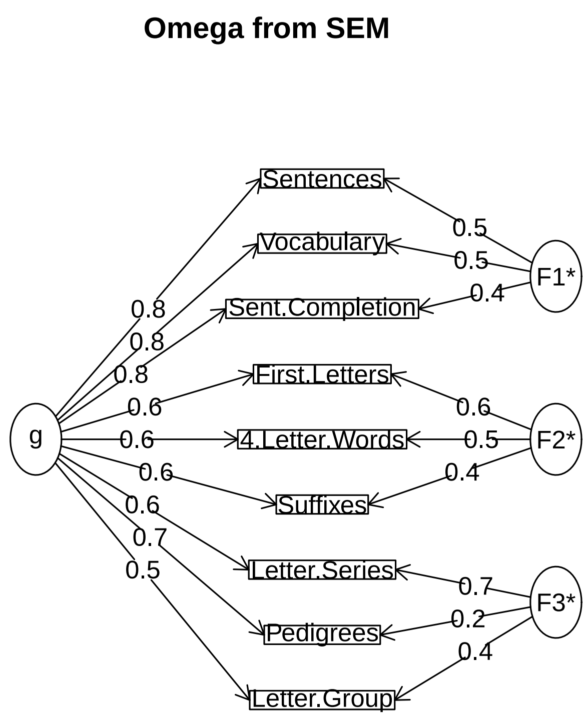

Figure 15: Confirmatory Omega structure using `omegaSem`

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

```
> sessionInfo()

R version 2.15.1 (2012-06-22)
Platform: i386-apple-darwin9.8.0/i386 (32-bit)

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

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

other attached packages:
[1] sem_2.1-1            matrixcalc_1.0-1     GPArotation_2010.07-1
[4] MASS_7.3-18          psych_1.2.8

loaded via a namespace (and not attached):
[1] tools_2.15.1
```

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.

Fox, J., Byrnes, J., with contributions from Michael Culbertson, Friendly, M., Kramer, A., and Monette., G. (2011). *sem: Structural Equation Models*. R package version 2.1-1.

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. (2012). *psych: Procedures for Personality and Psychological Research*. Northwestern University, Evanston. R package version 1.2.8.

Rosseel, Y. (2011). *lavaan: Latent Variable Analysis*. R package version 0.4-11.

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.