# Using the psych package to generate and test structural models

William Revelle
Northwestern University

Contents



---

Written to accompany the psych package.
Comments should be directed to William Revelle
`revelle@northwestern.edu`

# The psych package

## *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 complete user's manual and the relevant help pages. 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, 2008).

## *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, these two libraries are required for these examples. *Rgraphviz* is used for the graphical displays, but the analyses themselves require only the *sem* package to do the structural modeling

# 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.

## *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.

```
> 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 0.65 0.69 0.62
V2 0.65 1.00 0.71 0.65
V3 0.69 0.71 1.00 0.59
V4 0.62 0.65 0.59 1.00
```

```
> tau.samp <- sim.congeneric(loads = c(0.8, 0.8, 0.8, 0.8), N = 100, short = FALSE)
> tau.samp
```

```
 $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 0.68 0.66 0.68
V2 0.68 1.00 0.63 0.71
V3 0.66 0.63 1.00 0.64
V4 0.68 0.71 0.64 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 is 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 0.61 0.34 0.40
V2 0.61 1.00 0.42 0.27
V3 0.34 0.42 1.00 0.15
V4 0.40 0.27 0.15 1.00
```

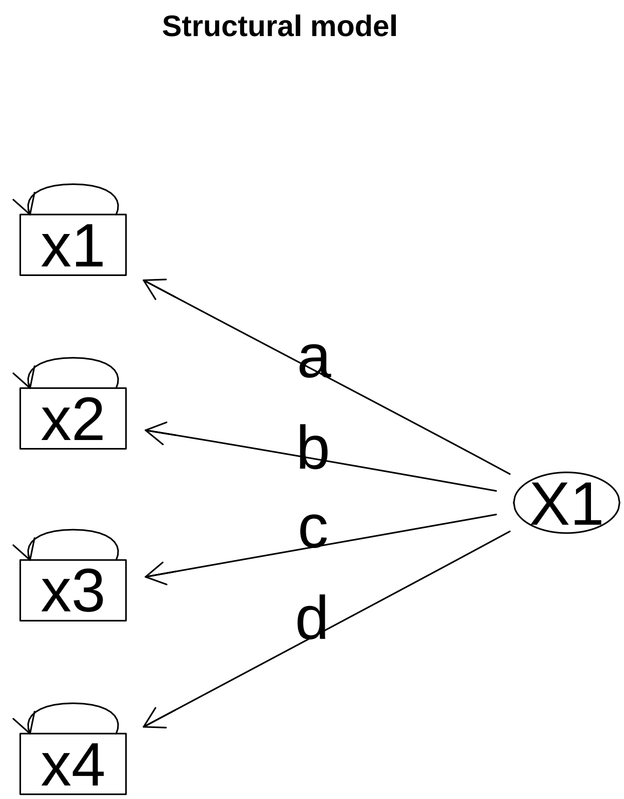


*Figure 1.* Tau equivalent tests are special cases of congeneric tests. Tau equivalence assumes a=b=c=d

*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 & Swineford (1937). A helpful description of what makes a good general factor is that of Jensen & Weng (1994).

```
> 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), nco
> 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) or *hierarchical* (Figure 3) factor solution.

### *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 & 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.

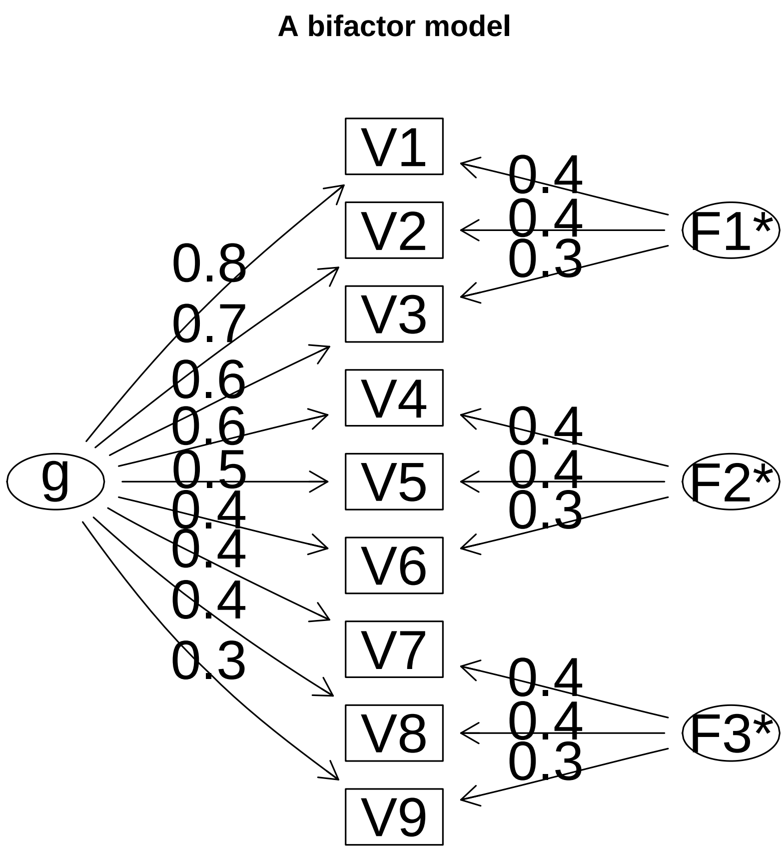


*Figure 2.* A bifactor solution represents each test in terms of a general factor and a residualized group factor.

*sim.structural*

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.

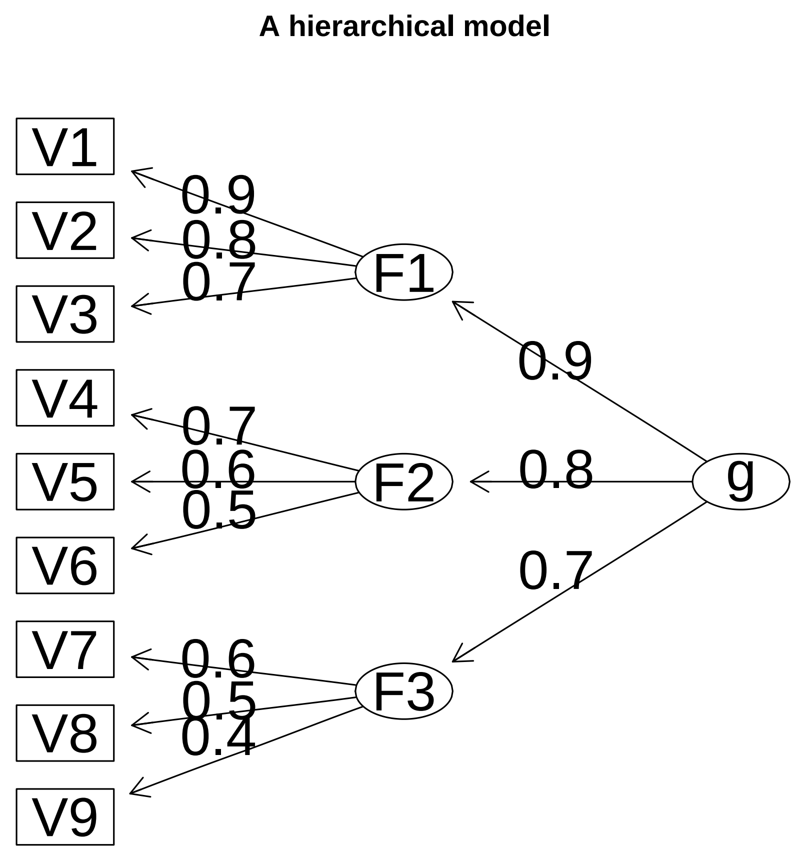


*Figure 3*. A hierarchical factor solution has g as a second order factor accounting for the correlations between the first order factors.

Consider the following examples:

1. $\vec{f_x}$ is a vector implies a congeneric model:

```
> fx <- c(0.9, 0.8, 0.7, 0.6)
> cong1 <- sim.structural(f = fx)
> cong1
 $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. $\vec{f}_x$ is a matrix implies an independent factors model:

```
> 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 =
> three.fact <- sim.structural(f = fx)
> three.fact
 $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
```

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)
> corf3 <- sim.structural(f = fx, Phi = Phi)
> fx
      [,1] [,2] [,3]
 [1,]  0.9  0.0  0.0
 [2,]  0.8  0.0  0.0
 [3,]  0.7  0.0  0.0
 [4,]  0.0  0.7  0.0
 [5,]  0.0  0.6  0.0
 [6,]  0.0  0.5  0.0
 [7,]  0.0  0.0  0.6
 [8,]  0.0  0.0  0.5
 [9,]  0.0  0.0  0.4
> Phi
     [,1] [,2] [,3]
[1,]  1.0  0.5  0.3
[2,]  0.5  1.0  0.2
[3,]  0.3  0.2  1.0
> corf3
```

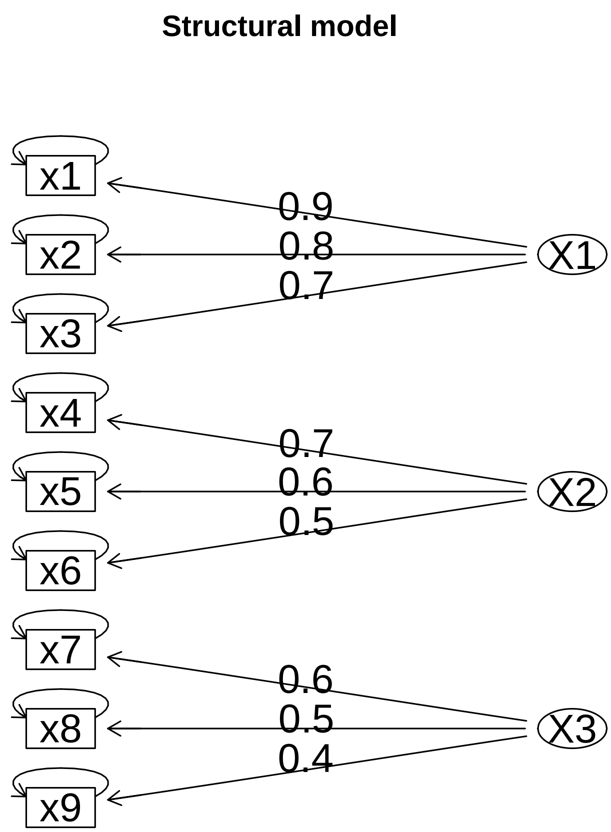


*Figure 4.* default

```
 $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
```

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.

4. $\vec{f}_x$ and $\vec{f}_y$ are matrices, and Phi ***neI*** represents their correlations.

```
> 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 =
> fy <- c(0.6, 0.5, 0.4)
> Phi <- matrix(c(1, 0.5, 0.3, 0.1, 0.5, 1, 0.2, 0.4, 0.3, 0.2, 1, 0.4, 0.1, 0.4, 0.4, 1),
> ls <- sim.structural(fx, fy, Phi)
> ls
 $model (Population correlation matrix)
       V1    V2    V3    V4    V5   V6    V7   V8    V9   V10   V11   V12
V1  1.000 0.720 0.630 0.315 0.270 0.23 0.162 0.14 0.108 0.054 0.045 0.036
V2  0.720 1.000 0.560 0.280 0.240 0.20 0.144 0.12 0.096 0.048 0.040 0.032
V3  0.630 0.560 1.000 0.245 0.210 0.17 0.126 0.10 0.084 0.042 0.035 0.028
V4  0.315 0.280 0.245 1.000 0.420 0.35 0.084 0.07 0.056 0.168 0.140 0.112
V5  0.270 0.240 0.210 0.420 1.000 0.30 0.072 0.06 0.048 0.144 0.120 0.096
V6  0.225 0.200 0.175 0.350 0.300 1.00 0.060 0.05 0.040 0.120 0.100 0.080
V7  0.162 0.144 0.126 0.084 0.072 0.06 1.000 0.30 0.240 0.144 0.120 0.096
V8  0.135 0.120 0.105 0.070 0.060 0.05 0.300 1.00 0.200 0.120 0.100 0.080
V9  0.108 0.096 0.084 0.056 0.048 0.04 0.240 0.20 1.000 0.096 0.080 0.064
V10 0.054 0.048 0.042 0.168 0.144 0.12 0.144 0.12 0.096 1.000 0.300 0.240
V11 0.045 0.040 0.035 0.140 0.120 0.10 0.120 0.10 0.080 0.300 1.000 0.200
V12 0.036 0.032 0.028 0.112 0.096 0.08 0.096 0.08 0.064 0.240 0.200 1.000

$reliability (population reliability)
 [1] 0.81 0.64 0.49 0.49 0.36 0.25 0.36 0.25 0.16 0.36 0.25 0.16
```

This may be seen by specifying a symbolic model seen in Figure 5.

## Functions for analyzing structure

Given a correlation matrix such as seen above for congeneric or bifactor models, 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.

### *Exploratory 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

```
> 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"
> corf3.mod <- structure.graph(fxs, Phi = Phis)
```

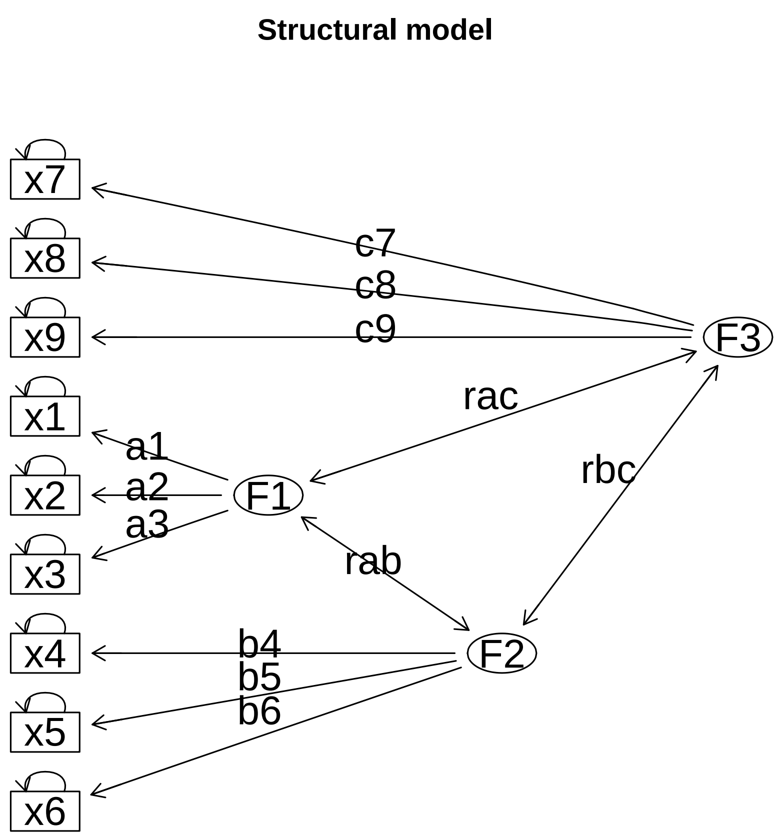


*Figure 5.* Three correlated factors with symbolic paths. Created using structure.graph and structure.list and phi.list for ease of input.

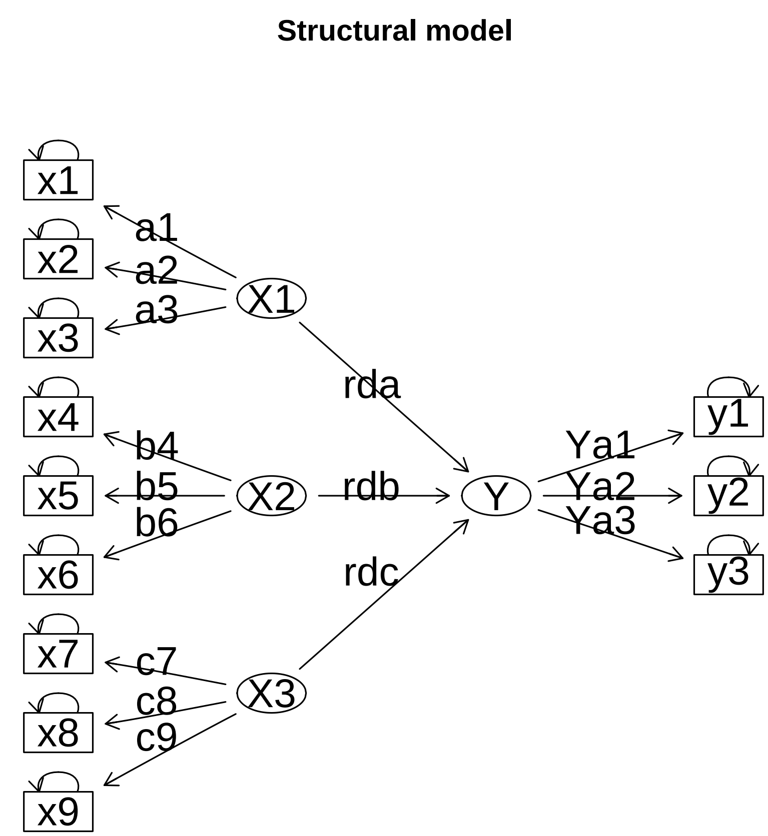


*Figure 6.* A symbolic structural model. Three independent latent variables are regressed on a latent Y.

*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)

  V  PA1
1 1 0.89
2 2 0.85
3 3 0.80
4 4 0.73
```

```
                PA1
SS loadings    2.69
Proportion Var 0.67

Test of the hypothesis that 1 factor is sufficient.

The degrees of freedom for the model is 2  and the fit was  0.14
```

```
> factor.pa(cong1$model)
```

```
  V PA1
1 1 0.9
2 2 0.8
3 3 0.7
4 4 0.6

                PA1
SS loadings    2.30
Proportion Var 0.58

Test of the hypothesis that 1 factor is sufficient.

The degrees of freedom for the model is 2  and the fit was  0
```

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 factor analysis, does.

Consider the case of three underlying factors as seen in the bifact example above.

```
> pc3 <- principal(bifact, 3)
> pa3 <- factor.pa(bifact, 3)
> ml3 <- factanal(covmat = bifact, factors = 3)
> pc3
```

```
   V  PC1  PC3  PC2
V1 1 0.82
V2 2 0.82
V3 3 0.82
V4 4 0.32 0.68
V5 5      0.70
V6 6      0.77
```

```
V7 7           0.66
V8 8           0.68
V9 9           0.71

                PC1  PC3  PC2
SS loadings    2.26 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 model is 12  and the fit was  0.71
```

> *pa3*

```
   V  PA1   PA3  PA2
V1 1 0.78 -0.34
V2 2 0.70 -0.30
V3 3 0.61
V4 4      -0.63
V5 5      -0.54
V6 6      -0.45
V7 7            0.55
V8 8            0.47
V9 9            0.37

                PA1  PA3  PA2
SS loadings    1.66 1.21 0.93
Proportion Var 0.18 0.13 0.10
Cumulative Var 0.18 0.32 0.42

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the model is 12  and the fit was  0
```

> *ml3*

```
Call:
factanal(factors = 3, covmat = bifact)

Uniquenesses:
  V1   V2   V3   V4   V5   V6   V7   V8   V9
0.19 0.36 0.51 0.51 0.64 0.75 0.64 0.75 0.84
```

```
Loadings:
   Factor1 Factor2 Factor3
V1 0.785   0.336   0.286
V2 0.697   0.298   0.254
V3 0.610   0.261   0.222
V4 0.242   0.631   0.182
V5 0.207   0.541   0.156
V6 0.173   0.451   0.130
V7 0.167   0.148   0.557
V8 0.139   0.124   0.464
V9 0.112           0.371

               Factor1 Factor2 Factor3
SS loadings      1.666   1.212   0.933
Proportion Var   0.185   0.135   0.104
Cumulative Var   0.185   0.320   0.423

The degrees of freedom for the model is 12 and the fit was 0
```

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

```
     PA1   PA3  PA2
PC1 0.99 -0.70 0.65
PC3 0.57 -0.96 0.51
PC2 0.45 -0.43 0.95
```

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

```
    Factor1 Factor2 Factor3
PA1    1.00    0.72    0.67
PA3   -0.72   -1.00   -0.63
PA2    0.67    0.63    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
```

```
   V Factor1 Factor2 Factor3
V1 1  0.8329
```

```
V2 2  0.7403
V3 3  0.6478
V4 4          0.6913
V5 5          0.5925
V6 6          0.4938
V7 7                   0.598
V8 8                   0.498
V9 9                   0.399

               Factor1 Factor2 Factor3
SS loadings       1.66    1.08    0.77
Proportion Var    0.18    0.12    0.09
Cumulative Var    0.18    0.30    0.39

 With factor correlations of
        Factor1 Factor2 Factor3
Factor1    1.00    0.67    0.59
Factor2    0.67    1.00    0.55
Factor3    0.59    0.55    1.00
```

*Hierarchical models*

An exploratory hierarchical model can be applied to this data structure using the `omega` function. Graphic options include drawing a Schmid - Leiman bifactor solution (Figure 7) or drawing a hierarchical factor solution f(Figure 8).

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
Alpha:  0.7899659
Lambda.6:
Omega Hierarchical:   0.715484
Omega Total   0.828264

Schmid Leiman Factor loadings greater than  0.2
   g    F1*  F2*  F3*  h2   u2
```

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

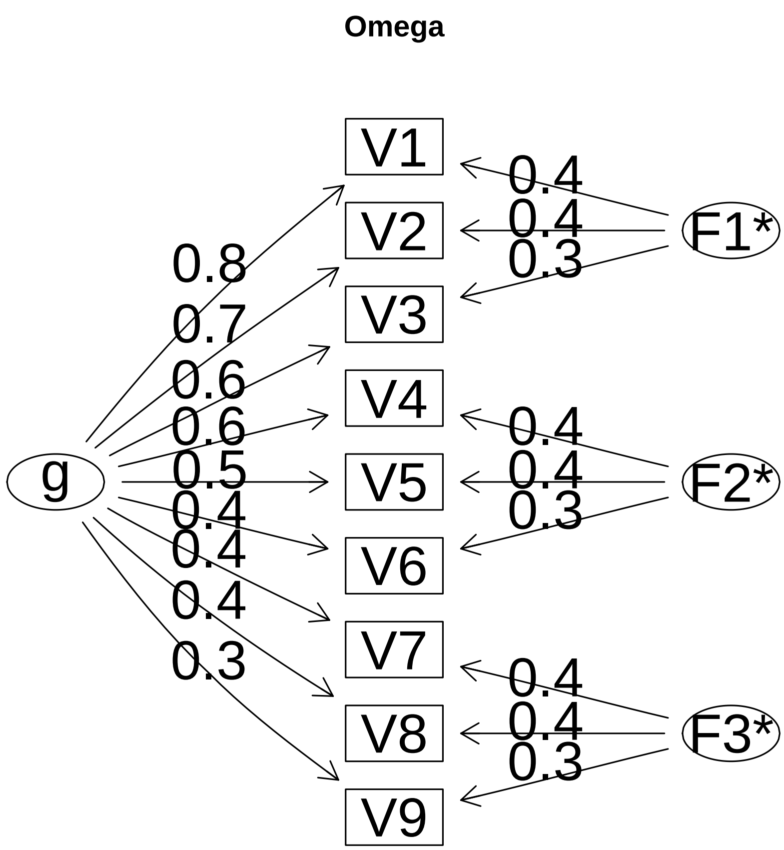


*Figure 7.* An exploratory bifactor solution to the nine variable problem

```
V1 0.81 0.39           0.81
V2 0.72 0.35           0.64 0.36
V3 0.63 0.31           0.49 0.51
V4 0.56      0.42      0.49 0.51
V5 0.48      0.36      0.36 0.64
V6 0.40      0.30      0.25 0.75
V7 0.42           0.43 0.36 0.64
V8 0.35           0.36 0.25 0.75
V9 0.28           0.29      0.84

With eigenvalues of:
```

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

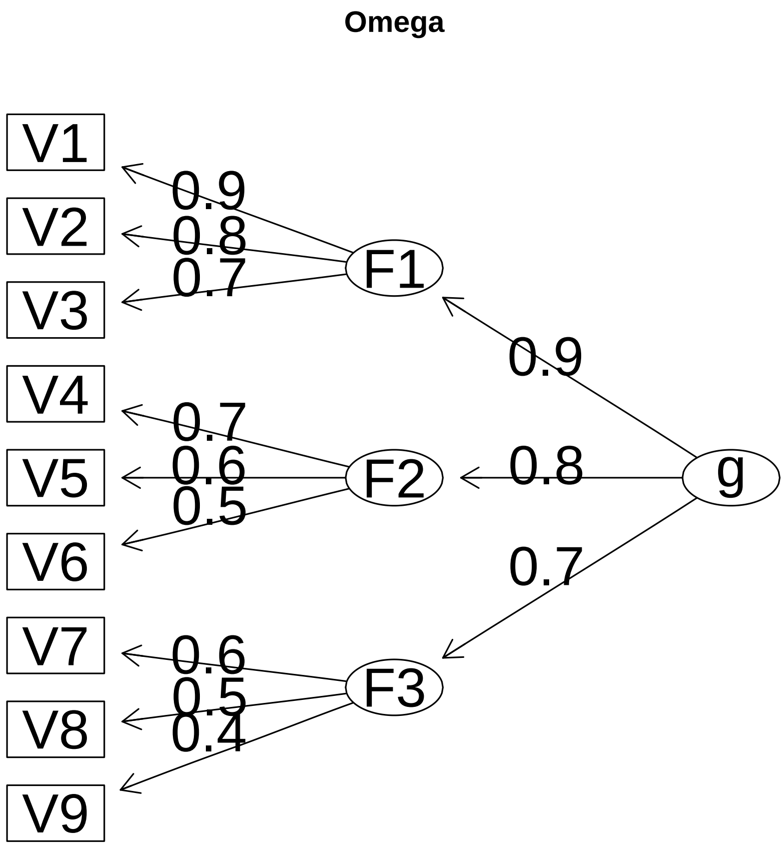


*Figure 8.* An exploratory hierarchical solution to the nine variable problem

```
   g  F1*  F2*  F3*
2.65 0.37 0.40 0.40

general/max  6.66   max/min =   1.08
The degrees of freedom for the model is 12  and the fit was  0
```

Yet one more way to show the hierarchical structure of a data set is to consider hierarchical cluster analysis using the `ICLUST` algorithm (Figure 9).

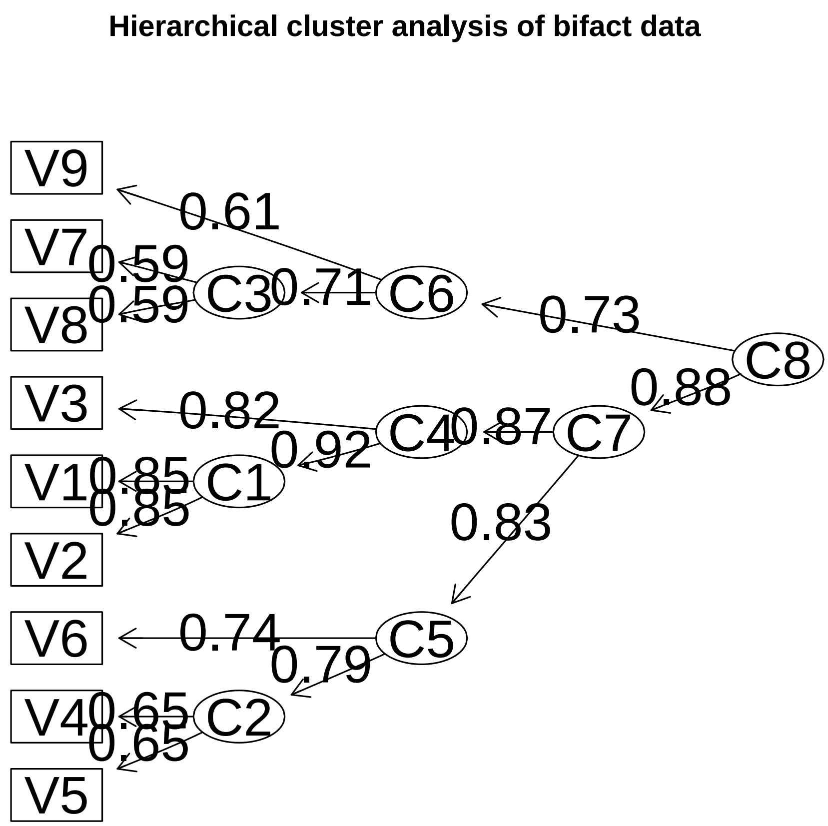


*Figure 9.* A hierarchical cluster analysis of the bifact data set using ICLUST

## Confirmatory models

Although the exploratory models shown above do estimate the goodness of fit of the model and compare the residual matrix to a zero matrix using a $\chi^2$ statistic, they estimate more parameters than are necessary if there is indeed a simple structure, and they do not allow for tests of competing models. The `sem` function in the *sem* package by John Fox allows for confirmatory tests. The interested reader is referred to the *sem* manual for more detail (Fox, 2008).

### *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.graph`, `structure.sem`, and `omega.graph` may be used as a front end to `sem`.

### *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. Either the `structure.graph` function which requires `Rgraphviz` or the `structure.sem` function may be used.

The following example tests the hypothesis (which is actually false) that the correlations found in the cong data set (see **??** 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.graph(c("a", "a", "a", "a"), labels = paste("V", 1:4, sep = ""))
> mod.tau
```

```
      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"
```

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

```
 Model Chisquare =  16.580   Df =  5 Pr(>Chisq) = 0.0053688
 Chisquare (null model) =  83.639   Df =  6
 Goodness-of-fit index =  0.92482
 Adjusted goodness-of-fit index =  0.84965
 RMSEA index =  0.15295   90% CI: (0.075456, 0.23745)
 Bentler-Bonnett NFI =  0.80177
 Tucker-Lewis NNFI =  0.82102
```

```
 Bentler CFI =  0.85085
 SRMR =  0.14708
 BIC =  -6.4457

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -2.120  -1.170  -0.358  -0.273   0.234   2.000

 Parameter Estimates
    Estimate Std Error z value Pr(>|z|)
a   0.64052  0.065250  9.8163  0.0000e+00 V1 <--- X1
x1e 0.47151  0.090509  5.2095  1.8933e-07 V1 <--> V1
x2e 0.49253  0.092883  5.3027  1.1412e-07 V2 <--> V2
x3e 0.73645  0.123996  5.9393  2.8617e-09 V3 <--> V3
x4e 0.78517  0.131206  5.9843  2.1735e-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 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"

> sem.cong <- sem(mod.cong, cong, 100)
> summary(sem.cong)

 Model Chisquare =  3.9815   Df =  2 Pr(>Chisq) = 0.13659
 Chisquare (null model) =  83.639   Df =  6
 Goodness-of-fit index =  0.9793
 Adjusted goodness-of-fit index =  0.89653
 RMSEA index =  0.10004   90% CI: (NA, 0.24494)
```

```
 Bentler-Bonnett NFI =  0.9524
 Tucker-Lewis NNFI =  0.92343
 Bentler CFI =  0.97448
 SRMR =  0.038429
 BIC =  -5.2288

 Normalized Residuals
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.5330 -0.3910 -0.0177 -0.0372  0.1300  0.6150

 Parameter Estimates
    Estimate Std Error z value Pr(>|z|)
a   0.80803  0.11395   7.0910  1.3318e-12 V1 <--- X1
b   0.75677  0.11291   6.7025  2.0484e-11 V2 <--- X1
c   0.46427  0.11199   4.1457  3.3874e-05 V3 <--- X1
d   0.43180  0.11054   3.9063  9.3728e-05 V4 <--- X1
x1e 0.34709  0.13633   2.5460  1.0897e-02 V1 <--> V1
x2e 0.42730  0.12800   3.3383  8.4298e-04 V2 <--> V2
x3e 0.78446  0.12442   6.3049  2.8841e-10 V3 <--> V3
x4e 0.81355  0.12514   6.5013  7.9653e-11 V4 <--> V4

 Iterations =  14

> anova(sem.cong, sem.tau)

LR Test for Difference Between Models

        Model Df Model Chisq Df LR Chisq Pr(>Chisq)
Model 1        2      3.9815
Model 2        5     16.5802  3  12.5986    0.00559 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

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

```
      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"
```

```
> bifact <- round(bifact, 5)
> sem.one <- sem(mod.one, bifact, 100)
> summary(sem.one)
```

```
 Model Chisquare =  18.729   Df =  27 Pr(>Chisq) = 0.87967
 Chisquare (null model) =  234.74   Df =  36
 Goodness-of-fit index =  0.95526
 Adjusted goodness-of-fit index =  0.92543
 RMSEA index =  0   90% CI: (NA, 0.039523)
 Bentler-Bonnett NFI =  0.92022
 Tucker-Lewis NNFI =  1.0555
 Bentler CFI =  1
 SRMR =  0.052506
 BIC =  -105.61

 Normalized Residuals
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-2.67e-01 -1.85e-01 -1.40e-06  1.37e-01  1.20e-01  1.61e+00

 Parameter Estimates
    Estimate Std Error z value Pr(>|z|)
```

```
a   0.88014  0.084098  10.4657 0.0000e+00 V1 <--- X1
b   0.79786  0.087665   9.1013 0.0000e+00 V2 <--- X1
c   0.69867  0.092309   7.5688 3.7748e-14 V3 <--- X1
d   0.54016  0.099013   5.4555 4.8843e-08 V4 <--- X1
e   0.46911  0.101142   4.6381 3.5161e-06 V5 <--- X1
f   0.39443  0.102945   3.8315 1.2738e-04 V6 <--- X1
g   0.40361  0.102583   3.9344 8.3390e-05 V7 <--- X1
h   0.34005  0.103944   3.2714 1.0701e-03 V8 <--- X1
i   0.27422  0.105061   2.6101 9.0526e-03 V9 <--- X1
x1e 0.22535  0.061293   3.6765 2.3644e-04 V1 <--> V1
x2e 0.36342  0.068545   5.3019 1.1461e-07 V2 <--> V2
x3e 0.51186  0.083791   6.1087 1.0042e-09 V3 <--> V3
x4e 0.70822  0.107282   6.6015 4.0701e-11 V4 <--> V4
x5e 0.77993  0.115697   6.7412 1.5708e-11 V5 <--> V5
x6e 0.84442  0.123326   6.8471 7.5369e-12 V6 <--> V6
x7e 0.83710  0.122367   6.8409 7.8686e-12 V7 <--> V7
x8e 0.88437  0.128072   6.9053 5.0109e-12 V8 <--> V8
x9e 0.92481  0.132971   6.9549 3.5274e-12 V9 <--> V9

 Iterations =  14
```

*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)
> summary(sem.f1)
```

```
 Model Chisquare =  18.729   Df =  27 Pr(>Chisq) = 0.87967
 Chisquare (null model) =  234.74   Df =  36
 Goodness-of-fit index =  0.95526
 Adjusted goodness-of-fit index =  0.92543
 RMSEA index =  0   90% CI: (NA, 0.039523)
 Bentler-Bonnett NFI =  0.92022
 Tucker-Lewis NNFI =  1.0555
 Bentler CFI =  1
 SRMR =  0.052506
 BIC =  -105.61
```

```
 Normalized Residuals
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-2.67e-01 -1.85e-01 -1.40e-06  1.37e-01  1.20e-01  1.61e+00

 Parameter Estimates
    Estimate Std Error z value Pr(>|z|)
V1  0.88014  0.084098  10.4657 0.0000e+00 V1 <--- Factor1
V2  0.79786  0.087665   9.1013 0.0000e+00 V2 <--- Factor1
V3  0.69867  0.092309   7.5688 3.7748e-14 V3 <--- Factor1
V4  0.54016  0.099013   5.4555 4.8843e-08 V4 <--- Factor1
V5  0.46911  0.101142   4.6381 3.5161e-06 V5 <--- Factor1
V6  0.39443  0.102945   3.8315 1.2738e-04 V6 <--- Factor1
V7  0.40361  0.102583   3.9344 8.3390e-05 V7 <--- Factor1
V8  0.34005  0.103944   3.2714 1.0701e-03 V8 <--- Factor1
V9  0.27422  0.105061   2.6101 9.0526e-03 V9 <--- Factor1
x1e 0.22535  0.061293   3.6765 2.3644e-04 V1 <--> V1
x2e 0.36342  0.068545   5.3019 1.1461e-07 V2 <--> V2
x3e 0.51186  0.083791   6.1087 1.0042e-09 V3 <--> V3
x4e 0.70822  0.107282   6.6015 4.0701e-11 V4 <--> V4
x5e 0.77993  0.115697   6.7412 1.5708e-11 V5 <--> V5
x6e 0.84442  0.123326   6.8471 7.5369e-12 V6 <--> V6
x7e 0.83710  0.122367   6.8409 7.8686e-12 V7 <--> V7
x8e 0.88437  0.128072   6.9053 5.0109e-12 V8 <--> V8
x9e 0.92481  0.132971   6.9549 3.5274e-12 V9 <--> V9

 Iterations =  14
```

*Specifying a three factor model*

An alternative model is to extract three factors and try this solution. The `factor.pa` factor analysis function is used for variety.

```
> f3 <- factor.pa(bifact, 3)
> mod.f3 <- structure.sem(f3)
> sem.f3 <- sem(mod.f3, bifact, 100)
> summary(sem.f3)
```

```
 Model Chisquare =  49.362   Df =  26 Pr(>Chisq) = 0.0037439
 Chisquare (null model) =  234.74   Df =  36
 Goodness-of-fit index =  0.89584
 Adjusted goodness-of-fit index =  0.81972
 RMSEA index =  0.095268   90% CI: (0.053304, 0.13543)
```

```
 Bentler-Bonnett NFI =  0.78972
 Tucker-Lewis NNFI =  0.83724
 Bentler CFI =  0.88245
 SRMR =  0.19571
 BIC =  -70.373

 Normalized Residuals
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-2.04e-05  1.92e-05  1.76e+00  1.66e+00  2.63e+00  4.01e+00

 Parameter Estimates
     Estimate Std Error z value Pr(>|z|)
F1V1 0.79231  0.093980  8.4306  0.0000e+00 V1 <--- PA1
F2V1 0.23013  0.089392  2.5744  1.0043e-02 V1 <--- PA3
F1V2 0.80000  0.093251  8.5790  0.0000e+00 V2 <--- PA1
F1V3 0.70000  0.095002  7.3683  1.7275e-13 V3 <--- PA1
F2V4 0.70000  0.129238  5.4164  6.0827e-08 V4 <--- PA3
F2V5 0.60000  0.123717  4.8498  1.2359e-06 V5 <--- PA3
F2V6 0.50000  0.120027  4.1657  3.1037e-05 V6 <--- PA3
F3V7 0.60000  0.189530  3.1657  1.5470e-03 V7 <--- PA2
F3V8 0.50000  0.167439  2.9862  2.8250e-03 V8 <--- PA2
F3V9 0.40000  0.146908  2.7228  6.4733e-03 V9 <--- PA2
x1e  0.19428  0.073779  2.6333  8.4554e-03 V1 <--> V1
x2e  0.36000  0.085408  4.2150  2.4973e-05 V2 <--> V2
x3e  0.51000  0.089431  5.7028  1.1787e-08 V3 <--> V3
x4e  0.51000  0.151833  3.3589  7.8242e-04 V4 <--> V4
x5e  0.64000  0.135626  4.7188  2.3721e-06 V5 <--> V5
x6e  0.75000  0.130148  5.7627  8.2777e-09 V6 <--> V6
x7e  0.64000  0.219308  2.9183  3.5198e-03 V7 <--> V7
x8e  0.75000  0.174853  4.2893  1.7921e-05 V8 <--> V8
x9e  0.84000  0.148755  5.6468  1.6343e-08 V9 <--> V9

 Iterations =  34
```

*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`. Unfortunately, this model does not converge.

```
> bifact.s <- sim.hierarchical()
> bifact.s <- round(bifact.s, 5)
> f3 <- factor.pa(bifact.s, 3)
> f3.p <- Promax(f3)
> mod.f3p <- structure.sem(f3.p)
> mod.f3p
```

```
      Path         Parameter Value
 [1,] "PA1->V1"   "F1V1"    NA
 [2,] "PA1->V2"   "F1V2"    NA
 [3,] "PA1->V3"   "F1V3"    NA
 [4,] "PA3->V4"   "F2V4"    NA
 [5,] "PA3->V5"   "F2V5"    NA
 [6,] "PA3->V6"   "F2V6"    NA
 [7,] "PA2->V7"   "F3V7"    NA
 [8,] "PA2->V8"   "F3V8"    NA
 [9,] "PA2->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,] "PA3<->PA1" "rF2F1"   NA
[20,] "PA2<->PA1" "rF3F1"   NA
[21,] "PA2<->PA3" "rF3F2"   NA
[22,] "PA1<->PA1" NA        "1"
[23,] "PA3<->PA3" NA        "1"
[24,] "PA2<->PA2" NA        "1"
```

Unfortunately, this model seems to fail and can not be shown.

```
> sem.f3p <- try(sem(mod.f3p, bifact.s, 100))
> try(summary(sem.f3p))
```

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

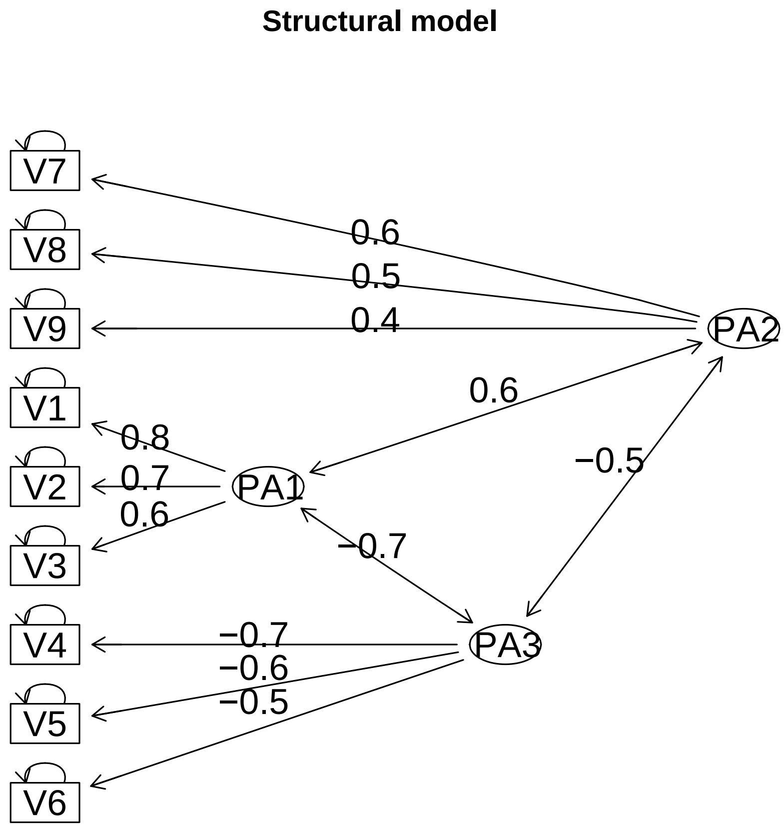


*Figure 10.* A three factor, oblique solution.

*Extract a bifactor solution using omega and then test that model using sem*

A bifactor solution has previously been shown (Figure 7). The output from the `omega` function includes the sem commands for the analysis. For completeness, the `std.coef` from *sem* is used as well as the `summary` function.

```
> mod.bi <- om.bi$model
> sem.bi <- sem(mod.bi, bifact.s, 100)
> summary(sem.bi)
```

```
 Model Chisquare =  9.514e-10   Df =  18 Pr(>Chisq) = 1
 Chisquare (null model) =  234.74   Df =  36
 Goodness-of-fit index =  1
```

```
 Adjusted goodness-of-fit index =  1
 RMSEA index =  0   90% CI: (NA, NA)
 Bentler-Bonnett NFI =  1
 Tucker-Lewis NNFI =  1.1811
 Bentler CFI =  1
 SRMR =  5.8264e-07
 BIC =  -82.893

 Normalized Residuals
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-5.62e-06 -1.15e-07  2.47e-06  2.84e-06  5.29e-06  1.33e-05

 Parameter Estimates
      Estimate Std Error z value Pr(>|z|)
V1     0.81000  0.131114  6.1778  6.4990e-10 V1 <--- g
V2     0.72000  0.133088  5.4100  6.3039e-08 V2 <--- g
V3     0.63000  0.134805  4.6734  2.9625e-06 V3 <--- g
V4     0.56000  0.118547  4.7239  2.3141e-06 V4 <--- g
V5     0.48000  0.118196  4.0611  4.8851e-05 V5 <--- g
V6     0.40000  0.117898  3.3928  6.9189e-04 V6 <--- g
V7     0.42000  0.113206  3.7100  2.0723e-04 V7 <--- g
V8     0.35000  0.113875  3.0736  2.1153e-03 V8 <--- g
V9     0.28000  0.114418  2.4472  1.4399e-02 V9 <--- g
F1*V1 0.39230  0.239526  1.6378  1.0146e-01 V1 <--- F1*
F1*V2 0.34871  0.236741  1.4730  1.4076e-01 V2 <--- F1*
F1*V3 0.30512  0.234186  1.3029  1.9261e-01 V3 <--- F1*
F2*V4 0.42000  0.226424  1.8549  6.3607e-02 V4 <--- F2*
F2*V5 0.36000  0.205124  1.7550  7.9253e-02 V5 <--- F2*
F2*V6 0.30000  0.185166  1.6202  1.0520e-01 V6 <--- F2*
F3*V7 0.42849  0.254996  1.6804  9.2887e-02 V7 <--- F3*
F3*V8 0.35707  0.221805  1.6099  1.0743e-01 V8 <--- F3*
F3*V9 0.28566  0.190372  1.5005  1.3348e-01 V9 <--- F3*
e1     0.19000  0.084306  2.2537  2.4215e-02 V1 <--> V1
e2     0.36000  0.081251  4.4307  9.3921e-06 V2 <--> V2
e3     0.51000  0.087132  5.8532  4.8217e-09 V3 <--> V3
e4     0.51000  0.173932  2.9322  3.3659e-03 V4 <--> V4
e5     0.64000  0.147558  4.3373  1.4426e-05 V5 <--> V5
e6     0.75000  0.133709  5.6092  2.0327e-08 V6 <--> V6
e7     0.64000  0.219288  2.9185  3.5168e-03 V7 <--> V7
e8     0.75000  0.174848  4.2894  1.7912e-05 V8 <--> V8
e9     0.84000  0.148758  5.6468  1.6349e-08 V9 <--> V9
```

```
 Iterations =  61
```

```
> std.coef(sem.bi)
```

```
            Std. Estimate
V1    V1    0.81000       V1 <--- g
V2    V2    0.72000       V2 <--- g
V3    V3    0.63000       V3 <--- g
V4    V4    0.56000       V4 <--- g
V5    V5    0.48000       V5 <--- g
V6    V6    0.40000       V6 <--- g
V7    V7    0.42000       V7 <--- g
V8    V8    0.35000       V8 <--- g
V9    V9    0.28000       V9 <--- g
F1*V1 F1*V1 0.39230       V1 <--- F1*
F1*V2 F1*V2 0.34871       V2 <--- F1*
F1*V3 F1*V3 0.30512       V3 <--- F1*
F2*V4 F2*V4 0.42000       V4 <--- F2*
F2*V5 F2*V5 0.36000       V5 <--- F2*
F2*V6 F2*V6 0.30000       V6 <--- F2*
F3*V7 F3*V7 0.42849       V7 <--- F3*
F3*V8 F3*V8 0.35707       V8 <--- F3*
F3*V9 F3*V9 0.28566       V9 <--- F3*
```

*Examining a hierarchical solution*

A hierarchical solution to this data set was previously found by the `omega` function (Figure 8). 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.

```
> mod.hi <- om.hi$model
> sem.hi <- sem(mod.hi, bifact.s, 100)
> summary(sem.hi)
```

```
 Model Chisquare =  1.0105e-09   Df =  24 Pr(>Chisq) = 1
 Chisquare (null model) =  234.74   Df =  36
 Goodness-of-fit index =  1
 Adjusted goodness-of-fit index =  1
 RMSEA index =  0   90% CI: (NA, NA)
 Bentler-Bonnett NFI =  1
 Tucker-Lewis NNFI =  1.1811
 Bentler CFI =  1
```

```
 SRMR =  4.9184e-07
 BIC =  -110.52

 Normalized Residuals
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-1.17e-05 -2.52e-06  4.34e-07 -2.00e-07  2.28e-06  7.16e-06

 Parameter Estimates
     Estimate Std Error z value Pr(>|z|)
gF1  2.06475  1.425404  1.4485  1.4747e-01 F1 <--- g
gF2  1.33333  0.566569  2.3533  1.8605e-02 F2 <--- g
gF3  0.98020  0.374366  2.6183  8.8374e-03 F3 <--- g
F1V1 0.39230  0.221454  1.7715  7.6482e-02 V1 <--- F1
F1V2 0.34871  0.196030  1.7789  7.5261e-02 V2 <--- F1
F1V3 0.30512  0.173013  1.7636  7.7803e-02 V3 <--- F1
F2V4 0.42000  0.135678  3.0956  1.9643e-03 V4 <--- F2
F2V5 0.36000  0.117757  3.0572  2.2345e-03 V5 <--- F2
F2V6 0.30000  0.104721  2.8647  4.1735e-03 V6 <--- F2
F3V7 0.42849  0.132690  3.2292  1.2412e-03 V7 <--- F3
F3V8 0.35707  0.114753  3.1116  1.8605e-03 V8 <--- F3
F3V9 0.28566  0.105215  2.7150  6.6277e-03 V9 <--- F3
e1   0.19000  0.066521  2.8562  4.2869e-03 V1 <--> V1
e2   0.36000  0.071636  5.0254  5.0231e-07 V2 <--> V2
e3   0.51000  0.084167  6.0594  1.3665e-09 V3 <--> V3
e4   0.51000  0.116365  4.3827  1.1719e-05 V4 <--> V4
e5   0.64000  0.116066  5.5141  3.5060e-08 V5 <--> V5
e6   0.75000  0.121870  6.1541  7.5499e-10 V6 <--> V6
e7   0.64000  0.143096  4.4725  7.7295e-06 V7 <--> V7
e8   0.75000  0.134879  5.5605  2.6893e-08 V8 <--> V8
e9   0.84000  0.135210  6.2126  5.2131e-10 V9 <--> V9

 Iterations =  40
```

```
> std.coef(sem.hi)
```

```
          Std. Estimate
gF1  gF1  0.9           F1 <--- g
gF2  gF2  0.8           F2 <--- g
gF3  gF3  0.7           F3 <--- g
F1V1 F1V1 0.9           V1 <--- F1
F1V2 F1V2 0.8           V2 <--- F1
F1V3 F1V3 0.7           V3 <--- F1
```

```
F2V4 F2V4 0.7           V4 <--- F2
F2V5 F2V5 0.6           V5 <--- F2
F2V6 F2V6 0.5           V6 <--- F2
F3V7 F3V7 0.6           V7 <--- F3
F3V8 F3V8 0.5           V8 <--- F3
F3V9 F3V9 0.4           V9 <--- F3
```

The use of exploratory and confirmatory models for understanding real data structures is an important advance in psychological research. 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*.

## References

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

Fox, J. (2008). *sem: Structural equation models.* (R package version 0.9-13)

Holzinger, K., & Swineford, F. (1937, 03 27). The bi-factor method. *Psychometrika*, *2*(1), 41–54.

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

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

Revelle, W. (2009). *psych: Procedures for personality and psychological research.* (R package version 1.0-63)

## Index