An overview of the psych package

William Revelle
Department of Psychology
Northwestern University

March 31, 2011

Contents

1 Overview of this and related documents **2**

2 Getting started **3**

3 Basic data analysis **4**
3.1 Data input and descriptive statistics 4
3.1.1 Basic data cleaning using `scrub` 7
3.2 Simple descriptive graphics 8
3.2.1 Scatter Plot Matrices 8
3.2.2 Means and error bars 9
3.2.3 Back to back histograms 9
3.2.4 Correlational structure 13
3.3 Testing correlations 15
3.4 Polychoric, tetrachoric, polyserial, and biserial correlations 17

4 Item and scale analysis **17**
4.1 Dimension reduction through factor analysis and cluster analysis 19
4.1.1 Minimum Residual Factor Analysis 21
4.1.2 Principal Axis Factor Analysis 21
4.1.3 Weighted Least Squares Factor Analysis 24
4.1.4 Principal Components analysis 24
4.1.5 Hierarchical and bi-factor solutions 24
4.1.6 Item Cluster Analysis: iclust 29
4.2 Comparing factor/component/cluster solutions 32
4.3 Determining the number of dimensions to extract. 35
4.3.1 Very Simple Structure 36

4.3.2 Parallel Analysis . . . 38
4.4 Reliability analysis . . . 38
4.4.1 Reliability of a single scale . . . 41
4.4.2 Using omega to find the reliability of a single scale . . . 45
4.4.3 Estimating ω_h using Confirmatory Factor Analysis . . . 49
4.4.4 Other estimates of reliability . . . 51
4.4.5 Reliability of multiple scales within an inventory . . . 52
4.5 Item analysis . . . 56

5 Item Response Theory analysis **57**

6 Multiple Regression from the correlation matrix **58**

7 Simulation functions **63**

8 Graphical Displays **65**

9 Data sets **65**

10 Development version and a users guide **68**

11 Psychometric Theory **68**

12 SessionInfo **68**

1 Overview of this and related documents

The *psych* package (Revelle, 2011) has been developed at Northwestern University to include functions most useful for personality, psychometric, and psychological research. Some of the functions (e.g., `read.clipboard`, `describe`, `pairs.panels`, `error.bars`) are useful for basic data entry and descriptive analyses.

Psychometric applications include routines for five types of *factor analysis* (`fa` does *minimum residual*, *principal axis*, *weighted least squares*, *generalized least squares* and *maximum likelihood* factor analysis). Determining the number of factors or components to extract may be done by using the Very Simple Structure (Revelle and Rocklin, 1979) (`vss`), Minimum Average Partial correlation (Velicer, 1976) (`MAP`) or parallel analysis `fa.parallel` criteria. Item Response Theory models for dichotomous or polytomous items may be found by factoring `tetrachoric` or `polychoric` correlation matrices and expressing the resulting parameters in terms of location and discrimination `irt.fa`. Bifactor and hierarchical factor structures may be estimated by using Schmid Leiman transformations (Schmid and

Leiman, 1957) (schmid) to transform a hierarchical factor structure into a bifactor solution (Holzinger and Swineford, 1937). Scale construction can be done using the Item Cluster Analysis (Revelle, 1979) (iclust) function to determine the structure and to calculate reliability coefficients α Cronbach (1951)(alpha, score.items, score.multiple.choice), β (Revelle, 1979; Revelle and Zinbarg, 2009) (iclust) and McDonald's ω_h and ω_t (McDonald, 1999) (omega). Guttman's six estimates of internal consistency reliability (Guttman (1945), as well as additional estimates (Revelle and Zinbarg, 2009) are in the guttman function and the six measures of Intraclass correlation coefficients (ICC) discussed by Shrout and Fleiss (1979) are also available.

Graphical displays include Scatter Plot Matrix (SPLOM) plots using pairs.panels, factor, cluster, and structural diagrams using fa.diagram, iclust.diagram, structure.diagram, as well as item response characteristics and item and test information characteristic curves plot.irt.

This vignette is meant to give an overview of the *psych* package. That is, it is meant to give a summary of the main functions in the *psych* package with examples of how they are used for data description, dimension reduction, and scale construction. The extended user manual at psych_manual.pdf includes examples of graphic output and more extensive demonstrations than are found in the help menus. (Also available at http://personality-project.org/r/psych_manual.pdf). The vignette, psych for sem, at psych_for_sem.pdf, discusses how to use psych as a front end to the *sem* package of John Fox (Fox, 2009). (The vignette is also available at http://personality-project.org/r/book/psych_for_sem.pdf).

For a step by step tutorial in the use of the psych package and the base functions in R for basic personality research, see the guide for using R for personality research at http://personalitytheory.org/r/r.short.html. For an *introduction to psychometric theory with applications in R*, see the draft chapters at http://personality-project.org/r/book).

2 Getting started

Some of the functions described in this overview require other packages. Particularly useful for rotating the results of factor analyses (from e.g., fa, factor.minres, factor.pa, factor.wls, or principal) or hierarchical factor models using omega or schmid, is the *GPArotation* package. For some analyses of correlations of dichotomous data, the *polycor* package is suggested in order to use either the poly.mat or phi2poly functions. However, these functions have been supplement with tetrachoric and polychoric which do not require the *polycor* package. These and other useful packages may be installed by first

installing and then using the task views (*ctv*) package to install the "Psychometrics" task view, but doing it this way is not necessary.

```
install.packages("ctv")
library(ctv)
task.views("Psychometrics")
```

Because of the difficulty of installing the package *Rgraphviz*, alternative graphics have been developed and are available as *diagram* functions. If *Rgraphviz* is available, some functions will take advantage of it.

3 Basic data analysis

A number of *psych* functions facilitate the entry of data and finding basic descriptive statistics.

Remember, to run any of the *psych* functions, it is necessary to make the package active by using the `library` command:

```
> library(psych)
```

The other packages, once installed, will be called automatically by *psych*.

It is possible to automatically load *psych* and other functions by creating and then saving a ".First" function:

```
.First <-  function(x) {library(psych)}
```

3.1 Data input and descriptive statistics

There are of course many ways to enter data into R. Reading from a local file using `read.table` is perhaps the most preferred. However, many users will enter their data in a text editor or spreadsheet program and then want to copy and paste into R. This may be done by using `read.table` and specifying the input file as "clipboard" (PCs) or "pipe(pbpaste)" (Macs). Alternatively, the `read.clipboard` set of functions are perhaps more user friendly:

`read.clipboard` is the base function for reading data from the clipboard.

`read.clipboard.csv` for reading text that is comma delimited.

`read.clipboard.tab` for reading text that is tab delimited.

`read.clipboard.lower` for reading input of a lower triangular matrix with or without a diagonal. The resulting object is a square matrix.

`read.clipboard.upper` for reading input of an upper triangular matrix.

`read.clipboard.fwf` for reading in fixed width fields (some very old data sets)

For example, given a data set copied to the clipboard from a spreadsheet, just enter the command

```
> my.data <- read.clipboard()
```

This will work if every data field has a value and even missing data are given some values (e.g., NA or -999). If the data were entered in a spreadsheet and the missing values were just empty cells, then the data should be read in as a tab delimited or by using the `read.clipboard.tab` function.

```
> my.data <- read.clipboard(sep="\t")   #define the tab option, or
> my.tab.data <- read.clipboard.tab()   #just use the alternative function
```

For the case of data in fixed width fields (some old data sets tend to have this format), copy to the clipboard and then specify the width of each field (in the example below, the first variable is 5 columns, the second is 2 columns, the next 5 are 1 column the last 4 are 3 columns).

```
> my.data <- read.clipboard.fwf(widths=c(5,2,rep(1,5),rep(3,4))
```

Once the data are read in, then `describe` or `describe.by` will provide basic descriptive statistics arranged in a data frame format. Consider the data set `sat.act` which includes data from 700 web based participants on 3 demographic variables and 3 ability measures.

`describe` reports means, standard deviations, medians, min, max, range, skew, kurtosis and standard errors for integer or real data. Non-numeric data will produce an error.

`describe.by` reports descriptive statistics broken down by some categorizing variable (e.g., gender, age, etc.)

```
> library(psych)
> data(sat.act)
> describe(sat.act)
```

```
          var   n   mean     sd median trimmed    mad min max range  skew
gender      1 700   1.65   0.48      2    1.68   0.00   1   2     1 -0.61
education   2 700   3.16   1.43      3    3.31   1.48   0   5     5 -0.68
age         3 700  25.59   9.50     22   23.86   5.93  13  65    52  1.64
ACT         4 700  28.55   4.82     29   28.84   4.45   3  36    33 -0.66
```

```
SATV        5 700 612.23 112.90    620  619.45 118.61 200 800   600 -0.64
SATQ        6 687 610.22 115.64    620  617.25 118.61 200 800   600 -0.59
          kurtosis   se
gender       -1.62 0.02
education    -0.06 0.05
age           2.47 0.36
ACT           0.56 0.18
SATV          0.35 4.27
SATQ          0.00 4.41
```

These data may then be analyzed by groups defined in a logical statement or by some other variable. E.g., break down the descriptive data for males or females. These descriptive data can also be seen graphically using the `error.bars.by` function (Figure 3). By setting skew=FALSE and ranges=FALSE, the output is limited to the most basic statistics.

```
> describe.by(sat.act, sat.act$gender, skew = FALSE, ranges = FALSE)

group: 1
          var   n   mean     sd   se
gender      1 247   1.00   0.00 0.00
education   2 247   3.00   1.54 0.10
age         3 247  25.86   9.74 0.62
ACT         4 247  28.79   5.06 0.32
SATV        5 247 615.11 114.16 7.26
SATQ        6 245 635.87 116.02 7.41
------------------------------------------------------------
group: 2
          var   n   mean     sd   se
gender      1 453   2.00   0.00 0.00
education   2 453   3.26   1.35 0.06
age         3 453  25.45   9.37 0.44
ACT         4 453  28.42   4.69 0.22
SATV        5 453 610.66 112.31 5.28
SATQ        6 442 596.00 113.07 5.38
```

The output from the describe.by function can be forced into a matrix form for easy analysis by other programs. In addition, describe.by can group by several grouping variables at the same time.

```
> sa.mat <- describe.by(sat.act, list(sat.act$gender, sat.act$education),
+     skew = FALSE, ranges = FALSE, mat = TRUE)
> head(sa.mat)
```

```
        item group1 group2 var  n mean sd se
gender1    1      1      0   1 27    1  0  0
gender2    2      2      0   1 30    2  0  0
gender3    3      1      1   1 20    1  0  0
gender4    4      2      1   1 25    2  0  0
gender5    5      1      2   1 23    1  0  0
gender6    6      2      2   1 21    2  0  0
```

```
> tail(sa.mat)
```

```
       item group1 group2 var   n     mean        sd        se
SATQ7    67      1      3   6  79 642.5949 118.28329 13.307910
SATQ8    68      2      3   6 190 590.8895 114.46472  8.304144
SATQ9    69      1      4   6  51 635.9020 104.12190 14.579982
SATQ10   70      2      4   6  86 597.5930 106.24393 11.456578
SATQ11   71      1      5   6  46 657.8261  89.60811 13.211995
SATQ12   72      2      5   6  93 606.7204 105.55108 10.945137
```

3.1.1 Basic data cleaning using `scrub`

If, after describing the data it is apparent that there were data entry errors that need to be globally replaced with NA, or only certain ranges of data will be analyzed, the data can be "cleaned" using the `scrub` function.

Consider the `attitude` data set (which does not need to be cleaned, but will be used as an example). All values of columns 3 - 5 that are less than 30, 40, or 50 respectively, or greater than 70 in any column will be replaced with NA. In addition, any value exactly equal to 45 will be set to NA. (max and isvalue are set to one value here, but they could be a different value for every column).

```
> x <- matrix(1:120, ncol = 10, byrow = TRUE)
> new.x <- scrub(x, 3:5, min = c(30, 40, 50), max = 70, isvalue = 45)
> new.x
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    1    2   NA   NA   NA    6    7    8    9    10
 [2,]   11   12   NA   NA   NA   16   17   18   19    20
 [3,]   21   22   NA   NA   NA   26   27   28   29    30
 [4,]   31   32   33   34   35   36   37   38   39    40
 [5,]   41   42   43   44   NA   46   47   48   49    50
 [6,]   51   52   53   54   55   56   57   58   59    60
 [7,]   61   62   63   64   65   66   67   68   69    70
 [8,]   71   72   NA   NA   NA   76   77   78   79    80
```

```
 [9,]   81   82   NA   NA   NA   86   87   88   89   90
[10,]   91   92   NA   NA   NA   96   97   98   99  100
[11,]  101  102   NA   NA   NA  106  107  108  109  110
[12,]  111  112   NA   NA   NA  116  117  118  119  120
```

Note that the number of subjects for those columns has decrease, and the minimums have gone up but the maximums down. Data cleaning and examination for outliers should be a routine part of any data analysis.

3.2 Simple descriptive graphics

Graphic descriptions of data are very helpful both for understanding the data as well as communicating important results. Scatter Plot Matrices (SPLOMS) using the `pairs.panels` function are useful ways to look for strange effects involving outliers and non-linearities. `error.bars.bi` will show group means with 95% confidence boundaries.

3.2.1 Scatter Plot Matrices

Scatter Plot Matrices (SPLOMS) are very useful for describing the data. The `pairs.panels` function, adapted from the help menu for the `pairs` function produces xy scatter plots of each pair of variables below the diagonal, shows the histogram of each variable on the diagonal, and shows the *lowess* locally fit regression line as well. An ellipse around the mean with the axis length reflecting one standard deviation of the first and second principal components is also drawn. The x axis in each scatter plot represents the column variable, the y axis the row variable (Figure 1).

`pairs.panels` will show the pairwise scatter plots of all the variables as well as histograms, locally smoothed regressions, and the Pearson correlation.

```
> png("pairspanels.png")
> pairs.panels(sat.act)
> dev.off()
null device
          1
```

Figure 1: Using the `pairs.panels` function to graphically show relationships. The x axis in each scatter plot represents the column variable, the y axis the row variable. Note the extreme outlier for the ACT.

Another example of `pairs.panels` is to show differences between experimental groups. Consider the data in the `affect` data set. The scores reflect post test scores on positive

and negative affect and energetic and tense arousal. The colors show the results for four movie conditions: depressing, frightening movie, neutral, and a comedy.

```
> data(affect)
> png("affect.png")
> pairs.panels(flat[15:18], bg = c("red", "black", "white", "blue")[maps$Film],
+     pch = 21, main = "Affect varies by movies (study 'flat')")
> dev.off()
null device
          1
```

Figure 2: Using the `pairs.panels` function to graphically show relationships. The x axis in each scatter plot represents the column variable, the y axis the row variable. The coloring represent four different movie conditions.

3.2.2 Means and error bars

Additional descriptive graphics include the ability to draw *error bars* on sets of data, as well as to draw error bars in both the x and y directions for paired data. These are the functions

`error.bars` show the 95 % confidence intervals for each variable in a data frame or matrix.

`error.bars.by` does the same, but grouping the data by some condition.

`error.crosses` draw the confidence intervals for an x set and a y set of the same size.

The use of the `error.bars.by` function allows for graphic comparisons of different groups (see Figure 3). Five personality measures are shown as a function of high versus low scores on a "lie" scale. People with higher lie scores tend to report being more agreeable, conscientious and less neurotic than people with lower lie scores. The error bars are based upon normal theory and thus are symmetric rather than reflect any skewing in the data.

Although not recommended, it is possible to use the `error.bars` function to draw bar graphs with associated error bars. (This kind of *dynamite plot* (Figure 4) can be very misleading in that the scale is arbitrary. Go to a discussion of the problems in presenting data this way at http://emdbolker.wikidot.com/blog:dynamite.

3.2.3 Back to back histograms

The `bi.bars` function summarize the characteristics of two groups (e.g., males and females) on a second variable (e.g., age) by drawing back to back histograms (see Figure 5).

```
> data(epi.bfi)
> error.bars.by(epi.bfi[, 6:10], epi.bfi$epilie < 4)
```

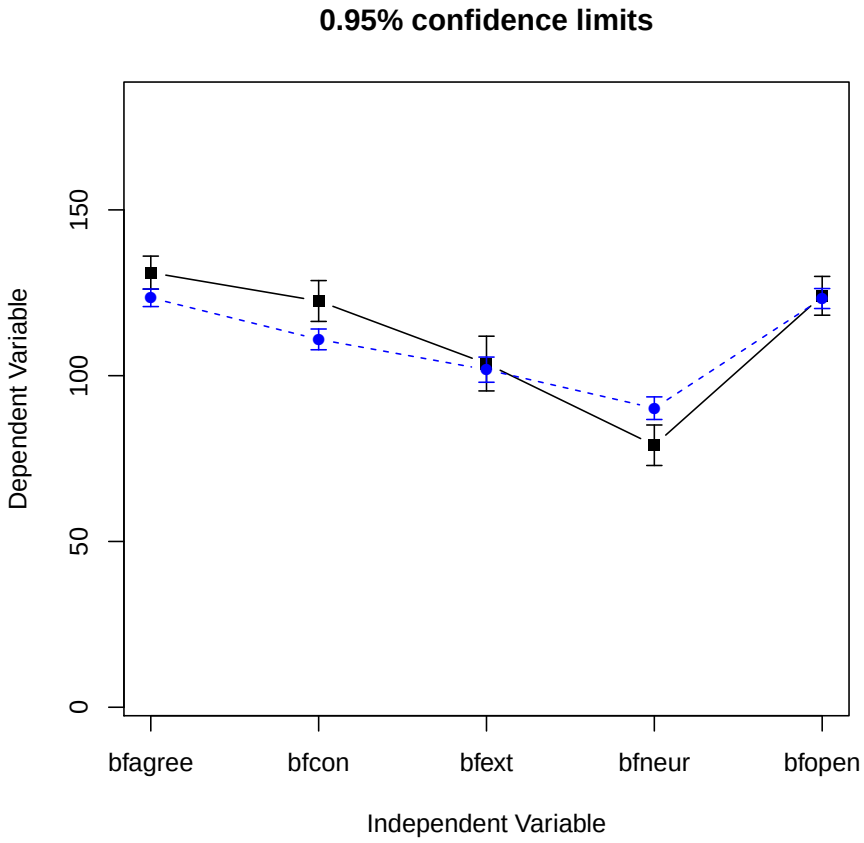

Figure 3: Using the `error.bars.by` function shows that self reported personality scales on the Big Five Inventory vary as a function of the Lie scale on the EPI.

```
> error.bars.by(sat.act[5:6], sat.act$gender, bars = TRUE, labels = c("Male",
+     "Female"), ylab = "SAT score", xlab = "")
```

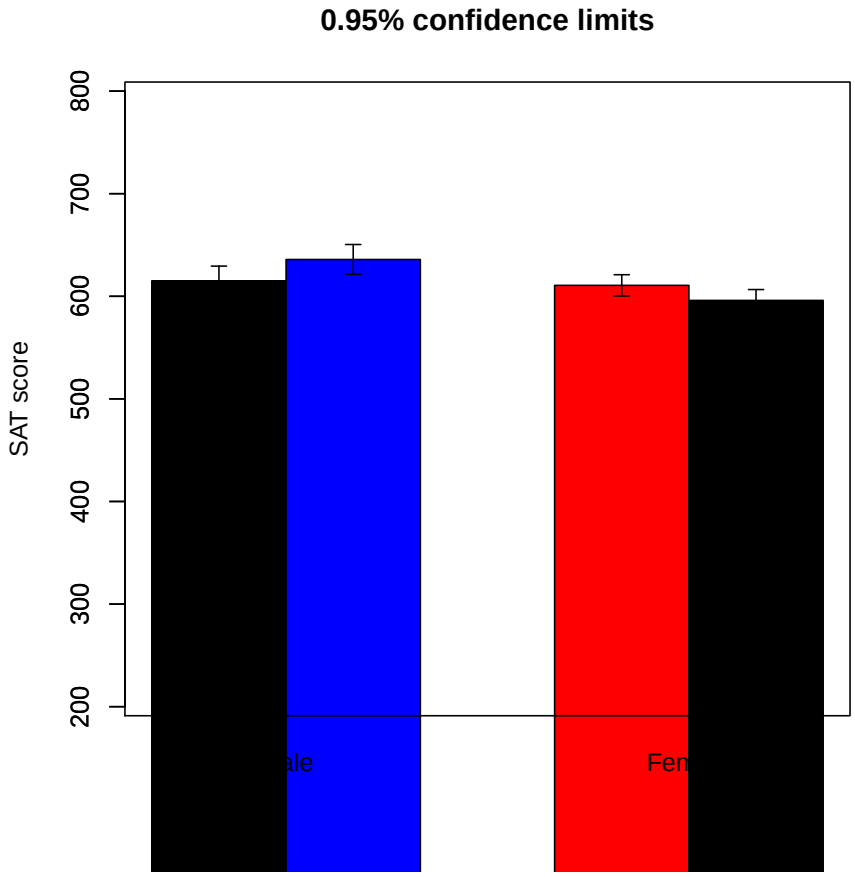

Figure 4: A "Dynamite plot" of SAT scores as a function of gender is one way of misleading the reader. By using a bar graph, the range of scores is ignored.

```
> data(bfi)
> with(bfi, {
+     bi.bars(age, gender, ylab = "Age", main = "Age by males and females")
+ })
```

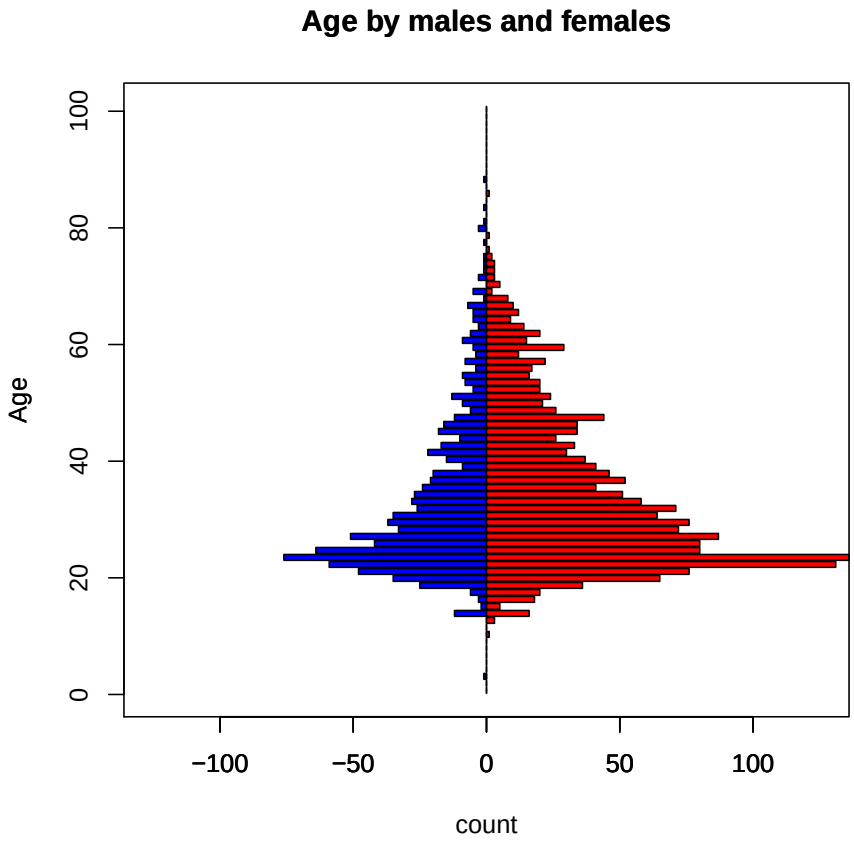

Figure 5: A bar plot of the age distribution for males and females shows the use of `bi.bars`. The data are males and females from 2800 cases collected using the SAPA procedure and are available as part of the `bfi` data set.

3.2.4 Correlational structure

It is also possible to see the structure in a correlation matrix by forming a matrix shaded to represent the magnitude of the correlation. This is useful when considering the number of factors in a data set. Consider the `Thurstone` data set which has a clear 3 factor solution (Figure 6) or a simulated data set of 24 variables with a circumplex structure (Figure 7).

```
> cor.plot(Thurstone, color = TRUE, main = "9 cognitive variables from Thurstone")
```

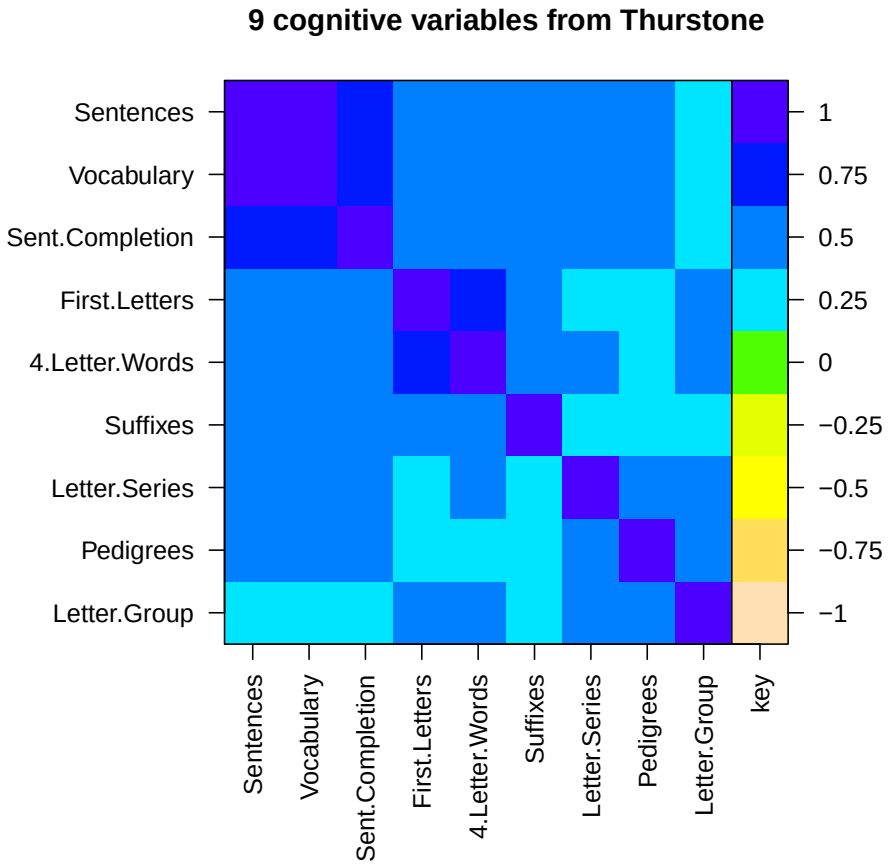

Figure 6: The structure of correlation matrix can be seen more clearly if the variables are grouped by factor and then the correlations are shown by color.

```
> circ <- sim.circ(24)
> r.circ <- cor(circ)
> cor.plot(r.circ, color = TRUE, zlim = c(-1, 1), main = "24 variables in a circumplex")
```

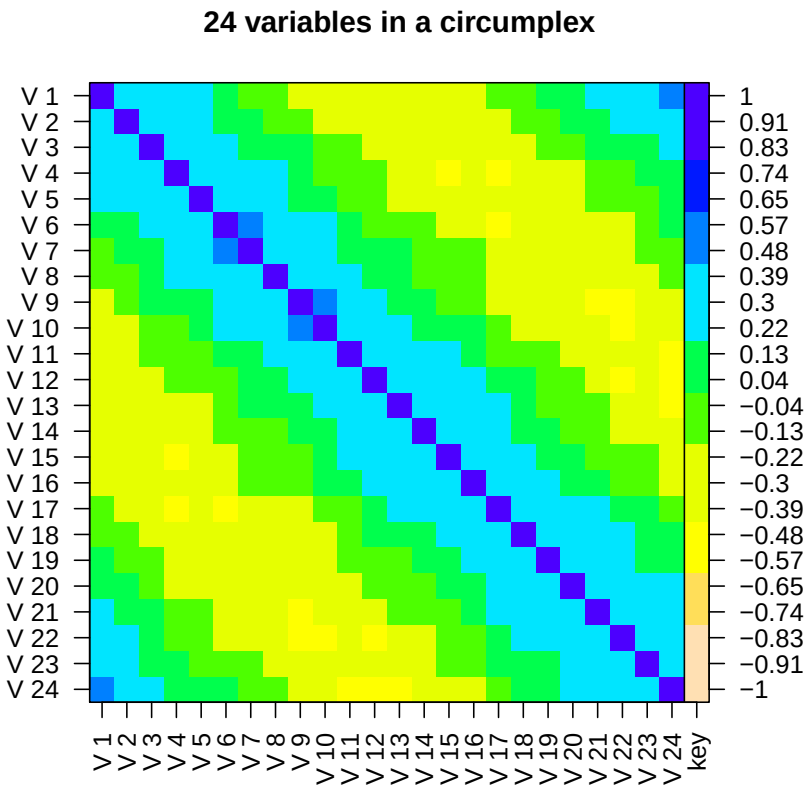

Figure 7: Using the cor.plot function to show the correlations in a circumplex. Correlations are highest near the diagonal, diminish to zero further from the diagonal, and the increase again towards the corners of the matrix. Circumplex structures are common in the study of affect.

3.3 Testing correlations

Correlations are wonderful descriptive statistics of the data but some people like to test whether these correlations differ from zero, or differ from each other. The `cor.test` function (in the *stats* package) will test the significance of a single correlation, and the `rcorr` function in the *Hmisc* package will do this for many correlations. In the *psych* package, the `corr.test` function reports the correlation (Pearson or Spearman) between all variables in either one or two data frames or matrices, as well as the number of observations for each case, and the (two-tailed) probability for each correlation. These probability values have not been corrected for multiple comparisons and so should be taken with a great deal of salt.

```
> corr.test(sat.act)

Call:corr.test(x = sat.act)
Correlation matrix
          gender education   age   ACT  SATV  SATQ
gender      1.00      0.09 -0.02 -0.04 -0.02 -0.17
education   0.09      1.00  0.55  0.15  0.05  0.03
age        -0.02      0.55  1.00  0.11 -0.04 -0.03
ACT        -0.04      0.15  0.11  1.00  0.56  0.59
SATV       -0.02      0.05 -0.04  0.56  1.00  0.64
SATQ       -0.17      0.03 -0.03  0.59  0.64  1.00
Sample Size
          gender education age ACT SATV SATQ
gender       700       700 700 700  700  687
education    700       700 700 700  700  687
age          700       700 700 700  700  687
ACT          700       700 700 700  700  687
SATV         700       700 700 700  700  687
SATQ         687       687 687 687  687  687
Probability value
          gender education  age  ACT SATV SATQ
gender      0.00      0.02 0.58 0.33 0.62 0.00
education   0.02      0.00 0.00 0.00 0.22 0.36
age         0.58      0.00 0.00 0.00 0.26 0.37
ACT         0.33      0.00 0.00 0.00 0.00 0.00
SATV        0.62      0.22 0.26 0.00 0.00 0.00
SATQ        0.00      0.36 0.37 0.00 0.00 0.00
```

Testing the difference between any two correlations can be done using the `r.test` function. The function actually does four different tests, depending upon the input:

1) For a sample size n, find the t and p value for a single correlation as well as the confidence interval.

```
> r.test(50, 0.3)

Correlation tests
Call:r.test(n = 50, r12 = 0.3)
Test of significance of a  correlation
 t value 2.18    with probability < 0.034
 and confidence interval  0.02 0.53
```

2) For sample sizes of n and n2 (n2 = n if not specified) find the z of the difference between the z transformed correlations divided by the standard error of the difference of two z scores.

```
> r.test(30, 0.4, 0.6)

Correlation tests
Call:r.test(n = 30, r12 = 0.4, r34 = 0.6)
Test of difference between two independent correlations
 z value 0.99    with probability  0.32
```

3) For sample size n, and correlations ra= r12, rb= r23 and r13 specified, test for the difference of two dependent correlations (Steiger case A).

```
> r.test(103, 0.4, 0.5, 0.1)

Correlation tests
Call:r.test(n = 103, r12 = 0.4, r34 = 0.5, r23 = 0.1)
Test of difference between two correlated  correlations
 t value -0.89    with probability < 0.37
```

4) For sample size n, test for the difference between two dependent correlations involving different variables. (Steiger case B).

```
> r.test(103, 0.5, 0.6, 0.7, 0.5, 0.5, 0.8)

Correlation tests
Call:r.test(n = 103, r12 = 0.5, r34 = 0.6, r23 = 0.7, r13 = 0.5, r14 = 0.5,
    r24 = 0.8)
Test of difference between two dependent correlations
 z value -1.2    with probability  0.23
```

To test whether a matrix of correlations differs from what would be expected if the population correlations were all zero, the function `cortest` follows Steiger (1980) who pointed out that the sum of the squared elements of a correlation matrix, or the Fisher z score equivalents, is distributed as chi square under the null hypothesis that the values are zero

(i.e., elements of the identity matrix). This is particularly useful for examining whether correlations in a single matrix differ from zero or for comparing two matrices. Although obvious, `cortest` can be used to test whether the `sat.act` data matrix produces non-zero correlations (it does). This is a much more appropriate test when testing whether a residual matrix differs from zero.

```
> cortest(sat.act)
```

```
Tests of correlation matrices
Call:cortest(R1 = sat.act)
 Chi Square value 1325.42  with df =  15   with probability < 1.8e-273
```

3.4 Polychoric, tetrachoric, polyserial, and biserial correlations

The Pearson correlation of dichotomous data is also known as the ϕ coefficient. If the data, e.g., ability items, are thought to represent an underlying continuous although latent variable, the ϕ will underestimate the value of the Pearson applied to these latent variables. One solution to this problem is to use the `tetrachoric` correlation which is based upon the assumption of a bivariate normal distribution that has been cut at certain points. The `draw.tetra` function demonstrates the process (Figure 8). A simple generalization of this to the case of the multiple cuts is the `polychoric` correlation.

Other estimated correlations based upon the assumption of bivariate normality with cut points include the `biserial` and `polyserial` correlation.

If the data are a mix of continuous, polytomous and dichotomous variables, the `mixed.cor` function will calculate the appropriate mixture of Pearson, polychoric, tetrachoric, biserial, and polyserial correlations.

4 Item and scale analysis

The main functions in the *psych* package are for analyzing the structure of items and of scales and for finding various estimates of scale reliability. These may be considered as problems of dimension reduction (e.g., factor analysis, cluster analysis, principal components analysis) and of forming and estimating the reliability of the resulting composite scales.

```
> draw.tetra()
```

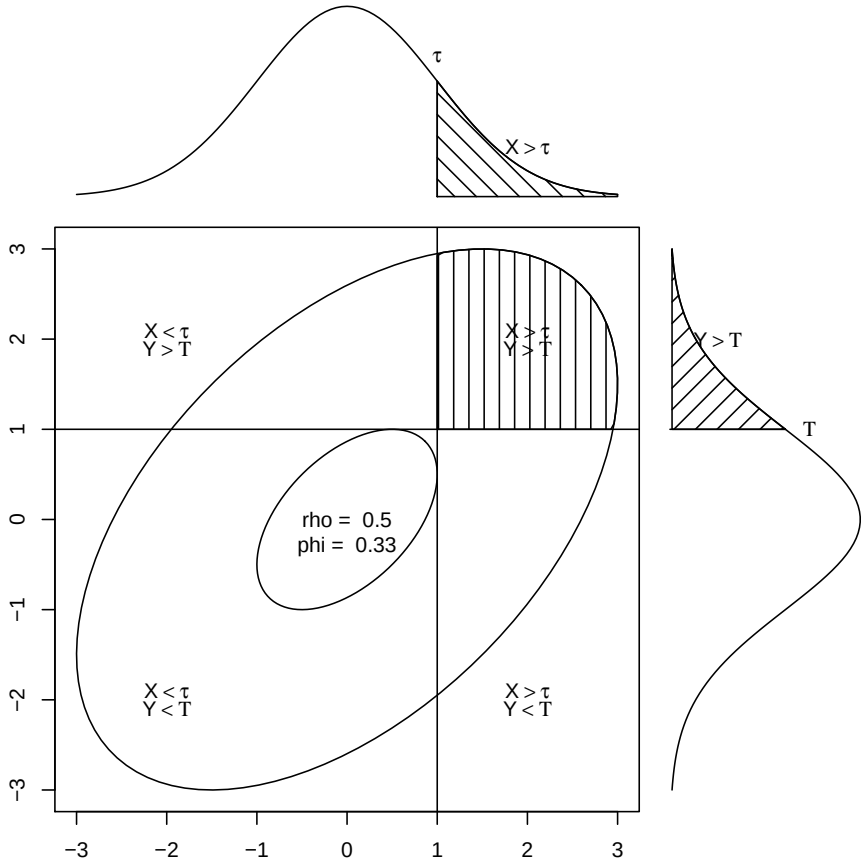

Figure 8: The tetrachoric correlation estimates what a Pearson correlation would be given a two by two table of observed values assumed to be sampled from a bivariate normal distribution. The ϕ correlation is just a Pearson r performed on the observed values.

4.1 Dimension reduction through factor analysis and cluster analysis

Parsimony of description has been a goal of science since at least the famous dictum commonly attributed to William of Ockham to not multiply entities beyond necessity[1]. The goal for parsimony is seen in psychometrics as an attempt either to describe (components) or to explain (factors) the relationships between many observed variables in terms of a more limited set of components or latent factors.

The typical data matrix represents multiple items or scales usually thought to reflect fewer underlying constructs[2]. At the most simple, a set of items can be be thought to represent a random sample from one underlying domain or perhaps a small set of domains. The question for the psychometrician is how many domains are represented and how well does each item represent the domains. Solutions to this problem are examples of *factor analysis* (*FA*), *principal components analysis* (*PCA*), and *cluster analysis* (*CA*). All of these procedures aim to reduce the complexity of the observed data. In the case of FA, the goal is to identify fewer underlying constructs to explain the observed data. In the case of PCA, the goal can be mere data reduction, but the interpretation of components is frequently done in terms similar to those used when describing the latent variables estimated by FA. Cluster analytic techniques, although usually used to partition the subject space rather than the variable space, can also be used to group variables to reduce the complexity of the data by forming fewer and more homogeneous sets of tests or items.

At the data level the data reduction problem may be solved as a *Singular Value Decomposition* of the original matrix, although the more typical solution is to find either the *principal components* or *factors* of the covariance or correlation matrices. Given the pattern of regression weights from the variables to the components or from the factors to the variables, it is then possible to find (for components) individual *component* or *cluster scores* or estimate (for factors) *factor scores*.

Several of the functions in *psych* address the problem of data reduction.

`fa` incorporates five alternative algorithms: *minres factor analysis*, *principal axis factor analysis*, *weighted least squares factor analysis*, *generalized least squares factor analysis* and *maximum likelihood factor analysis*. That is, it includes the functionality of three other functions that will be eventually phased out.

`factor.minres` Minimum residual factor analysis is a least squares, iterative solution to the factor problem. minres attempts to minimize the residual (off-diagonal) correlation matrix. It produces solutions similar to maximum likelihood solutions, but will

[1] Although probably neither original with Ockham nor directly stated by him (Thorburn, 1918), Ockham's razor remains a fundamental principal of science.

[2] Cattell (1978) as well as MacCallum et al. (2007) argue that the data are the result of many more factors than observed variables, but are willing to estimate the major underlying factors.

work even if the matrix is singular.

`factor.pa` Principal Axis factor analysis is a least squares, iterative solution to the factor problem. PA will work for cases where maximum likelihood techniques (`factanal`) will not work. The original communality estimates are either the squared multiple correlations (`smc`) for each item or 1.

`factor.wls` Weighted least squares factor analysis is a least squares, iterative solution to the factor problem. It minimizes the (weighted) squared residual matrix. The weights are based upon the independent contribution of each variable.

`principal` Principal Components Analysis reports the largest n eigen vectors rescaled by the square root of their eigen values.

`factor.congruence` The congruence between two factors is the cosine of the angle between them. This is just the cross products of the loadings divided by the sum of the squared loadings. This differs from the correlation coefficient in that the mean loading is not subtracted before taking the products. `factor.congruence` will find the cosines between two (or more) sets of factor loadings.

`vss` Very Simple Structure Revelle and Rocklin (1979) applies a goodness of fit test to determine the optimal number of factors to extract. It can be thought of as a quasi-confirmatory model, in that it fits the very simple structure (all except the biggest c loadings per item are set to zero where c is the level of complexity of the item) of a factor pattern matrix to the original correlation matrix. For items where the model is usually of complexity one, this is equivalent to making all except the largest loading for each item 0. This is typically the solution that the user wants to interpret. The analysis includes the `MAP` criterion of Velicer (1976) and a χ^2 estimate.

`fa.parallel` The parallel factors technique compares the observed eigen values of a correlation matrix with those from random data.

`factor.plot` will plot the loadings from a factor, principal components, or cluster analysis (just a call to plot will suffice). If there are more than two factors, then a SPLOM of the loadings is generated.

`fa.diagram` replaces `fa.graph` and will draw a path diagram representing the factor structure. It does not require Rgraphviz and thus is probably preferred.

`fa.graph` requires `Rgraphviz` and will draw a graphic representation of the factor structure. If factors are correlated, this will be represented as well.

`iclust` is meant to do item cluster analysis using a hierarchical clustering algorithm specifically asking questions about the reliability of the clusters (Revelle, 1979). Clusters are formed until either coefficient α Cronbach (1951) or β Revelle (1979) fail to increase.

4.1.1 Minimum Residual Factor Analysis

The factor model is an approximation of a correlation matrix by a matrix of lower rank. That is, can the correlation matrix, ${}_n\vec{R}_n$ be approximated by the product of a factor matrix, ${}_n\vec{F}_k$ and its transpose plus a diagonal matrix of uniqueness.

$$R = FF' + U^2 \tag{1}$$

The maximum likelihood solution to this equation is found by `factanal` in the *stats* package. Five alternatives are provided in *psych*, all of them are included in the `fa` function and are called by specifying the factor method (e.g., fm="minres", fm="pa", fm=""wls", fm="gls" and fm="ml"). In the discussion of the other algorithms, the calls shown are to the `fa` function specifying the appropriate method.

`factor.minres` attempts to minimize the off diagonal residual correlation matrix by adjusting the eigen values of the original correlation matrix. This is similar to what is done in `factanal`, but uses an ordinary least squares instead of a maximum likelihood fit function. The solutions tend to be more similar to the MLE solutions than are the `factor.pa` solutions. *min.res* is the default for the `fa` function.

A classic data set, collected by Thurstone and Thurstone (1941) and then reanalyzed by Bechtoldt (1961) and discussed by McDonald (1999), is a set of 9 cognitive variables with a clear bi-factor structure Holzinger and Swineford (1937). The minimum residual solution was transformed into an oblique solution using the default option on rotate which uses an oblimin transformation (Table 1). Alternative rotations and transformations include "none", "varimax", "quartimax", "bentlerT", and "geominT" (all of which are orthogonal rotations). as well as "promax", "oblimin", "simplimax", "bentlerQ, and"geominQ" and "cluster" which are possible oblique transformations of the solution. The default is to do a oblimin transformation, although prior versions defaulted to varimax. The measures of factor adequacy reflect the multiple correlations of the factors with the best fitting linear regression estimates of the factor scores (Grice, 2001).

4.1.2 Principal Axis Factor Analysis

An alternative, least squares algorithm, `factor.pa`, does a Principal Axis factor analysis by iteratively doing an eigen value decomposition of the correlation matrix with the diagonal replaced by the values estimated by the factors of the previous iteration. This OLS solution is not as sensitive to improper matrices as is the maximum likelihood method, and will sometimes produce more interpretable results.

The solutions from the `fa`, the `factor.minres` and `factor.pa` as well as the `principal` functions can be rotated or transformed with a number of options. Some of these call

Table 1: Three correlated factors from the Thurstone 9 variable problem. By default, the solution is transformed obliquely using oblimin. The extraction method is (by default) minimum residual.

```
> f3t <- fa(Thurstone, 3, n.obs = 213)
> f3t
Factor Analysis using method =  minres
Call: fa(r = Thurstone, nfactors = 3, n.obs = 213)
Standardized loadings based upon correlation matrix
                   MR1   MR2   MR3   h2   u2
Sentences         0.91 -0.04  0.04 0.82 0.18
Vocabulary        0.89  0.06 -0.03 0.84 0.16
Sent.Completion   0.83  0.04  0.00 0.73 0.27
First.Letters     0.00  0.86  0.00 0.73 0.27
4.Letter.Words   -0.01  0.74  0.10 0.63 0.37
Suffixes          0.18  0.63 -0.08 0.50 0.50
Letter.Series     0.03 -0.01  0.84 0.72 0.28
Pedigrees         0.37 -0.05  0.47 0.50 0.50
Letter.Group     -0.06  0.21  0.64 0.53 0.47

                 MR1  MR2  MR3
SS loadings    2.64 1.86 1.50
Proportion Var 0.29 0.21 0.17
Cumulative Var 0.29 0.50 0.67

 With factor correlations of
     MR1  MR2  MR3
MR1 1.00 0.59 0.54
MR2 0.59 1.00 0.52
MR3 0.54 0.52 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  5.2 with
The degrees of freedom for the model are 12  and the objective function was  0.01

The root mean square of the residuals is  0
The df corrected root mean square of the residuals is  0.01
The number of observations was  213  with Chi Square =  2.82  with prob <  1

Tucker Lewis Index of factoring reliability =  1.027
RMSEA index =  0  and the 90 % confidence intervals are  0 0.023
BIC =  -61.51
Fit based upon off diagonal values = 1
Measures of factor score adequacy
                                                MR1  MR2  MR3
Correlation of scores with factors             0.96 0.92 0.90
Multiple R square of scores with factors       0.93 0.85 0.81
Minimum correlation of possible factor scores  0.86 0.71 0.63
```

the *GPArotation* package. Orthogonal rotations are `varimax` and `quartimax`. Oblique transformations include `oblimin`, `quartimin` and then two targeted rotation functions `Promax` and `target.rot`. The latter of these will transform a loadings matrix towards an arbitrary target matrix. The default is to transform towards an independent cluster solution.

Using the Thurstone data set, three factors were requested and then transformed into an independent clusters solution using `target.rot` (Table 2).

Table 2: The 9 variable problem from Thurstone is a classic example of factoring where there is a higher order factor, g, that accounts for the correlation between the factors. The extraction method was principal axis. The transformation was a targeted transformation to a simple cluster solution.

```
> f3 <- fa(Thurstone, 3, n.obs = 213, fm = "pa")
> f3o <- target.rot(f3)
> f3o
Call: NULL
Standardized loadings based upon correlation matrix
                   PA1   PA2   PA3   h2   u2
Sentences         0.89 -0.03  0.07 0.81 0.19
Vocabulary        0.89  0.07  0.00 0.80 0.20
Sent.Completion   0.83  0.04  0.03 0.70 0.30
First.Letters    -0.02  0.85 -0.01 0.73 0.27
4.Letter.Words   -0.05  0.74  0.09 0.57 0.43
Suffixes          0.17  0.63 -0.09 0.43 0.57
Letter.Series    -0.06 -0.08  0.84 0.69 0.31
Pedigrees         0.33 -0.09  0.48 0.37 0.63
Letter.Group     -0.14  0.16  0.64 0.45 0.55

                 PA1  PA2  PA3
SS loadings    2.45 1.72 1.37
Proportion Var 0.27 0.19 0.15
Cumulative Var 0.27 0.46 0.62

 With factor correlations of
     PA1  PA2  PA3
PA1 1.00 0.02 0.08
PA2 0.02 1.00 0.09
PA3 0.08 0.09 1.00
```

4.1.3 Weighted Least Squares Factor Analysis

Similar to the minres approach of minimizing the squared residuals, factor method "wls" weights the squared residuals by their uniquenesses. This tends to produce slightly smaller overall residuals. In the example of weighted least squares, the output is shown by using the `print` function with the cut option set to 0. That is, all loadings are shown (Table 3).

The unweighted least squares solution may be shown graphically using the `factor.plot` function which is called by the generic `plot` function (Figure 9. Factors were transformed obliquely using a oblimin. These solutions may be shown as item by factor plots (Figure 9 or by a structure diagram (Figure 10.

A comparison of these three approaches suggests that the minres solution is more similar to a maximum likelihood solution and fits slightly better than the pa or wls solutions. Comparisons with SPSS suggest that the pa solution matches the SPSS OLS solution, but that the minres solution is slightly better. At least in one test data set, the weighted least squares solutions, although fitting equally well, had slightly different structure loadings.

4.1.4 Principal Components analysis

An alternative to factor analysis, which is unfortunately frequently confused with factor analysis, is principal components analysis. Although the goals of PCA and FA are similar, PCA is a descriptive model of the data, while FA is a structural model. Psychologists typically use PCA in a manner similar to factor analysis and thus the `principal` function produces output that is perhaps more understandable than that produced by `princomp` in the *stats* package. Table 4 shows a PCA of the Thurstone 9 variable problem rotated using the `Promax` function. Note how the loadings from the factor model are similar but smaller than the principal component loadings. This is because the PCA model attempts to account for the entire variance of the correlation matrix, while FA accounts for just the *common variance*. This distinction becomes most important for small correlation matrices. Also note how the goodness of fit statistics, based upon the residual off diagonal elements, is much worse than the `fa` solution.

4.1.5 Hierarchical and bi-factor solutions

For a long time structural analysis of the ability domain have considered the problem of factors that are themselves correlated. These correlations may themselves be factored to produce a higher order, general factor. An alternative (Holzinger and Swineford, 1937; Jensen and Weng, 1994) is to consider the general factor affecting each item, and then

Table 3: The 9 variable problem from Thurstone is a classic example of factoring where there is a higher order factor, g, that accounts for the correlation between the factors. The factors were extracted using a weighted least squares algorithm. All loadings are shown by using the cut=0 option in the `print.psych` function.

```
> f3w <- fa(Thurstone, 3, n.obs = 213, fm = "wls")
> print(f3w, cut = 0, digits = 3)
Factor Analysis using method =  wls
Call: fa(r = Thurstone, nfactors = 3, n.obs = 213, fm = "wls")
Standardized loadings based upon correlation matrix
                   WLS1   WLS2   WLS3    h2    u2
Sentences         0.905 -0.034  0.040 0.822 0.178
Vocabulary        0.890  0.066 -0.031 0.835 0.165
Sent.Completion   0.833  0.034  0.007 0.735 0.265
First.Letters    -0.002  0.855  0.003 0.731 0.269
4.Letter.Words   -0.016  0.743  0.106 0.629 0.371
Suffixes          0.180  0.626 -0.082 0.496 0.504
Letter.Series     0.033 -0.015  0.838 0.719 0.281
Pedigrees         0.381 -0.051  0.464 0.505 0.495
Letter.Group     -0.062  0.209  0.632 0.527 0.473

                 WLS1  WLS2  WLS3
SS loadings    2.647 1.864 1.488
Proportion Var 0.294 0.207 0.165
Cumulative Var 0.294 0.501 0.667

 With factor correlations of
      WLS1  WLS2  WLS3
WLS1 1.000 0.591 0.535
WLS2 0.591 1.000 0.516
WLS3 0.535 0.516 1.000

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the null model are  36  and the objective function was  5.198 wi
The degrees of freedom for the model are 12  and the objective function was  0.014

The root mean square of the residuals is  0.004
The df corrected root mean square of the residuals is  0.01
The number of observations was  213  with Chi Square =  2.886  with prob <  0.996

Tucker Lewis Index of factoring reliability =  1.0264
RMSEA index =  0  and the 90 % confidence intervals are  0 0.0232
BIC =  -61.45
Fit based upon off diagonal values = 1
Measures of factor score adequacy
                                                 WLS1  WLS2  WLS3
Correlation of scores with factors              0.964 0.923 0.902
Multiple R square of scores with factors        0.929 0.853 0.814
Minimum correlation of possible factor scores   0.858 0.706 0.627
```

```
> plot(f3t)
```

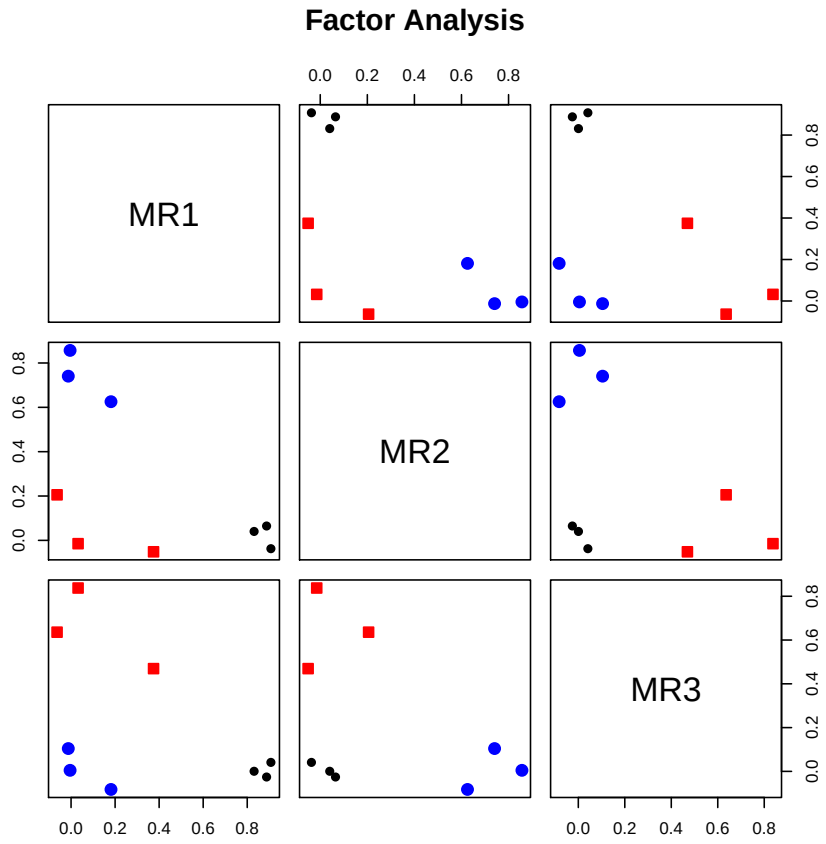

Figure 9: A graphic representation of the 3 oblique factors from the Thurstone data using `plot`. Factors were transformed to an oblique solution using the oblimin function from the GPArotation package.

```
> fa.diagram(f3t)
```

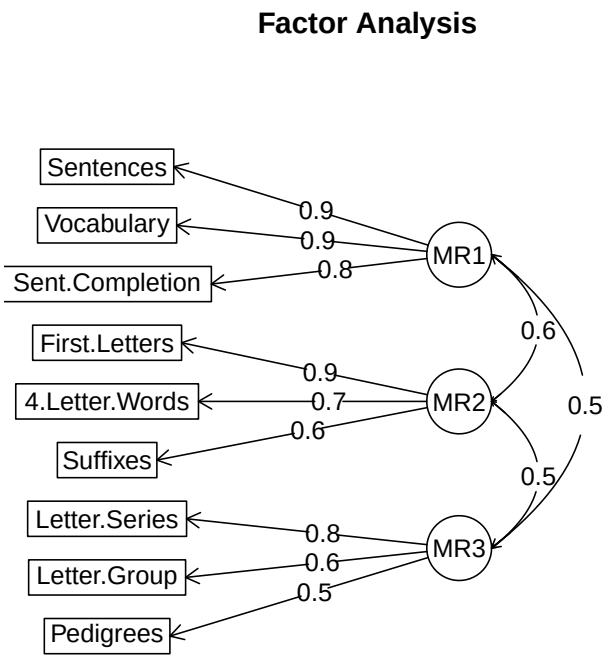

Figure 10: A graphic representation of the 3 oblique factors from the Thurstone data using `fa.diagram`. Factors were transformed to an oblique solution using oblimin.

Table 4: The Thurstone problem can also be analyzed using Principal Components Analysis. Compare this to Table 2. The loadings are higher for the PCA because the model accounts for the unique as well as the common variance.The fit of the off diagonal elements, however, is much worse than the `fa` results.

```
> p3p <- principal(Thurstone, 3, n.obs = 213, rotate = "Promax")
> p3p
Principal Components Analysis
Call: principal(r = Thurstone, nfactors = 3, rotate = "Promax", n.obs = 213)
Standardized loadings based upon correlation matrix
                   PC1   PC2   PC3   h2   u2
Sentences         0.92  0.01  0.01 0.86 0.14
Vocabulary        0.90  0.10 -0.05 0.86 0.14
Sent.Completion   0.91  0.04 -0.04 0.83 0.17
First.Letters     0.01  0.84  0.07 0.78 0.22
4.Letter.Words   -0.05  0.81  0.17 0.75 0.25
Suffixes          0.18  0.79 -0.15 0.70 0.30
Letter.Series     0.03 -0.03  0.88 0.78 0.22
Pedigrees         0.45 -0.16  0.57 0.67 0.33
Letter.Group     -0.19  0.19  0.86 0.75 0.25

                 PC1  PC2  PC3
SS loadings    2.83 2.19 1.96
Proportion Var 0.31 0.24 0.22
Cumulative Var 0.31 0.56 0.78

 With factor correlations of
     PC1  PC2  PC3
PC1 1.00 0.51 0.53
PC2 0.51 1.00 0.44
PC3 0.53 0.44 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  5.2
The degrees of freedom for the model are 12  and the objective function was  0.62
The number of observations was  213  with Chi Square =  127.9  with prob <  1.6e-21

Fit based upon off diagonal values = 0.98
```

to have group factors account for the residual variance. Exploratory factor solutions to produce a hierarchical or a bifactor solution are found using the `omega` function. This technique has more recently been applied to the personality domain to consider such things as the structure of neuroticism (treated as a general factor, with lower order factors of anxiety, depression, and aggression).

Consider the 9 Thurstone variables analyzed in the prior factor analyses. The correlations between the factors (as shown in Figure 10 can themselves be factored. This results in a higher order factor model (Figure 11). An an alternative solution is to take this higher order model and then solve for the general factor loadings as well as the loadings on the residualized lower order factors using the *Schmid-Leiman* procedure. (Figure 12). Yet another solution is to use structural equation modeling to directly solve for the general and group factors.

4.1.6 Item Cluster Analysis: iclust

An alternative to factor or components analysis is *cluster analysis*. The goal of cluster analysis is the same as factor or components analysis (reduce the complexity of the data and attempt to identify homogeneous subgroupings). Mainly used for clustering people or objects (e.g., projectile points if an anthropologist, DNA if a biologist, galaxies if an astronomer), clustering may be used for clustering items or tests as well. Introduced to psychologists by Tryon (1939) in the 1930's, the cluster analytic literature exploded in the 1970s and 1980s (Blashfield, 1980; Blashfield and Aldenderfer, 1988; Everitt, 1974; Hartigan, 1975). Much of the research is in taxonmetric applications in biology Sneath and Sokal (1973); Sokal and Sneath (1963) and marketing (Cooksey and Soutar, 2006) where clustering remains very popular. It is also used for taxonomic work in forming clusters of people in family (Henry et al., 2005) and clinical psychology (Martinent and Ferrand, 2007; Mun et al., 2008). Interestingly enough it has has had limited applications to psychometrics. This is unfortunate, for as has been pointed out by e.g. (Tryon, 1935; Loevinger et al., 1953), the theory of factors, while mathematically compelling, offers little that the geneticist or behaviorist or perhaps even non-specialist finds compelling. Cooksey and Soutar (2006) reviews why the `iclust` algorithm is particularly appropriate for scale construction in marketing.

Hierarchical cluster analysis forms clusters that are nested within clusters. The resulting *tree diagram* (also known somewhat pretentiously as a *rooted dendritic structure*) shows the nesting structure. Although there are many hierarchical clustering algorithms in R (e.g., `agnes`, `hclust`, and `iclust`), the one most applicable to the problems of scale construction is `iclust` (Revelle, 1979).

1. Find the proximity (e.g. correlation) matrix,

```
> om.h <- omega(Thurstone, n.obs = 213, sl = FALSE)
> op <- par(mfrow = c(1, 1))
```

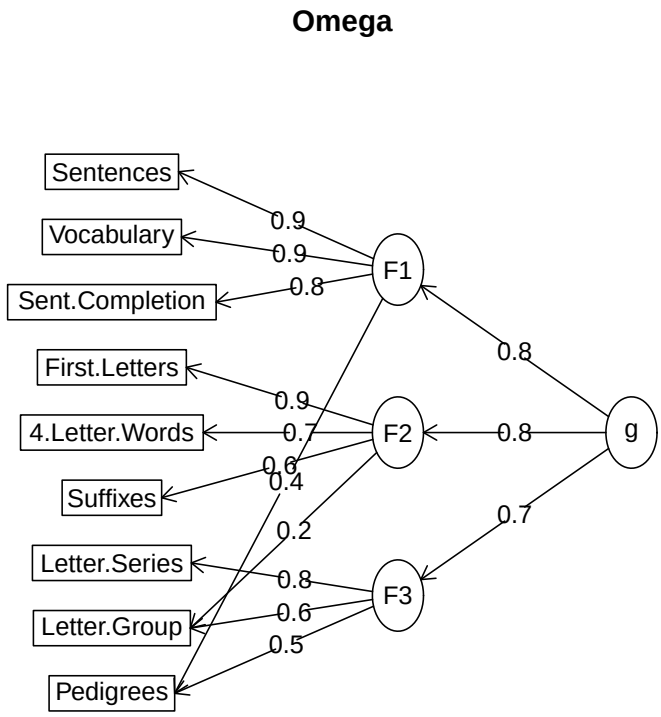

Figure 11: A higher order factor solution to the Thurstone 9 variable problem

```
> om <- omega(Thurstone, n.obs = 213)
```

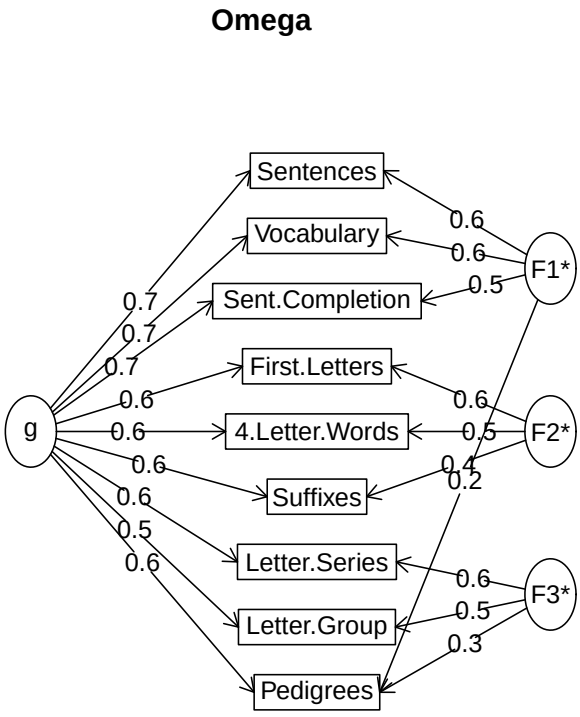

Figure 12: A bifactor factor solution to the Thurstone 9 variable problem

2. Identify the most similar pair of items
3. Combine this most similar pair of items to form a new variable (cluster),
4. Find the similarity of this cluster to all other items and clusters,
5. Repeat steps 2 and 3 until some criterion is reached (e.g., typicallly, if only one cluster remains or in `iclust` if there is a failure to increase reliability coefficients α or β).
6. Purify the solution by reassigning items to the most similar cluster center.

iclust forms clusters of items using a hierarchical clustering algorithm until one of two measures of internal consistency fails to increase (Revelle, 1979). The number of clusters may be specified a priori, or found empirically. The resulting statistics include the average split half reliability, α (Cronbach, 1951), as well as the worst split half reliability, β (Revelle, 1979), which is an estimate of the general factor saturation of the resulting scale.

The previous analysis was done using the Pearson correlation. A somewhat cleaner structure is obtained when using the `polychoric` function to find polychoric correlations.

```
data(bfi)
r.poly <- polychoric(bfi[1:25])
ic.poly <- iclust(r.poly)$rho
iclust.diagram(ic.poly)
```

4.2 Comparing factor/component/cluster solutions

Cluster analysis, factor analysis, and principal components analysis all produce structure matrices (matrices of correlations between the dimensions and the variables) that can in turn be compared in terms of the *congruence coefficient* which is just cosine of the angle between the dimensions. This is done using the `factor.congruence` function.

Consider the case of a four factor solution and four cluster solution to the Big Five problem.

```
> f4 <- fa(bfi[1:25], 4, fm = "pa")
> factor.congruence(f4, ic)
```

```
      C20   C16   C15   C21
PA1  0.92 -0.32 -0.44 -0.40
PA2 -0.26  0.95  0.33  0.12
PA3  0.35 -0.24 -0.88 -0.37
PA4  0.29 -0.12 -0.27 -0.90
```

Now consider the solutions to the Thurstone problem, considering the factor, component, and bifactor solution. Note the use of the list of solutions to compare.

```
> data(bfi)
> ic <- iclust(bfi[1:25])
```

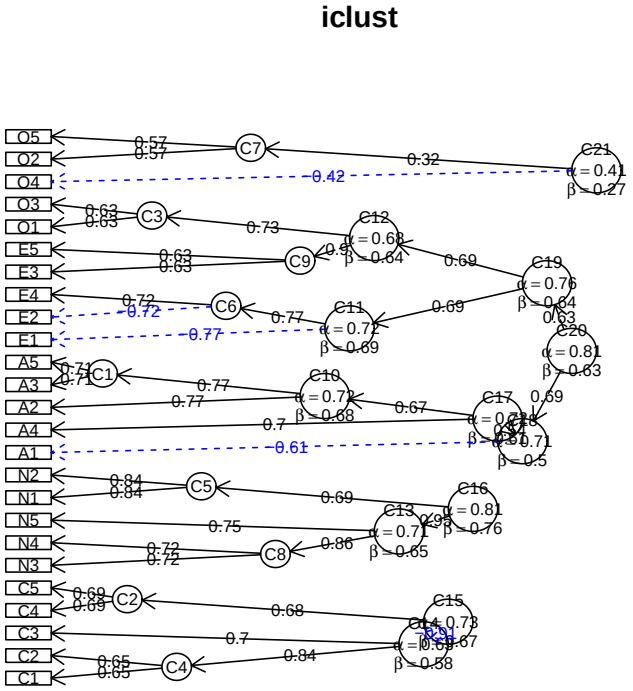

Figure 13: Using the `iclust` function to find the cluster structure of 25 personality items (the three demographic variables were excluded from this analysis.

Table 5: The summary statistics from an iclust analysis shows three large clusters and smaller cluster.

```
> summary(ic)
ICLUST (Item Cluster Analysis)Call: ICLUST(r.mat = r.mat, nclusters = nclusters, alpha = al
    beta.size = beta.size, alpha.size = alpha.size, correct = correct,
    correct.cluster = correct.cluster, reverse = reverse, beta.min = beta.min,
    output = output, digits = digits, labels = labels, cut = cut,
    n.iterations = n.iterations, title = title, plot = plot,
    weighted = weighted, cor.gen = cor.gen, SMC = SMC)
iclust

Purified Alpha:
 C20  C16  C15  C21
0.80 0.81 0.73 0.61

 Guttman Lambda6  *
 C20  C16  C15  C21
0.82 0.81 0.72 0.61

Original Beta:
      C20       C16       C15       C21
0.6293288 0.7616692 0.6736569 0.2685913

Cluster size:
C20 C16 C15 C21
 10   5   5   5

Purified scale intercorrelations
 reliabilities on diagonal
 correlations corrected for attenuation above diagonal:
      C20   C16   C15   C21
C20  0.80 -0.29 -0.40 -0.33
C16 -0.24  0.81  0.29  0.11
C15 -0.30  0.22  0.73  0.30
C21 -0.23  0.07  0.20  0.61
```

```
> factor.congruence(list(f3t, f3o, om, p3p))
```

	MR1	MR2	MR3	PA1	PA2	PA3	g	F1*	F2*	F3*	h2	PC1	PC2	PC3
MR1	1.00	0.06	0.09	1.00	0.06	0.13	0.72	1.00	0.06	0.09	0.74	1.00	0.08	0.04
MR2	0.06	1.00	0.08	0.03	1.00	0.06	0.60	0.06	1.00	0.08	0.57	0.04	0.99	0.12
MR3	0.09	0.08	1.00	0.01	0.01	1.00	0.52	0.09	0.08	1.00	0.51	0.06	0.02	0.99
PA1	1.00	0.03	0.01	1.00	0.04	0.05	0.67	1.00	0.03	0.01	0.69	1.00	0.06	-0.04
PA2	0.06	1.00	0.01	0.04	1.00	0.00	0.57	0.06	1.00	0.01	0.54	0.04	0.99	0.05
PA3	0.13	0.06	1.00	0.05	0.00	1.00	0.54	0.13	0.06	1.00	0.53	0.10	0.01	0.99
g	0.72	0.60	0.52	0.67	0.57	0.54	1.00	0.72	0.60	0.52	0.99	0.69	0.58	0.50
F1*	1.00	0.06	0.09	1.00	0.06	0.13	0.72	1.00	0.06	0.09	0.74	1.00	0.08	0.04
F2*	0.06	1.00	0.08	0.03	1.00	0.06	0.60	0.06	1.00	0.08	0.57	0.04	0.99	0.12
F3*	0.09	0.08	1.00	0.01	0.01	1.00	0.52	0.09	0.08	1.00	0.51	0.06	0.02	0.99
h2	0.74	0.57	0.51	0.69	0.54	0.53	0.99	0.74	0.57	0.51	1.00	0.71	0.56	0.49
PC1	1.00	0.04	0.06	1.00	0.04	0.10	0.69	1.00	0.04	0.06	0.71	1.00	0.06	0.00
PC2	0.08	0.99	0.02	0.06	0.99	0.01	0.58	0.08	0.99	0.02	0.56	0.06	1.00	0.05
PC3	0.04	0.12	0.99	-0.04	0.05	0.99	0.50	0.04	0.12	0.99	0.49	0.00	0.05	1.00

4.3 Determining the number of dimensions to extract.

How many dimensions to use to represent a correlation matrix is an unsolved problem in psychometrics. There are many solutions to this problem, none of which is uniformly the best. Henry Kaiser once said that "a solution to the number-of factors problem in factor analysis is easy, that he used to make up one every morning before breakfast. But the problem, of course is to find *the* solution, or at least a solution that others will regard quite highly not as the best" Horn and Engstrom (1979).

Techniques most commonly used include

1) Extracting factors until the chi square of the residual matrix is not significant.

2) Extracting factors until the change in chi square from factor n to factor n+1 is not significant.

3) Extracting factors until the eigen values of the real data are less than the corresponding eigen values of a random data set of the same size (parallel analysis) `fa.parallel` (Horn, 1965).

4) Plotting the magnitude of the successive eigen values and applying the scree test (a sudden drop in eigen values analogous to the change in slope seen when scrambling up the talus slope of a mountain and approaching the rock face (Cattell, 1966).

5) Extracting factors as long as they are interpretable.

6) Using the Very Structure Criterion (`vss` (Revelle and Rocklin, 1979).

7) Using Wayne Velicer's Minimum Average Partial (`MAP`) criterion (Velicer, 1976).

8) Extracting principal components until the eigen value < 1.

Each of the procedures has its advantages and disadvantages. Using either the chi square test or the change in square test is, of course, sensitive to the number of subjects and leads to the nonsensical condition that if one wants to find many factors, one simply runs more subjects. Parallel analysis is partially sensitive to sample size in that for large samples the eigen values of random factors will be very small. The scree test is quite appealing but can lead to differences of interpretation as to when the scree "breaks". Extracting interpretable factors means that the number of factors reflects the investigators creativity more than the data. vss, while very simple to understand, will not work very well if the data are very factorially complex. (Simulations suggests it will work fine if the complexities of some of the items are no more than 2). The eigen value of 1 rule, although the default for many programs, seems to be a rough way of dividing the number of variables by 3 and is probably the worst of all criteria.

An additional problem in determining the number of factors is what is considered a factor. Many treatments of factor analysis assume that the residual correlation matrix after the factors of interest are extracted is composed of just random error. An alternative concept is that the matrix is formed from major factors of interest but that there are also numerous minor factors of no substantive interest but that account for some of the shared covariance between variables. The presence of such minor factors can lead one to extract too many factors and to reject solutions on statistical grounds of misfit that are actually very good fits to the data. This problem is partially addressed later in the discussion of simulating complex structures using `sim.structure` and of small extraneous factors using the `sim.minor` function.

4.3.1 Very Simple Structure

The `vss` function compares the fit of a number of factor analyses with the loading matrix "simplified" by deleting all except the c greatest loadings per item, where c is a measure of factor complexity citerevelle:vss. Included in `vss` is the MAP criterion (Minimum Absolute Partial correlation) of Velicer (1976).

Using the Very Simple Structure criterion for the bfi data suggests that 4 factors are optimal (Figure 14). However, the MAP criterion suggests that 5 is optimal.

```
> vss
```

```
Very Simple Structure of  Very Simple Structure of a Big 5 inventory
Call: VSS(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
```

```
> vss <- vss(bfi[1:25], title = "Very Simple Structure of a Big 5 inventory")
```

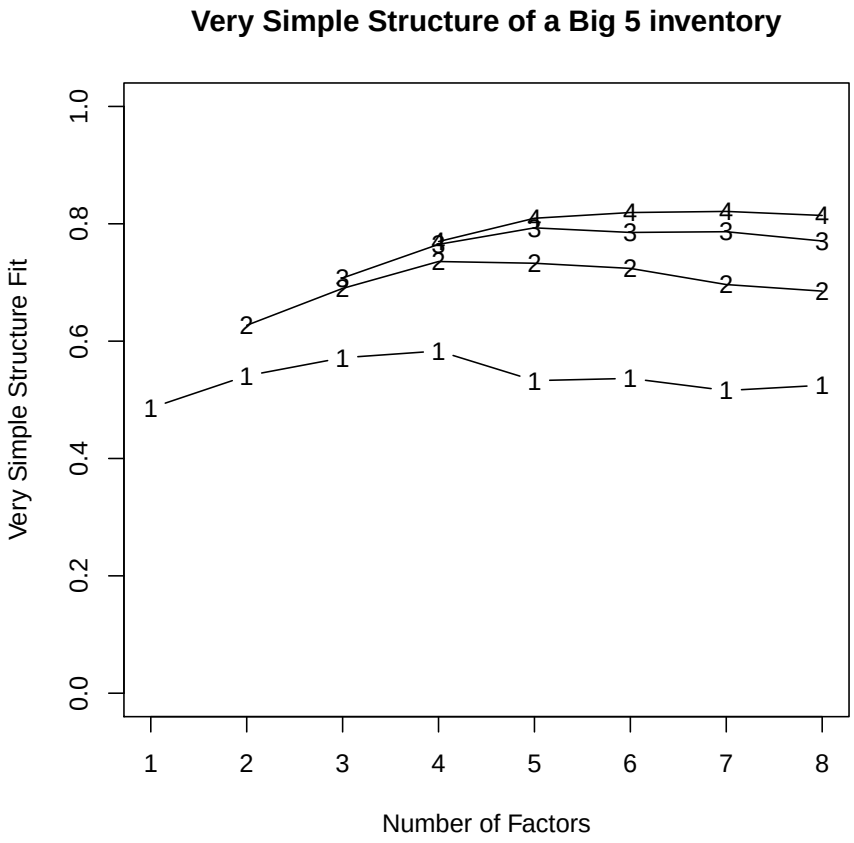

Figure 14: The Very Simple Structure criterion for the number of factors compares solutions for various levels of item complexity and various numbers of factors.

```
    n.obs = n.obs, plot = plot, title = title)
VSS complexity 1 achieves a maximimum of 0.58  with  4  factors
VSS complexity 2 achieves a maximimum of 0.74  with  4  factors

The Velicer MAP criterion achieves a minimum of 0.01  with  5  factors

Velicer MAP
[1] 0.02 0.02 0.02 0.02 0.01 0.02 0.02 0.02

Very Simple Structure Complexity 1
[1] 0.49 0.54 0.57 0.58 0.53 0.54 0.52 0.52

Very Simple Structure Complexity 2
[1] 0.00 0.63 0.69 0.74 0.73 0.72 0.70 0.69
```

4.3.2 Parallel Analysis

An alternative way to determine the number of factors is to compare the solution to random data with the same properties as the real data set. If the input is a data matrix, the comparison includes random samples from the real data, as well as normally distributed random data with the same number of subjects and variables. For the BFI data, parallel analysis suggests that 6 factors might be most appropriate (Figure 15). It is interesting to compare `fa.parallel` with the `paran` from the *paran* package. This latter uses smcs to estimate communalities. Simulations of known structures with a particular number of major factors but with the presence of trivial, minor (but not zero) factors, show that using smcs will tend to lead to too many factors.

4.4 Reliability analysis

Surprisingly, 105 years after Spearman (1904) introduced the concept of reliability to psychologists, there are still multiple approaches for measuring it. Although very popular, Cronbach's α (1951) underestimates the reliability of a test and over estimates the first factor saturation (Revelle and Zinbarg, 2009).

α (Cronbach, 1951) is the same as Guttman's $\lambda 3$ (Guttman, 1945) and may be found by

$$\lambda_3 = \frac{n}{n-1}\left(1 - \frac{tr(\vec{V})_x}{V_x}\right) = \frac{n}{n-1}\frac{V_x - tr(\vec{V}_x)}{V_x} = \alpha$$

Perhaps because it is so easy to calculate and is available in most commercial programs,

```
> fa.parallel(bfi[1:25], main = "Parallel Analysis of a Big 5 inventory")
Parallel analysis suggests that the number of factors =  6  and the number of components =
```

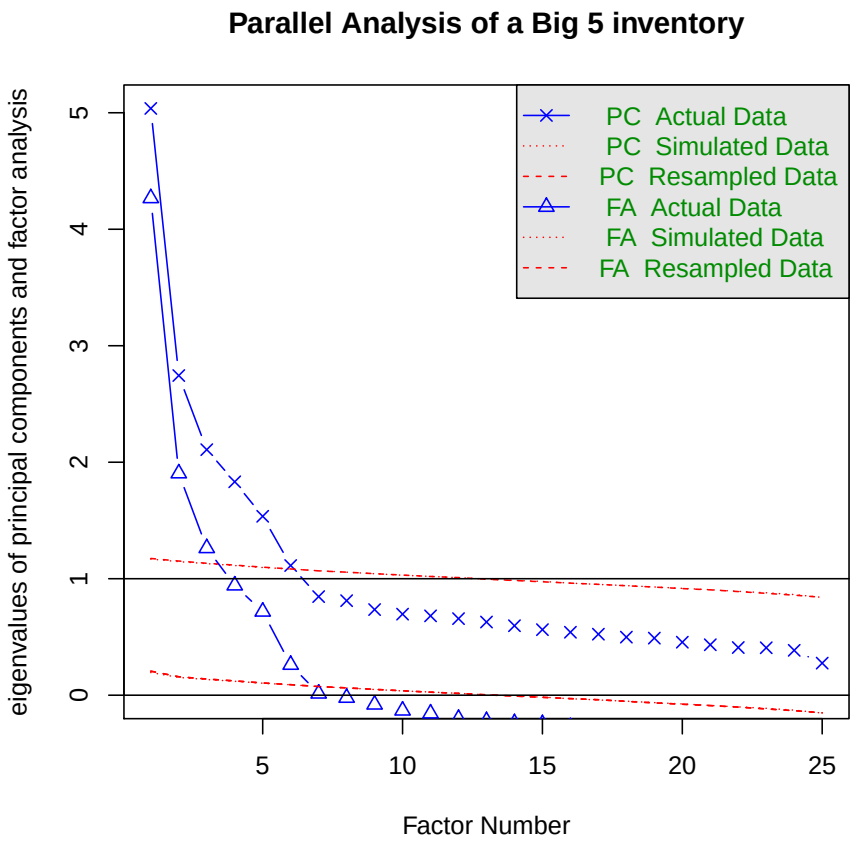

Figure 15: Parallel analysis compares factor and principal components solutions to the real data as well as resampled data. Although vss suggests 4 factors, MAP 5, parallel analysis suggests 6. One more demonstration of Kaiser's dictum.

alpha is without doubt the most frequently reported measure of internal consistency reliability. Alpha is the mean of all possible spit half reliabilities (corrected for test length). For a unifactorial test, it is a reasonable estimate of the first factor saturation, although if the test has any microstructure (i.e., if it is "lumpy") coefficients β (Revelle, 1979; see `iclust`) and ω_h (see `omega`) are more appropriate estimates of the general factor saturation. $\omega_t$is a better estimate of the reliability of the total test.

Guttman's λ_6 (G6) considers the amount of variance in each item that can be accounted for the linear regression of all of the other items (the squared multiple correlation or smc), or more precisely, the variance of the errors, e_j^2, and is

$$\lambda_6 = 1 - \frac{\sum e_j^2}{V_x} = 1 - \frac{\sum(1 - r_{smc}^2)}{V_x}.$$

The squared multiple correlation is a lower bound for the item communality and as the number of items increases, becomes a better estimate.

G6 is also sensitive to lumpiness in the test and should not be taken as a measure of unifactorial structure. For lumpy tests, it will be greater than alpha. For tests with equal item loadings, alpha > G6, but if the loadings are unequal or if there is a general factor, G6 > alpha. G6 estimates item reliability by the squared multiple correlation of the other items in a scale. A modification of G6, G6*, takes as an estimate of an item reliability the smc with all the items in an inventory, including those not keyed for a particular scale. This will lead to a better estimate of the reliable variance of a particular item.

Alpha, G6 and G6* are positive functions of the number of items in a test as well as the average intercorrelation of the items in the test. When calculated from the item variances and total test variance, as is done here, raw alpha is sensitive to differences in the item variances. Standardized alpha is based upon the correlations rather than the covariances.

More complete reliability analyses of a single scale can be done using the `omega` function which finds ω_h and ω_t based upon a hierarchical factor analysis.

Alternative functions `score.items` and `cluster.cor` will also score multiple scales and report more useful statistics. "Standardized" alpha is calculated from the inter-item correlations and will differ from raw alpha.

Functions for examining the reliability of a single scale or a set of scales include:

alpha Internal consistency measures of reliability range from ω_h to α to ω_t. The `alpha` function reports two estimates: Cronbach's coefficient α and Guttman's λ_6. Also reported are item - whole correlations, α if an item is omitted, and item means and standard deviations.

guttman Eight alternative estimates of test reliability include the six discussed by Guttman (1945), four discussed by ten Berge and Zergers (1978) ($\mu_0 \ldots \mu_3$) as well as β (the worst split half, Revelle, 1979), the glb (greatest lowest bound) discussed by Bentler and Woodward (1980), and ω_h andω_t (McDonald, 1999; Zinbarg et al., 2005).

omega Calculate McDonald's omega estimates of general and total factor saturation. (Revelle and Zinbarg (2009) compare these coefficients with real and artificial data sets.)

cluster.cor Given a n x c cluster definition matrix of -1s, 0s, and 1s (the keys) , and a n x n correlation matrix, find the correlations of the composite clusters.

score.items Given a matrix or data.frame of k keys for m items (-1, 0, 1), and a matrix or data.frame of items scores for m items and n people, find the sum scores or average scores for each person and each scale. If the input is a square matrix, then it is assumed that correlations or covariances were used, and the raw scores are not available. In addition, report Cronbach's alpha, coefficient G6*, the average r, the scale intercorrelations, and the item by scale correlations (both raw and corrected for item overlap and scale reliability). Replace missing values with the item median or mean if desired. Will adjust scores for reverse scored items.

score.multiple.choice Ability tests are typically multiple choice with one right answer. score.multiple.choice takes a scoring key and a data matrix (or data.frame) and finds total or average number right for each participant. Basic test statistics (alpha, average r, item means, item-whole correlations) are also reported.

4.4.1 Reliability of a single scale

A conventional (but non-optimal) estimate of the internal consistency reliability of a test is coefficient α (Cronbach, 1951). Alternative estimates are Guttman's λ_6, Revelle's β, McDonald's ω_h and ω_t. Consider a simulated data set, representing 9 items with a hierarchical structure and the following correlation matrix. Then using the `alpha` function, the α and λ_6 estimates of reliability may be found for all 9 items, as well as the if one item is dropped at a time.

```
> set.seed(42)
> r9 <- sim.hierarchical(n = 500, raw = TRUE)$observed
> round(cor(r9), 2)
```

```
     V1   V2   V3   V4   V5   V6   V7   V8   V9
V1 1.00 0.52 0.39 0.39 0.31 0.28 0.23 0.30 0.22
V2 0.52 1.00 0.41 0.37 0.33 0.27 0.22 0.22 0.18
V3 0.39 0.41 1.00 0.28 0.23 0.25 0.18 0.16 0.08
```

```
V4 0.39 0.37 0.28 1.00 0.44 0.36 0.21 0.21 0.09
V5 0.31 0.33 0.23 0.44 1.00 0.31 0.19 0.13 0.16
V6 0.28 0.27 0.25 0.36 0.31 1.00 0.15 0.07 0.08
V7 0.23 0.22 0.18 0.21 0.19 0.15 1.00 0.25 0.20
V8 0.30 0.22 0.16 0.21 0.13 0.07 0.25 1.00 0.22
V9 0.22 0.18 0.08 0.09 0.16 0.08 0.20 0.22 1.00
```

```
> alpha(r9)
```

```
Reliability analysis
Call: alpha(x = r9)

  raw_alpha std.alpha G6(smc) average_r   mean   sd
      0.75      0.75    0.74      0.25 -0.019 0.57

 Reliability if an item is dropped:
   raw_alpha std.alpha G6(smc) average_r
V1      0.70      0.70    0.69      0.22
V2      0.70      0.70    0.69      0.23
V3      0.72      0.72    0.72      0.25
V4      0.71      0.71    0.70      0.23
V5      0.72      0.72    0.71      0.24
V6      0.73      0.73    0.73      0.25
V7      0.74      0.74    0.73      0.26
V8      0.74      0.74    0.73      0.26
V9      0.75      0.75    0.74      0.27

 Item statistics
     n    r r.cor r.drop    mean   sd
V1 500 0.70  0.68   0.58 -0.0272 0.96
V2 500 0.68  0.64   0.55 -0.0030 1.01
V3 500 0.58  0.50   0.42  0.0202 0.97
V4 500 0.65  0.60   0.51 -0.0397 0.98
V5 500 0.60  0.53   0.45 -0.0205 1.03
V6 500 0.53  0.44   0.37 -0.0165 0.97
V7 500 0.51  0.40   0.34 -0.0351 1.03
V8 500 0.50  0.39   0.33 -0.0028 1.00
V9 500 0.43  0.30   0.26 -0.0494 1.03
```

Some scales have items that need to be reversed before being scored. Rather than reversing the items in the raw data, it is more convenient to just specify which items need to be reversed scored. This may be done in `alpha` by specifying a *keys* vector of 1s and -1s.

(This concept of keys vector is more useful when scoring multiple scale inventories, see below.) As an example, consider scoring the 7 attitude items in the attitude data set. Assume a conceptual mistake in that item 2 is to be scored (incorrectly) negatively.

```
> keys <- c(1, -1, 1, 1, 1, 1, 1)
> alpha(attitude, keys)

Reliability analysis
Call: alpha(x = attitude, keys = keys)

  raw_alpha std.alpha G6(smc) average_r mean  sd
      0.43      0.52    0.75      0.14   60 8.2

 Reliability if an item is dropped:
           raw_alpha std.alpha G6(smc) average_r
rating          0.32      0.44    0.67     0.114
complaints      0.80      0.80    0.82     0.394
privileges      0.27      0.41    0.72     0.103
learning        0.14      0.31    0.64     0.069
raises          0.14      0.27    0.61     0.059
critical        0.36      0.47    0.76     0.130
advance         0.21      0.34    0.66     0.079

 Item statistics
             n     r r.cor r.drop mean   sd
rating      30  0.60  0.60   0.33   65 12.2
complaints 30 -0.58 -0.74  -0.74   67 13.3
privileges 30  0.65  0.54   0.42   53 12.2
learning   30  0.79  0.78   0.64   56 11.7
raises     30  0.83  0.85   0.69   65 10.4
critical   30  0.53  0.35   0.27   75  9.9
advance    30  0.75  0.71   0.58   43 10.3
```

Note how the reliability of the 7 item scales with an incorrectly reversed item is very poor, but if the item 2 is dropped then the reliability is improved substantially. This suggests that item 2 was incorrectly scored. Doing the analysis again with item 2 positively scored produces much more favorable results.

```
> keys <- c(1, 1, 1, 1, 1, 1, 1)
> alpha(attitude, keys)

Reliability analysis
Call: alpha(x = attitude, keys = keys)
```

```
  raw_alpha std.alpha G6(smc) average_r mean  sd
      0.84      0.84    0.88      0.43   60 8.2

 Reliability if an item is dropped:
           raw_alpha std.alpha G6(smc) average_r
rating          0.81      0.81    0.83      0.41
complaints      0.80      0.80    0.82      0.39
privileges      0.83      0.82    0.87      0.44
learning        0.80      0.80    0.84      0.40
raises          0.80      0.78    0.83      0.38
critical        0.86      0.86    0.89      0.51
advance         0.84      0.83    0.86      0.46

 Item statistics
            n    r r.cor r.drop mean   sd
rating     30 0.76  0.75   0.67   65 12.2
complaints 30 0.81  0.82   0.74   67 13.3
privileges 30 0.68  0.60   0.56   53 12.2
learning   30 0.80  0.78   0.71   56 11.7
raises     30 0.86  0.85   0.79   65 10.4
critical   30 0.45  0.31   0.27   75  9.9
advance    30 0.62  0.56   0.46   43 10.3
```

It is useful when considering items for a potential scale to examine the item distribution. This is done in `score.items` as well as in `alpha`.

```
> items <- sim.congeneric(N = 500, short = FALSE, low = -2, high = 2,
+     categorical = TRUE)
> alpha(items$observed)

Reliability analysis
Call: alpha(x = items$observed)

  raw_alpha std.alpha G6(smc) average_r   mean   sd
      0.69      0.69    0.63      0.36 -0.002 0.71

 Reliability if an item is dropped:
   raw_alpha std.alpha G6(smc) average_r
V1      0.57      0.57    0.47      0.30
V2      0.62      0.62    0.52      0.35
V3      0.65      0.64    0.56      0.38
V4      0.66      0.66    0.58      0.40

 Item statistics
     n    r r.cor r.drop   mean   sd
V1 500 0.77  0.68   0.56 -0.004 0.99
V2 500 0.73  0.60   0.49  0.020 0.99
```

```
V3 500 0.70  0.53   0.44 -0.002 0.98
V4 500 0.68  0.49   0.41 -0.022 0.98

Non missing response frequency for each item
     -2   -1    0    1    2 miss
V1 0.06 0.25 0.38 0.25 0.06    0
V2 0.05 0.26 0.37 0.26 0.06    0
V3 0.07 0.23 0.39 0.26 0.05    0
V4 0.06 0.25 0.40 0.23 0.06    0
```

4.4.2 Using `omega` to find the reliability of a single scale

Two alternative estimates of reliability that take into account the hierarchical structure of the inventory are McDonald's ω_h and ω_t. These may be found using the `omega` function for an exploratory analysis (See Figure 16) or `omegaSem` for a confirmatory analysis using the *sem* based upon the exploratory solution from `omega`.

McDonald has proposed coefficient omega (hierarchical) (ω_h) as an estimate of the general factor saturation of a test. Zinbarg, Revelle, Yovel and Li (2005) http://personality-project.org/revelle/publications/zinbarg.revelle.pmet.05.pdf compare McDonald's ω_h to Cronbach's α and Revelle's β. They conclude that ω_h is the best estimate. (See also Zinbarg et al., 2006 and Revelle and Zinbarg (2009) http://personality-project.org/revelle/publications/revelle.zinbarg.08.pdf).

One way to find ω_h is to do a factor analysis of the original data set, rotate the factors obliquely, factor that correlation matrix, do a Schmid-Leiman (`schmid`) transformation to find general factor loadings, and then find ω_h

ω_h differs slightly as a function of how the factors are estimated. Four options are available, the default will do a minimum residual factor analysis, fm="pa" does a principal axes factor analysis (`factor.pa`), fm="mle" uses the factanal function, and fm="pc" does a principal components analysis (`principal`).

For ability items, it is typically the case that all items will have positive loadings on the general factor. However, for non-cognitive items it is frequently the case that some items are to be scored positively, and some negatively. Although probably better to specify which directions the items are to be scored by specifying a key vector, if flip =TRUE (the default), items will be reversed so that they have positive loadings on the general factor. The keys are reported so that scores can be found using the `score.items` function. Arbitrarily reversing items this way can overestimate the general factor. (See the example with a simulated circumplex).

β, an alternative to ω, is defined as the worst split half reliability. It can be estimated by using `iclust` (Item Cluster analysis: a hierarchical clustering algorithm). For a very complimentary review of why the iclust algorithm is useful in scale construction, see Cooksey

and Soutar, 2005.

The `omega` function uses exploratory factor analysis to estimate the ω_h coefficient. It is important to remember that "A recommendation that should be heeded, regardless of the method chosen to estimate ω_h, is to always examine the pattern of the estimated general factor loadings prior to estimating ω_h. Such an examination constitutes an informal test of the assumption that there is a latent variable common to all of the scale's indicators that can be conducted even in the context of EFA. If the loadings were salient for only a relatively small subset of the indicators, this would suggest that there is no true general factor underlying the covariance matrix. Just such an informal assumption test would have afforded a great deal of protection against the possibility of misinterpreting the misleading ω_h estimates occasionally produced in the simulations reported here." (Zinbarg et al., 2006, p 137).

Although ω_h is uniquely defined only for cases where 3 or more subfactors are extracted, it is sometimes desired to have a two factor solution. By default this is done by forcing the `schmid` extraction to treat the two subfactors as having equal loadings.

There are three possible options for this condition: setting the general factor loadings between the two lower order factors to be "equal" which will be the $\sqrt{r_{ab}}$ where r_{ab} is the oblique correlation between the factors) or to "first" or "second" in which case the general factor is equated with either the first or second group factor. A message is issued suggesting that the model is not really well defined. This solution discussed in Zinbarg et al., 2007. To do this in omega, add the option="first" or option="second" to the call.

Although obviously not meaningful for a 1 factor solution, it is of course possible to find the sum of the loadings on the first (and only) factor, square them, and compare them to the overall matrix variance. This is done, with appropriate complaints.

In addition to ω_h, another of McDonald's coefficients is ω_t. This is an estimate of the total reliability of a test.

McDonald's ω_t, which is similar to Guttman's λ_6, (see `guttman`) uses the estimates of uniqueness u^2 from factor analysis to find e_j^2. This is based on a decomposition of the variance of a test score, V_x into four parts: that due to a general factor, $\vec{g}$, that due to a set of group factors, $\vec{f}$, (factors common to some but not all of the items), specific factors, $\vec{s}$ unique to each item, and $\vec{e}$, random error. (Because specific variance can not be distinguished from random error unless the test is given at least twice, some combine these both into error).

Letting $\vec{x} = \vec{cg} + \vec{Af} + \vec{Ds} + \vec{e}$ then the communality of item$_j$, based upon general as well as group factors, $h_j^2 = c_j^2 + \sum f_{ij}^2$ and the unique variance for the item $u_j^2 = \sigma_j^2(1 - h_j^2)$ may be used to estimate the test reliability. That is, if h_j^2 is the communality of item$_j$, based upon

general as well as group factors, then for standardized items, $e_j^2 = 1 - h_j^2$ and

$$\omega_t = \frac{\vec{1}\vec{cc'}\vec{1} + \vec{1}\vec{AA'}\vec{1}'}{V_x} = 1 - \frac{\sum(1-h_j^2)}{V_x} = 1 - \frac{\sum u^2}{V_x}$$

Because $h_j^2 \geq r_{smc}^2$, $\omega_t \geq \lambda_6$.

It is important to distinguish here between the two ω coefficients of McDonald, 1978 and Equation 6.20a of McDonald, 1999, ω_t and ω_h. While the former is based upon the sum of squared loadings on all the factors, the latter is based upon the sum of the squared loadings on the general factor.

$$\omega_h = \frac{\vec{1}\vec{cc'}\vec{1}}{V_x}$$

Another estimate reported is the omega for an infinite length test with a structure similar to the observed test. This is found by

$$\omega_{\inf} = \frac{\vec{1}\vec{cc'}\vec{1}}{\vec{1}\vec{cc'}\vec{1} + \vec{1}\vec{AA'}\vec{1}'}$$

In the case of these simulated 9 variables, the amount of variance attributable to a general factor (ω_h) is quite large, and the reliability of the set of 9 items is somewhat greater than that estimated by α or λ_6.

```
> om.9

9 simulated variables
Call: omega(m = r9, title = "9 simulated variables")
Alpha:                 0.75
G.6:                   0.74
Omega Hierarchical:    0.66
Omega H asymptotic:    0.84
Omega Total            0.78

Schmid Leiman Factor loadings greater than  0.2
      g   F1*   F2*   F3*   h2   u2   p2
V1 0.70                    0.53 0.47 0.93
V2 0.70                    0.52 0.48 0.94
V3 0.54                    0.32 0.68 0.91
V4 0.53  0.46              0.50 0.50 0.57
V5 0.44  0.44              0.39 0.61 0.50
V6 0.40  0.32              0.26 0.74 0.59
V7 0.31               0.31 0.21 0.79 0.48
```

```
> om.9 <- omega(r9, title = "9 simulated variables")
```

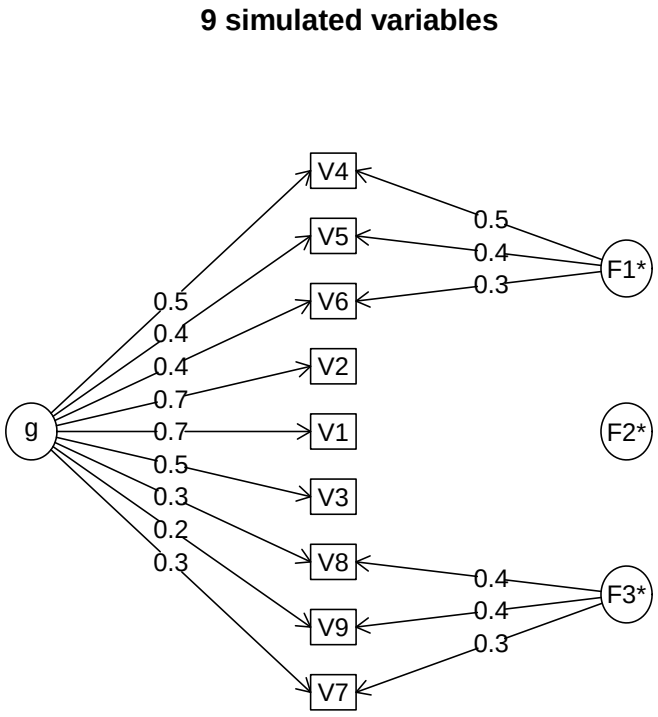

Figure 16: A bifactor solution for 9 simulated variables with a hierarchical structure.

```
V8 0.34              0.44 0.30 0.70 0.37
V9 0.24              0.36 0.19 0.81 0.32

With eigenvalues of:
   g  F1*  F2*  F3*
2.18 0.52 0.08 0.44

general/max  4.21   max/min =   6.17
mean percent general =  0.62    with sd =  0.24 and cv of  0.39

The degrees of freedom are 12  and the fit is  0.03
The number of observations was  500  with Chi Square =  14.23  with prob <  0.29
The root mean square of the residuals is  0.01
The df corrected root mean square of the residuals is  0.03
RMSEA index =  0.02  and the 90 % confidence intervals are  0.02 0.026
BIC =  -60.35

Compare this with the adequacy of just a general factor and no group factors
The degrees of freedom for just the general factor are 27  and the fit is  0.21
The number of observations was  500  with Chi Square =  103.64  with prob <  6.4e-11
The root mean square of the residuals is  0.05
The df corrected root mean square of the residuals is  0.08

RMSEA index =  0.076  and the 90 % confidence intervals are  0.075 0.078
BIC =  -64.15

Measures of factor score adequacy
                                                 g   F1*   F2*   F3*
Correlation of scores with factors            0.86  0.63  0.25  0.59
Multiple R square of scores with factors      0.74  0.39  0.06  0.35
Minimum correlation of factor score estimates 0.48 -0.21 -0.88 -0.30
```

4.4.3 Estimating ω_h using Confirmatory Factor Analysis

The `omegaSem` function will do an exploratory analysis and then take the highest loading items on each factor and do a confirmatory factor analysis using the *sem* package. These results can produce slightly different estimates of ω_h, primarily because cross loadings are modeled as part of the general factor.

```
> omegaSem(r9, n.obs = 500)
```

```
Call: omegaSem(m = r9, n.obs = 500)
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:                 0.75
G.6:                   0.74
Omega Hierarchical:    0.66
Omega H asymptotic:    0.84
Omega Total            0.78

Schmid Leiman Factor loadings greater than  0.2
      g   F1*   F2*   F3*   h2   u2   p2
V1 0.70                    0.53 0.47 0.93
V2 0.70                    0.52 0.48 0.94
V3 0.54                    0.32 0.68 0.91
V4 0.53  0.46              0.50 0.50 0.57
V5 0.44  0.44              0.39 0.61 0.50
V6 0.40  0.32              0.26 0.74 0.59
V7 0.31              0.31 0.21 0.79 0.48
V8 0.34              0.44 0.30 0.70 0.37
V9 0.24              0.36 0.19 0.81 0.32

With eigenvalues of:
   g  F1*  F2*  F3*
2.18 0.52 0.08 0.44

general/max  4.21   max/min =   6.17
mean percent general =  0.62    with sd =  0.24 and cv of  0.39

The degrees of freedom are 12  and the fit is  0.03
The number of observations was  500  with Chi Square =  14.23  with prob <  0.29
The root mean square of the residuals is  0.01
The df corrected root mean square of the residuals is  0.03
RMSEA index =  0.02  and the 90 % confidence intervals are  0.02 0.026
BIC =  -60.35

Compare this with the adequacy of just a general factor and no group factors
The degrees of freedom for just the general factor are 27  and the fit is  0.21
The number of observations was  500  with Chi Square =  103.64  with prob <  6.4e-11
The root mean square of the residuals is  0.05
```

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

RMSEA index =  0.076  and the 90 % confidence intervals are  0.075 0.078
BIC =  -64.15

Measures of factor score adequacy
                                                 g   F1*   F2*   F3*
Correlation of scores with factors            0.86  0.63  0.25  0.59
Multiple R square of scores with factors      0.74  0.39  0.06  0.35
Minimum correlation of factor score estimates 0.48 -0.21 -0.88 -0.30

 Omega Hierarchical from a confirmatory model using sem =  0.68
 Omega Total  from a confirmatory model using sem =  0.78
With loadings of
      g  F1*  F2*  F3*   h2   u2
V1 0.73                0.54 0.46
V2 0.68      0.29      0.54 0.46
V3 0.51      0.22      0.31 0.69
V4 0.54 0.47           0.51 0.49
V5 0.45 0.42           0.38 0.62
V6 0.39 0.31           0.25 0.75
V7 0.34           0.34 0.23 0.77
V8 0.36           0.39 0.28 0.72
V9 0.26           0.33 0.18 0.82

With eigenvalues of:
   g  F1*  F2*  F3*
2.21 0.49 0.14 0.38
```

4.4.4 Other estimates of reliability

Other estimates of reliability are found by the `guttman` function. These are described in more detail in Revelle and Zinbarg (2009). They include the 6 estimates from Guttman, four from TenBerge, and an estimate of the greatest lower bound.

```
> guttman(r9)

Call: guttman(r = r9)

Alternative estimates of reliability
Beta =  0.52  This is an estimate of the worst split half reliability
```

```
Guttman bounds
L1 =  0.66
L2 =  0.75
L3 (alpha) =  0.75
L4 (max) =  0.79
L5 =  0.74
L6 (smc) =  0.74
TenBerge bounds
mu0 =  0.75 mu1 =  0.75 mu2 =  0.76 mu3 =  0.76

alpha of first PC =  0.76
estimated greatest lower bound based upon communalities=  0.84
```

4.4.5 Reliability of multiple scales within an inventory

A typical research question in personality involves an inventory of multiple items purporting to measure multiple constructs. For example, the data set `bfi` includes 25 items thought to measure five dimensions of personality (Extraversion, Emotional Stability, Conscientiousness, Agreeableness, and Openness). The data may either be the raw data or a correlation matrix (`score.items`) or just a correlation matrix of the items (`cluster.cor` and `cluster.loadings`). When finding reliabilities for multiple scales, item reliabilities can be estimated using the squared multiple correlation of an item with all other items, not just those that are keyed for a particular scale. This leads to an estimate of G6*.

Scoring from raw data To score these five scales from the 25 items, use the `score.items` function with the helper function `make.keys`. Logically, scales are merely the weighted composites of a set of items. The weights used are -1, 0, and 1. 0 implies do not use that item in the scale, 1 implies a positive weight (add the item to the total score), -1 a negative weight (subtract the item from the total score, i.e., reverse score the item). Reverse scoring an item is equivalent to subtracting the item from the maximum + minimum possible value for that item. The minima and maxima can be estimated from all the items, or can be specified by the user.

There are two different ways that scale scores tend to be reported. Social psychologists and educational psychologists tend to report the scale score as the *average item score* while many personality psychologists tend to report the *total item score*. The default option for `score.items` is to report item averages (which thus allows interpretation in the same metric as the items) but totals can be found as well.

The printed output includes coefficients α and G6*, the average correlation of the items

within the scale (corrected for item ovelap and scale relliability), as well as the correlations between the scales (below the diagonal, the correlations above the diagonal are corrected for attenuation. As is the case for most of the *psych* functions, additional information is returned as part of the object.

First, create keys matrix using the `make.keys` function. (The keys matrix could also be prepared externally using a spreadsheet and then copying it into R). Although not normally necessary, show the keys to understand what is happening.

Note that the number of items to specify in the `make.keys` function is the total number of items in the inventory. That is, if scoring just 5 items from a 25 item inventory, `make.keys` should be told that there are 25 items. `make.keys` just changes a list of items on each scale to make up a scoring matrix. Because the `bfi` data set has 25 items as well as 3 demographic items, the number of variables is specified as 28.

```
> keys <- make.keys(nvars = 28, list(Agree = c(-1, 2:5), Conscientious = c(6:8,
+     -9, -10), Extraversion = c(-11, -12, 13:15), Neuroticism = c(16:20),
+     Openness = c(21, -22, 23, 24, -25)), item.labels = colnames(bfi))
> keys

          Agree Conscientious Extraversion Neuroticism Openness
A1           -1             0            0           0        0
A2            1             0            0           0        0
A3            1             0            0           0        0
A4            1             0            0           0        0
A5            1             0            0           0        0
C1            0             1            0           0        0
C2            0             1            0           0        0
C3            0             1            0           0        0
C4            0            -1            0           0        0
C5            0            -1            0           0        0
E1            0             0           -1           0        0
E2            0             0           -1           0        0
E3            0             0            1           0        0
E4            0             0            1           0        0
E5            0             0            1           0        0
N1            0             0            0           1        0
N2            0             0            0           1        0
N3            0             0            0           1        0
N4            0             0            0           1        0
N5            0             0            0           1        0
O1            0             0            0           0        1
O2            0             0            0           0       -1
O3            0             0            0           0        1
O4            0             0            0           0        1
O5            0             0            0           0       -1
gender        0             0            0           0        0
education     0             0            0           0        0
age           0             0            0           0        0
```

The use of multiple key matrices for different inventories is facilitated by using the `super.matrix` function to combine matrices. This allows convenient scoring of large data sets combining multiple inventories with keys based upon each individual inventory. Pre-

tend for the moment that the big 5 items were made up of two inventories, one consisting of the first 10 items, the second the last 18 items. (15 personality items + 3 demographic items.) Then the following code would work:

```
> keys.1 <- make.keys(10, list(Agree = c(-1, 2:5), Conscientious = c(6:8,
+     -9, -10)))
> keys.2 <- make.keys(15, list(Extraversion = c(-1, -2, 3:5), Neuroticism = c(6:10),
+     Openness = c(11, -12, 13, 14, -15)))
> keys.25 <- super.matrix(keys.1, keys.2)
```

The resulting keys matrix is identical to that found above except that it does not include the extra 3 demographic items. This is useful when scoring the raw items because the response frequencies for each category are reported, and for the demographic data,

This use of making multiple key matrices and then combining them into one super matrix of keys is particularly useful when combining demographic information with items to be scores. A set of demographic keys can be made and then these can be combined with the keys for the particular scales.

Now use these keys in combination with the raw data to score the items, calculate basic reliability and intercorrelations, and find the item-by scale correlations for each item and each scale. By default, missing data are replaced by the median for that variable.

```
> scores <- score.items(keys, bfi)
> scores

Call: score.items(keys = keys, items = bfi)

(Unstandardized) Alpha:
      Agree Conscientious Extraversion Neuroticism Openness
alpha   0.7          0.72         0.76        0.81      0.6

Average item correlation:
          Agree Conscientious Extraversion Neuroticism Openness
average.r  0.32          0.34         0.39        0.46     0.23

 Guttman 6* reliability:
         Agree Conscientious Extraversion Neuroticism Openness
Lambda.6   0.7          0.72         0.76        0.81      0.6

Scale intercorrelations corrected for attenuation
 raw correlations below the diagonal, alpha on the diagonal
 corrected correlations above the diagonal:
             Agree Conscientious Extraversion Neuroticism Openness
Agree          0.70          0.36         0.63      -0.245     0.23
```

```
Conscientious  0.26          0.72         0.35      -0.305     0.30
Extraversion   0.46          0.26         0.76      -0.284     0.32
Neuroticism   -0.18         -0.23        -0.22       0.812    -0.12
Openness       0.15          0.19         0.22      -0.086     0.60

Item by scale correlations:
 corrected for item overlap and scale reliability
          Agree Conscientious Extraversion Neuroticism Openness
A1        -0.40         -0.06        -0.11        0.14    -0.14
A2         0.67          0.23         0.40       -0.07     0.17
A3         0.70          0.22         0.48       -0.11     0.17
A4         0.49          0.29         0.30       -0.14     0.01
A5         0.62          0.23         0.55       -0.23     0.18
C1         0.13          0.53         0.19       -0.08     0.28
C2         0.21          0.61         0.17        0.00     0.20
C3         0.21          0.54         0.14       -0.09     0.08
C4        -0.24         -0.66        -0.23        0.31    -0.23
C5        -0.26         -0.59        -0.29        0.36    -0.10
E1        -0.30         -0.06        -0.59        0.11    -0.16
E2        -0.39         -0.25        -0.70        0.34    -0.15
E3         0.44          0.20         0.60       -0.10     0.37
E4         0.51          0.23         0.68       -0.22     0.04
E5         0.34          0.40         0.55       -0.10     0.31
N1        -0.22         -0.21        -0.11        0.76    -0.12
N2        -0.22         -0.19        -0.12        0.74    -0.06
N3        -0.14         -0.20        -0.14        0.74    -0.03
N4        -0.22         -0.30        -0.39        0.62    -0.02
N5        -0.04         -0.14        -0.19        0.55    -0.18
O1         0.16          0.20         0.31       -0.09     0.52
O2        -0.01         -0.18        -0.07        0.19    -0.45
O3         0.26          0.20         0.42       -0.07     0.61
O4         0.06         -0.02        -0.10        0.21     0.32
O5        -0.09         -0.14        -0.11        0.11    -0.53
gender     0.25          0.11         0.12        0.14    -0.07
education  0.06          0.03         0.01       -0.06     0.13
age        0.22          0.14         0.07       -0.13     0.10
```

To see the additional information (the raw correlations, the individual scores, etc.), they may be specified by name. Then, to visualize the correlations between the raw scores, use the `pairs.panels` function on the scores values of scores.

```
> png("scores.png")
> pairs.panels(scores$scores)
> dev.off()
pdf
  2
```

Figure 17: A graphic analysis of the Big Five scales found by using the score.items function.

Forming scales from a correlation matrix There are some situations when the raw data are not available, but the correlation matrix between the items is available. In this case, it is not possible to find individual scores, but it is possible to find the reliability and intercorrelations of the scales. This may be done using the `cluster.cor` function or the `score.items` function. The use of a keys matrix is the same as in the raw data case.

Consider the same `bfi` data set, but first find the correlations, and then use `cluster.cor`.

```
> r.bfi <- cor(bfi, use = "pairwise")
> scales <- cluster.cor(keys, r.bfi)
> summary(scales)

Call: cluster.cor(keys = keys, r.mat = r.bfi)

Scale intercorrelations corrected for attenuation
 raw correlations below the diagonal, (standardized) alpha on the diagonal
 corrected correlations above the diagonal:
             Agree Conscientious Extraversion Neuroticism Openness
Agree         0.71          0.35         0.64       -0.24     0.25
Conscientious 0.25          0.73         0.36       -0.29     0.30
Extraversion  0.47          0.27         0.76       -0.28     0.35
Neuroticism  -0.18         -0.22        -0.22        0.81    -0.11
Openness      0.16          0.20         0.24       -0.07     0.61
```

To find the correlations of the items with each of the scales (the "structure" matrix) or the correlations of the items controlling for the other scales (the "pattern" matrix), use the `cluster.loadings` function. To do both at once (e.g., the correlations of the scales as well as the item by scale correlations, it is also possible to just use `score.items`.

4.5 Item analysis

Basic item analysis starts with describing the data (`describe`, finding the number of dimensions using factor analysis (`fa`), perhaps using the Very Simple Structure criterion (`vss`),

or perhaps parallel analysis `fa.parallel`. Item whole correlations may then be found for scales scored on one dimension (`alpha` or many scales simultaneously (`score.items`). Scales can be modified by changing the keys matrix (i.e., dropping particular items, changing the scale on which an item is to be scored).

5 Item Response Theory analysis

The use of Item Response Theory has become is said to be the "new psychometrics". The emphasis is upon item properties, particularly those of item difficulty or location and item discrimination. These two parameters are easily found from classic techniques when using factor analyses of correlation matrices formed by `polychoric` or `tetrachoric` correlations. The `irt.fa` function does this and then graphically displays item discrimination and item location as well as item and test information (see Figure 18).

If the correlations of all of the items reflect one underlying latent variable, then factor analysis of the matrix of tetrachoric correlations should allow for the identification of the regression slopes (α) of the items on the latent variable. These regressions are, of course just the factor loadings. Item difficulty, δ_j and item discrimination, α_j may be found from factor analysis of the tetrachoric correlations where λ_j is just the factor loading on the first factor and τ_j is the normal threshold reported by the `tetrachoric` function.

$$\delta_j = \frac{D\tau}{\sqrt{1-\lambda_j^2}}, \qquad \alpha_j = \frac{\lambda_j}{\sqrt{1-\lambda_j^2}} \tag{2}$$

where D is a scaling factor used when converting to the parameterization of *logistic* model and is 1.702 in that case and 1 in the case of the normal ogive model. Thus, in the case of the normal model, factor loadings (λ_j) and item medians (τ) are just

$$\lambda_j = \frac{\alpha_j}{\sqrt{1+\alpha_j^2}}, \qquad \tau_j = \frac{\delta_j}{\sqrt{1+\alpha_j^2}}.$$

Consider 9 dichotomous items representing one factor but differing in their levels of difficulty

```
> set.seed(17)
> d9 <- sim.irt(9, 1000, -2.5, 2.5, mod = "normal")
> test <- irt.fa(d9$items)
> test

Item Response Analysis using Factor Analysis =
Call: irt.fa(x = d9$items)
```

```
Item discrimination and location for factor  MR1
   discrimination location
V1           1.03    -2.45
V2           1.00    -1.84
V3           1.04    -1.22
V4           1.03    -0.69
V5           1.18    -0.03
V6           1.05     0.63
V7           1.10     1.43
V8           1.01     1.85
V9           0.90     2.31
```

Similar analyses can be done for polytomous items such as those of the bfi extraversion scale:

```
> data(bfi)
> e.irt <- irt.fa(bfi[11:15])
> e.irt

Item Response Analysis using Factor Analysis =
Call: irt.fa(x = bfi[11:15])

Item discrimination and location for factor  MR1
   discrimination location.1 location.2 location.3 location.4 location.5
E1          -0.84      -0.93      -0.09       0.40       1.01       1.78
E2          -1.20      -1.37      -0.27       0.22       1.16       2.09
E3           0.75      -2.01      -1.25      -0.63       0.33       1.43
E4           1.09      -2.43      -1.57      -1.04      -0.37       0.95
E5           0.65      -2.17      -1.44      -0.93      -0.18       0.91
```

The item information functions show that not all of items are equally good (Figure 19):

These procedures can be generalized to more than one factor by specifying the number of factors in `irt.fa`. The plots can be limited to those items with discriminations greater than some value of cut.

More extensive IRT packages include the *ltm* and *eRm* and should be used for serious Item Response Theory analysis.

6 Multiple Regression from the correlation matrix

Although the standard multiple regression (using the `lm` function in base R) uses the raw data, it is sometimes useful to do multiple regression from a correlation matrix. This may be done using the `mat.regress` function.

Consider the correlations of the 6 variables in the `sat.act` data set. First do the normal

```
> op <- par(mfrow = c(3, 1))
> plot(test, type = "ICC")
> plot(test, type = "IIC")
> plot(test, type = "test")
> op <- par(mfrow = c(1, 1))
```

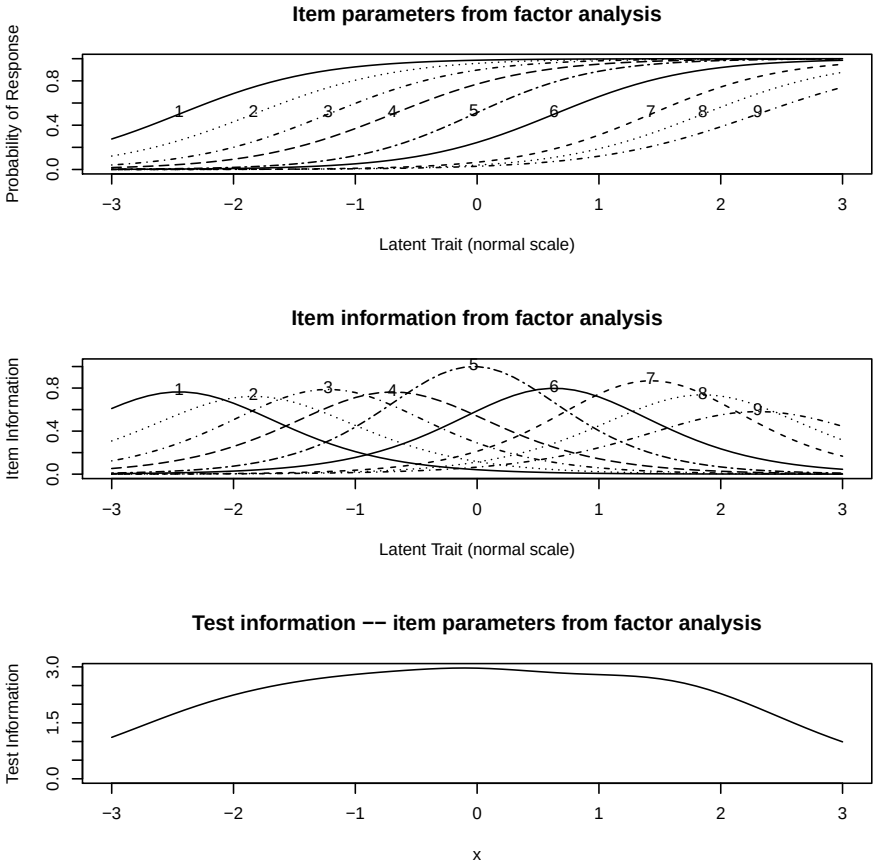

Figure 18: A graphic analysis of 9 dichotomous (simulated) items. The top panel shows the probability of item endorsement as the value of the latent trait increases. Items differ in their location (difficulty) and discrimination (slope). The middle panel shows the information in each item as a function of latent trait level. An item is most informative when the probability of endorsement is 50%. The lower panel shows the total test information. These items form a test that is most informative (most accurate) at the middle range of the latent trait.

```
> plot(e.irt, type = "IIC")
```

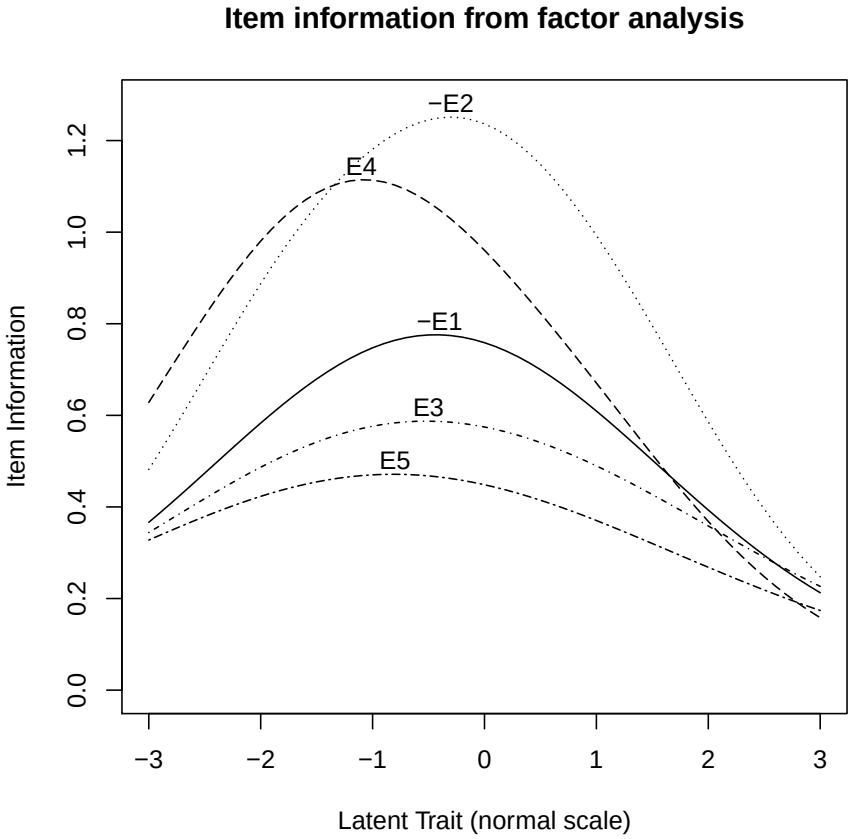

Figure 19: A graphic analysis of 5 extraversion items from the bfi. The curves represent the amount of information in the item as a function of the latent score for an individual. That is, each item is maximally discriminating at a different part of the latent continuum.

multiple regression, and then compare it with the results using `mat.regress`. Two things to notice. `mat.regress` works on the *correlation* or *covariance* matrix, and thus if using the correlation matrix, will report standardized β weights. Secondly, it is possible to do several multiple regressions simultaneously. If the number of observations is specified, statistical tests of significance are applied.

```
> data(sat.act)
> C <- cov(sat.act, use = "pairwise")
> model1 <- lm(ACT ~ gender + education + age, data = sat.act)
> summary(model1)

Call:
lm(formula = ACT ~ gender + education + age, data = sat.act)

Residuals:
     Min       1Q   Median       3Q      Max
-25.2458  -3.2133   0.7769   3.5921   9.2630

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 27.41706    0.82140  33.378  < 2e-16 ***
gender      -0.48606    0.37984  -1.280  0.20110
education    0.47890    0.15235   3.143  0.00174 **
age          0.01623    0.02278   0.712  0.47650
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.768 on 696 degrees of freedom
Multiple R-squared: 0.0272,        Adjusted R-squared: 0.02301
F-statistic: 6.487 on 3 and 696 DF,  p-value: 0.0002476
```

Compare this with the output from `mat.regress`.

```
> mat.regress(C, c(1:3), c(4:6), n.obs = 700)

Call: mat.regress(m = C, x = c(1:3), y = c(4:6), n.obs = 700)

Multiple Regression from matrix input

Beta weights
            ACT  SATV   SATQ
gender    -0.49 -7.08 -42.67
education  0.48  8.23   8.19
```

```
age        0.02 -1.19  -1.13

Multiple R
 ACT SATV SATQ
0.16 0.10 0.19

Multiple R2
   ACT   SATV   SATQ
0.0272 0.0096 0.0359

 SE of Beta weights
           ACT SATV SATQ
gender    0.38 8.97 9.07
education 0.15 3.60 3.64
age       0.02 0.54 0.54

 t of Beta Weights
            ACT  SATV  SATQ
gender    -1.28 -0.79 -4.71
education  3.14  2.29  2.25
age        0.71 -2.21 -2.08

Probability of t <
             ACT  SATV    SATQ
gender    0.2000 0.430 3.0e-06
education 0.0017 0.022 2.5e-02
age       0.4800 0.028 3.8e-02

 Shrunken R2
   ACT   SATV   SATQ
0.0230 0.0054 0.0317

Standard Error of R2
   ACT   SATV   SATQ
0.0120 0.0073 0.0137

F
 ACT SATV SATQ
6.49 2.26 8.63

Probability of F <
```

```
     ACT     SATV     SATQ
2.48e-04 8.08e-02 1.24e-05

 degrees of freedom of regression
[1]   3 696
```

7 Simulation functions

It is particularly helpful, when teaching psychometric concepts, to be able to generate sample data sets that meet certain specifications. By knowing "truth" it is possible to see how well various algorithms can capture it. Several of the `sim` functions create artificial data sets with known structures.

A number of functions in the psych package will generate simulated data. These functions include `sim` for a factor simplex, and `sim.simplex` for a data simplex, `sim.circ` for a circumplex structure, `sim.congeneric` for a one factor factor congeneric model, `sim.dichot` to simulate dichotomous items, `sim.hierarchical` to create a hierarchical factor model, `sim.item` is a more general item simulation, `sim.minor` to simulate major and minor factors, `sim.omega` to test various examples of omega, `sim.parallel` to compare the efficiency of various ways of determining the number of factors, `sim.rasch` to create simulated rasch data, `sim.irt`to create general 1 to 4 parameter IRT data by calling `sim.npl` 1 to 4 parameter logistic IRT or `sim.npn` 1 to 4 paramater normal IRT, `sim.structural` a general simulation of structural models, and `sim.anova` for ANOVA and lm simulations, and `sim.vss`. Some of these functions are separately documented and are listed here for ease of the help function. See each function for more detailed help.

`sim` The default version is to generate a four factor simplex structure over three occasions, although more general models are possible.

`sim.simple` Create major and minor factors. The default is for 12 variables with 3 major factors and 6 minor factors.

`sim.structure` To combine a measurement and structural model into one data matrix. Useful for understanding structural equation models.

`sim.hierarchical` To create data with a hierarchical (bifactor) structure.

`sim.congeneric` To create congeneric items/tests for demonstrating classical test theory. This is just a special case of sim.structure.

`sim.circ` To create data with a circumplex structure.

`sim.item` To create items that either have a simple structure or a circumplex structure.

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

`sim.rasch` Simulate a 1 parameter logistic (Rasch) model.

`sim.irt` Simulate a 2 parameter logistic (2PL) or 2 parameter Normal model. Will also do 3 and 4 PL and PN models.

Some of these functions are described in more detail in the companion vignette: psych for sem.

The default values for `sim.structure` is to generate a 4 factor, 12 variable data set with a simplex structure between the factors.

Two data structures that are particular challenges to exploratory factor analysis are the simplex structure and the presence of minor factors. Simplex structures `sim.simplex` will typically occur in developmental or learning contexts and have a correlation structure of r between adjacent variables and r^n for variables n apart. Although just one latent variable (r) needs to be estimated, the structure will have nvar-1 factors.

Many simulations of factor structures assume that except for the major factors, all residuals are normally distributed around 0. An alternative, and perhaps more realistic situation, is that the there are a few major (big) factors and many minor (small) factors. The challenge is thus to identify the major factors. `sim.minor` generates such structures. The structures generated can be thought of as having a a major factor structure with some small correlated residuals.

Although coefficient ω_h is a very useful indicator of the general factor saturation of a unifactorial test (one with perhaps several sub factors), it has problems with the case of multiple, independent factors. In this situation, one of the factors is labelled as "general" and the omega estimate is too large. This situation may be explored using the `sim.omega` function.

The four irt simulations, `sim.rasch`, `sim.irt`, `sim.npl` and `sim.npn`, simulate dichotomous items following the Item Response model. `sim.irt` just calls either `sim.npl` (for logistic models) or `sim.npn` (for normal models) depending upon the specification of the model.

The logistic model is

$$P(x|\theta_i,\delta_j,\gamma_j,\zeta_j) = \gamma_j + \frac{\zeta_j - \gamma_j}{1+e^{\alpha_j(\delta_j-\theta_i}}. \tag{3}$$

where γ is the lower asymptote or guessing parameter, ζ is the upper asymptote (normally 1), α_j is item discrimination and δ_j is item difficulty. For the 1 Paramater Logistic (Rasch) model, gamma=0, zeta=1, alpha=1 and item difficulty is the only free parameter to specify.

(Graphics of these may be seen in the demonstrations for the logistic function.)

The normal model (`irt.npn` calculates the probability using `pnorm` instead of the logistic function used in `irt.npl`, but the meaning of the parameters are otherwise the same. With the a = α parameter = 1.702 in the logiistic model the two models are practically identical.

8 Graphical Displays

Many of the functions in the *psych* package include graphic output and examples have been shown in the previous figures. After running `fa`, `iclust`, `omega`, `irt.fa`, plotting the resulting object is done by the `plot.psych` function as well as specific diagram functions. e.g.,

```
f3 <- fa(Thurstone,3)
plot(f3)
fa.diagram(f3)
c <- iclust(Thurstone)
plot(c)  #a pretty boring plot
iclust.diagram(c)  #a better diagram
c3 <- iclust(Thurstone,3)
plot(c3)  #a more interesting plot
data(bfi)
e.irt <- irt.fa(bfi[11:15])
plot(e.irt)
ot <- omega(Thurstone)
plot(ot)
omega.diagram(ot)
```

The ability to show path diagrams to represent factor analytic and structural models is discussed in somewhat more detail in the accompanying vignette, psych for sem. Basic routines to draw path diagrams are included in the `dia.rect` and accompanying functions. These are used by the `fa.diagram`, `structure.diagram` and `iclust.diagram` functions.

9 Data sets

A number of data sets for demonstrating psychometric techniques are included in the *psych* package. These include six data sets showing a hierarchical factor structure (five cognitive examples, `Thurstone`, `Thurstone.33`, `Holzinger`, `Bechtoldt.1`, `Bechtoldt.2`, and one from health psychology `Reise`). One of these (`Thurstone`) is used as an example

```
> xlim = c(0, 10)
> ylim = c(0, 10)
> plot(NA, xlim = xlim, ylim = ylim, main = "Demontration of dia functions",
+     axes = FALSE, xlab = "", ylab = "")
> ul <- dia.rect(1, 9, labels = "upper left", xlim = xlim, ylim = ylim)
> ll <- dia.rect(1, 3, labels = "lower left", xlim = xlim, ylim = ylim)
> lr <- dia.ellipse(9, 3, "lower right", xlim = xlim, ylim = ylim)
> ur <- dia.ellipse(9, 9, "upper right", xlim = xlim, ylim = ylim)
> ml <- dia.ellipse(3, 6, "middle left", xlim = xlim, ylim = ylim)
> mr <- dia.ellipse(7, 6, "middle right", xlim = xlim, ylim = ylim)
> bl <- dia.ellipse(1, 1, "bottom left", xlim = xlim, ylim = ylim)
> br <- dia.rect(9, 1, "bottom right", xlim = xlim, ylim = ylim)
> dia.arrow(from = lr, to = ul, labels = "right to left")
> dia.arrow(from = ul, to = ur, labels = "left to right")
> dia.curved.arrow(from = lr, to = ll$right, labels = "right to left")
> dia.curved.arrow(to = ur, from = ul$right, labels = "left to right")
> dia.curve(ll$top, ul$bottom, "double")
> dia.curved.arrow(mr, ur, "up")
> dia.curve(ml, mr, "across")
> dia.arrow(ur, lr, "top down")
> dia.curved.arrow(br$top, lr$bottom, "up")
> dia.curved.arrow(bl, br, "left to right")
> dia.arrow(bl, ll$bottom)
> dia.curved.arrow(ml, ll$right)
> dia.curved.arrow(mr, lr$top)
```

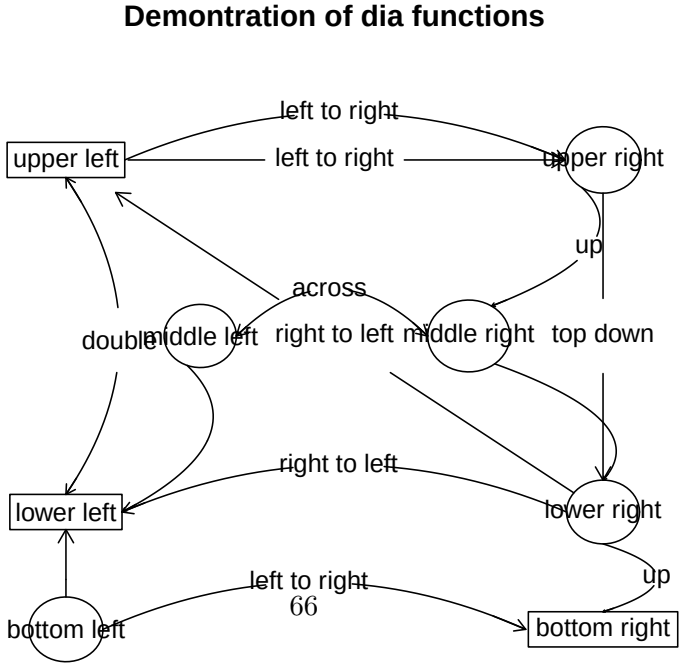

Figure 20: The drawing capabilities of the dia functions are shown in this figure.

in the *sem* package as well as McDonald (1999). The original data are from Thurstone and Thurstone (1941) and reanalyzed by Bechtoldt (1961). Personality item data representing five personality factors on 25 items (`bfi`) or 13 personality inventory scores (`epi.bfi`), and 14 multiple choice iq items (`iqitems`). The `vegetables` example has paired comparison preferences for 9 vegetables. This is an example of Thurstonian scaling used by Guilford (1954) and Nunnally (1967). Other data sets include `cubits`, `peas`, and `heights` from Galton.

Thurstone Holzinger-Swineford (1937) introduced the bifactor model of a general factor and uncorrelated group factors. The Holzinger correlation matrix is a 14 * 14 matrix from their paper. The Thurstone correlation matrix is a 9 * 9 matrix of correlations of ability items. The Reise data set is 16 * 16 correlation matrix of mental health items. The Bechtholdt data sets are both 17 x 17 correlation matrices of ability tests.

bfi 25 personality self report items taken from the International Personality Item Pool (ipip.ori.org) were included as part of the Synthetic Aperture Personality Assessment (SAPA) web based personality assessment project. The data from 2800 subjects are included here as a demonstration set for scale construction, factor analysis and Item Response Theory analyses.

sat.act Self reported scores on the SAT Verbal, SAT Quantitative and ACT were collected as part of the Synthetic Aperture Personality Assessment (SAPA) web based personality assessment project. Age, gender, and education are also reported. The data from 700 subjects are included here as a demonstration set for correlation and analysis.

epi.bfi A small data set of 5 scales from the Eysenck Personality Inventory, 5 from a Big 5 inventory, a Beck Depression Inventory, and State and Trait Anxiety measures. Used for demonstrations of correlations, regressions, graphic displays.

iq 14 multiple choice ability items were included as part of the Synthetic Aperture Personality Assessment (SAPA) web based personality assessment project. The data from 1000 subjects are included here as a demonstration set for scoring multiple choice inventories and doing basic item statistics.

galton Two of the earliest examples of the correlation coefficient were Francis Galton's data sets on the relationship between mid parent and child height and the similarity of parent generation peas with child peas. `galton` is the data set for the Galton height. `peas` is the data set Francis Galton used to ntroduce the correlation coefficient with an analysis of the similarities of the parent and child generation of 700 sweet peas.

Dwyer Dwyer (1937) introduced a method for *factor extension* (see `fa.extension` that finds loadings on factors from an original data set for additional (extended) variables. This data set includes his example.

miscellaneous `cities` is a matrix of airline distances between 11 US cities and may be used for demonstrating multiple dimensional scaling. `vegetables` is a classic data set for demonstrating Thurstonian scaling and is the preference matrix of 9 vegetables from Guilford (1954). Used by Guilford (1954); Nunnally (1967); Nunnally and Bernstein (1984), this data set allows for examples of basic scaling techniques.

10 Development version and a users guide

The most recent development version is available as a source file at the repository maintained at http://personality-project.org/r. That version will have removed the most recently discovered bugs (but perhaps introduced other, yet to be discovered ones). To download that version, go to the repository http://personality-project.org/r/src/contrib/ and wander around. For a Mac, this version can be installed directly using the "other repository" option in the package installer. For a PC, the zip file for the most recent release has been created using the win-builder facility at CRAN.

Although the individual help pages for the *psych* package are available as part of R and may be accessed directly (e.g. ?psych) , the full manual for the `psych` package is also available as a pdf at http://personality-project.org/r/psych_manual.pdf

News and a history of changes are available in the NEWS and CHANGES files in the source files.

11 Psychometric Theory

The *psych* package has been developed to help psychologists do basic research. Many of the functions were developed to supplement a book (http://personality-project.org/r/book An introduction to Psychometric Theory with Applications in R (Revelle, in prep). More information about the use of some of the functions may be found in the book .

For more extensive discussion of the use of psych in particular and R in general, consult http://personality-project.org/r/r.guide.html A short guide to R.

12 SessionInfo

This document was prepared using the following settings.

```
> sessionInfo()
```

```
R version 2.13.0 beta (2011-03-29 r55167)
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

locale:
[1] C

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

other attached packages:
[1] sem_0.9-21            MASS_7.3-12           GPArotation_2010.07-1
[4] mvtnorm_0.9-96        psych_1.0-95

loaded via a namespace (and not attached):
[1] tools_2.13.0
```

References

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

Blashfield, R. K. (1980). The growth of cluster analysis: Tryon, Ward, and Johnson. *Multivariate Behavioral Research*, 15(4):439 – 458.

Blashfield, R. K. and Aldenderfer, M. S. (1988). The methods and problems of cluster analysis. In Nesselroade, J. R. and Cattell, R. B., editors, *Handbook of multivariate experimental psychology (2nd ed.)*, pages 447–473. Plenum Press, New York, NY.

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, 1(2):245–276.

Cattell, R. B. (1978). *The scientific use of factor analysis*. Plenum Press, New York.

Cooksey, R. and Soutar, G. (2006). Coefficient beta and hierarchical item clustering - an analytical procedure for establishing and displaying the dimensionality and homogeneity of summated scales. *Organizational Research Methods*, 9:78–98.

Cronbach, L. J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, 16:297–334.

Dwyer, P. S. (1937). The determination of the factor loadings of a given test from the known factor loadings of other tests. *Psychometrika*, 2(3):173–178.

Everitt, B. (1974). *Cluster analysis*. John Wiley & Sons, Cluster analysis. 122 pp. Oxford, England.

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

Grice, J. W. (2001). Computing and evaluating factor scores. *Psychological Methods*, 6(4):430–450.

Guilford, J. P. (1954). *Psychometric Methods*. McGraw-Hill, New York, 2nd edition.

Guttman, L. (1945). A basis for analyzing test-retest reliability. *Psychometrika*, 10(4):255–282.

Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA.

Henry, D. B., Tolan, P. H., and Gorman-Smith, D. (2005). Cluster analysis in family psychology research. *Journal of Family Psychology*, 19(1):121–132.

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

Horn, J. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–185.

Horn, J. L. and Engstrom, R. (1979). Cattell's scree test in relation to bartlett's chi-square test and other observations on the number of factors problem. *Multivariate Behavioral Research*, 14(3):283–300.

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

Loevinger, J., Gleser, G., and DuBois, P. (1953). Maximizing the discriminating power of a multiple-score test. *Psychometrika*, 18(4):309–317.

MacCallum, R. C., Browne, M. W., and Cai, L. (2007). Factor analysis models as approximations. In Cudeck, R. and MacCallum, R. C., editors, *Factor analysis at 100: Historical developments and future directions*, pages 153–175. Lawrence Erlbaum Associates Publishers, Mahwah, NJ.

Martinent, G. and Ferrand, C. (2007). A cluster analysis of precompetitive anxiety: Relationship with perfectionism and trait anxiety. *Personality and Individual Differences*, 43(7):1676–1686.

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

Mun, E. Y., von Eye, A., Bates, M. E., and Vaschillo, E. G. (2008). Finding groups using model-based cluster analysis: Heterogeneous emotional self-regulatory processes and heavy alcohol use risk. *Developmental Psychology*, 44(2):481–495.

Nunnally, J. C. (1967). *Psychometric theory*. McGraw-Hill, New York,.

Nunnally, J. C. and Bernstein, I. H. (1984). *Psychometric theory*. McGraw-Hill, New York,, 3rd edition.

Revelle, W. (1979). Hierarchical cluster-analysis and the internal structure of tests. *Multivariate Behavioral Research*, 14(1):57–74.

Revelle, W. (2011). *psych: Procedures for Personality and Psychological Research*. Northwestern University, Evanston, 1.0-95 edition. R package version 1.0-95.

Revelle, W. and Rocklin, T. (1979). Very Simple Structure - alternative procedure for estimating the optimal number of interpretable factors. *Multivariate Behavioral Research*, 14(4):403–414.

Revelle, W. and Zinbarg, R. E. (2009). Coefficients alpha, beta, omega and the glb: comments on Sijtsma. *Psychometrika*, 74(1):145–154.

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

Shrout, P. E. and Fleiss, J. L. (1979). Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin*, 86(2):420–428.

Sneath, P. H. A. and Sokal, R. R. (1973). *Numerical taxonomy: the principles and practice of numerical classification*. A Series of books in biology. W. H. Freeman, San Francisco.

Sokal, R. R. and Sneath, P. H. A. (1963). *Principles of numerical taxonomy*. A Series of books in biology. W. H. Freeman, San Francisco.

Steiger, J. H. (1980). Tests for comparing elements of a correlation matrix. *Psychological Bulletin*, 87(2):245–251.

Thorburn, W. M. (1918). The myth of occam's razor. *Mind*, 27:345–353.

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

Tryon, R. C. (1935). A theory of psychological components–an alternative to "mathematical factors.". *Psychological Review*, 42(5):425–454.

Tryon, R. C. (1939). *Cluster analysis*. Edwards Brothers, Ann Arbor, Michigan.

Velicer, W. (1976). Determining the number of components from the matrix of partial correlations. *Psychometrika*, 41(3):321–327.

Index

affect, 8
agnes, 29
alpha, 3, 40–42, 44, 57
attitude, 7

Bechtoldt.1, 65
Bechtoldt.2, 65
bfi, 12, 52, 53, 56, 67
bi.bars, 9, 12
biserial, 17

cities, 68
cluster analysis, 19, 29
cluster scores, 19
cluster.cor, 40, 52, 56
cluster.loadings, 52, 56
common variance, 24
component scores, 19
congruence coefficient, 32
cor.test, 15
corr.test, 15
cortest, 16, 17
ctv, 4
cubits, 67

describe, 2, 5, 56
describe.by, 5
dia.rect, 65
diagram, 4
draw.tetra, 17
dynamite plot, 9

epi.bfi, 67
eRm, 58
error bars, 9
error.bars, 2, 9
error.bars.bi, 8
error.bars.by, 6, 9, 10
error.crosses, 9

FA, 19
fa, 2, 3, 19, 21, 24, 28, 56, 65
fa.diagram, 3, 20, 27, 65
fa.extension, 67
fa.graph, 20
fa.parallel, 2, 20, 35, 38, 57
factanal, 20, 21
factor analysis, 2, 19
factor scores, 19
factor.congruence, 20, 32
factor.minres, 3, 19, 21
factor.pa, 3, 20, 21, 45
factor.plot, 20, 24
factor.wls, 3, 20
factors, 19

galton, 67
generalized least squares, 2
generalized least squares factor analysis, 19
GPArotation, 3, 23
guttman, 3, 46, 51

hclust, 29
heights, 67
hierarchical cluster analysis, 29
Hmisc, 15
Holzinger, 65

ICC, 3
iclust, 3, 20, 29, 32, 33, 40, 45, 65
iclust.diagram, 3, 65
introduction to psychometric theory with applications in R, 3
iqitems, 67
irt.fa, 2, 57, 58, 65
irt.npl, 65
irt.npn, 65

keys, 42

library, 4
lm, 58
logistic, 57
lowess, 8
ltm, 58

make.keys, 52, 53
MAP, 2, 20, 36
mat.regress, 58, 61
maximum likelihood, 2
maximum likelihood factor analysis, 19
min.res, 21
minimum residual, 2
minres factor analysis, 19
mixed.cor, 17

oblimin, 23
omega, 3, 29, 40, 45, 46, 65
omegaSem, 45, 49

pairs, 8
pairs.panels, 2, 3, 8, 9, 55
paran, 38
PCA, 19
peas, 67
phi2poly, 3
plot, 24, 26
plot.irt, 3
plot.psych, 65
pnorm, 65
poly.mat, 3
polychoric, 2, 3, 17, 32, 57
polycor, 3
polyserial, 17
principal, 3, 20, 21, 24, 45
principal axis, 2
principal axis factor analysis, 19
principal components, 19
principal components analysis, 19
princomp, 24
print, 24
print.psych, 25

Promax, 23, 24
psych, 2–4, 15, 17, 19, 21, 53, 65, 68

quartimax, 23
quartimin, 23

R function
  affect, 8
  agnes, 29
  alpha, 3, 40–42, 44, 57
  attitude, 7
  Bechtoldt.1, 65
  Bechtoldt.2, 65
  bfi, 12, 52, 53, 56, 67
  bi.bars, 9, 12
  biserial, 17
  cities, 68
  cluster.cor, 40, 52, 56
  cluster.loadings, 52, 56
  cor.test, 15
  corr.test, 15
  cortest, 16, 17
  cubits, 67
  describe, 2, 5, 56
  describe.by, 5
  dia.rect, 65
  draw.tetra, 17
  epi.bfi, 67
  error.bars, 2, 9
  error.bars.bi, 8
  error.bars.by, 6, 9, 10
  error.crosses, 9
  fa, 2, 3, 19, 21, 24, 28, 56, 65
  fa.diagram, 3, 20, 27, 65
  fa.extension, 67
  fa.graph, 20
  fa.parallel, 2, 20, 35, 38, 57
  factanal, 20, 21
  factor.congruence, 20, 32
  factor.minres, 3, 19, 21
  factor.pa, 3, 20, 21, 45

factor.plot, 20, 24
factor.wls, 3, 20
galton, 67
guttman, 3, 46, 51
hclust, 29
heights, 67
Holzinger, 65
ICC, 3
iclust, 3, 20, 29, 32, 33, 40, 45, 65
iclust.diagram, 3, 65
iqitems, 67
irt.fa, 2, 57, 58, 65
irt.npl, 65
irt.npn, 65
library, 4
lm, 58
make.keys, 52, 53
MAP, 2, 20, 36
mat.regress, 58, 61
mixed.cor, 17
oblimin, 23
omega, 3, 29, 40, 45, 46, 65
omegaSem, 45, 49
pairs, 8
pairs.panels, 2, 3, 8, 9, 55
paran, 38
peas, 67
phi2poly, 3
plot, 24, 26
plot.irt, 3
plot.psych, 65
pnorm, 65
poly.mat, 3
polychoric, 2, 3, 17, 32, 57
polyserial, 17
principal, 3, 20, 21, 24, 45
princomp, 24
print, 24
print.psych, 25
Promax, 23, 24
psych, 68
psych package
  affect, 8
  alpha, 3, 40–42, 44, 57
  Bechtoldt.1, 65
  Bechtoldt.2, 65
  bfi, 12, 52, 53, 56, 67
  bi.bars, 9, 12
  biserial, 17
  cities, 68
  cluster.cor, 40, 52, 56
  cluster.loadings, 52, 56
  corr.test, 15
  cortest, 16, 17
  cubits, 67
  describe, 2, 5, 56
  describe.by, 5
  dia.rect, 65
  draw.tetra, 17
  epi.bfi, 67
  error.bars, 2, 9
  error.bars.bi, 8
  error.bars.by, 6, 9, 10
  error.crosses, 9
  fa, 2, 3, 19, 21, 24, 28, 56, 65
  fa.diagram, 3, 20, 27, 65
  fa.extension, 67
  fa.graph, 20
  fa.parallel, 2, 20, 35, 38, 57
  factor.congruence, 20, 32
  factor.minres, 3, 19, 21
  factor.pa, 3, 20, 21, 45
  factor.plot, 20, 24
  factor.wls, 3, 20
  galton, 67
  guttman, 3, 46, 51
  heights, 67
  Holzinger, 65
  ICC, 3
  iclust, 3, 20, 29, 32, 33, 40, 45, 65
  iclust.diagram, 3, 65
  iqitems, 67

irt.fa, 2, 57, 58, 65
irt.npl, 65
irt.npn, 65
make.keys, 52, 53
MAP, 2, 20, 36
mat.regress, 58, 61
mixed.cor, 17
omega, 3, 29, 40, 45, 46, 65
omegaSem, 45, 49
pairs.panels, 2, 3, 8, 9, 55
peas, 67
phi2poly, 3
plot, 26
plot.irt, 3
plot.psych, 65
poly.mat, 3
polychoric, 2, 3, 17, 32, 57
polyserial, 17
principal, 3, 20, 21, 24, 45
print, 24
print.psych, 25
Promax, 23, 24
psych, 68
r.test, 15
read.clipboard, 2, 4
read.clipboard.csv, 4
read.clipboard.fwf, 5
read.clipboard.lower, 5
read.clipboard.tab, 4, 5
read.clipboard.upper, 5
Reise, 65
sat.act, 5, 17, 58
schmid, 3, 45, 46
score.items, 3, 40, 44, 45, 52, 56, 57
score.multiple.choice, 3
scrub, 7
sim, 63
sim.anova, 63
sim.circ, 63
sim.congeneric, 63
sim.dichot, 63, 64
sim.hierarchical, 63
sim.irt, 63, 64
sim.item, 63
sim.minor, 36, 63, 64
sim.npl, 63, 64
sim.npn, 63, 64
sim.omega, 63, 64
sim.parallel, 63
sim.rasch, 63, 64
sim.simple, 63
sim.simplex, 63, 64
sim.structural, 63
sim.structure, 36, 63, 64
sim.vss, 63
smc, 20
structure.diagram, 3, 65
super.matrix, 53
target.rot, 23
tetrachoric, 2, 3, 17, 57
Thurstone, 13, 65
Thurstone.33, 65
vegetables, 67, 68
vss, 2, 20, 36, 56
quartimax, 23
quartimin, 23
r.test, 15
rcorr, 15
read.clipboard, 2, 4
read.clipboard.csv, 4
read.clipboard.fwf, 5
read.clipboard.lower, 5
read.clipboard.tab, 4, 5
read.clipboard.upper, 5
read.table, 4
Reise, 65
Rgraphviz, 20
sat.act, 5, 17, 58
schmid, 3, 45, 46
score.items, 3, 40, 44, 45, 52, 56, 57
score.multiple.choice, 3
scrub, 7

sim, 63
sim.anova, 63
sim.circ, 63
sim.congeneric, 63
sim.dichot, 63, 64
sim.hierarchical, 63
sim.irt, 63, 64
sim.item, 63
sim.minor, 36, 63, 64
sim.npl, 63, 64
sim.npn, 63, 64
sim.omega, 63, 64
sim.parallel, 63
sim.rasch, 63, 64
sim.simple, 63
sim.simplex, 63, 64
sim.structural, 63
sim.structure, 36, 63, 64
sim.vss, 63
smc, 20
structure.diagram, 3, 65
super.matrix, 53
target.rot, 23
tetrachoric, 2, 3, 17, 57
Thurstone, 13, 65
Thurstone.33, 65
varimax, 23
vegetables, 67, 68
vss, 2, 20, 36, 56
R package
ctv, 4
eRm, 58
GPArotation, 3, 23
Hmisc, 15
ltm, 58
paran, 38
polycor, 3
psych, 2–4, 15, 17, 19, 21, 53, 65, 68
Rgraphviz, 4
sem, 3, 45, 49, 67
stats, 15, 21, 24
r.test, 15
rcorr, 15
read.clipboard, 2, 4
read.clipboard.csv, 4
read.clipboard.fwf, 5
read.clipboard.lower, 5
read.clipboard.tab, 4, 5
read.clipboard.upper, 5
read.table, 4
Reise, 65
Rgraphviz, 4, 20
rooted dendritic structure, 29

sat.act, 5, 17, 58
schmid, 3, 45, 46
Schmid-Leiman, 29
score.items, 3, 40, 44, 45, 52, 56, 57
score.multiple.choice, 3
scrub, 7
sem, 3, 45, 49, 67
sim, 63
sim.anova, 63
sim.circ, 63
sim.congeneric, 63
sim.dichot, 63, 64
sim.hierarchical, 63
sim.irt, 63, 64
sim.item, 63
sim.minor, 36, 63, 64
sim.npl, 63, 64
sim.npn, 63, 64
sim.omega, 63, 64
sim.parallel, 63
sim.rasch, 63, 64
sim.simple, 63
sim.simplex, 63, 64
sim.structural, 63
sim.structure, 36, 63, 64
sim.vss, 63
Singular Value Decomposition, 19
smc, 20

stats, 15, 21, 24
structure.diagram, 3, 65
super.matrix, 53

target.rot, 23
tetrachoric, 2, 3, 17, 57
Thurstone, 13, 65
Thurstone.33, 65
tree diagram, 29

varimax, 23
vegetables, 67, 68
vss, 2, 20, 36, 56

weighted least squares, 2
weighted least squares factor analysis, 19