An overview of the psych package

William Revelle

May 25, 2009

Contents

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

2 Basic data analysis **3**
2.1 Data input and descriptive statistics 3
2.2 Simple descriptive graphics 5
2.3 Testing correlations 5

3 Item and scale analysis **12**
3.1 Dimension reduction through factor analysis and cluster analysis 12
3.1.1 Item Cluster Analysis: ICLUST 14
3.1.2 Minimum Residual Factor Analysis 15
3.1.3 Principal Axis Factor Analysis 17
3.1.4 Principal Components analysis 17
3.1.5 Hierarchical and bi-factor solutions 22
3.2 Comparing factor/component/cluster solutions 22
3.3 Determining the number of dimensions to extract. 24
3.3.1 Very Simple Structure 25
3.3.2 Parallel Analysis 25
3.4 Reliability analysis 28
3.4.1 Reliability of a single scale 29
3.4.2 Reliability of multiple scales within an inventory 35
3.5 Item analysis 40
3.6 Multiple Regression from the correlation matrix 40

4 Simulation functions **42**

5 Data sets **43**

6 Development version and a users guide **44**

7 Psychometric Theory **45**

8 sessionInfo **45**

1 Overview of this document and related documents

The *psych* package (Revelle, 2009) 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 Very Simple Structure (Revelle and Rocklin, 1979) (`VSS`), Minimum Average Partial correlation (Velicer, 1976) (`MAP`), Item Cluster Analysis (Revelle, 1979) (`ICLUST`) and principal axes factor analysis (`factor.pa`), as well as functions to do Schmid Leiman transformations (Schmid and Leiman, 1957) (`schmid`) to transform a hierarchical factor structure into a bifactor solution (Holzinger and Swineford, 1937), 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.

This vignette is meant to give an overview of the *psych* package; it is meant to give an overview of the main functions in the psych package and how they are used for data description, dimension reduction, and scale construction. The extended user manual 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, 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.

2 Basic data analysis

2.1 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 "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 is that is comma delimited.

`read.clipboard.lower` for reading input of a lower triangular matrix with or without a diagonal.

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

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.

```
> my.data <- read.clipboard(sep="\t")
```

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    -0.07 0.05
age           2.42 0.36
ACT           0.53 0.18
SATV          0.33 4.27
SATQ         -0.02 4.41
```

These data can 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 2). 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)

$`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

$`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
```

2.2 Simple descriptive graphics

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.

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 2). 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.

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" 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`.

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 in the `bifactor` data set.

2.3 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

```
> pairs.panels(sat.act)
```

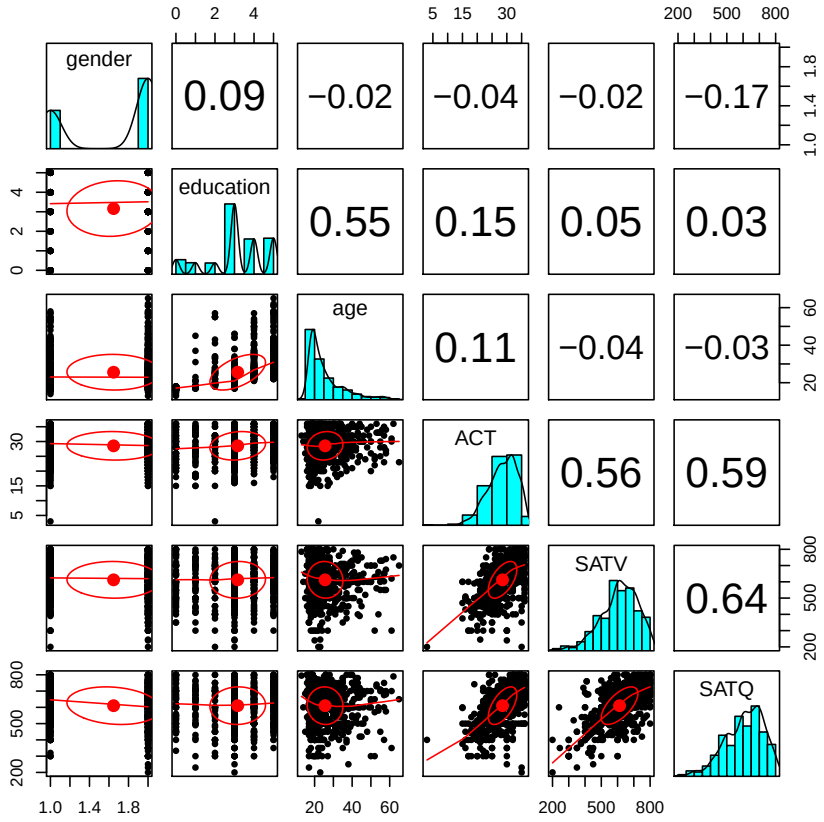

Figure 1: Using the `pairs.panels` function to graphically show relationships. The x axis in each scatter plot represents the column variable, the y axis the row variable. Note the extreme outlier for the ACT.

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

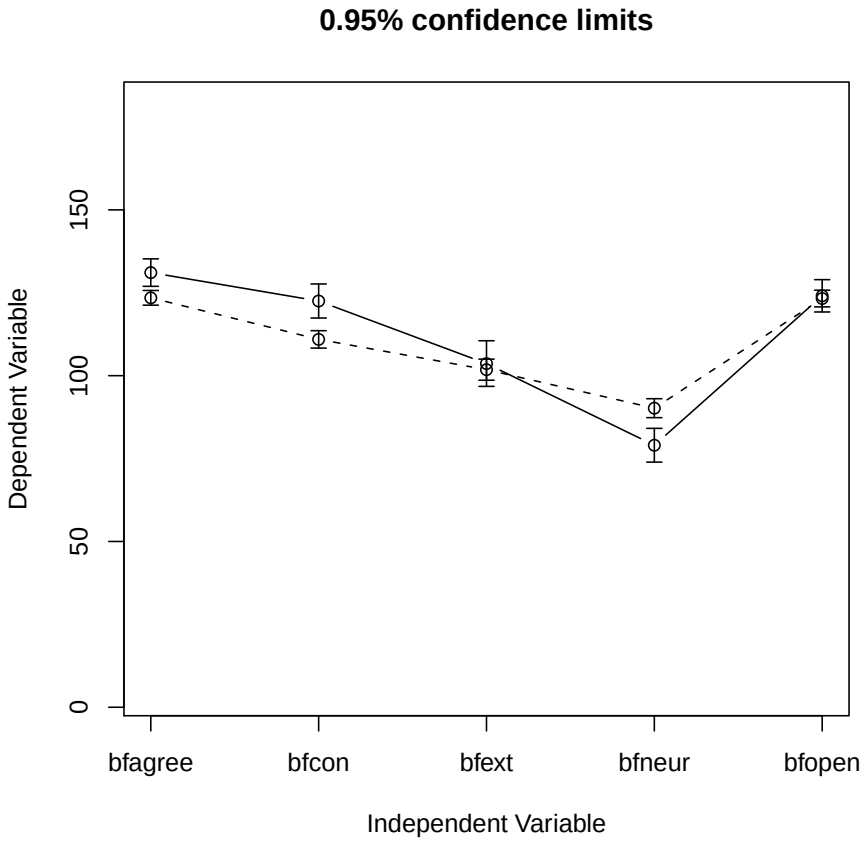

Figure 2: Using the `error.bars.by` function shows that self reported personality scales on the Big Five Inventory vary as a function of the Lie scale on the EPI.

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

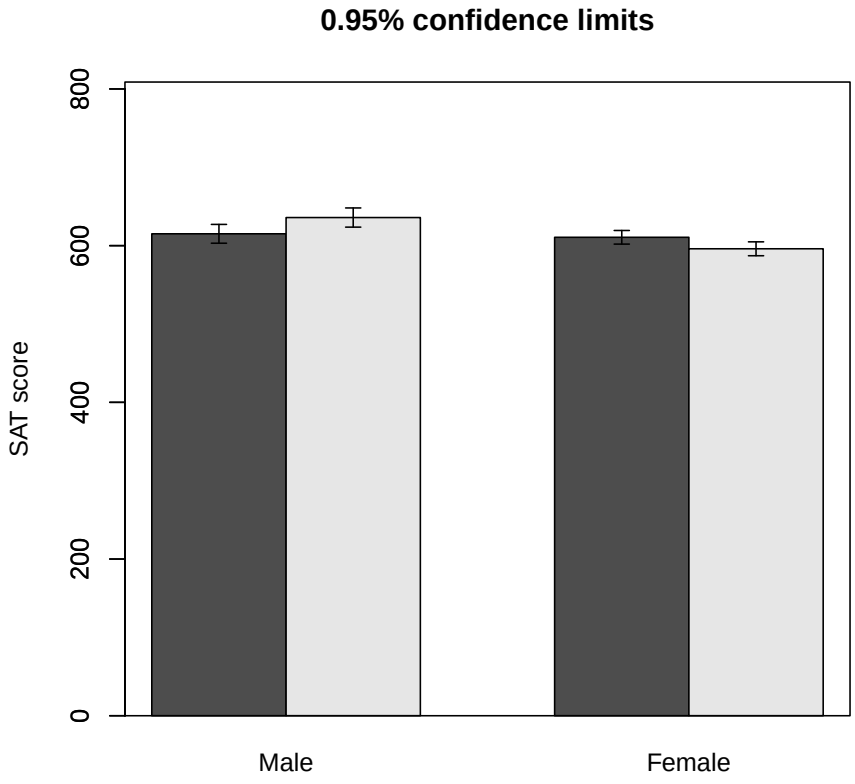

Figure 3: A "Dynamite plot" of SAT scores as a function of gender is one way of misleading the reader. By using a bar graph, the range of scores is ignored.

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

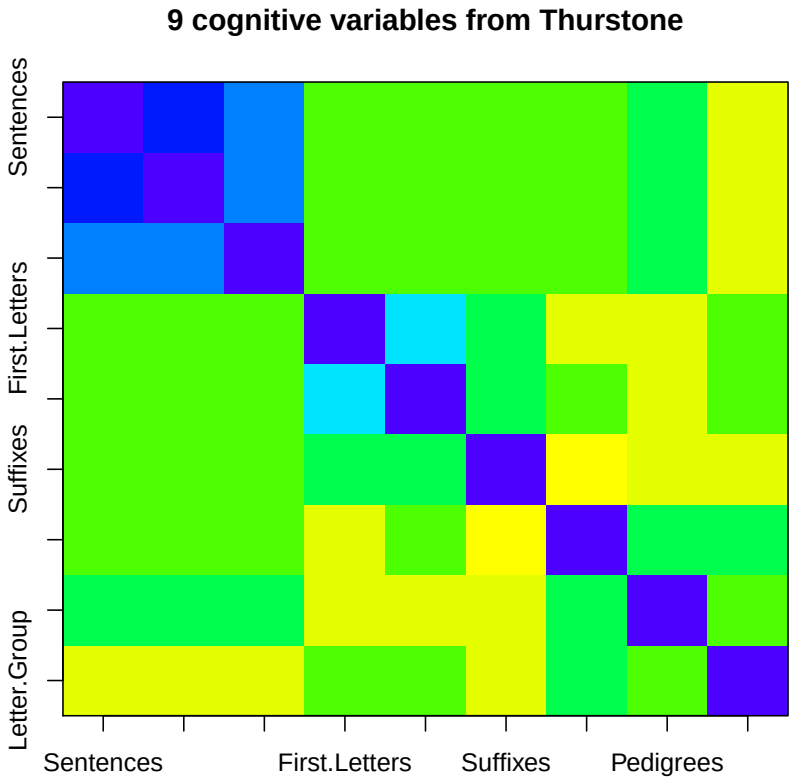

Figure 4: The structure of correlation matrix can be seen more clearly if the variables are grouped by factor and then the correlations are shown by color.

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 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.

3.1 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 of representing random samples 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

[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.

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.

`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.

`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 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.

`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. If there are more than two factors, then a SPLOM of the loadings is generated.

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

3.1.1 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,
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.

Table 1: The summary statistics from an ICLUST analysis shows three large clusters and one item that is not grouped with the other clusters.

```
> summary(ic)
ICLUST (Item Cluster Analysis)Call: ICLUST(r.mat = bfi)
ICLUST

Purified Alpha:
 C21  C15  C16  V22
0.80 0.81 0.74 0.33

 Guttman Lambda6  *
 C21  C15  C16  V22
0.83 0.82 0.75 0.31

Original Beta:
 C21  C15  C16  V22
0.58 0.73 0.65   NA

Cluster size:
C21 C15 C16 V22
 12   5   5   3

Purified scale intercorrelations
 reliabilities on diagonal
 correlations corrected for attenuation above diagonal:
      C21  C15   C16   V22
C21  0.80 0.24  0.38 -0.26
C15  0.19 0.81  0.20 -0.01
C16  0.29 0.16  0.74 -0.31
V22 -0.13 0.00 -0.15  0.33
```

3.1.2 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,

```
> data(bfi)
> if (require(Rgraphviz)) {
+     ic <- ICLUST(bfi)
+ } else {
+     ic <- ICLUST(bfi, plot = FALSE)
+     cluster.plot(ic)
+ }
```

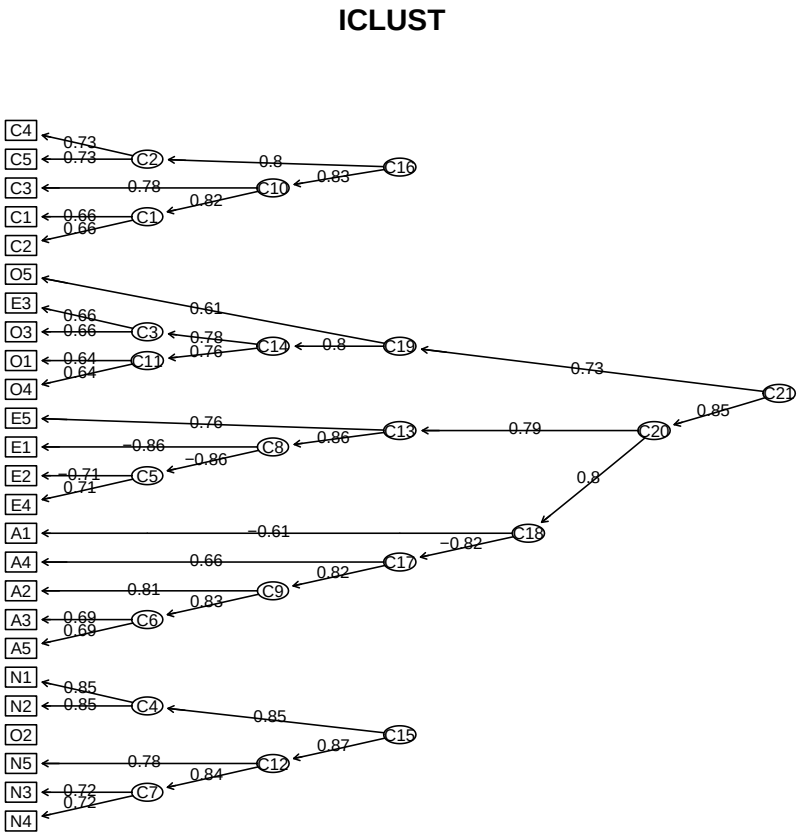

Figure 5: Using the `ICLUST` function to find the cluster structure of 25 personality items.

${}_n\vec{F}_k$ 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. Two alternatives are provided in *psych*, `factor.minres` and `factor.pa`. `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.

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 independent clusters solution using the "cluster" option on rotate (Table 2). 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).

3.1.3 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.

Both the `factor.minres` and `factor.pa` as well as the `principal` solutions can be rotated or transformed with a number of options. Some of these call 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 3).

3.1.4 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,

Table 2: Three correlated factors from the Thurstone 9 variable problem. The solution is transformed obliquely using a targeted rotation towards an independent cluster solution.

```
> f3t <- factor.minres(Thurstone, 3, n.obs = 213, rotate = "cluster")
> f3t
                V   MR1   MR2   MR3
Sentences       1  0.88
Vocabulary      2  0.87
Sent.Completion 3  0.81
First.Letters   4        0.87
4.Letter.Words  5        0.76
Suffixes        6        0.65
Letter.Series   7              0.93
Pedigrees       8  0.32        0.53
Letter.Group    9              0.69

                MR1  MR2  MR3
SS loadings    2.33 1.81 1.65
Proportion Var 0.26 0.20 0.18
Cumulative Var 0.26 0.46 0.64

 With factor correlations of
    MR1  MR2  MR3
MR1 1.0 0.60 0.60
MR2 0.6 1.00 0.61
MR3 0.6 0.61 1.00

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the model is 12  and the fit was  0.01
The number of observations was  213  with Chi Square =  2.87  with prob <  1

Measures of factor score adequacy              MR1 MR2 MR3
 Correlation of scores with factors            0.96 0.93 0.92
Multiple R square of scores with factors       0.93 0.86 0.85
Minimum correlation of factor score estimates  0.85 0.72 0.69
Validity of unit weighted factor scores        0.96 0.91 0.89
```

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.

```
> data(bifactor)
> f3 <- factor.pa(Thurstone, 3, n.obs = 213)
> f3t <- target.rot(f3)
> f3t
                V   PA1   PA2   PA3
Sentences       1  0.87
Vocabulary      2  0.87
Sent.Completion 3  0.81
First.Letters   4        0.87
4.Letter.Words  5        0.76
Suffixes        6        0.65
Letter.Series   7              0.92
Pedigrees       8  0.32        0.53
Letter.Group    9              0.70

                PA1  PA2  PA3
SS loadings    2.33 1.81 1.64
Proportion Var 0.26 0.20 0.18
Cumulative Var 0.26 0.46 0.64

 With factor correlations of
    PA1  PA2  PA3
PA1 1.0 0.60 0.60
PA2 0.6 1.00 0.61
PA3 0.6 0.61 1.00
```

```
> factor.plot(f3t)
```

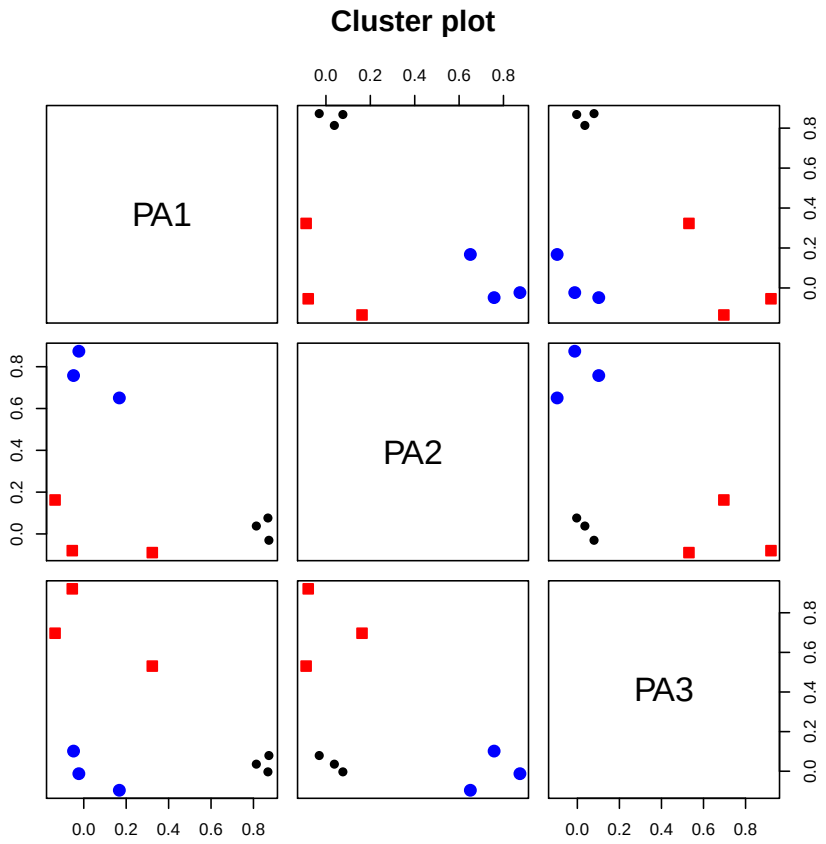

Figure 6: A graphic representation of the 3 oblique factors from the Thurstone data using `factor.plot`. Factors were transformed to an oblique solution using the `target.rot` function.

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. 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.

Table 4: The Thurstone problem can also be analyzed using Principal Components Analysis. Compare this to Table 3. The loadings are higher for the PCA because the model accounts for the unique as well as the common variance.

```
> data(bifactor)
> p3 <- principal(Thurstone, 3, n.obs = 213)
> p3p <- Promax(p3)
> p3p
                 V   PC1   PC2   PC3
Sentences        1  0.92
Vocabulary       2  0.90
Sent.Completion  3  0.91
First.Letters    4        0.84
4.Letter.Words   5        0.81
Suffixes         6        0.79
Letter.Series    7              0.88
Pedigrees        8  0.45        0.57
Letter.Group     9              0.86

                 PC1  PC2  PC3
SS loadings     2.76 2.07 1.91
Proportion Var  0.31 0.23 0.21
Cumulative Var  0.31 0.54 0.75

 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
```

3.1.5 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 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 agression).

Consider the 9 Thurstone variables from above (Figure 7). The figure shows both a bi-factor solution as well as a hierarchical solution with a g factor.

3.2 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 <- factor.pa(bfi, 4)
> factor.congruence(f4, ic)
```

```
      C21   C15   C16  V22
PA1 -0.98 -0.31 -0.53 0.20
PA2  0.34  0.99  0.34 0.29
PA3 -0.47 -0.24 -0.96 0.38
PA4  0.18  0.00 -0.04 0.79
```

Now consider the solutions to the Thurstone problem, considering the factor, component, and bifactor solution.

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

```
       g  F1*  F2*  F3*  PC1  PC2   PC3   PA1  PA2  PA3
g   1.00 0.75 0.65 0.58 0.69 0.58  0.50  0.67 0.57 0.54
F1* 0.75 1.00 0.13 0.17 0.99 0.10  0.09  0.99 0.08 0.18
F2* 0.65 0.13 1.00 0.19 0.09 0.98  0.16  0.08 0.99 0.11
```

```
> if (require(Rgraphviz)) {
+     op <- par(mfrow = c(1, 2))
+     om <- omega(Thurstone, n.obs = 213)
+     om.h <- omega(Thurstone, n.obs = 213, sl = FALSE)
+     op <- par(mfrow = c(1, 1))
+ } else {
+     plot(1:10, main = "Figure missing", typ = "l")
+     points(10:1, typ = "l")
+     text(5, 2, "Rgraphviz needs to be installed")
+     om <- omega(Thurstone, n.obs = 213)
+ }
```

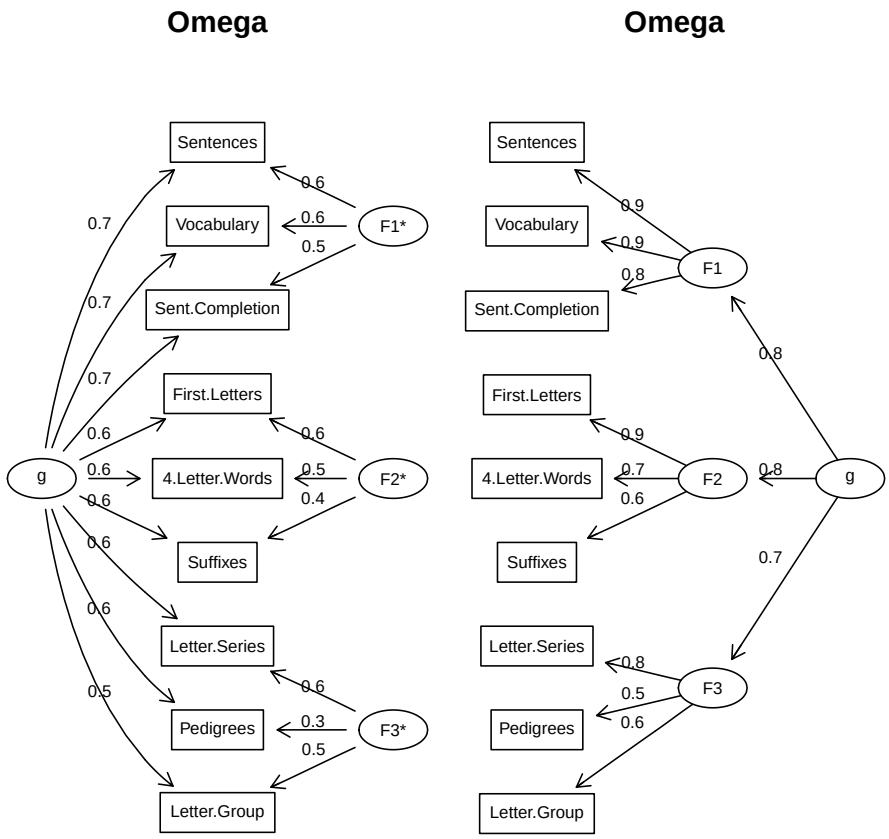

Figure 7: A bifactor and hierarchical factor solution to the Thurstone 9 variable problem

```
F3* 0.58 0.17 0.19 1.00 0.10 0.10  0.97  0.05 0.08 0.99
PC1 0.69 0.99 0.09 0.10 1.00 0.06  0.00  1.00 0.04 0.10
PC2 0.58 0.10 0.98 0.10 0.06 1.00  0.05  0.06 0.99 0.01
PC3 0.50 0.09 0.16 0.97 0.00 0.05  1.00 -0.04 0.05 0.99
PA1 0.67 0.99 0.08 0.05 1.00 0.06 -0.04  1.00 0.04 0.05
PA2 0.57 0.08 0.99 0.08 0.04 0.99  0.05  0.04 1.00 0.00
PA3 0.54 0.18 0.11 0.99 0.10 0.01  0.99  0.05 0.00 1.00
```

3.3 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`.

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.

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

6) Extracting factors as long as they are interpetable.

7) Using the Very Structure Criterion (`VSS`).

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

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 appealling but can lead to differences of interpretation as to when the scree "breaks". 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. 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).

3.3.1 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 8).

```
> vss

Very Simple Structure of  Very Simple Structure of a Big 5 inventory
Call: VSS(x = bfi, title = "Very Simple Structure of a Big 5 inventory")
VSS complexity 1 achieves a maximimum of 0.58  with  4  factors
VSS complexity 2 achieves a maximimum of 0.72  with  8  factors

The Velicer MAP criterion achieves a minimum of 0.03  with  5  factors

Velicer MAP
[1] 0.03 0.02 0.02 0.02 0.02 0.02 0.02 0.02

Very Simple Structure Complexity 1
[1] 0.44 0.53 0.57 0.58 0.52 0.52 0.53 0.53

Very Simple Structure Complexity 2
[1] 0.00 0.61 0.69 0.69 0.70 0.71 0.71 0.72
```

3.3.2 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 9).

```
> vss <- VSS(bfi, title = "Very Simple Structure of a Big 5 inventory")
```

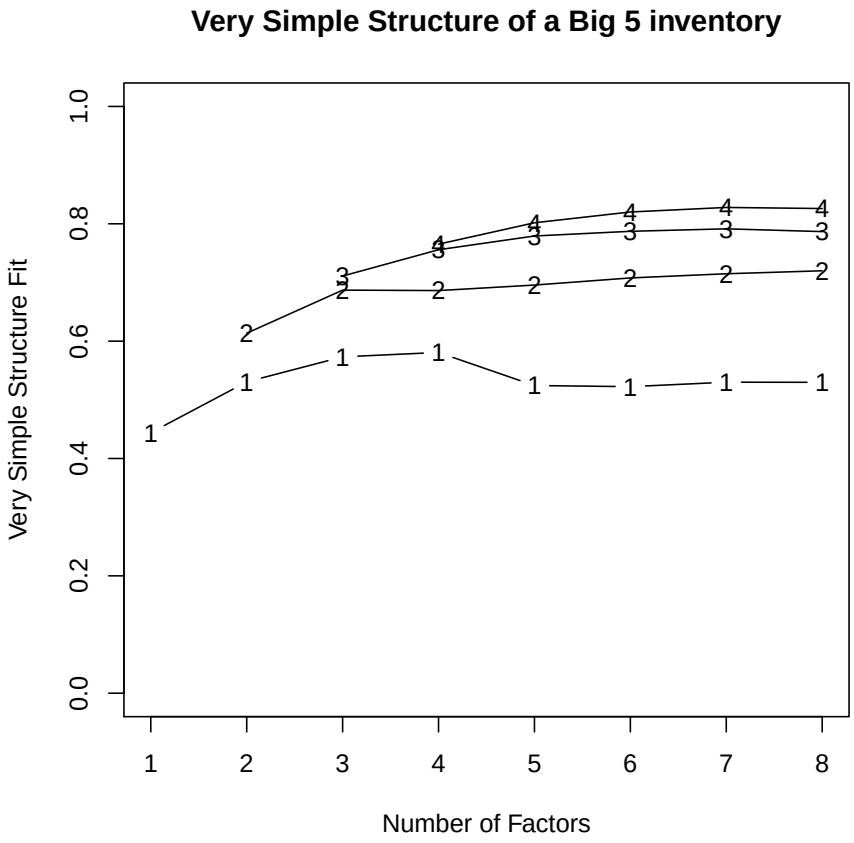

Figure 8: The Very Simple Structure criterion for the number of factors compares solutions for various levels of item complexity and various numbers of factors.

```
> fa.parallel(bfi, main = "Parallel Analysis of a Big 5 inventory")
```

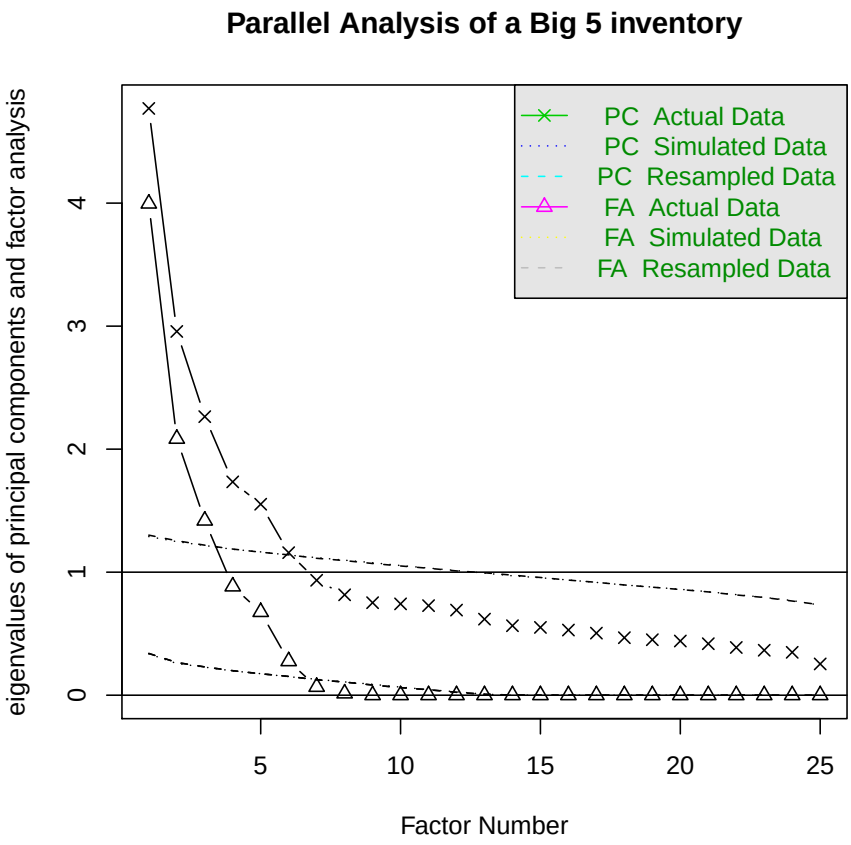

Figure 9: Parallel analysis compares factor and principal components solutions to the real data as well as resampled data.

3.4 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

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

Perhaps because it is so easy to calculate and is available in most commercial programs, 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 lumpyness 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 asβ (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.

3.4.1 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 -0.17 5.2

 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    mean   sd
V1 500 0.70  0.68 -0.0272 0.96
```

```
V2 500 0.68  0.64 -0.0030 1.01
V3 500 0.58  0.50  0.0202 0.97
V4 500 0.65  0.60 -0.0397 0.98
V5 500 0.60  0.53 -0.0205 1.03
V6 500 0.53  0.44 -0.0165 0.97
V7 500 0.51  0.40 -0.0351 1.03
V8 500 0.50  0.39 -0.0028 1.00
V9 500 0.43  0.30 -0.0494 1.03
```

Some scales have items that need to be reversed before being 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  423 58

 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 mean   sd
rating     30  0.60  0.60   65 12.2
complaints 30 -0.58 -0.74   67 13.3
privileges 30  0.65  0.54   53 12.2
learning   30  0.79  0.78   56 11.7
raises     30  0.83  0.85   65 10.4
critical   30  0.53  0.35   75  9.9
```

```
advance    30  0.75  0.71   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  423 58

 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 mean   sd
rating     30 0.76  0.75   65 12.2
complaints 30 0.81  0.82   67 13.3
privileges 30 0.68  0.60   53 12.2
learning   30 0.80  0.78   56 11.7
raises     30 0.86  0.85   65 10.4
critical   30 0.45  0.31   75  9.9
advance    30 0.62  0.56   43 10.3
```

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 (See Figure 10).

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.

```
> if (require(Rgraphviz)) {
+     om.9 <- omega(r9, title = "9 simulated variables")
+ } else {
+     plot(1:10, main = "Figure missing", typ = "l")
+     points(10:1, typ = "l")
+     text(5, 2, "Rgraphviz needs to be installed")
+     om.9 <- omega(r9, title = "9 simulated variables")
+ }
```

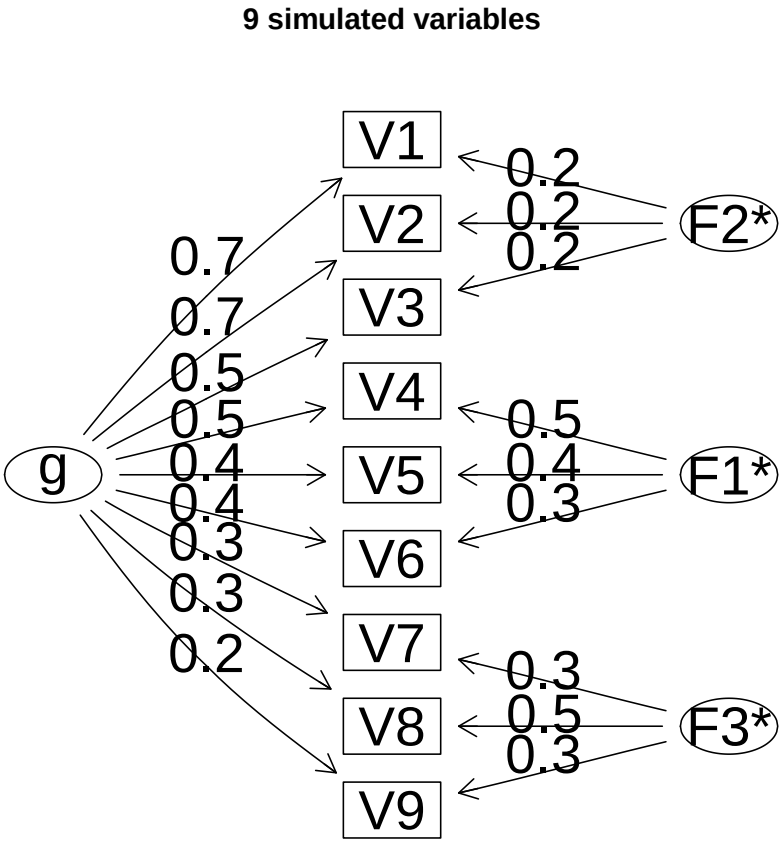

Figure 10: A bifactor solution for 9 simulated variables with a hierarchical structure.

```
> 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
V1 0.70                0.53 0.47
V2 0.70                0.52 0.48
V3 0.54                0.32 0.68
V4 0.53 0.47           0.51 0.49
V5 0.44 0.43           0.38 0.62
V6 0.40 0.31           0.26 0.74
V7 0.31           0.31 0.20 0.80
V8 0.34           0.46 0.33 0.67
V9 0.24           0.34      0.83

With eigenvalues of:
    g   F1*   F2*   F3*
2.175 0.510 0.082 0.452

general/max  4.26   max/min =   6.24
The degrees of freedom for the model is 12  and the fit was  0.03
The number of observations was  500  with Chi Square =  14.17  with prob <  0.29
```

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 =  0.54  This is an estimate of the worst split half reliability
Guttman bounds
L1 =  0.66
L2 =  0.75
```

```
L3 (alpha) =  0.75
L4 (max) =  0.8
L5 =  0.74
L6 (smc) =  0.74
TenBerge bounds
mu0 =  mu1 =  0.75 mu2 =  0.76 mu3 =  0.76

alpha of first PC =  0.76
estimated greatest lower bound =  0.8

beta estimated by first and second PC =  0.51  This is an exploratory statistic
```

3.4.2 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.

```
> keys <- make.keys(25, 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

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 15 items. Then the following code would work:

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

The resulting keys matrix is identical to that found above.

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  0.65          0.74         0.77        0.81     0.47

Average item correlation:
          Agree Conscientious Extraversion Neuroticism Openness
average.r  0.27          0.36         0.41        0.46     0.15

 Guttman 6* reliability:
         Agree Conscientious Extraversion Neuroticism Openness
Lambda.6  0.66          0.74         0.78        0.82     0.53

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         0.65          0.36         0.62       -0.30     0.33
Conscientious 0.25          0.74         0.28       -0.22     0.24
Extraversion  0.44          0.21         0.77       -0.24     0.57
Neuroticism  -0.22         -0.17        -0.19        0.81     0.11
Openness      0.18          0.14         0.34        0.07     0.47

Item by scale correlations:
```

```
 corrected for item overlap and scale reliability
   Agree Conscientious Extraversion Neuroticism Openness
A1 -0.40         -0.12        -0.07        0.16    -0.16
A2  0.63          0.20         0.41       -0.09     0.26
A3  0.63          0.20         0.41       -0.13     0.30
A4  0.39          0.21         0.22       -0.17    -0.05
A5  0.61          0.19         0.54       -0.23     0.17
C1  0.12          0.59         0.11       -0.03     0.23
C2  0.19          0.56         0.08        0.07     0.14
C3  0.24          0.61         0.23       -0.03     0.15
C4 -0.29         -0.70        -0.22        0.30    -0.17
C5 -0.22         -0.57        -0.20        0.32    -0.01
E1 -0.36         -0.07        -0.65        0.09    -0.42
E2 -0.35         -0.21        -0.69        0.28    -0.23
E3  0.44          0.19         0.61       -0.09     0.50
E4  0.53          0.12         0.68       -0.23     0.10
E5  0.27          0.34         0.58       -0.05     0.49
N1 -0.26         -0.17        -0.06        0.77     0.13
N2 -0.26         -0.11        -0.06        0.74     0.09
N3 -0.19         -0.16        -0.10        0.71     0.10
N4 -0.24         -0.27        -0.37        0.63     0.08
N5 -0.06         -0.04        -0.23        0.54    -0.03
O1  0.21          0.17         0.30       -0.01     0.54
O2  0.13          0.13        -0.22       -0.06    -0.09
O3  0.29          0.21         0.43        0.01     0.63
O4  0.17          0.10         0.00        0.22     0.42
O5 -0.12         -0.14        -0.15        0.06    -0.42
```

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

Forming scales from a correlation matrix There are some situations when the raw data are not available, but the correlation matrix is. 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")
```

```
> pairs.panels(scores$scores)
```

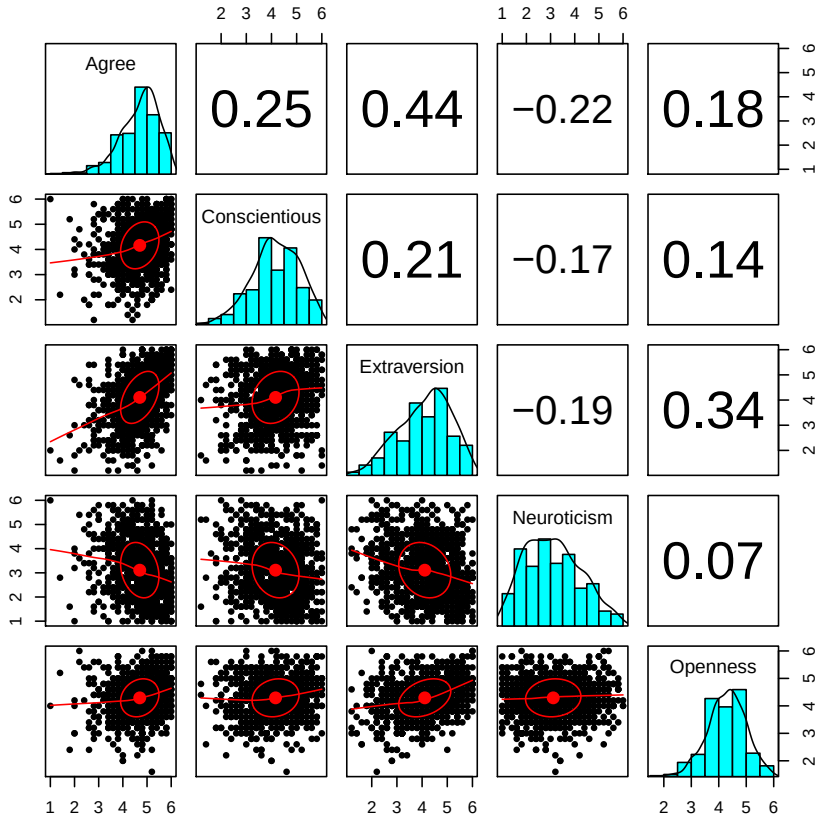

Figure 11: A graphic analysis of the Big Five scales found by using the score.items function.

```
> 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         0.66          0.35         0.63       -0.30     0.34
Conscientious 0.25          0.74         0.29       -0.20     0.25
Extraversion  0.45          0.22         0.78       -0.23     0.57
Neuroticism  -0.22         -0.16        -0.19        0.81     0.11
Openness      0.19          0.15         0.35        0.07     0.48
```

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.

3.5 Item analysis

Basic item analysis starts with describing the data (`describe`, finding the number of dimensions using `factor.pa`, `VSS`, and parallel analysis `fa.parallel`. Item whole correlations may then be found for scales scored on one dimension (`alpha` or many scales simultaneously (`score.items`).

3.6 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 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
-25.246  -3.213   0.777   3.592   9.263

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

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

$beta
            ACT  SATV   SATQ
gender    -0.49 -7.08 -42.67
education  0.48  8.23   8.19
age        0.02 -1.19  -1.13

$se
           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
            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
           ACT SATV SATQ
gender    0.20 0.43 0.00
education 0.00 0.02 0.02
age       0.48 0.03 0.04

$R
 ACT SATV SATQ
0.16 0.10 0.19

$R2
 ACT SATV SATQ
0.03 0.01 0.04

$shrunkenR2
 ACT SATV SATQ
0.02 0.01 0.03

$seR2
 ACT SATV SATQ
0.01 0.01 0.01

$F
 ACT SATV SATQ
6.49 2.26 8.63

$probF
  ACT  SATV  SATQ
0.000 0.081 0.000

$df
[1]   3 696
```

4 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.

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

`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.

These functions are described in more detail in the companion vignette: psych for sem.

5 Data sets

A number of data sets for demonstrating psychometric techniques are included in the *psych* package. These include six data sets in `bifactor` 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 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.

bifactor 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 1000 subjects are included here as a demonstration set for scale construction and factor analysis.

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. item [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.

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 Guiford, Nunnally, and Nunnally and Bernstein, this data set allows for examples of basic scaling techniques.

6 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.

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 .

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

files.

7 Psychometric Theory

The *psych* package has been developed to help psychologists do basic research. Many of the functions were developed to supplement a 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 A short guide to R

8 sessionInfo

This document was prepared using the following settings.

```
> sessionInfo()

R version 2.10.0 Under development (unstable) (2009-05-22 r48594)
i386-apple-darwin9.6.0

locale:
[1] C

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

other attached packages:
[1] MASS_7.3-0            GPArotation_2009.02-1 Rgraphviz_1.21.10
[4] graph_1.21.7          psych_1.0-70

loaded via a namespace (and not attached):
[1] cluster_1.12.0
```

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. (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.

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. 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,.

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

Revelle, W. (2009). *psych: Procedures for Personality and Psychological Research*. R package version 1.0-67.

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

agnes, 14
alpha, 2, 29–31, 40

Bechtoldt.1, 43
Bechtoldt.2, 43
bfi, 35, 38, 43
bifactor, 5, 43

cities, 44
cluster analysis, 12, 14
cluster scores, 13
cluster.cor, 29, 35, 38
cluster.loadings, 35, 40
common variance, 21
component scores, 13
congruence coefficient, 22
cor.test, 5
corr.test, 5
cortest, 11
cubits, 43

describe, 2, 3, 40
describe.by, 3

epi.bfi, 43
error bars, 5
error.bars, 2, 5
error.bars.by, 4, 5, 7
error.crosses, 5

FA, 12
fa.graph, 14
fa.parallel, 13, 24, 40
factanal, 13, 17
factor analysis, 12
factor scores, 13
factor.congruence, 13, 22
factor.minres, 13, 17
factor.pa, 2, 13, 17, 40
factor.plot, 13, 20
factors, 13

galton, 44
GPArotation, 17
guttman, 2, 34

hclust, 14
heights, 43
hierarchical cluster analysis, 14
Hmisc, 5
Holzinger, 43

ICC, 2
ICLUST, 2, 13, 14, 16, 28
iqitems, 43

keys, 31

lm, 40

make.keys, 35, 36
MAP, 2, 13, 24
mat.regress, 40

oblimin, 17
omega, 2, 22, 28, 29, 32

pairs, 5
pairs.panels, 2, 5, 6, 38
PCA, 12
peas, 43, 44
principal, 13, 17, 21
principal components, 13
principal components analysis, 12
princomp, 21
Promax, 17
psych, 2, 5, 12, 13, 17, 36, 43–45

quartimax, 17
quartimin, 17

R function
agnes, 14
alpha, 2, 29–31, 40
Bechtoldt.1, 43
Bechtoldt.2, 43
bfi, 35, 38, 43
bifactor, 5, 43
cities, 44
cluster.cor, 29, 35, 38
cluster.loadings, 35, 40
cor.test, 5
corr.test, 5
cortest, 11
cubits, 43
describe, 2, 3, 40
describe.by, 3
epi.bfi, 43
error.bars, 2, 5
error.bars.by, 4, 5, 7
error.crosses, 5
fa.graph, 14
fa.parallel, 13, 24, 40
factanal, 13, 17
factor.congruence, 13, 22
factor.minres, 13, 17
factor.pa, 2, 13, 17, 40
factor.plot, 13, 20
galton, 44
guttman, 2, 34
hclust, 14
heights, 43
Holzinger, 43
ICC, 2
ICLUST, 2, 13, 14, 16, 28
iqitems, 43
lm, 40
make.keys, 35, 36
MAP, 2, 13, 24
mat.regress, 40
oblimin, 17
omega, 2, 22, 28, 29, 32
pairs, 5
pairs.panels, 2, 5, 6, 38
peas, 43, 44
principal, 13, 17, 21
princomp, 21
Promax, 17
psych, 44
psych package
alpha, 2, 29–31, 40
Bechtoldt.1, 43
Bechtoldt.2, 43
bfi, 35, 38, 43
bifactor, 5, 43
cities, 44
cluster.cor, 29, 35, 38
cluster.loadings, 35, 40
corr.test, 5
cortest, 11
cubits, 43
describe, 2, 3, 40
describe.by, 3
epi.bfi, 43
error.bars, 2, 5
error.bars.by, 4, 5, 7
error.crosses, 5
fa.graph, 14
fa.parallel, 13, 24, 40
factor.congruence, 13, 22
factor.minres, 13, 17
factor.pa, 2, 13, 17, 40
factor.plot, 13, 20
galton, 44
guttman, 2, 34
heights, 43
Holzinger, 43
ICC, 2
ICLUST, 2, 13, 14, 16, 28
iqitems, 43
make.keys, 35, 36
MAP, 2, 13, 24
mat.regress, 40

omega, 2, 22, 28, 29, 32
pairs.panels, 2, 5, 6, 38
peas, 43, 44
principal, 13, 17, 21
Promax, 17
psych, 44
r.test, 10
read.clipboard, 2, 3
read.clipboard.csv, 3
read.clipboard.lower, 3
read.clipboard.upper, 3
Reise, 43
sat.act, 3, 11, 40
schmid, 2
score.items, 2, 29, 35, 38, 40
score.multiple.choice, 2
sim, 42
sim.circ, 43
sim.congeneric, 43
sim.dichot, 43
sim.hierarchical, 43
sim.item, 43
sim.structure, 43
smc, 13
super.matrix, 36
target.rot, 17, 20
Thurstone, 5, 43
Thurstone.33, 43
vegetables, 43, 44
VSS, 2, 13, 24, 25, 40
quartimax, 17
quartimin, 17
r.test, 10
rcorr, 5
read.clipboard, 2, 3
read.clipboard.csv, 3
read.clipboard.lower, 3
read.clipboard.upper, 3
read.table, 3
Reise, 43
Rgraphviz, 14
sat.act, 3, 11, 40
schmid, 2
score.items, 2, 29, 35, 38, 40
score.multiple.choice, 2
sim, 42
sim.circ, 43
sim.congeneric, 43
sim.dichot, 43
sim.hierarchical, 43
sim.item, 43
sim.structure, 43
smc, 13
super.matrix, 36
target.rot, 17, 20
Thurstone, 5, 43
Thurstone.33, 43
varimax, 17
vegetables, 43, 44
VSS, 2, 13, 24, 25, 40
R package
GPArotation, 17
Hmisc, 5
psych, 2, 5, 12, 13, 17, 36, 43–45
sem, 2, 43
stats, 5, 17, 21
r.test, 10
rcorr, 5
read.clipboard, 2, 3
read.clipboard.csv, 3
read.clipboard.lower, 3
read.clipboard.upper, 3
read.table, 3
Reise, 43
Rgraphviz, 14
rooted dendritic structure, 14

sat.act, 3, 11, 40
schmid, 2
score.items, 2, 29, 35, 38, 40
score.multiple.choice, 2
sem, 2, 43

sim, 42
sim.circ, 43
sim.congeneric, 43
sim.dichot, 43
sim.hierarchical, 43
sim.item, 43
sim.structure, 43
Singular Value Decomposition, 12
smc, 13
stats, 5, 17, 21
super.matrix, 36

target.rot, 17, 20
Thurstone, 5, 43
Thurstone.33, 43
tree diagram, 14

varimax, 17
vegetables, 43, 44
VSS, 2, 13, 24, 25, 40