An introduction to the psych package: Part I: data entry and data description

William Revelle
Department of Psychology
Northwestern University

March 18, 2021

Contents

0.1 Jump starting the *psych* package–a guide for the impatient

You have installed *psych* (section 2) and you want to use it without reading much more. What should you do?

1. Activate the *psych* package and the *psychTools* package:

R code
```
library(psych)
library(psychTools)
```

2. Input your data (section 3.1). There are two ways to do this:

   - Find and read standard files using `read.file`. This will open a search window for your operating system which you can use to find the file. If the file has a suffix of .text, .txt, .TXT, .csv, ,dat, .data, .sav, .xpt, .XPT, .r, .R, .rds, .Rds, .rda, .Rda, .rdata, Rdata, or .RData, then the file will be opened and the data will be read in (or loaded in the case of Rda files)

R code
```
myData <- read.file() # find the appropriate file using
                              #   your normal operating system
```

   - Alternatively, go to your friendly text editor or data manipulation program (e.g., Excel) and copy the data to the clipboard. Include a first line that has the variable labels. Paste it into *psych* using the `read.clipboard.tab` command:

R code
```
myData <- read.clipboard.tab() # if on the clipboard
```

     Note that there are number of options for `read.clipboard` for reading in Excel based files, lower triangular files, etc.

3. Make sure that what you just read is right. Describe it (section 3.3) and perhaps look at the first and last few lines. If you have multiple groups, try `describeBy`.

R code
```
dim(myData)    #What are the dimensions of the data?
describe(myData)  # or
describeBy(myData,groups="mygroups")   #for descriptive statistics by groups
headTail(myData)  #show the first and last n lines of a file
```

4. Look at the patterns in the data. If you have fewer than about 12 variables, look at the SPLOM (Scatter Plot Matrix) of the data using `pairs.panels` (section 3.4.1)

Then, use the `outlier` function to detect outliers.

R code

```
pairs.panels(myData)
outlier(myData)
```

5. Note that you might have some weird subjects, probably due to data entry errors. Either edit the data by hand (use the `edit` command) or just `scrub` the data (section 3.3.2).

R code

```
cleaned <- scrub(myData,  max=9)  #e.g., change anything great than 9 to NA
```

6. Graph the data with error bars for each variable (section 3.4.3).

R code

```
error.bars(myData)
```

7. Find the correlations of all of your data. `lowerCor` will by default find the pairwise correlations, round them to 2 decimals, and display the lower off diagonal matrix.
   - Descriptively (just the values) (section 3.4.7)

     R code

     ```
     r  <-  lowerCor(myData) #The correlation matrix, rounded to 2 decimals
     ```

   - Graphically (section 3.4.8). Another way is to show a heat map of the correlations with the correlation values included.

     R code

     ```
     corPlot(r)   #examine the many options for this function.
     ```

   - Inferentially (the values, the ns, and the p values) (section 3.5)

     R code

     ```
     corr.test(myData)
     ```

8. Apply various regression models.

   Several functions are meant to do multiple regressions, either from the raw data or from a variance/covariance matrix, or a correlation matrix. This is discussed in more detail in the "How To use `mediate` and `setCor` to do mediation, moderation and regression analysis tutorial.

- `setCor` will take raw data or a correlation matrix and find (and graph the path diagram) for multiple y variables depending upon multiple x variables. If we have the raw data, we can also find the interaction term (x1 * x2). Although we can find the regressions from just a correlation matrix, we can not find the interaction (moderation effect) unless given raw data.

R code

```
myData <- sat.act
colnames(myData) <- c("mod1","med1","x1","x2","y1","y2")
setCor(y1 + y2 ~  x1 + x2 + x1*x2, data = myData)
```

- `mediate` will take raw data or a correlation matrix and find (and graph the path diagram) for multiple y variables depending upon multiple x variables mediated through a mediation variable. It then tests the mediation effect using a boot strap. We specify the mediation variable by enclosing it in parentheses, and show the moderation by the standard multiplication. For the purpose of this demonstration, we do the boot strap with just 50 iterations. The default is 5,000. We use the data from Tal-Or et al. (2010) which was downloaded from the supplementary material for Hayes (2013) https://www.afhayes.com/public/hayes2013data.zip.

R code

```
mediate(reaction ~ cond + (import) + (pmi), data =Tal_Or,n.iter=50)
```

We can also find the moderation effect by adding in a product term.

- `mediate` will take raw data and find (and graph the path diagram) a moderated multiple regression model for multiple y variables depending upon multiple x variables mediated through a mediation variable. It then tests the mediation effect using a boot strap. By default, we find the raw regressions and mean center. If we specify zero=FALSE, we do not mean center the data. If we specify std=TRUE, we find the standardized regressions.

R code

```
mediate(respappr ~ prot * sexism +(sexism),data=Garcia,zero=FALSE, n.iter=50,
    main="Moderated  mediation (not mean centered)")
```

0.2 Psychometric functions are summarized in the second vignette

Many additional functions, particularly designed for basic and advanced psychometrics are discussed more fully in the *Overview Vignette*, which may be downloaded from https://personality-project.org/r/psych/vignettes/overview.pdf . A brief review of the functions available is included here. In addition, there are helpful

tutorials for *Finding omega*, *How to score scales and find reliability*, and for *Using psych for factor analysis* at https://personality-project.org/r.

- Test for the number of factors in your data using parallel analysis (`fa.parallel`) or Very Simple Structure (`vss`) .

R code

```
fa.parallel(myData)
vss(myData)
```

- Factor analyze (see section 4.1) the data with a specified number of factors (the default is 1), the default method is minimum residual, the default rotation for more than one factor is oblimin. There are many more possibilities such as minres (section 4.1.1), alpha factoring, and wls. Compare the solution to a hierarchical cluster analysis using the ICLUST algorithm (Revelle, 1979) (see section 4.1.6). Also consider a hierarchical factor solution to find coefficient ω).

R code

```
fa(myData)
iclust(myData)
omega(myData)
```

  If you prefer to do a principal components analysis you may use the `principal` function. The default is one component.

R code

```
principal(myData)
```

- Some people like to find coefficient α as an estimate of reliability. This may be done for a single scale using the `alpha` function. Perhaps more useful is the ability to create several scales as unweighted averages of specified items using the `scoreItems` function and to find various estimates of internal consistency for these scales, find their intercorrelations, and find scores for all the subjects.

R code

```
alpha(myData)   #score all of the items as part of one scale.
myKeys  <- make.keys(nvar=20,list(first = c(1,-3,5,-7,8:10),
            second=c(2,4,-6,11:15,-16)))
my.scores <- scoreItems(myKeys,myData)  #form several scales
my.scores  #show the highlights of the results
```

At this point you have had a chance to see the highlights of the *psych* package and to do some basic (and advanced) data analysis. You might find reading this entire vignette as well as the Overview Vignette to be helpful to get a broader understanding of what can be done in R using the *psych*. Remember that the help command (?) is available for every function. Try running the examples for each help page.

1 Overview of this and related documents

The *psych* package (Revelle, 2018) has been developed at Northwestern University since 2005 to include functions most useful for personality, psychometric, and psychological research. The package is also meant to supplement a text on psychometric theory (Revelle, prep), a draft of which is available at https://personality-project.org/r/book/.

Some of the functions (e.g., `read.file`, `read.clipboard`, `describe`, `pairs.panels`, `scatter.hist`, `error.bars`, `multi.hist`, `bi.bars`) are useful for basic data entry and descriptive analyses.

Psychometric applications emphasize techniques for dimension reduction including factor analysis, cluster analysis, and principal components analysis. The `fa` function includes six methods of *factor analysis* (*minimum residual*, *principal axis*, *alpha factoring*, *weighted least squares*, *generalized least squares* and *maximum likelihood* factor analysis). Principal Components Analysis (PCA) is also available through the use of the `principal` or `pca` functions. Rotations and transformations of these solutions are done by calling the many rotations available in the *GPArotation* (**?**).

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. These and several other criteria are included in the `nfactors` function. Two parameter Item Response Theory (IRT) 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 using `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). Higher order models can also be found using `fa.multi`.

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`, `scoreItems`, `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. The six measures of Intraclass correlation coefficients (`ICC`) discussed by Shrout and Fleiss (1979) are also available.

For data with a a multilevel structure (e.g., items within subjects across time, or items within subjects across groups), the `describeBy`, `statsBy` functions will give basic descriptives by group. `StatsBy` also will find within group (or subject) correlations as well as the between group correlation.

`multilevel.reliability` (`mlr`) will find various generalizability statistics for subjects over time and items. `mlPlot` will graph items over for each subject, `mlArrange` converts wide data frames to long data frames suitable for multilevel modeling.

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

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 `https://personality-project.org/r/psych_manual.pdf`). The vignette, psych for sem, at `https://personalty-project.org/r/psych_for_sem.pdf`, discusses how to use psych as a front end to the *sem* package of John Fox (Fox et al., 2012). (The vignette is also available at `https://personality-project.org/r/psych/vignettes/psych_for_sem.pdf`).

In addition, there are a growing number of "HowTo"s at the personality project. Currently these include:

1. An introduction (vignette) of the *psych* package
2. An overview (vignette) of the *psych* package
3. Installing R and some useful packages
4. Using R and the *psych* package to find $omega_h$ and ω_t.
5. Using R and the *psych* for factor analysis and principal components analysis.
6. Using the `scoreItems` function to find scale scores and scale statistics.
7. Using `mediate` and `setCor` to do mediation, moderation and regression analysis.

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 `https://personalitytheory.org/r/r.short.html`. For an *introduction to psychometric theory with applications in R*, see the draft chapters at `https://personality-project.org/r/book`).

2 Getting started

Some of the functions described in the Overview Vignette require other packages. This is not the case for the functions listed in this Introduction. 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. 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.

The "Psychometrics" task view will install a large number of useful packages. To install the bare minimum for the examples in this vignette, it is necessary to install just 3 packages:

R code

```
install.packages(list(c("GPArotation","mnormt")
```

Alternatively, many packages for psychometric can be downloaded at once using the "Psychometrics" task view:

R code

```
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. An alternative is to use "dot" output of commands for any external graphics package that uses the dot language.

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:

R code

```
library(psych)
library(psychTools)
```

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: e.g.,

R code

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

3.1 Getting the data by using read.file

Although many find copying the data to the clipboard and then using the `read.clipboard` functions (see below), a helpful alternative is to read the data in directly. This can be done using the `read.file` function which calls `file.choose` to find the file and then based upon the suffix of the file, chooses the appropriate way to read it. For files with suffixes of .text, .txt, .TXT, .csv, ,dat, .data, .sav, .xpt, .XPT, .r, .R, .rds, .Rds, .rda, .Rda, .rdata, Rdata, or .RData, the file will be read correctly.

R code

```
my.data <- read.file()
```

If the file contains Fixed Width Format (fwf) data, the column information can be specified with the widths command.

R code

```
my.data <- read.file(widths = c(4,rep(1,35)) #will read in a file without a header row
# and 36 fields, the first of which is 4 colums, the rest of which are 1 column each.
```

If the file is a .RData file (with suffix of .RData, .Rda, .rda, .Rdata, or .rdata) the object will be loaded. Depending what was stored, this might be several objects. If the file is a .sav file from SPSS, it will be read with the most useful default options (converting the file to a data.frame and converting character fields to numeric). Alternative options may be specified. If it is an export file from SAS (.xpt or .XPT) it will be read. .csv files (comma separated files), normal .txt or .text files, .data, or .dat files will be read as well. These are assumed to have a header row of variable labels (header=TRUE). If the data do not have a header row, you must specify read.file(header=FALSE).

To read SPSS files and to keep the value labels, specify use.value.labels=TRUE.

R code

```
#this will keep the value labels for .sav files
 my.spss  <- read.file(use.value.labels=TRUE)
```

3.2 Data input from the clipboard

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 (e.g., copied directly from an Excel file).

`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

R code

```
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.

R code

```
> 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).

R code

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

3.3 Basic descriptive statistics

Once the data are read in, then `describe` or `describeBy` 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, although the statistics are meaningless, will be treated as if numeric (based upon the categorical coding of the data), and will be flagged with an *.

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

R code

```
> library(psych)   #need to make psych active the first time you call it
> library(psychTools)  #additional tools and data are here
> data(sat.act)
> describe(sat.act)  #basic descriptive statistics
```

```
          vars   n   mean     sd median trimmed    mad min max range  skew kurtosis   se
gender       1 700   1.65   0.48      2    1.68   0.00   1   2     1 -0.61    -1.62 0.02
education    2 700   3.16   1.43      3    3.31   1.48   0   5     5 -0.68    -0.07 0.05
age          3 700  25.59   9.50     22   23.86   5.93  13  65    52  1.64     2.42 0.36
ACT          4 700  28.55   4.82     29   28.84   4.45   3  36    33 -0.66     0.53 0.18
SATV         5 700 612.23 112.90    620  619.45 118.61 200 800   600 -0.64     0.33 4.27
SATQ         6 687 610.22 115.64    620  617.25 118.61 200 800   600 -0.59    -0.02 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 5). By setting skew=FALSE and ranges=FALSE, the output is limited to the most basic statistics. Here we use formula mode.

R code

```
>  #basic descriptive statistics by a grouping variable.
> describeBy(sat.act ~ gender,skew=FALSE,ranges=FALSE)
```

```
 Descriptive statistics by group
gender: 1
          vars   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
------------------------------------------------------------------------
gender: 2
          vars   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 `describeBy` function can be forced into a matrix form for easy analysis by other programs. In addition, describeBy can group by several grouping variables at the same time.

R code

```
> sa.mat <-  describeBy(sat.act ~ gender + education,
+  skew=FALSE,ranges=FALSE,mat=TRUE)
> headTail(sa.mat)
```

```
        item group1 group2 vars   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
...      <NA>   <NA>   <NA>  ... ...    ...    ...   ...
SATQ9     69      1      4    6  51  635.9 104.12 14.58
SATQ10    70      2      4    6  86 597.59 106.24 11.46
SATQ11    71      1      5    6  46 657.83  89.61 13.21
SATQ12    72      2      5    6  93 606.72 105.55 10.95
```

3.3.1 Outlier detection using `outlier`

One way to detect unusual data is to consider how far each data point is from the multivariate centroid of the data. That is, find the squared Mahalanobis distance for each data point and then compare these to the expected values of χ^2. This produces a Q-Q (quantle-quantile) plot with the n most extreme data points labeled (Figure 1). The outlier values are in the vector d2.

3.3.2 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.

R code

```
> png( 'outlier.png' )
> d2 <- outlier(sat.act,cex=.8)
> dev.off()
```

```
null device
          1
```

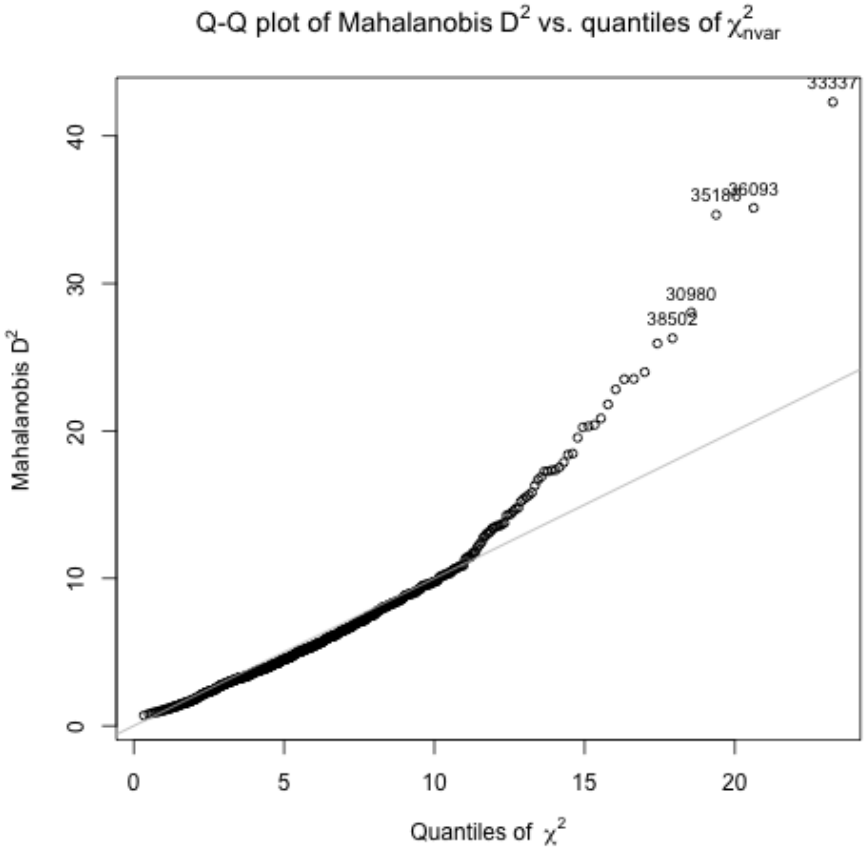

Figure 1: Using the `outlier` function to graphically show outliers. The y axis is the Mahalanobis D^2, the X axis is the distribution of χ^2 for the same number of degrees of freedom. The outliers detected here may be shown graphically using `pairs.panels` (see 2, and may be found by sorting d2.

Consider a data set of 10 rows of 12 columns with values from 1 - 120. All values of columns 3 - 5 that are less than 30, 40, or 50 respectively, or greater than 70 in any of the three columns 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).

R code

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

```
      V1  V2 V3 V4 V5  V6  V7  V8  V9 V10
 [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 NA NA  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 decreased, 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.

3.3.3 Recoding categorical variables into dummy coded variables

Sometimes categorical variables (e.g., college major, occupation, ethnicity) are to be analyzed using correlation or regression. To do this, one can form "dummy codes" which are merely binary variables for each category. This may be done using `dummy.code`. Subsequent analyses using these dummy coded variables may be using `biserial` or point biserial (regular Pearson r) to show effect sizes and may be plotted in e.g., `spider` plots.

Alternatively, sometimes data were coded originally as categorical (Male/Female, High School, some College, in college, etc.) and you want to convert these columns of data to numeric. This is done by `char2numeric`.

3.4 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.by will show group means with 95% confidence boundaries. By default, error.bars.by and error.bars will show "cats eyes" to graphically show the confidence limits (Figure 5) This may be turned off by specifying eyes=FALSE. densityBy or violinBy may be used to show the distribution of the data in "violin" plots (Figure 4). (These are sometimes called "lava-lamp" plots.)

3.4.1 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 x and y variables is also drawn. The x axis in each scatter plot represents the column variable, the y axis the row variable (Figure 2). When plotting many subjects, it is both faster and cleaner to set the plot character (pch) to be '.'. (See Figure 2 for an example.)

pairs.panels will show the pairwise scatter plots of all the variables as well as histograms, locally smoothed regressions, and the Pearson correlation. When plotting many data points (as in the case of the sat.act data, it is possible to specify that the plot character is a period to get a somewhat cleaner graphic. However, in this figure, to show the outliers, we use colors and a larger plot character. If we want to indicate 'significance' of the correlations by the conventional use of 'magic astricks' we can set the stars=TRUE option.

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.

Yet another demonstration of pairs.panels is useful when you have many subjects and want to show the density of the distributions. To do this we will use the make.keys and scoreItems functions (discussed in the second vignette) to create scales measuring Energetic Arousal, Tense Arousal, Positive Affect, and Negative Affect (see the msq help file). We then show a pairs.panels scatter plot matrix where we smooth the data points and show the density of the distribution by color.

R code

```
> keys <- make.keys(msq[1:75],list(
+ EA = c("active", "energetic", "vigorous", "wakeful", "wide.awake", "full.of.pep",
+        "lively", "-sleepy", "-tired", "-drowsy"),
+  TA =c("intense", "jittery", "fearful", "tense", "clutched.up", "-quiet", "-still",
+        "-placid", "-calm", "-at.rest") ,
+ PA =c("active", "excited", "strong", "inspired", "determined", "attentive",
+           "interested", "enthusiastic", "proud", "alert"),
```

R code

```
> png( 'pairspanels.png' )
> sat.d2 <- data.frame(sat.act,d2) #combine the d2 statistics from before with the sat.act data.frame
> pairs.panels(sat.d2,bg=c("yellow","blue")[(d2 > 25)+1],pch=21,stars=TRUE)
> dev.off()
```

```
null device
          1
```

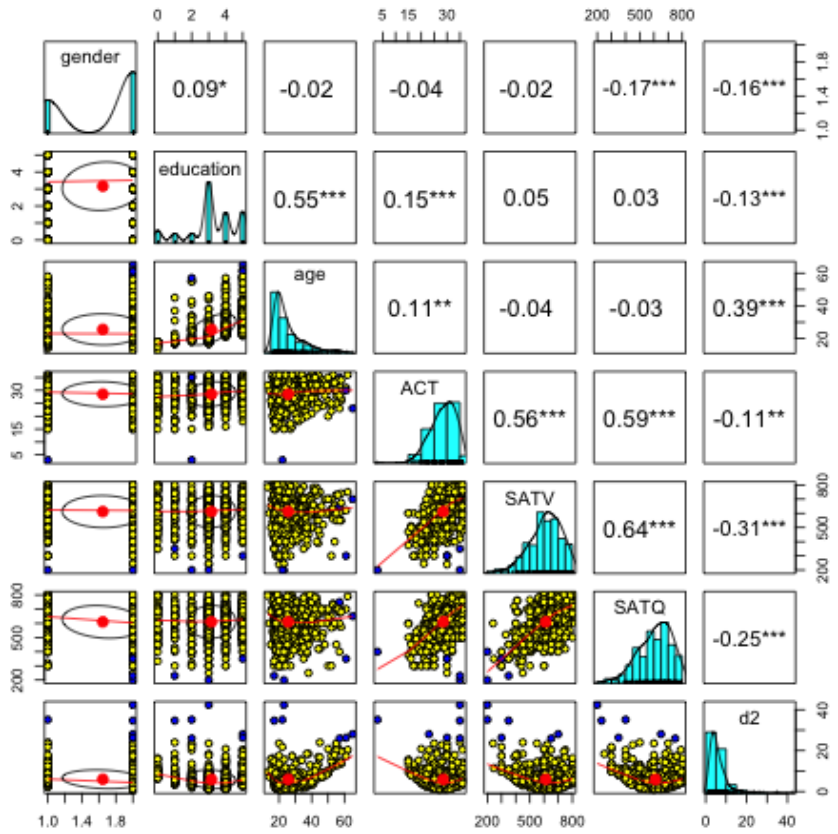

Figure 2: Using the `pairs.panels` function to graphically show relationships. The x axis in each scatter plot represents the column variable, the y axis the row variable. Note the extreme outlier for the ACT. If the plot character were set to a period (pch='.') it would make a cleaner graphic, but in to show the outliers in color we use the plot characters 21 and 22.

R code

```
> png('affect.png')
> pairs.panels(affect[14:17],bg=c("red","black","white","blue")[affect$Film],pch=21,
+      main="Affect varies by movies ")
> dev.off()
```

```
null device
          1
```

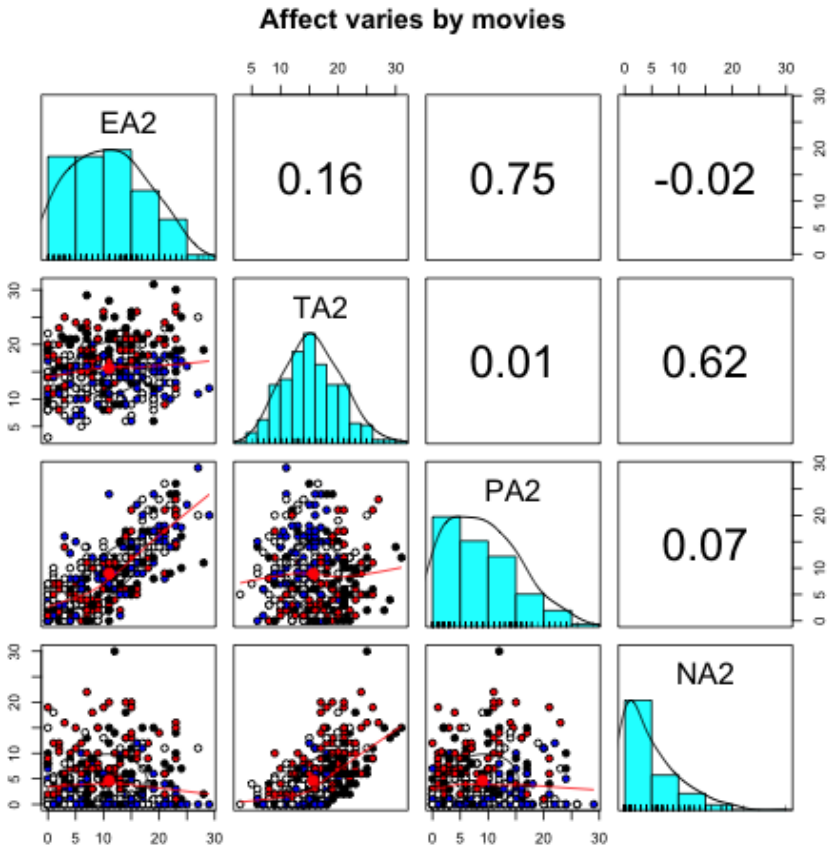

Figure 3: Using the `pairs.panels` function to graphically show relationships. The x axis in each scatter plot represents the column variable, the y axis the row variable. The coloring represent four different movie conditions.

```
+ NAf =c("jittery", "nervous", "scared", "afraid", "guilty", "ashamed", "distressed",
+          "upset", "hostile", "irritable" )) )
> scores <- scoreItems(keys,msq[,1:75])
> #png('msq.png')
> # pairs.panels(scores$scores,smoother=TRUE,
> #  main ="Density distributions of four measures of affect" )
>
> #dev.off()
```

Using the `pairs.panels` function to graphically show relationships. (Not shown in the interests of space.) The x axis in each scatter plot represents the column variable, the y axis the row variable. The variables are four measures of motivational state for 3896 participants. Each scale is the average score of 10 items measuring motivational state. Compare this a plot with smoother set to FALSE.

3.4.2 Density or violin plots

Graphical presentation of data may be shown using box plots to show the median and 25th and 75th percentiles. A powerful alternative is to show the density distribution using the `violinBy` function (Figure 4) or the more conventional density plot for multiple groups (Figure 10 .

R code

```
> png('violin.png')
> data(sat.act)
> violinBy(SATV+SATQ ~ gender, data=sat.act,grp.name=cs(Verbal.M,Verbal.F, Quan.M,Quant.F),  main="Density Plot by gende
> dev.off()
```

```
null device
          1
```

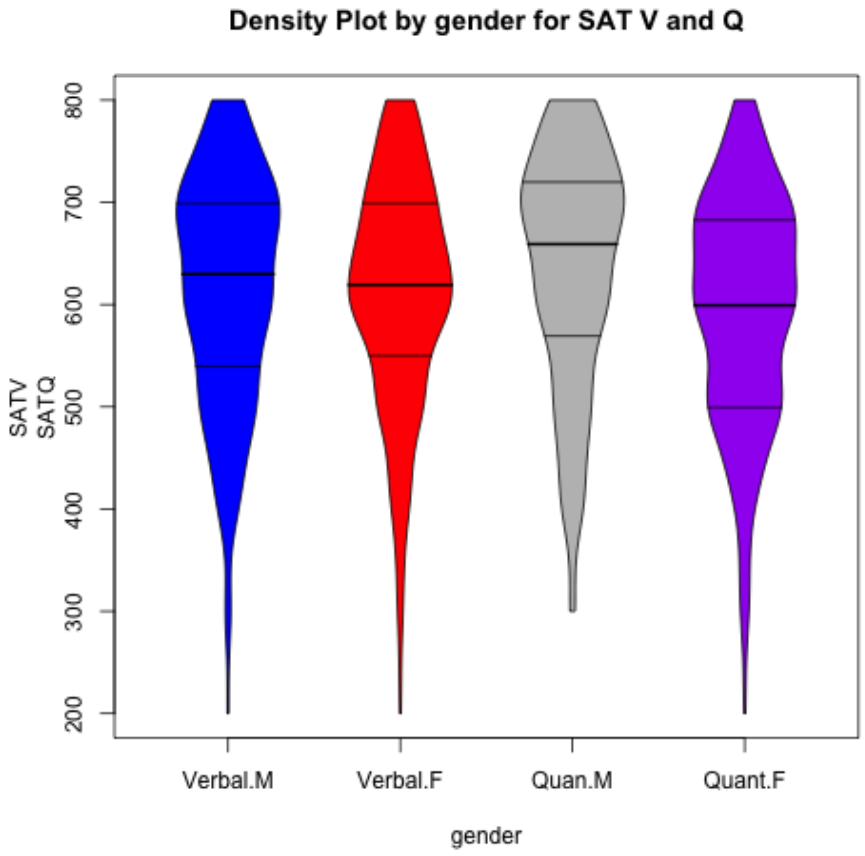

Figure 4: Using the `violinBy` function to show the distribution of SAT V and Q for males and females. The plot shows the medians, and 25th and 75th percentiles, as well as the entire range and the density distribution.

3.4.3 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`, `error.bars.by`, `error.bars.tab`, and `error.crosses`.

`error.bars` show the 95 % confidence intervals for each variable in a data frame or matrix. These errors are based upon normal theory and the standard errors of the mean. Alternative options include +/- one standard deviation or 1 standard error. If the data are repeated measures, the error bars will be reflect the between variable correlations. By default, the confidence intervals are displayed using a "cats eyes" plot which emphasizes the distribution of confidence within the confidence interval.

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

`error.bars.tab` draws bar graphs from tabular data with error bars based upon the standard error of proportion ($\sigma_p = \sqrt{pq/N}$)

`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 5). 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.

R code

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

Figure 5: 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. The "cats eyes" show the distribution of the confidence.

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 7) can be very misleading in that the scale is arbitrary. Go to a discussion of the problems in presenting data this way at https://emdbolker.wikidot.com/blog:dynamite. In the example shown, note that the graph starts at 0, although is out of the range. This is a function of using bars, which always are assumed to start at zero. Consider other ways of showing your data.

R code

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

Figure 6: 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. Bar graphs start from 0.

3.4.4 Error bars for tabular data

However, it is sometimes useful to show error bars for tabular data, either found by the `table` function or just directly input. These may be found using the `error.bars.tab` function.

R code

```
> T <- with(sat.act,table(gender,education))
> rownames(T) <- c("M","F")
> error.bars.tab(T,way="both",ylab="Proportion of Education Level",xlab="Level of Education",
+ main="Proportion of sample by education level")
```

Figure 7: The proportion of each education level that is Male or Female. By using the way="both" option, the percentages and errors are based upon the grand total. Alternatively, way="columns" finds column wise percentages, way="rows" finds rowwise percentages. The data can be converted to percentages (as shown) or by total count (raw=TRUE). The function invisibly returns the probabilities and standard errors. See the help menu for an example of entering the data as a data.frame.

3.4.5 Two dimensional displays of means and errors

Yet another way to display data for different conditions is to use the `errorCrosses` function. For instance, the effect of various movies on both "Energetic Arousal" and "Tense Arousal" can be seen in one graph and compared to the same movie manipulations on "Positive Affect" and "Negative Affect". Note how Energetic Arousal is increased by three of the movie manipulations, but that Positive Affect increases following the Happy movie only.

R code

```
> op <- par(mfrow=c(1,2))
>   data(affect)
> colors <- c("black","red","white","blue")
>  films <- c("Sad","Horror","Neutral","Happy")
> affect.stats <- errorCircles("EA2","TA2",data=affect[-c(1,20)],group="Film",labels=films,
+ xlab="Energetic Arousal",   ylab="Tense Arousal",ylim=c(10,22),xlim=c(8,20),pch=16,
+ cex=2,colors=colors, main =' Movies effect on arousal')
>  errorCircles("PA2","NA2",data=affect.stats,labels=films,xlab="Positive Affect",
+   ylab="Negative Affect", pch=16,cex=2,colors=colors,  main ="Movies effect on affect")
> op <- par(mfrow=c(1,1))
```

Figure 8: The use of the `errorCircles` function allows for two dimensional displays of means and error bars. The first call to `errorCircles` finds descriptive statistics for the *affect* data.frame based upon the grouping variable of Film. These data are returned and then used by the second call which examines the effect of the same grouping variable upon different measures. The size of the circles represent the relative sample sizes for each group. The data are from the PMC lab and reported in Smillie et al. (2012).

3.4.6 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 9).

```
data(bfi)
```
R code

```
> png( 'bibars.png' )
> bi.bars(bfi,"age","gender",ylab="Age",main="Age by males and females")
> dev.off()
```

```
pdf
  2
```

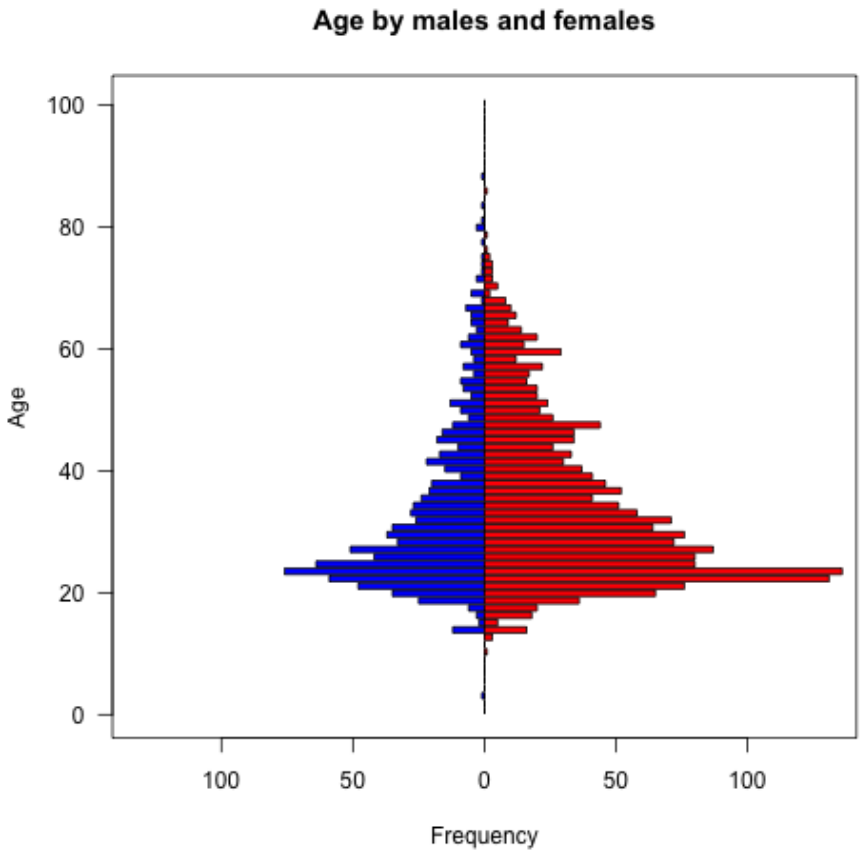

Figure 9: A bar plot of the age distribution for males and females shows the use of `bi.bars`. The data are males and females from 2800 cases collected using the *SAPA* procedure and are available as part of the `bfi` data set. An alternative way of displaying these data is in the `densityBy` in the next figure.

R code

```
> png('histo.png')
> data(sat.act)
> densityBy(bfi,"age",grp="gender")
> dev.off()
```

```
pdf
  2
```

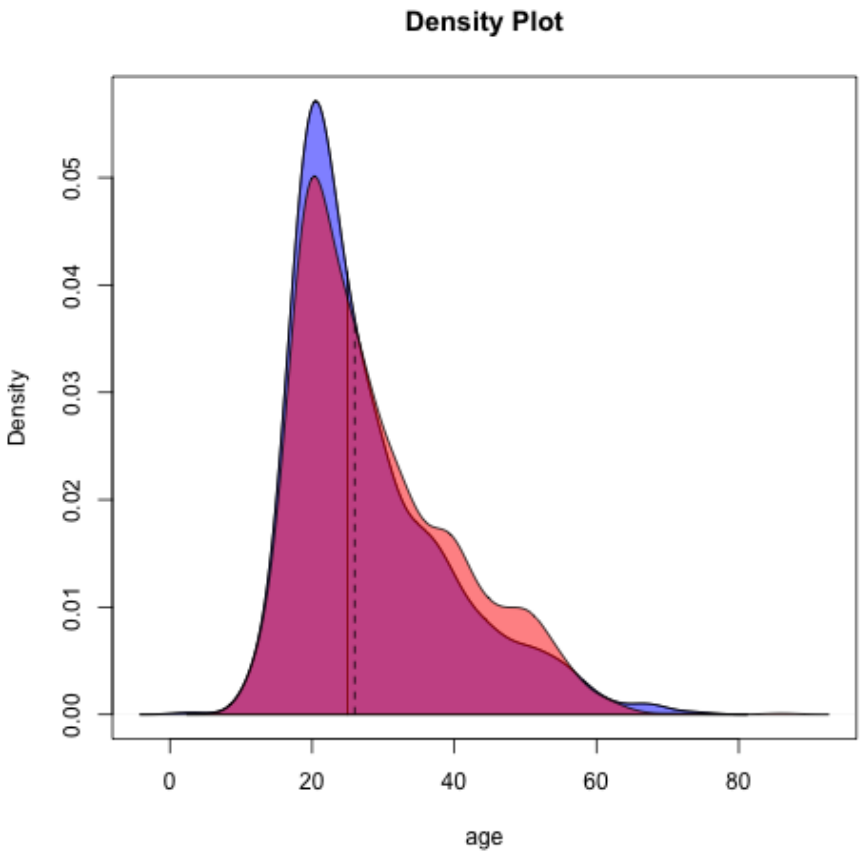

Figure 10: Using the `densitynBy` function to show the age distribution for males and females. The plot is a conventional density diagram for two two groups. Compare this to the `bi.bars` plot in the previous figure. By plotting densities, we can see that the males are slightly over represented in the younger ranges.

3.4.7 Correlational structure

There are many ways to display correlations. Tabular displays are probably the most common. The output from the `cor` function in core R is a rectangular matrix. `lowerMat` will round this to (2) digits and then display as a lower off diagonal matrix. `lowerCor` calls `cor` with *use='pairwise', method='pearson'* as default values and returns (invisibly) the full correlation matrix and displays the lower off diagonal matrix.

R code

```
> lowerCor(sat.act)
```

```
          gendr edctn age   ACT   SATV  SATQ
gender     1.00
education  0.09  1.00
age       -0.02  0.55  1.00
ACT       -0.04  0.15  0.11  1.00
SATV      -0.02  0.05 -0.04  0.56  1.00
SATQ      -0.17  0.03 -0.03  0.59  0.64  1.00
```

When comparing results from two different groups, it is convenient to display them as one matrix, with the results from one group below the diagonal, and the other group above the diagonal. Use `lowerUpper` to do this:

R code

```
> female <- subset(sat.act,sat.act$gender==2)
>  male <- subset(sat.act,sat.act$gender==1)
> lower <- lowerCor(male[-1])
```

```
          edctn age   ACT   SATV  SATQ
education  1.00
age        0.61  1.00
ACT        0.16  0.15  1.00
SATV       0.02 -0.06  0.61  1.00
SATQ       0.08  0.04  0.60  0.68  1.00
```

R code

```
> upper <- lowerCor(female[-1])
```

```
          edctn age   ACT   SATV  SATQ
education  1.00
age        0.52  1.00
ACT        0.16  0.08  1.00
SATV       0.07 -0.03  0.53  1.00
SATQ       0.03 -0.09  0.58  0.63  1.00
```

R code

```
> both <- lowerUpper(lower,upper)
> round(both,2)
```

```
          education   age  ACT  SATV  SATQ
education        NA  0.52 0.16  0.07  0.03
age            0.61    NA 0.08 -0.03 -0.09
ACT            0.16  0.15   NA  0.53  0.58
SATV           0.02 -0.06 0.61    NA  0.63
SATQ           0.08  0.04 0.60  0.68    NA
```

It is also possible to compare two matrices by taking their differences and displaying one (below the diagonal) and the difference of the second from the first above the diagonal:

R code

```
> diffs <-  lowerUpper(lower,upper,diff=TRUE)
> round(diffs,2)
```

```
          education   age  ACT  SATV SATQ
education        NA  0.09 0.00 -0.05 0.05
age            0.61    NA 0.07 -0.03 0.13
ACT            0.16  0.15   NA  0.08 0.02
SATV           0.02 -0.06 0.61    NA 0.05
SATQ           0.08  0.04 0.60  0.68   NA
```

3.4.8 Heatmap displays of correlational structure

Perhaps a better way to see the structure in a correlation matrix is to display a *heat map* of the correlations. This is just a matrix color coded 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 11) or a simulated data set of 24 variables with a circumplex structure (Figure 12). The color coding represents a "heat map" of the correlations, with darker shades of red representing stronger negative and darker shades of blue stronger positive correlations. As an option, the value of the correlation can be shown.

Yet another way to show structure is to use "spider" plots. Particularly if variables are ordered in some meaningful way (e.g., in a circumplex), a spider plot will show this structure easily. This is just a plot of the magnitude of the correlation as a radial line, with length ranging from 0 (for a correlation of -1) to 1 (for a correlation of 1). (See Figure 13).

3.5 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, Spearman, or Kendall) 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. Unfortunately,

R code

```
> png('corplot.png')
> corPlot(Thurstone,numbers=TRUE,upper=FALSE,diag=FALSE,main="9 cognitive variables from Thurstone")
> dev.off()
```

```
pdf
  2
```

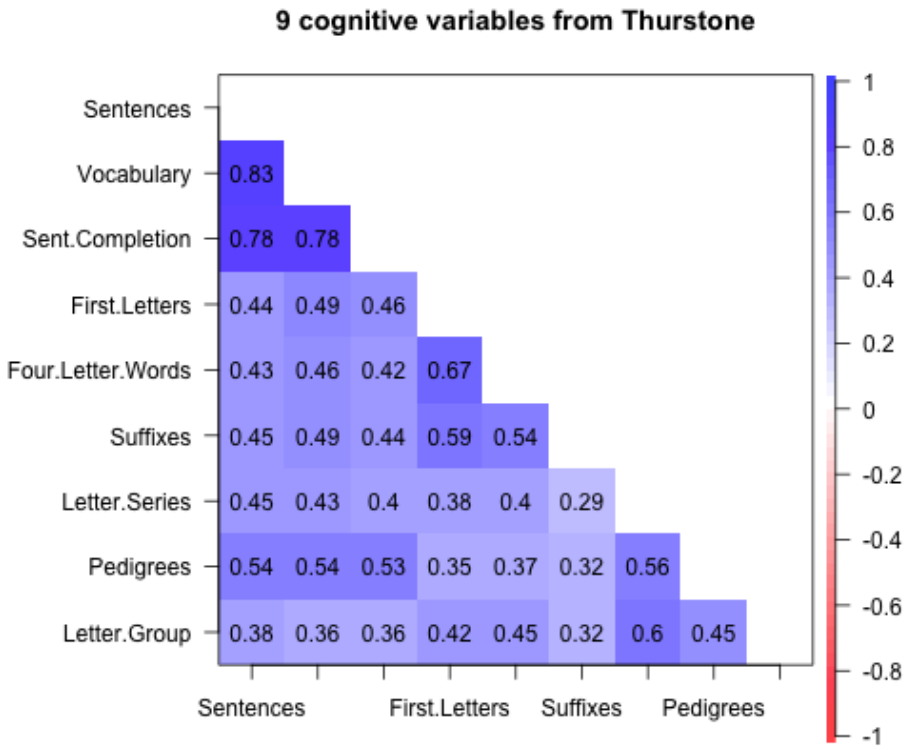

Figure 11: The structure of correlation matrix can be seen more clearly if the variables are grouped by factor and then the correlations are shown by color. By using the 'numbers' option, the values are displayed as well. By default, the complete matrix is shown. Setting upper=FALSE and diag=FALSE shows a cleaner figure.

R code

```
> png('circplot.png')
> circ <- sim.circ(24)
> r.circ <- cor(circ)
> corPlot(r.circ,main='24 variables in a circumplex')
> dev.off()
```

```
pdf
  2
```

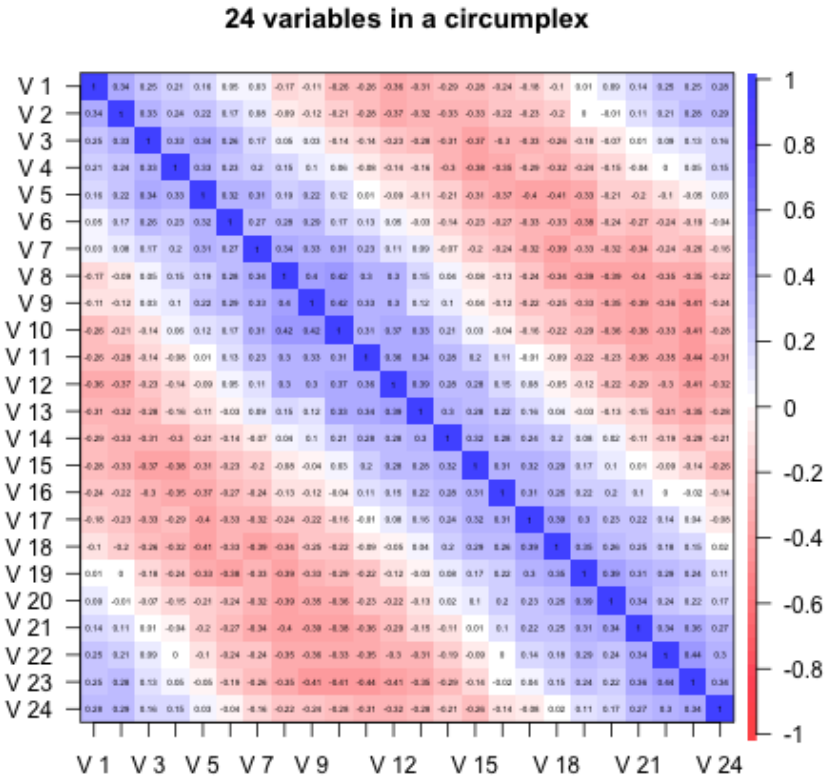

Figure 12: Using the corPlot function to show the correlations in a circumplex. Correlations are highest near the diagonal, diminish to zero further from the diagonal, and the increase again towards the corners of the matrix. Circumplex structures are common in the study of affect. For circumplex structures, it is perhaps useful to show the complete matrix.

R code

```
> png('spider.png')
> op<- par(mfrow=c(2,2))
> spider(y=c(1,6,12,18),x=1:24,data=r.circ,fill=TRUE,main="Spider plot of 24 circumplex variables")
> op <- par(mfrow=c(1,1))
> dev.off()
```

```
pdf
  2
```

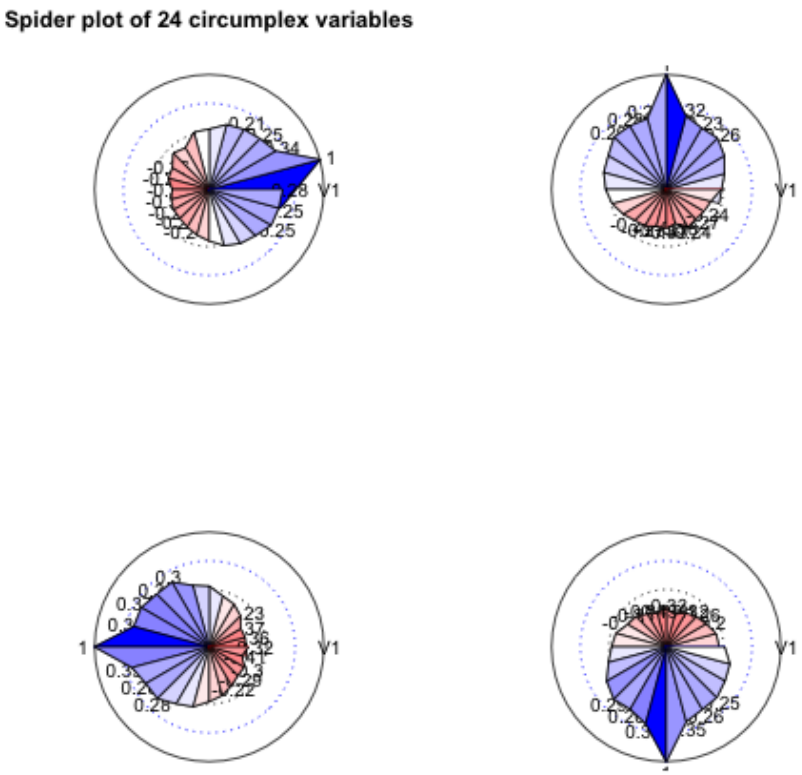

Figure 13: A spider plot can show circumplex structure very clearly. Circumplex structures are common in the study of affect.

these probability values have not been corrected for multiple comparisons and so should be taken with a great deal of salt. Thus, in `corr.test` and `corr.p` the raw probabilities are reported below the diagonal and the probabilities adjusted for multiple comparisons using (by default) the Holm correction are reported above the diagonal (Table 1). (See the `p.adjust` function for a discussion of Holm (1979) and other corrections.)

Table 1: The `corr.test` function reports correlations, cell sizes, and raw and adjusted probability values. `corr.p` reports the probability values for a correlation matrix. By default, the adjustment used is that of Holm (1979).

R code

```
> 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 values (Entries above the diagonal are adjusted for multiple tests.)
          gender education  age  ACT SATV SATQ
gender      0.00      0.17 1.00 1.00    1    0
education   0.02      0.00 0.00 0.00    1    1
age         0.58      0.00 0.00 0.03    1    1
ACT         0.33      0.00 0.00 0.00    0    0
SATV        0.62      0.22 0.26 0.00    0    0
SATQ        0.00      0.36 0.37 0.00    0    0

 To see confidence intervals of the correlations, print with the short=FALSE option
```

Testing the difference between any two correlations can be done using the `r.test` function. The function actually does four different tests (based upon an article by Steiger (1980), 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 code

```
> r.test(50,.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 code

```
> r.test(30,.4,.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 code

```
> r.test(103,.4,.5,.1)
```

```
Correlation tests
Call:[1] "r.test(n =  103 ,  r12 =  0.4 ,  r23 =  0.1 ,  r13 =  0.5 )"
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 code

```
> r.test(103,.5,.6,.7,.5,.5,.8)  #steiger Case B
```

```
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.

R code

```
> 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
```

3.6 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 14). This is also shown in terms of dichotomizing the bivariate normal density function using the `draw.cor` function. A simple generalization of this to the case of the multiple cuts is the `polychoric` correlation.

R code

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

Figure 14: 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.

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. It is found (laboriously) by optimizing the fit of the bivariate normal for various values of the correlation to the observed cell frequencies. In the interests of space, we do not show the next figure but it can be created by

```
draw.cor(expand=20,cuts=c(0,0))
```

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.

The correlation matrix resulting from a number of tetrachoric or polychoric correlation matrix sometimes will not be positive semi-definite. This will sometimes happen if the correlation matrix is formed by using pair-wise deletion of cases. The `cor.smooth` function will adjust the smallest eigen values of the correlation matrix to make them positive, rescale all of them to sum to the number of variables, and produce a "smoothed" correlation matrix. An example of this problem is a data set of `burt` which probably had a typo in the original correlation matrix. Smoothing the matrix corrects this problem.

4 Multilevel modeling

Correlations between individuals who belong to different natural groups (based upon e.g., ethnicity, age, gender, college major, or country) reflect an unknown mixture of the pooled correlation within each group as well as the correlation of the means of these groups. These two correlations are independent and do not allow inferences from one level (the group) to the other level (the individual). When examining data at two levels (e.g., the individual and by some grouping variable), it is useful to find basic descriptive statistics (means, sds, ns per group, within group correlations) as well as between group statistics (over all descriptive statistics, and overall between group correlations). Of particular use is the ability to decompose a matrix of correlations at the individual level into correlations within group and correlations between groups.

4.1 Decomposing data into within and between level correlations using `statsBy`

There are at least two very powerful packages (*nlme* and *multilevel*) which allow for complex analysis of hierarchical (multilevel) data structures. `statsBy` is a much simpler function to give some of the basic descriptive statistics for two level models. (*nlme* and *multilevel* allow for statistical inference, but the descriptives of `statsBy` are useful.)

This follows the decomposition of an observed correlation into the pooled correlation within groups (rwg) and the weighted correlation of the means between groups which is discussed by Pedhazur (1997) and by Bliese (2009) in the multilevel package.

$$r_{xy} = \eta_{x_{wg}} * \eta_{y_{wg}} * r_{xy_{wg}} + \eta_{x_{bg}} * \eta_{y_{bg}} * r_{xy_{bg}} \tag{1}$$

where r_{xy} is the normal correlation which may be decomposed into a within group and between group correlations $r_{xy_{wg}}$ and $r_{xy_{bg}}$ and η (eta) is the correlation of the data with the within group values, or the group means.

4.2 Generating and displaying multilevel data

`withinBetween` is an example data set of the mixture of within and between group correlations. The within group correlations between 9 variables are set to be 1, 0, and -1 while those between groups are also set to be 1, 0, -1. These two sets of correlations are crossed such that V1, V4, and V7 have within group correlations of 1, as do V2, V5 and V8, and V3, V6 and V9. V1 has a within group correlation of 0 with V2, V5, and V8, and a -1 within group correlation with V3, V6 and V9. V1, V2, and V3 share a between group correlation of 1, as do V4, V5 and V6, and V7, V8 and V9. The first group has a 0 between group correlation with the second and a -1 with the third group. See the help file for `withinBetween` to display these data.

`sim.multilevel` will generate simulated data with a multilevel structure.

The `statsBy.boot` function will randomize the grouping variable ntrials times and find the statsBy output. This can take a long time and will produce a great deal of output. This output can then be summarized for relevant variables using the `statsBy.boot.summary` function specifying the variable of interest.

Consider the case of the relationship between various tests of ability when the data are grouped by level of education (statsBy(sat.act)) or when affect data are analyzed within and between an affect manipulation (statsBy(affect)).

4.3 Factor analysis by groups

Confirmatory factor analysis comparing the structures in multiple groups can be done in the *lavaan* package. However, for exploratory analyses of the structure within each of multiple groups, the `faBy` function may be used in combination with the `statsBy` function. First run pfunstatsBy with the correlation option set to TRUE, and then run `faBy` on the resulting output.

R code

```
sb <- statsBy(bfi[c(1:25,27)], group="education",cors=TRUE)
faBy(sb,nfactors=5)  #find the 5 factor solution for each education level
```

5 Multiple Regression, mediation, moderation, and set correlations

The typical application of the `lm` function is to do a linear model of one Y variable as a function of multiple X variables. Because `lm` is designed to analyze complex interactions, it requires raw data as input. It is, however, sometimes convenient to do *multiple regression*

from a correlation or covariance matrix. This is done using the `setCor` which will work with either raw data, covariance matrices, or correlation matrices.

5.1 Multiple regression from data or correlation matrices

The `setCor` function will take a set of y variables predicted from a set of x variables, perhaps with a set of z covariates removed from both x and y. Consider the *Thurstone* correlation matrix and find the multiple correlation of the last five variables as a function of the first 4.

R code

```
> setCor(y = 5:9,x=1:4,data=Thurstone)
```

```
Call: setCor(y = 5:9, x = 1:4, data = Thurstone)

Multiple Regression from matrix input

 DV =  Four.Letter.Words
                slope  VIF
Sentences        0.09 3.69
Vocabulary       0.09 3.88
Sent.Completion  0.02 3.00
First.Letters    0.58 1.35

 Multiple Regression
                     R   R2  Ruw R2uw
Four.Letter.Words 0.69 0.48 0.59 0.34

 DV =  Suffixes
                slope  VIF
Sentences        0.07 3.69
Vocabulary       0.17 3.88
Sent.Completion  0.05 3.00
First.Letters    0.45 1.35

 Multiple Regression
            R  R2  Ruw R2uw
Suffixes 0.63 0.4 0.58 0.34

 DV =  Letter.Series
                slope  VIF
Sentences        0.25 3.69
Vocabulary       0.09 3.88
Sent.Completion  0.04 3.00
First.Letters    0.21 1.35

 Multiple Regression
                R   R2  Ruw R2uw
Letter.Series 0.5 0.25 0.49 0.24

 DV =  Pedigrees
                slope  VIF
Sentences        0.21 3.69
Vocabulary       0.16 3.88
```

```
Sent.Completion  0.21 3.00
First.Letters    0.08 1.35

 Multiple Regression
             R   R2  Ruw R2uw
Pedigrees 0.58 0.34 0.58 0.33

 DV =  Letter.Group
                slope  VIF
Sentences        0.20 3.69
Vocabulary      -0.02 3.88
Sent.Completion  0.08 3.00
First.Letters    0.31 1.35

 Multiple Regression
                R   R2  Ruw R2uw
Letter.Group 0.48 0.23 0.45  0.2

Various estimates of between set correlations
Squared Canonical Correlations
[1] 0.6280 0.1478 0.0076 0.0049

 Average squared canonical correlation =  0.2
 Cohen's Set Correlation R2 =  0.69
Unweighted correlation between the two sets =  0.73
```

By specifying the number of subjects in correlation matrix, appropriate estimates of standard errors, t-values, and probabilities are also found. The next example finds the regressions with variables 1 and 2 used as covariates. The $\hat{\beta}$ weights for variables 3 and 4 do not change, but the multiple correlation is much less. It also shows how to find the residual correlations between variables 5-9 with variables 1-4 removed.

R code

```
> sc <- setCor(y = 5:9,x=3:4,data=Thurstone,z=1:2)
> round(sc$residual,2)
```

```
                 Four.Letter.Words Suffixes Letter.Series Pedigrees Letter.Group
Four.Letter.Words              0.53     0.12          0.11      0.08         0.14
Suffixes                       0.12     0.61          0.01      0.03         0.04
Letter.Series                  0.11     0.01          0.79      0.31         0.39
Pedigrees                      0.08     0.03          0.31      0.70         0.23
Letter.Group                   0.14     0.04          0.39      0.23         0.79
```

5.2 Mediation and Moderation analysis

Although multiple regression is a straightforward method for determining the effect of multiple predictors ($x_{1,2,...i}$) on a criterion variable, y, some prefer to think of the effect of one predictor, x, as mediated by another variable, m (Preacher and Hayes, 2004). Thus, we we may find the indirect path from x to m, and then from m to y as well as the direct path from x to y. Call these paths a, b, and c, respectively. Then the indirect effect of x on y through m is just ab and the direct effect is c. Statistical tests of the ab effect are

best done by bootstrapping. This is discussed in detail in the "How To use `mediate` and `setCor` to do mediation, moderation and regression analysis tutorial.

Consider the example from Preacher and Hayes (2004) as analyzed using the `mediate` function and the subsequent graphic from `mediate.diagram`. The data are found in the example for `mediate`.

R code

```
> mediate.diagram(preacher)
```

Figure 15: A mediated model taken from Preacher and Hayes, 2004 and solved using the `mediate` function. The direct path from Therapy to Satisfaction has a an effect of .76, while the indirect path through Attribution has an effect of .33. Compare this to the normal regression graphic created by setCor.diagram.

R code

```
> preacher <- setCor(SATIS ~ THERAPY + ATTRIB,data =sobel,std=FALSE)
> setCor.diagram(preacher)
```

Figure 16: The conventional regression model for the Preacher and Hayes, 2004 data set solved using the `sector` function. Compare this to the previous figure.

- `setCor` will take raw data or a correlation matrix and find (and graph the path diagram) for multiple y variables depending upon multiple x variables.

  R code

  ```
  setCor(SATV +      SATQ ~ education + age, data = sat.act, std=TRUE)
  ```

- `mediate` will take raw data or a correlation matrix and find (and graph the path diagram) for multiple y variables depending upon multiple x variables mediated through a mediation variable. It then tests the mediation effect using a boot strap.

  R code

  ```
  mediate( SATV ~ education+ age + (ACT), data =sat.act,std=TRUE,n.iter=50)
  ```

- `mediate` will also take raw data and find (and graph the path diagram) a moderated multiple regression model for multiple y variables depending upon multiple x variables mediated through a mediation variable. It will form the product term either from the mean centered data or from the raw data. It then tests the mediation effect using a boot strap. The data set is taken from Garcia et al. (2010). The number of iterations

for the boot strap was set to 50 for speed. The default number of boot straps is 5000. See the help page for the `mediate` function for more details. For a much longer discussion of how to use the `mediate` function, see the "HowTo" Using `mediate` and `setCor` to do mediation, moderation and regression analysis.

```
Mediation/Moderation Analysis
Call: mediate(y = respappr ~ prot2 * sexism + (sexism), data = Garcia,
    n.iter = 50, main = "Moderated mediation (mean centered)")

The DV (Y) was  respappr . The IV (X) was  prot2 prot2*sexism . The mediating variable(s) =  sexism .

Total effect(c) of  prot2  on  respappr  =  1.46   S.E. =  0.22  t  =  6.77  df=  126   with p =  4.4e-10
Direct effect (c') of  prot2  on  respappr  removing  sexism  =  1.46   S.E. =  0.22  t  =  6.73  df=  125   with p =  5.5e
Indirect effect (ab) of  prot2  on  respappr  through  sexism   =  0
Mean bootstrapped indirect effect =  0  with standard error =  0.03  Lower CI =  -0.04    Upper CI =  0.07

Total effect(c) of  prot2*sexism  on  respappr  =  0.81   S.E. =  0.28  t  =  2.89  df=  126   with p =  0.0045
Direct effect (c') of  prot2*sexism  on  respappr  removing  sexism  =  0.81   S.E. =  0.28  t  =  2.87  df=  125   with p
Indirect effect (ab) of  prot2*sexism  on  respappr  through  sexism   =  0
Mean bootstrapped indirect effect =  0.01  with standard error =  0.04  Lower CI =  -0.07    Upper CI =  0.12
R = 0.54 R2 = 0.3   F = 17.53 on 3 and 125 DF   p-value:  1.91e-11

 To see the longer output, specify short = FALSE in the print statement or ask for the summary
```

Figure 17: Moderated multiple regression requires the raw data. By default, the data are mean centered before find the product term.

5.3 Set Correlation

An important generalization of multiple regression and multiple correlation is *set correlation* developed by Cohen (1982) and discussed by Cohen et al. (2003). Set correlation is a multivariate generalization of multiple regression and estimates the amount of variance shared between two sets of variables. Set correlation also allows for examining the relationship between two sets when controlling for a third set. This is implemented in the `setCor` function. Set correlation is

$$R^2 = 1 - \prod_{i=1}^{n}(1-\lambda_i)$$

where λ_i is the ith eigen value of the eigen value decomposition of the matrix

$$R = R_{xx}^{-1} R_{xy} R_{xx}^{-1} R_{xy}^{-1}.$$

Unfortunately, there are several cases where set correlation will give results that are much too high. This will happen if some variables from the first set are highly related to those in the second set, even though most are not. In this case, although the set correlation can be very high, the degree of relationship between the sets is not as high. In this case, an alternative statistic, based upon the average canonical correlation might be more appropriate.

`setCor` has the additional feature that it will calculate multiple and partial correlations from the correlation or covariance matrix rather than the original data.

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 `setCor`. Two things to notice. `setCor` works on the *correlation* or *covariance* or *raw data* matrix, and thus if using the correlation matrix, will report standardized or raw $\hat{\beta}$ weights. Secondly, it is possible to do several multiple regressions simultaneously. If the number of observations is specified, or if the analysis is done on raw data, statistical tests of significance are applied.

For this example, the analysis is done on the correlation matrix rather than the raw data.

R code

```
> 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 `setCor`.

R code

```
> #compare with sector
> setCor(c(4:6),c(1:3),C, n.obs=700)
```

```
Call: setCor(y = c(4:6), x = c(1:3), data = C, n.obs = 700)

Multiple Regression from raw data

 DV =  ACT
           slope   se     t     p lower.ci upper.ci  VIF
```

```
(Intercept)  0.00 0.07  0.00 1.000    -0.28     0.28 1.00
gender      -0.26 0.08 -3.17 0.087    -0.61     0.09 1.55
education    0.56 0.07  7.72 0.016     0.25     0.87 1.22
age         -0.64 0.08 -8.28 0.014    -0.97    -0.31 1.38

Residual Standard Error =  0.15  with  2  degrees of freedom

 Multiple Regression
   R   R2  Ruw R2uw Shrunken R2 SE of R2 overall F df1 df2     p
ACT 1 0.99 0.98 0.96        0.98        0     76.34   3   2 0.013

 DV =  SATV
            slope   se     t     p lower.ci upper.ci  VIF
(Intercept)  0.00 0.07  0.00 1.000    -0.31     0.31 1.00
gender       0.10 0.09  1.11 0.380    -0.28     0.48 1.55
education    0.72 0.08  9.21 0.012     0.39     1.06 1.22
age         -0.82 0.08 -9.76 0.010    -1.18    -0.46 1.38

Residual Standard Error =  0.16  with  2  degrees of freedom

 Multiple Regression
       R   R2  Ruw R2uw Shrunken R2 SE of R2 overall F df1 df2      p
SATV 0.99 0.99 0.89 0.79        0.97        0     65.18   3   2 0.0151

 DV =  SATQ
            slope   se     t     p lower.ci upper.ci  VIF
(Intercept)  0.00 0.04  0.00 1.000    -0.19     0.19 1.00
gender      -0.52 0.05 -9.73 0.010    -0.76    -0.29 1.55
education    0.40 0.05  8.32 0.014     0.19     0.60 1.22
age         -0.47 0.05 -9.24 0.012    -0.69    -0.25 1.38

Residual Standard Error =  0.1  with  2  degrees of freedom

 Multiple Regression
     R R2 Ruw R2uw Shrunken R2 SE of R2 overall F df1 df2       p
SATQ 1  1   1 0.99        0.99        0    177.11   3   2 0.00562

Various estimates of between set correlations
Squared Canonical Correlations
[1] 1.000 0.988 0.013
Chisq of canonical correlations
[1] 32.517  4.433  0.013

 Average squared canonical correlation =  0.67
 Cohen's Set Correlation R2 =  1
 Shrunken Set Correlation R2 =  1
 F and df of Cohen's Set Correlation  -Inf 12 -12.94
Unweighted correlation between the two sets =  0.98
```

Note that the `setCor` analysis also reports the amount of shared variance between the predictor set and the criterion (dependent) set. This set correlation is symmetric. That is, the R^2 is the same independent of the direction of the relationship.

6 Converting output to APA style tables using LaTeX

Although for most purposes, using the *Sweave* or *KnitR* packages produces clean output, some prefer output pre formatted for APA style tables. This can be done using the *xtable* package for almost anything, but there are a few simple functions in *psych* for the most common tables. `fa2latex` will convert a factor analysis or components analysis output to a LaTeXtable, `cor2latex` will take a correlation matrix and show the lower (or upper diagonal), `irt2latex` converts the item statistics from the `irt.fa` function to more convenient LaTeXoutput, and finally, `df2latex` converts a generic data frame to LaTeX.

An example of converting the output from `fa` to LaTeXappears in Table 2.

Table 2: fa2latex
A factor analysis table from the psych package in R

Variable	MR1	MR2	MR3	h2	u2	com
Sentences	0.91	-0.04	0.04	0.82	0.18	1.01
Vocabulary	0.89	0.06	-0.03	0.84	0.16	1.01
Sent.Completion	0.83	0.04	0.00	0.73	0.27	1.00
First.Letters	0.00	0.86	0.00	0.73	0.27	1.00
4.Letter.Words	-0.01	0.74	0.10	0.63	0.37	1.04
Suffixes	0.18	0.63	-0.08	0.50	0.50	1.20
Letter.Series	0.03	-0.01	0.84	0.72	0.28	1.00
Pedigrees	0.37	-0.05	0.47	0.50	0.50	1.93
Letter.Group	-0.06	0.21	0.64	0.53	0.47	1.23
SS loadings	2.64	1.86	1.5			
MR1	1.00	0.59	0.54			
MR2	0.59	1.00	0.52			
MR3	0.54	0.52	1.00			

7 Miscellaneous functions

A number of functions have been developed for some very specific problems that don't fit into any other category. The following is an incomplete list. Look at the *Index* for *psych* for a list of all of the functions.

`block.random` Creates a block randomized structure for n independent variables. Useful for teaching block randomization for experimental design.

`df2latex` is useful for taking tabular output (such as a correlation matrix or that of `describe` and converting it to a LaTeX table. May be used when Sweave is not convenient.

`cor2latex` Will format a correlation matrix in APA style in a LaTeX table. See also `fa2latex` and `irt2latex`.

`cosinor` One of several functions for doing *circular statistics*. This is important when studying mood effects over the day which show a diurnal pattern. See also `circadian.mean`, `circadian.cor` and `circadian.linear.cor` for finding circular means, circular correlations, and correlations of circular with linear data.

`fisherz` Convert a correlation to the corresponding Fisher z score.

`geometric.mean` also `harmonic.mean` find the appropriate mean for working with different kinds of data.

`ICC` and `cohen.kappa` are typically used to find the reliability for raters.

`headtail` combines the `head` and `tail` functions to show the first and last lines of a data set or output.

`topBottom` Same as headtail. Combines the `head` and `tail` functions to show the first and last lines of a data set or output, but does not add ellipsis between.

`mardia` calculates univariate or multivariate (Mardia's test) skew and kurtosis for a vector, matrix, or data.frame

`p.rep` finds the probability of replication for an F, t, or r and estimate effect size.

`partial.r` partials a y set of variables out of an x set and finds the resulting partial correlations. (See also `set.cor`.)

`rangeCorrection` will correct correlations for restriction of range.

`reverse.code` will reverse code specified items. Done more conveniently in most *psych* functions, but supplied here as a helper function when using other packages.

`superMatrix` Takes two or more matrices, e.g., A and B, and combines them into a "Super matrix" with A on the top left, B on the lower right, and 0s for the other two quadrants. A useful trick when forming complex keys, or when forming example problems.

8 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 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`), 135 items for 4,000 participants (`spi`) or 13 personality inventory scores (`epi.bfi`), and 16 multiple choice iq items (`iqitems`, `ability`). 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.

spi 135 personality items and 10 demographic items for 4,000 subjects are taken from the Synthetic Aperture Personality Assessment (*SAPA*) web based personality assessment project Revelle et al. (2016). These 135 items form part of the SAPA Personality Inventory **?**.

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.

iqitems 16 multiple choice ability items were included as part of the Synthetic Aperture Personality Assessment (*SAPA*) web based personality assessment project. The data from 1525 subjects are included here as a demonstration set for scoring multiple choice inventories and doing basic item statistics.

ability The same 16 items, converted to 0,1 scores are used for examples of various IRT procedures. These data are from the *International Cognitive Ability Resource* (ICAR) Condon & Revelle (2014) and were collected as part of the SAPA web based assessment https://sapa-project.org project Revelle et al. (2016).

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 introduce 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.

9 Development version and a users guide

The most recent development version is available as a source file at the repository maintained at `https://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 both Macs and PC, this version can be installed directly using the "other repository" option in the package installer. Make sure to specify type="source"

R code

```
> install.packages("psych", repos="https://personality-project.org/r", type="source")
```

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 https://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. To view the most recent news,

R code

```
> news(Version >= "2.1.1",package="psych")
```

10 Psychometric Theory

The *psych* package has been developed to help psychologists do basic research. Many of the functions were developed to supplement a book (https://personality-project.org/r/book An introduction to Psychometric Theory with Applications in R (Revelle, 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 https://personality-project.org/r/r.guide.html A short guide to R.

11 SessionInfo

This document was prepared using the following settings.

R code

```
> sessionInfo()
```

```
R version 4.0.3 (2020-10-10)
Platform: x86_64-apple-darwin17.0 (64-bit)
Running under: macOS Big Sur 10.16

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRblas.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.0/Resources/lib/libRlapack.dylib

locale:
[1] C

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

other attached packages:
[1] psychTools_2.1.1 psych_2.1.4.0

loaded via a namespace (and not attached):
 [1] lattice_0.20-41 grid_4.0.3      nlme_3.1-149    magrittr_1.5    evaluate_0.14   highr_0.8
 [7] stringi_1.4.6   tools_4.0.3     stringr_1.4.0   foreign_0.8-80  xfun_0.16       parallel_4.0.3
[13] compiler_4.0.3  mnormt_2.0.1    tmvnsim_1.0-2   knitr_1.29
```

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.

Bliese, P. D. (2009). Multilevel modeling in r (2.3) a brief introduction to r, the multilevel package and the nlme package.

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.

A. Bernaards Coen . & Robert I. Jennrich (2005). Gradient Projection Algorithms and Software for Arbitrary Rotation Criteria in Factor Analysis *Educational and Psychological Measurement*, *65*, 676-696.

Cohen, J. (1982). Set correlation as a general multivariate data-analytic method. *Multivariate Behavioral Research*, 17(3).

Cohen, J., Cohen, P., West, S. G., and Aiken, L. S. (2003). *Applied multiple regression/correlation analysis for the behavioral sciences*. L. Erlbaum Associates, Mahwah, N.J., 3rd ed edition.

Condon, D. M. & Revelle, W. (2014). The International Cognitive Ability Resource: Development and initial validation of a public-domain measure. *Intelligence*, *43*, 52–64.

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., Nie, Z., and Byrnes, J. (2012). *sem: Structural Equation Models*.

Garcia, D. M., Schmitt, M. T., Branscombe, N. R., and Ellemers, N. (2010). Women's reactions to ingroup members who protest discriminatory treatment: The importance of beliefs about inequality and response appropriateness. *European Journal of Social Psychology*, 40(5):733–745.

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.

Hayes, A. F. (2013). *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach*. Guilford Press, New York.

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.

Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):pp. 65–70.

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.

Jennrich, R. and Bentler, P. (2011). Exploratory bi-factor analysis. *Psychometrika*, pages 1–13. 10.1007/s11336-011-9218-4.

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.

Pedhazur, E. (1997). *Multiple regression in behavioral research: explanation and prediction*. Harcourt Brace College Publishers.

Preacher, K. J. and Hayes, A. F. (2004). SPSS and SAS procedures for estimating indirect effects in simple mediation models. *Behavior Research Methods, Instruments, & Computers*, 36(4):717–731.

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

Revelle, W. (2018). *psych: Procedures for Personality and Psychological Research*. Northwestern University, Evanston. R package version 1.8.6

Revelle, W. (in prep). *An introduction to psychometric theory with applications in R*. Springer.

Revelle, W. and Condon, D. M. (2014). Reliability. In Irwing, P., Booth, T., and Hughes, D., editors, *Wiley-Blackwell Handbook of Psychometric Testing*. Wiley-Blackwell (in press).

Revelle, W., Condon, D., and Wilt, J. (2011). Methodological advances in differential psychology. In Chamorro-Premuzic, T., Furnham, A., and von Stumm, S., editors, *Handbook of Individual Differences*, chapter 2, pages 39–73. Wiley-Blackwell.

Revelle, W., Condon, D. M., Wilt, J., French, J. A., Brown, A., & Elleman, L. G. (2016). Web and phone based data collection using planned missing designs. In N. G. Fielding, R. M. Lee, & G. Blank (Eds.), *SAGE Handbook of Online Research Methods* (2nd ed.). chapter 37, (pp. 578–595). Sage Publications, Inc.

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., Wilt, J., and Rosenthal, A. (2010). Personality and cognition: The personality-cognition link. In Gruszka, A., Matthews, G., and Szymura, B., editors, *Handbook of Individual Differences in Cognition: Attention, Memory and Executive Control*, chapter 2, pages 27–49. Springer.

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.

Smillie, L. D., Cooper, A., Wilt, J., and Revelle, W. (2012). Do extraverts get more bang for the buck? refining the affective-reactivity hypothesis of extraversion. *Journal of Personality and Social Psychology*, 103(2):306–326.

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.

Spearman, C. (1904). The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

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

Tal-Or, N., Cohen, J., Tsfati, Y., and Gunther, A. C. (2010). Testing causal direction in the influence of presumed media influence. *Communication Research*, 37(6):801–824.

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.

Zinbarg, R. E., Revelle, W., Yovel, I., and Li, W. (2005). Cronbach's α, Revelle's β, and McDonald's ω_H): Their relations with each other and two alternative conceptualizations of reliability. *Psychometrika*, 70(1):123–133.

Zinbarg, R. E., Yovel, I., Revelle, W., and McDonald, R. P. (2006). Estimating generalizability to a latent variable common to all of a scale's indicators: A comparison of estimators for ω_h. *Applied Psychological Measurement*, 30(2):121–144.

Index

r.test, 32
rangeCorrection, 44
rcorr, 28
read.clipboard, 3, 7, 10, 11
read.clipboard.csv, 11
read.clipboard.fwf, 11
read.clipboard.lower, 11
read.clipboard.tab, 3, 11
read.clipboard.upper, 11
read.file, 3, 7, 10
read.table, 11
Reise, 45
reverse.code, 44
sat.act, 12, 33, 41
scatter.hist, 7
schmid, 7, 9
score.multiple.choice, 7
scoreItems, 6–8, 16
scrub, 4, 13
sector, 39
set.cor, 44
setCor, 4, 5, 8, 37, 39–42
sim.multilevel, 36
spi, 45
spider, 15
stars, 16
StatsBy, 7
statsBy, 7, 35, 36
statsBy.boot, 36
statsBy.boot.summary, 36
structure.diagram, 8
superMatrix, 45