Psych: A Swiss Army Knife for psychology

William Revelle

Department of Psychology
Northwestern University
Evanston, Illinois USA

July, 2023
Outline

A Swiss Army knife for psychologists

Preliminaries

Data entry and description

Getting and cleaning data

Graphical displays

Multivariate analysis

The number of factors problem

Factors and clusters

Hierarchical models

True hierarchical

Seemingly hierarchical

Scale Construction

From raw data

From correlation matrices

The many forms of reliability

Seeing the objects

Alternative estimates of internal consistency: $\alpha$, $\beta$, $\omega_h$
The psych and psychTools packages

1. *psych* and *psychTools* have been developed to help further research in personality and individual differences.

2. Like a Swiss Army Knife, *psych* is not the best tool for anything, but it is a very helpful tool for many things.

3. *psych* is particularly aimed at the researcher in personality and individual differences.

4. Like core R *psych* helps researchers do open source science.

5. It is meant to be a relatively “light” package, in that it does not have many dependencies.

6. Unlike many other packages, the Help pages and Vignettes are fairly extensive (some would say wordy).
Installing the psych package (≥ 2.3.6)

# if you have not already done so, you first install the package
install.packages("psych",dependencies=TRUE)

library(psych)  # you need to do this every time you start R

# or automate the library(psych) call
# by creating and saving a function

.First <- function() {library(psych)}
quit() # with save option

# start R and psych will be automatically loaded
sessionInfo() # will tell you what version you are using

Good morning Bill.
Are you ready to have some fun?

> sessionInfo()
  version 4.3.1 (2023-06-16)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Ventura 13.4.1
To see the dependencies

```
sessionInfo()
```

```
sessionInfo()
R version 4.3.1 (2023-06-16)
Platform: aarch64-apple-darwin20 (64-bit)
Running under: macOS Ventura 13.4.1

Matrix products: default
BLAS: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib

Random number generation:
    RNG: Mersenne-Twister
    Normal: Inversion
    Sample: Rounding

locale:
    [1] en_US.UTF-8

time zone: America/Chicago
tzcode source: internal

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

other attached packages:
    [1] psychTools_2.3.6 psych_2.3.6

loaded via a namespace (and not attached):
    [1] compiler_4.3.1 tools_4.3.1 parallel_4.3.1 foreign_0.8-84 nlme_3.1-162 mnormt_2.1-1`
Show all the functions in the `psych` package

```r
objects("package:psych")
```

```r
[1] 
[2]  "%+%" 
[3]  "alpha2r" 
[4]  "bassAckward" 
[5]  "Bechtoldt.2" 
[6]  "bfi.keys" 
[7]  "biplot.psych" 
[8]  "bokc.table" 
[9]  "Chen" 
[10] "circ.simulation" 
[12] "circadian.sd" 
[13] "cluster.cor" 
[14] "cluster2keys" 
[15] "cohen.kappa" 
[16] "congeneric.sim" 
[17] "superMatrix" 
[18] "table2matrix" 
[19] "target2rot" 
[20] "test.all" 
[21] "tetrachoric" 
[22] "Thurstone.33G" 
[23] "Tucker" 
[24] "vgQ.bimin" 
[25] "violinBy" 
[26] "VSS.plot" 
[27] "West" 
[28] "winsor.sd" 
[29] "Yule" 
[30] "Yule2poly" 
[31] "YuleCor"
```
Objects in *psychTools*

```r
objects(package:psychTools)
[1]  "ability"       "ability.keys"               "affect"
[4]  "all.income"    "Athenstaedt"                 "Athenstaedt.dictionary"
[7] "Athenstaedt.keys" "bfi"                    "bfi.adjectives.dictionary"
[10] "bfi.adjectives.keys" "bfi.dictionary"     "bfi.keys"
[13] "big5.100.adjectives" "burt"               "blant"
[16] "blot"             "colom"                 "cities"
[19]  "city.location"   "colom.ed2"            "colom.ed0"
[22] "colom.ed1"       "cor2latex"             "colom.ed3"
[25] "combineMatrices"  "Damian"                "cubits"
[28]  "cushny"          "eminence"              "df2latex"
[31]    "dfOrder"        "epi"                   "
[34]  "epi.bfi"         "epi.bfi.adjectives.dictionary"    
[37]  "epiR"             "epi.dictionary"        "
[40]  "fileScan"        "fa2latex"               "epi.keys"
[43]  "galton"          "filesInfo"              "epiR"
[46]  "GERAS.keys"      "GERAS.dictionary"      "expresses"
[49]  "heights"         "GERAS.items"            "globalWarm"
[52]  "holzinger.swineford" "holzinger.dictionary"  "holzinger.raw"
[55]   "iqitems"        "ICC2latex"              "income"
[58]  "msq.keys"        "irt2latex"              "msg"
[61]  "omeg2latex"      "msqR"                  "neo"
[64]  "read.clipboard"  "peas"                  "Pollack"
[67]  "read.clipboard.lower"  "read.clipboard.csv"     "read.clipboard.fwf"
[70]  "read.file"       "read.clipboard.tab"      "read.clipboard.upper"
[73]  "recode"          "read.file.csv"          "read.https"
[76]  "sat.act"         "sai"                   "sai.dictionary"
[79]  "Spengler"        "Schutz"                "selectBy"
[82]  "spi.dictionary"  "Spengler.stat"        "splitBy"
[85]  "tai"             "spi.keys"               "spiR"
[88]  "vJoin"           "USAf"                  "splitBy"
[91]  "zola"            "write.file"             "veg"
[94]  "zola.dictionary" "write.file.csv"     "zola.keys"
[97]  "zola.keys"       "zola.dictionary"      "zola.keys"
```

Vignettes and How To’s

- How to’s and Vignettes
  1. An introduction to the `psych` package: Part I.
  2. Intro: Part II: Scale construction and psychometrics
  3. Installing Rand the `psych` package
  4. Using R and psych to find $\omega$
  5. How To: Use `psych` for factor analysis and data reduction
  6. Using R to score personality scales
  7. Using `psych` for regression and mediation analysis

- User manual for `psych`
- User manual for `psychTools`
- Help files for `psych`
- Help files for `psychTools`
Get your data: using read.file or read.clipboard

From a website: define the file name

```
fn <- "https://personality-project.org/r/datasets/glbwarm.sav"
fn #show it to check
fn
[1] "https://personality-project.org/r/datasets/glbwarm.sav"
mydata <- read.file(fn)
```

From a local file: find the file using read.file

```
> my.data <- read.file() #will open a search window, read the file
#depending upon the suffix, will read .sav, .csv, .txt, .rds, .rDa, etc.
```

From the clipboard: (first, go to the remote site, copy to the clipboard and then use the read.clipboard function).

```
mydata <- read.clipboard() #or
mydata <- read.clipboard.tab() #if an excel file
my.data <- read.clipboard.csv() #if a tab delimited file
```

(This example data set can also be accessed directly in glbwarm.)
```
R code

dim(mydata)  # how many rows and columns?
headTail(mydata)  # Show the top and bottom n rows and columns from c1 to c2
describe(mydata)  # basic descriptive statistics

dim(mydata)  # how many rows and columns?
[1] 815  7
> headTail(mydata)  # Show the top and bottom n rows and columns from c1 to c2
     govact  posemot  negemot ideology age  sex partyid
 1  3.6  3.67  4.67  6  61 0 2
 2  5 2.33  2.33 2  55 0 1
 3  6.6 2.33  3.67 1  85 1 1
 4  1  5  5  1  59 0 1
...
812  3.4 1 1 7  67 0 3
813  1.6 3.67 1.67 7  72 1 3
814  5.4 2.67 3.33 6  36 0 2
815  5.4 5.33 6  4  82 1 1
> describe(mydata)  # basic descriptive statistics
     vars n mean sd median trimmed mad min max range skew kurtosis se
govact     1 815 4.59 1.36  4.80  4.68  1.19  1 7  6 -0.63  0.22  0.05
posemot    2 815 3.13 1.35  3.00  3.11  1.48  1 6  5  0.09 -0.85  0.05
negemot    3 815 3.56 1.53  3.67  3.58  1.97  1 6  5 -0.15 -1.07  0.05
ideology   4 815 4.08 1.51  4.00  4.07  1.48  1 7  6  0.03 -0.43  0.05
age        5 815 49.54 16.33 51.00 49.66 19.27 17 87 70 -0.07 -1.03  0.57
sex        6 815 0.49 0.50  0.00  0.49  0.00  0 1  1  0.05 -2.00  0.02
partyid    7 815 1.88 0.87  2.00  1.85  1.48  1 3  2  0.23 -1.63  0.03
```
headTail of a bigger local file

```r
dim(msqR)
headTail(msqR, top=4, bottom=6, from=78, to=88)
```

```r
dim(msqR)
headTail(msqR, top=4, bottom=6, from=78, to=88)

[1] 6411 88

headTail(msqR, top=4, bottom=6, from=78, to=88)

<table>
<thead>
<tr>
<th>Lie</th>
<th>Sociability</th>
<th>Impulsivity</th>
<th>gender</th>
<th>TOD</th>
<th>drug</th>
<th>film</th>
<th>time</th>
<th>id</th>
<th>form</th>
<th>study</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>&lt;NA&gt;</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>9</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>1</td>
<td>&lt;NA&gt;</td>
<td>1</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>1</td>
<td>&lt;NA&gt;</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>11</td>
<td>4</td>
<td>2</td>
<td>9</td>
<td>2</td>
<td>&lt;NA&gt;</td>
<td>1</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

... ... ... ... ... ... ... ... ... ... ... ... ... ... ...

| 3941 | 2 | 9 | 6 | <NA> | 9 | 2 | 4 | 2 | 195 | 2 | XRAY |
| 3942 | 3 | 3 | 5 | <NA> | 9 | 1 | 2 | 2 | 196 | 2 | XRAY |
| 3943 | 2 | 7 | 2 | <NA> | 9 | 2 | 1 | 2 | 197 | 2 | XRAY |
| 3944 | 0 | 12| 5 | <NA> | 9 | 1 | 4 | 2 | 198 | 2 | XRAY |
| 3945 | 3 | 11| 3 | <NA> | 9 | 1 | 3 | 2 | 199 | 2 | XRAY |
| 3946 | 0 | 2 | 4 | <NA> | 9 | 2 | 4 | 2 | 200 | 2 | XRAY |
```

Notice that rowname although unique is not the line number
Descriptives by a grouping variable

**R code**

```
describeBy(mydata~sex)
```

<table>
<thead>
<tr>
<th>vars</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>govact</td>
<td>1</td>
<td>417</td>
<td>4.72</td>
<td>1.16</td>
<td>4.8</td>
<td>4.77</td>
<td>1.19</td>
<td>1</td>
<td>7</td>
<td>-0.52</td>
<td>0.55</td>
<td>0.06</td>
</tr>
<tr>
<td>posemot</td>
<td>2</td>
<td>417</td>
<td>3.03</td>
<td>1.39</td>
<td>3.0</td>
<td>3.00</td>
<td>1.48</td>
<td>1</td>
<td>6</td>
<td>0.17</td>
<td>-0.98</td>
<td>0.07</td>
</tr>
<tr>
<td>negemot</td>
<td>3</td>
<td>417</td>
<td>3.73</td>
<td>1.45</td>
<td>4.0</td>
<td>3.79</td>
<td>1.48</td>
<td>1</td>
<td>6</td>
<td>-0.26</td>
<td>-0.83</td>
<td>0.07</td>
</tr>
<tr>
<td>ideology</td>
<td>4</td>
<td>417</td>
<td>3.89</td>
<td>1.44</td>
<td>4.0</td>
<td>3.87</td>
<td>1.48</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>-0.29</td>
<td>0.07</td>
</tr>
<tr>
<td>age</td>
<td>5</td>
<td>417</td>
<td>46.90</td>
<td>14.95</td>
<td>44.0</td>
<td>46.77</td>
<td>16.31</td>
<td>18</td>
<td>83</td>
<td>0.13</td>
<td>-0.90</td>
<td>0.73</td>
</tr>
<tr>
<td>sex</td>
<td>6</td>
<td>417</td>
<td>0.00</td>
<td>0.00</td>
<td>0.0</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>NaN</td>
<td>NaN</td>
<td>0.00</td>
</tr>
<tr>
<td>partyid</td>
<td>7</td>
<td>417</td>
<td>1.79</td>
<td>0.86</td>
<td>2.0</td>
<td>1.74</td>
<td>1.48</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>-1.52</td>
<td>0.04</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>vars</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>govact</td>
<td>1</td>
<td>398</td>
<td>4.45</td>
<td>1.53</td>
<td>4.60</td>
<td>4.55</td>
<td>1.48</td>
<td>1</td>
<td>7</td>
<td>-0.56</td>
<td>-0.31</td>
<td>0.08</td>
</tr>
<tr>
<td>posemot</td>
<td>2</td>
<td>398</td>
<td>3.23</td>
<td>1.30</td>
<td>3.33</td>
<td>3.23</td>
<td>1.48</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>-0.68</td>
<td>0.07</td>
</tr>
<tr>
<td>negemot</td>
<td>3</td>
<td>398</td>
<td>3.37</td>
<td>1.59</td>
<td>3.67</td>
<td>3.36</td>
<td>1.98</td>
<td>1</td>
<td>6</td>
<td>5</td>
<td>-1.25</td>
<td>0.08</td>
</tr>
<tr>
<td>ideology</td>
<td>4</td>
<td>398</td>
<td>4.29</td>
<td>1.56</td>
<td>4.00</td>
<td>4.29</td>
<td>1.48</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>-0.56</td>
<td>0.08</td>
</tr>
<tr>
<td>age</td>
<td>5</td>
<td>398</td>
<td>52.30</td>
<td>17.25</td>
<td>55.00</td>
<td>52.85</td>
<td>19.27</td>
<td>17</td>
<td>87</td>
<td>70</td>
<td>-1.01</td>
<td>0.86</td>
</tr>
<tr>
<td>sex</td>
<td>6</td>
<td>398</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>NaN</td>
<td>NaN</td>
</tr>
<tr>
<td>partyid</td>
<td>7</td>
<td>398</td>
<td>1.98</td>
<td>0.87</td>
<td>2.00</td>
<td>1.98</td>
<td>1.48</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>-1.67</td>
<td>0.04</td>
</tr>
</tbody>
</table>
lowerCor is a nice example of the power of R to nest functions. It is just a call to cor with the use=“pairwise” option followed by a call to lowerMat which “prettifies” a correlation matrix.

R <- lowerCor(mydata)  #returns R invisibly

lowerCor(mydata)

govct  posmt  negmt  idlyg  age   sex   prtyd

govact   1.00
posemot  0.04  1.00
negemot  0.58  0.13  1.00
ideology -0.42 -0.03 -0.35  1.00
age      -0.10  0.04 -0.06  0.21  1.00
sex      -0.10  0.07 -0.12  0.13  0.17  1.00
partyid  -0.36 -0.04 -0.32  0.62  0.15  0.11  1.00

R  #show R (if you want to use it for something else. Note that it is not rounded.)

govact  1.00000000  0.04302895  0.57774582 -0.41831995 -0.09713873 -0.09861854 -0.36039647
posemot  0.04302895  1.00000000  0.12792202 -0.02937618  0.04235193  0.07429449 -0.03577099
negemot  0.57774582  0.12792202  1.00000000 -0.34878643 -0.05689493 -0.11735643 -0.32419141
ideology -0.41831995 -0.02937618 -0.34878643  1.00000000  0.21240565  0.13288895  0.61945381
age      -0.09713873  0.04235193 -0.05689493  0.21240565  1.00000000  0.16553039  0.15443184
sex      -0.09861854  0.07429449 -0.11735643  0.13288895  0.16553039  1.00000000  0.10875960
partyid  -0.36039647 -0.03577099 -0.32419141  0.61945381  0.15443184  0.10875960  1.00000000
**Correlations using** `corr.test`

```r
corr.test(mydata)
```

Call: `corr.test(x = mydata)`
Correlation matrix

```
govact  posemot  negemot ideology  age   sex  partyid
govact   1.00     0.04    0.58  -0.42 -0.10 -0.10 -0.36
posemot   0.04     1.00    0.13  -0.03  0.04  0.07 -0.04
negemot   0.58     0.13    1.00  -0.35 -0.06 -0.12 -0.32
ideology -0.42    -0.03   -0.35   1.00  0.21  0.13  0.62
age      -0.10     0.04   -0.06   0.21  1.00  0.17  0.15
sex      -0.10     0.07   -0.12   0.13  0.17  1.00  0.11
partyid  -0.36    -0.04   -0.32   0.62  0.15  0.11  1.00
```

Sample Size

```
[1] 815
```

Probability values (Entries above the diagonal are adjusted* for multiple tests.)

```
govact  posemot  negemot ideology  age   sex  partyid
govact   0.00     0.88     0.0   0.00  0.04  0.04  0.00
posemot   0.22     0.00     0.0   0.88  0.88  0.20  0.88
negemot   0.00     0.00     0.0   0.00  0.52  0.01  0.00
ideology  0.00     0.40     0.0   0.00  0.00  0.00  0.00
age       0.01     0.23     0.1   0.00  0.00  0.00  0.00
sex       0.00     0.03     0.0   0.00  0.00  0.00  0.02
partyid   0.00     0.31     0.0   0.00  0.00  0.00  0.00
```

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

*Adjustment using the Holm (1979) correction for multiple tests
long output from `corr.test` gives the normal theory CI

R code

```r
print(corr.test(mydata), short=FALSE)
```

Confidence intervals based upon normal theory. To get bootstrapped values, try `cor.ci`

<table>
<thead>
<tr>
<th></th>
<th>raw.lower</th>
<th>raw.r</th>
<th>raw.upper</th>
<th>raw.p</th>
<th>lower.adj</th>
<th>upper.adj</th>
</tr>
</thead>
<tbody>
<tr>
<td>govct-posmt</td>
<td>-0.03</td>
<td>0.04</td>
<td>0.11</td>
<td>0.22</td>
<td>-0.04</td>
<td>0.13</td>
</tr>
<tr>
<td>govct-negmt</td>
<td>0.53</td>
<td>0.58</td>
<td>0.62</td>
<td>0.00</td>
<td>0.50</td>
<td>0.64</td>
</tr>
<tr>
<td>govct-idlgy</td>
<td>-0.47</td>
<td>-0.42</td>
<td>-0.36</td>
<td>0.00</td>
<td>-0.50</td>
<td>-0.33</td>
</tr>
<tr>
<td>govct-age</td>
<td>-0.16</td>
<td>-0.10</td>
<td>-0.03</td>
<td>0.01</td>
<td>-0.19</td>
<td>0.00</td>
</tr>
<tr>
<td>govct-sex</td>
<td>-0.17</td>
<td>-0.10</td>
<td>-0.03</td>
<td>0.00</td>
<td>-0.19</td>
<td>0.00</td>
</tr>
<tr>
<td>govct-prtyd</td>
<td>-0.42</td>
<td>-0.36</td>
<td>-0.30</td>
<td>0.00</td>
<td>-0.45</td>
<td>-0.27</td>
</tr>
<tr>
<td>posmt-negmt</td>
<td>0.06</td>
<td>0.13</td>
<td>0.19</td>
<td>0.00</td>
<td>0.03</td>
<td>0.22</td>
</tr>
<tr>
<td>posmt-idlgy</td>
<td>-0.10</td>
<td>-0.03</td>
<td>0.04</td>
<td>0.40</td>
<td>-0.10</td>
<td>0.04</td>
</tr>
<tr>
<td>posmt-age</td>
<td>-0.03</td>
<td>0.04</td>
<td>0.11</td>
<td>0.23</td>
<td>-0.04</td>
<td>0.13</td>
</tr>
<tr>
<td>posmt-sex</td>
<td>0.01</td>
<td>0.07</td>
<td>0.14</td>
<td>0.03</td>
<td>-0.02</td>
<td>0.17</td>
</tr>
<tr>
<td>posmt-prtyd</td>
<td>-0.10</td>
<td>-0.04</td>
<td>0.03</td>
<td>0.31</td>
<td>-0.11</td>
<td>0.04</td>
</tr>
<tr>
<td>negmt-idlgy</td>
<td>-0.41</td>
<td>-0.35</td>
<td>-0.29</td>
<td>0.00</td>
<td>-0.44</td>
<td>-0.25</td>
</tr>
<tr>
<td>negmt-age</td>
<td>-0.13</td>
<td>-0.06</td>
<td>0.01</td>
<td>0.10</td>
<td>-0.15</td>
<td>0.03</td>
</tr>
<tr>
<td>negmt-sex</td>
<td>-0.18</td>
<td>-0.12</td>
<td>-0.05</td>
<td>0.00</td>
<td>-0.21</td>
<td>-0.02</td>
</tr>
<tr>
<td>negmt-prtyd</td>
<td>-0.38</td>
<td>-0.32</td>
<td>-0.26</td>
<td>0.00</td>
<td>-0.41</td>
<td>-0.23</td>
</tr>
<tr>
<td>idlgy-age</td>
<td>0.15</td>
<td>0.21</td>
<td>0.28</td>
<td>0.00</td>
<td>0.11</td>
<td>0.31</td>
</tr>
<tr>
<td>idlgy-sex</td>
<td>0.06</td>
<td>0.13</td>
<td>0.20</td>
<td>0.00</td>
<td>0.03</td>
<td>0.23</td>
</tr>
<tr>
<td>idlgy-prtyd</td>
<td>0.58</td>
<td>0.62</td>
<td>0.66</td>
<td>0.00</td>
<td>0.55</td>
<td>0.68</td>
</tr>
<tr>
<td>age-sex</td>
<td>0.10</td>
<td>0.17</td>
<td>0.23</td>
<td>0.00</td>
<td>0.06</td>
<td>0.26</td>
</tr>
<tr>
<td>age-prtyd</td>
<td>0.09</td>
<td>0.15</td>
<td>0.22</td>
<td>0.00</td>
<td>0.05</td>
<td>0.25</td>
</tr>
<tr>
<td>sex-prtyd</td>
<td>0.04</td>
<td>0.11</td>
<td>0.18</td>
<td>0.00</td>
<td>0.01</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Adjusted cis are given with Holm (1979) adjustment
Correlations with "magic astericks"

R code

```
print(corr.test(mydata)$stars, quote=FALSE)
```

<table>
<thead>
<tr>
<th></th>
<th>govact</th>
<th>posemot</th>
<th>negemot</th>
<th>ideology</th>
<th>age</th>
<th>sex</th>
<th>partyid</th>
</tr>
</thead>
<tbody>
<tr>
<td>govact</td>
<td>1***</td>
<td>0.04</td>
<td>0.58***</td>
<td>-0.42***</td>
<td>-0.1*</td>
<td>-0.1*</td>
<td>-0.36***</td>
</tr>
<tr>
<td>posemot</td>
<td>0.04</td>
<td>1***</td>
<td>0.13**</td>
<td>-0.03</td>
<td>0.04</td>
<td>0.07</td>
<td>-0.04</td>
</tr>
<tr>
<td>negemot</td>
<td>0.58***</td>
<td>0.13***</td>
<td>1***</td>
<td>-0.35***</td>
<td>-0.06</td>
<td>-0.12**</td>
<td>-0.32***</td>
</tr>
<tr>
<td>ideology</td>
<td>-0.42***</td>
<td>-0.03</td>
<td>-0.35***</td>
<td>1***</td>
<td>0.21***</td>
<td>0.13**</td>
<td>0.62***</td>
</tr>
<tr>
<td>age</td>
<td>-0.1**</td>
<td>0.04</td>
<td>-0.06</td>
<td>0.21***</td>
<td>1***</td>
<td>0.17***</td>
<td>0.15***</td>
</tr>
<tr>
<td>sex</td>
<td>-0.1**</td>
<td>0.07</td>
<td>-0.12***</td>
<td>0.13***</td>
<td>0.17***</td>
<td>1***</td>
<td>0.11*</td>
</tr>
<tr>
<td>partyid</td>
<td>-0.36***</td>
<td>-0.04</td>
<td>-0.32***</td>
<td>0.62***</td>
<td>0.15***</td>
<td>0.11**</td>
<td>1***</td>
</tr>
</tbody>
</table>

Once again, the p values above the diagonal are adjusted using the Holm (1979) correction for multiple tests.
**Graphical displays of data**

1. Can show basic means and ranges (error.bars and error.bars.by)
2. Can show correlations using pairs.panels, corPlot
3. densityBy for distribution information.
4. scatterHist to show bivariate plots with densities by group
5. cohen.d combines with error.dots to show effect sizes and confidence intervals
Showing group differences using `error.bars.by`

95% confidence limits

```r
error.bars.by(mydata[-5], "sex", by.var=FALSE, ylab="score", xlab="Variable")
```
Global warming data set – age by sex

Age distributions for Global Warming data

- **sex 0**
- **sex 1**
Finding the effect of caffeine on mood

1. What is the effect of caffeine on motivational/emotional state?

2. Motivational State Questionnaire (Revelle & Anderson, 1998) was given to participants before and after caffeine/movie/stress manipulations.

3. Data are pooled over 10 years of data (> 50 studies) in the PMC lab and available as the msqR data set.

4. Here we show how to select cases and find Cohen d (Cohen, 1988)

```r
table(msqR$time)
msq2 <- selectBy(msqR,"time=2") #just the time 2 data
msqd <- msq2[c(1:70,83)] #just the mood and drug data
cd <- cohen.d(msqd~drug) #find the Cohen d
error.dots(cd) #show the top and bottom 10 items
```

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>counts</td>
<td>3032</td>
<td>2086</td>
<td>1112</td>
<td>181</td>
</tr>
</tbody>
</table>

> msq2 <- selectBy(msqR,"time=2") #just the time 2 data
> msqd <- msq2[c(1:70,83)] #just the mood and drug data
> cd <- cohen.d(msqd~drug) #find the Cohen d
> error.dots(cd) #show the top and bottom 10 items
> summary(cd)
Multivariate (Mahalanobis) distance between groups 1.13
p-equivalent for each variable
Cohen d for caffeine/placebo on msqR data

Effect of caffeine on mood

- jittery
- alert
- anxious
- wakeful
- intense
- attentive
- wide.awake
- tense
- active
- energetic
- lively
- full.of.pep
- at.rest
- at.ease
- bored
- relaxed
- quiet
- still
- calm
- placid
- sluggish
- tired
- drowsy
- sleepy
**Showing sex differences in behavior**

1. Athenstaedt (2003) examined Gender Role Self-Concept. She reports two independent dimensions of Male and Female behaviors.

2. While there are large gender/sex differences on both of these dimensions, the two represent independent factors!

3. Eagly & Revelle (2022) used these data to explore the power of aggregation when examining sex differences.

4. Included as an example of various graphical displays.

---

**R code**

```r
# show error dots and ci
cd <- cohen.d(Athenstaedt[2:75], group="gender", dictionary=Athenstaedt.dictionary)
error.dots(cd, main="Cohen d for Athenstaedt data (with 95% CI)")
abline(v=0)

# show scatter plots and density
scatterHist(Femininity ~ Masculinity + gender, data =Athenstaedt, cex.point=.4,smooth=FALSE, correl=FALSE,d.arrow=TRUE,
col=c("red","blue"), lwd=4, cex.main=1.5,
main="Scatter Plot and Density",cex.axis=2)
```
Male/female differences on GERA-S items

Athenstaedt (2003)

Cohen $d$ for Athenstaedt data (with 95% CI)
Male/female differences on two scales

Combined M and F scales

\[ D = 1.68 \]

\[ r_{wg} = -0.05, \quad r = -0.29 \] (Athenstaedt, 2003; Eagly & Revelle, 2022)
Factor analysis, Cluster Analysis, Principal Components

1. Psychological data is typically of a high dimensionality.

2. One solution to this problem is the factor model which interprets observed manifest observed variables in terms of unobservable, latent variables. At the data level this is of course:

\[ X_i = \Lambda x_k + \Theta x_u \]  \hspace{1cm} (1)

3. At the level of covariances or correlations this is

\[ C \approx \Lambda \Lambda' + \Theta^2. \]  \hspace{1cm} (2)

4. For a fixed number of factors and fixed values of \( \Theta^2 \) this is solveteable as a simple eigen value decomposition.

5. However, the problem of how many factors is difficult (there is no one right answer).
Factor analysis as an iterative procedure

1. An initial estimate of communalities \((1 - \Theta^2)\)
2. Find the eigen vectors \((F)\) of \(R - \Theta^2\)
3. Find the residuals of \(R - F'F\)
   - ML for maximum likelihood
   - minres for minimum residual (default)
   - pa for principal factor
   - ...
4. Set the new communalities to diagonal of \(F'F\)
5. Iterate until communalities don’t change or until sum of squared residuals is a minimum or until Maximum Likelihood estimate is minimized.
6. If rotation (orthogonal) or transformation (oblique) apply the chosen algorithm.
   - “none”, “varimax”, “quartimax”, “BentlerT”, ”equamax”, “varimin”, ”geominT” and ”bifactor” are orthogonal rotations.
   - ”Promax”, ”promax”, ”oblimin”, ”simplimax”, ”bentlerQ”, ”geominQ”, ”biquartimin” and ”cluster” are oblique transformations
How many factors – no right answer, one wrong answer

1. Statistical
   - Extracting factors until the $\chi^2$ of the residual matrix is not significant.
   - Extracting factors until the change in $\chi^2$ from factor n to factor n+1 is not significant.

2. Rules of Thumb
   - Parallel: Extracting factors until the eigenvalues of the real data are less than the corresponding eigenvalues of a random data set of the same size (*parallel analysis*) fa.parallel
   - Plotting the magnitude of the successive eigenvalues and applying the *scree test*. scree

3. Interpretability
   - Extracting factors as long as they are interpretable.
   - Using the *Very Simple Structure Criterion* (VSS)
   - Using the Minimum Average Partial criterion (MAP).

4. Eigen Value of 1 rule (The worst rule)
nfactors applies many of these procedures
The number of factors problem is easy and hard

No best rule, one worst rule

“Solving the number of factors problem is easy, I do it everyday before breakfast. But knowing the right solution is harder.”

(attributed to Henry Kaiser by Horn & Engstrom (1979)

1. Parallel analysis (Extract factors until the eigen values are less than those of a random matrix).
   - Although a good rule for 100-500 subjects, this will not do as well with many (> 1000) subjects.

2. Velicer (1976) Minimum Average Partial (MAP) is pretty good

3. For items, the Very Simple Structure (VSS) (Revelle & Rocklin, 1979) criterion is pretty good.

4. Multiple statistical tests, many have problems with sample size.
   - If you want few factors, run few subjects
   - If you want many factors, run many subjects

5. One worst rule is the eigen value of 1.0 rule.
fa.parallel(bfi[1:25])

Parallel analysis suggests that the number of factors = 6
and the number of components = 6

Very Simple Structure
Call: vss(x = bfi[1:25])
VSS complexity 1 achieves a maximum of 0.58 with 4 factors
VSS complexity 2 achieves a maximum of 0.74 with 5 factors

The Velicer MAP achieves a minimum of 0.01 with 5 factors
BIC achieves a minimum of -513.09 with 8 factors
Sample Size adjusted BIC achieves a minimum of -106.39 with 8 factors
"The number of factors problem will break your heart"

```R
nfactors(bfi[1:25])
```

**Number of factors**

Call: `vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
      n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)`

VSS complexity 1 achieves a maximum of 0.58 with 4 factors
VSS complexity 2 achieves a maximum of 0.74 with 5 factors
The Velicer MAP achieves a minimum of 0.01 with 5 factors
Empirical BIC achieves a minimum of -737.9 with 8 factors
Sample Size adjusted BIC achieves a minimum of -205.18 with 12 factors

**Statistics by number of factors**

<table>
<thead>
<tr>
<th>vss1</th>
<th>vss2</th>
<th>map</th>
<th>dof</th>
<th>chisq</th>
<th>prob</th>
<th>sqresid</th>
<th>fit</th>
<th>RMSEA</th>
<th>BIC</th>
<th>SABIC</th>
<th>complex</th>
<th>eChisq</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.49</td>
<td>0.00</td>
<td>0.024</td>
<td>275</td>
<td>1.2e+04</td>
<td>0.0e+00</td>
<td>25.9</td>
<td>0.49</td>
<td>0.123</td>
<td>9680</td>
<td>10554</td>
<td>1.0</td>
</tr>
<tr>
<td>2</td>
<td>0.54</td>
<td>0.63</td>
<td>0.018</td>
<td>251</td>
<td>7.4e+03</td>
<td>0.0e+00</td>
<td>18.6</td>
<td>0.63</td>
<td>0.101</td>
<td>5370</td>
<td>6168</td>
<td>1.2</td>
</tr>
<tr>
<td>3</td>
<td>0.56</td>
<td>0.70</td>
<td>0.017</td>
<td>228</td>
<td>5.1e+03</td>
<td>0.0e+00</td>
<td>14.6</td>
<td>0.71</td>
<td>0.087</td>
<td>3286</td>
<td>4010</td>
<td>1.3</td>
</tr>
<tr>
<td>4</td>
<td>0.58</td>
<td>0.74</td>
<td>0.015</td>
<td>206</td>
<td>3.4e+03</td>
<td>0.0e+00</td>
<td>11.5</td>
<td>0.77</td>
<td>0.075</td>
<td>1787</td>
<td>2441</td>
<td>1.4</td>
</tr>
<tr>
<td>5</td>
<td>0.54</td>
<td>0.74</td>
<td>0.015</td>
<td>185</td>
<td>1.8e+03</td>
<td>4.3e-264</td>
<td>9.4</td>
<td>0.81</td>
<td>0.056</td>
<td>341</td>
<td>928</td>
<td>1.5</td>
</tr>
<tr>
<td>6</td>
<td>0.54</td>
<td>0.71</td>
<td>0.016</td>
<td>165</td>
<td>1.0e+03</td>
<td>1.8e-125</td>
<td>8.3</td>
<td>0.84</td>
<td>0.043</td>
<td>-277</td>
<td>247</td>
<td>1.8</td>
</tr>
<tr>
<td>7</td>
<td>0.51</td>
<td>0.70</td>
<td>0.019</td>
<td>146</td>
<td>7.1e+02</td>
<td>1.2e-74</td>
<td>7.9</td>
<td>0.85</td>
<td>0.037</td>
<td>-451</td>
<td>13</td>
<td>1.9</td>
</tr>
<tr>
<td>8</td>
<td>0.51</td>
<td>0.66</td>
<td>0.022</td>
<td>128</td>
<td>5.0e+02</td>
<td>7.1e-46</td>
<td>7.4</td>
<td>0.85</td>
<td>0.032</td>
<td>-513</td>
<td>-106</td>
<td>2.0</td>
</tr>
<tr>
<td>9</td>
<td>0.49</td>
<td>0.68</td>
<td>0.027</td>
<td>111</td>
<td>3.8e+02</td>
<td>8.2e-31</td>
<td>7.1</td>
<td>0.86</td>
<td>0.029</td>
<td>-503</td>
<td>-150</td>
<td>2.0</td>
</tr>
<tr>
<td>10</td>
<td>0.50</td>
<td>0.64</td>
<td>0.032</td>
<td>95</td>
<td>2.9e+02</td>
<td>4.5e-21</td>
<td>6.7</td>
<td>0.87</td>
<td>0.027</td>
<td>-468</td>
<td>-166</td>
<td>2.1</td>
</tr>
<tr>
<td>11</td>
<td>0.48</td>
<td>0.65</td>
<td>0.039</td>
<td>80</td>
<td>1.8e+02</td>
<td>2.2e-09</td>
<td>6.4</td>
<td>0.87</td>
<td>0.021</td>
<td>-457</td>
<td>-203</td>
<td>2.1</td>
</tr>
<tr>
<td>12</td>
<td>0.49</td>
<td>0.64</td>
<td>0.047</td>
<td>66</td>
<td>1.1e+02</td>
<td>6.9e-04</td>
<td>6.3</td>
<td>0.88</td>
<td>0.015</td>
<td>-415</td>
<td>-205</td>
<td>2.2</td>
</tr>
<tr>
<td>13</td>
<td>0.47</td>
<td>0.65</td>
<td>0.057</td>
<td>53</td>
<td>7.6e+01</td>
<td>2.0e-02</td>
<td>6.2</td>
<td>0.88</td>
<td>0.012</td>
<td>-345</td>
<td>-176</td>
<td>2.3</td>
</tr>
<tr>
<td>14</td>
<td>0.46</td>
<td>0.64</td>
<td>0.066</td>
<td>41</td>
<td>5.4e+01</td>
<td>7.9e-02</td>
<td>5.6</td>
<td>0.89</td>
<td>0.011</td>
<td>-271</td>
<td>-141</td>
<td>2.1</td>
</tr>
</tbody>
</table>
How many factors? What ever you want

![Graphs showing Very Simple Structure, Complexity, Empirical BIC, and Root Mean Residual vs. Number of Factors]
Exploratory Factor analysis using MinRes (minimum residual) as well as EFA by Principal Axis, Weighted Least Squares or Maximum Likelihood

Among the many ways to do latent variable exploratory factor analysis (EFA), one of the better is to use Ordinary Least Squares (OLS) to find the minimum residual (minres) solution. This produces solutions very similar to maximum likelihood even for badly behaved matrices. A variation on minres is to do weighted least squares (WLS). Perhaps the most conventional technique is principal axes (PAF). An eigen value decomposition of a correlation matrix is done and then the communalities for each variable are estimated by the first n factors. These communalities are entered onto the diagonal and the procedure is repeated until the sum(diag(r)) does not vary. Yet another estimate procedure is maximum likelihood. For well behaved matrices, maximum likelihood factor analysis (either in the fa or in the factanal function) is probably preferred. Bootstrapped confidence intervals of the loadings and interfactor correlations are found by fa with n.iter > 1.
factor the bfi data set, extract 5 factors

R code

f5 <- fa(bfi[1:25], 5)

Factor Analysis using method = minres
Call: fa(r = bfi[1:25], n=5)
Standardized loadings (pattern matrix) based upon correlation matrix

<table>
<thead>
<tr>
<th></th>
<th>MR2</th>
<th>MR1</th>
<th>MR3</th>
<th>MR5</th>
<th>MR4</th>
<th>h2</th>
<th>u2</th>
<th>com</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>0.21</td>
<td>0.17</td>
<td>0.07</td>
<td>-0.41</td>
<td>-0.06</td>
<td>0.19</td>
<td>0.81</td>
<td>2.0</td>
</tr>
<tr>
<td>A2</td>
<td>-0.02</td>
<td>0.00</td>
<td>0.08</td>
<td>0.64</td>
<td>0.03</td>
<td>0.45</td>
<td>0.55</td>
<td>1.0</td>
</tr>
<tr>
<td>A3</td>
<td>-0.03</td>
<td>0.12</td>
<td>0.02</td>
<td>0.66</td>
<td>0.03</td>
<td>0.52</td>
<td>0.48</td>
<td>1.1</td>
</tr>
<tr>
<td>A4</td>
<td>-0.06</td>
<td>0.06</td>
<td>0.19</td>
<td>0.43</td>
<td>-0.15</td>
<td>0.28</td>
<td>0.72</td>
<td>1.7</td>
</tr>
<tr>
<td>A5</td>
<td>-0.11</td>
<td>0.23</td>
<td>0.01</td>
<td>0.53</td>
<td>0.04</td>
<td>0.46</td>
<td>0.54</td>
<td>1.5</td>
</tr>
<tr>
<td>C1</td>
<td>0.07</td>
<td>-0.03</td>
<td>0.55</td>
<td>-0.02</td>
<td>0.15</td>
<td>0.33</td>
<td>0.67</td>
<td>1.2</td>
</tr>
<tr>
<td>C2</td>
<td>0.15</td>
<td>-0.09</td>
<td>0.67</td>
<td>0.08</td>
<td>0.04</td>
<td>0.45</td>
<td>0.55</td>
<td>1.2</td>
</tr>
<tr>
<td>C3</td>
<td>0.03</td>
<td>-0.06</td>
<td>0.57</td>
<td>0.09</td>
<td>-0.07</td>
<td>0.32</td>
<td>0.68</td>
<td>1.1</td>
</tr>
<tr>
<td>C4</td>
<td>0.17</td>
<td>0.00</td>
<td>-0.61</td>
<td>0.04</td>
<td>-0.05</td>
<td>0.45</td>
<td>0.55</td>
<td>1.2</td>
</tr>
<tr>
<td>C5</td>
<td>0.19</td>
<td>-0.14</td>
<td>-0.55</td>
<td>0.02</td>
<td>0.09</td>
<td>0.43</td>
<td>0.57</td>
<td>1.4</td>
</tr>
<tr>
<td>E1</td>
<td>-0.06</td>
<td>-0.56</td>
<td>0.11</td>
<td>-0.08</td>
<td>-0.10</td>
<td>0.35</td>
<td>0.65</td>
<td>1.2</td>
</tr>
<tr>
<td>E2</td>
<td>0.10</td>
<td>-0.68</td>
<td>-0.02</td>
<td>-0.05</td>
<td>-0.06</td>
<td>0.54</td>
<td>0.46</td>
<td>1.1</td>
</tr>
<tr>
<td>E3</td>
<td>0.08</td>
<td>0.42</td>
<td>0.00</td>
<td>0.25</td>
<td>0.28</td>
<td>0.44</td>
<td>0.56</td>
<td>2.6</td>
</tr>
<tr>
<td>E4</td>
<td>0.01</td>
<td>0.59</td>
<td>0.02</td>
<td>0.29</td>
<td>-0.08</td>
<td>0.53</td>
<td>0.47</td>
<td>1.5</td>
</tr>
<tr>
<td>E5</td>
<td>0.15</td>
<td>0.42</td>
<td>0.27</td>
<td>0.05</td>
<td>0.21</td>
<td>0.40</td>
<td>0.60</td>
<td>2.6</td>
</tr>
<tr>
<td>N1</td>
<td>0.81</td>
<td>0.10</td>
<td>0.00</td>
<td>-0.11</td>
<td>-0.05</td>
<td>0.65</td>
<td>0.35</td>
<td>1.1</td>
</tr>
<tr>
<td>N2</td>
<td>0.78</td>
<td>0.04</td>
<td>0.01</td>
<td>-0.09</td>
<td>0.01</td>
<td>0.60</td>
<td>0.40</td>
<td>1.0</td>
</tr>
<tr>
<td>N3</td>
<td>0.71</td>
<td>-0.10</td>
<td>-0.04</td>
<td>0.08</td>
<td>0.02</td>
<td>0.55</td>
<td>0.45</td>
<td>1.1</td>
</tr>
<tr>
<td>N4</td>
<td>0.47</td>
<td>-0.39</td>
<td>-0.14</td>
<td>0.09</td>
<td>0.08</td>
<td>0.49</td>
<td>0.51</td>
<td>2.3</td>
</tr>
<tr>
<td>N5</td>
<td>0.49</td>
<td>-0.20</td>
<td>0.00</td>
<td>0.21</td>
<td>-0.15</td>
<td>0.35</td>
<td>0.65</td>
<td>2.0</td>
</tr>
<tr>
<td>O1</td>
<td>0.02</td>
<td>0.10</td>
<td>0.07</td>
<td>0.02</td>
<td>0.51</td>
<td>0.31</td>
<td>0.69</td>
<td>1.1</td>
</tr>
<tr>
<td>O2</td>
<td>0.19</td>
<td>0.06</td>
<td>-0.08</td>
<td>0.16</td>
<td>-0.46</td>
<td>0.26</td>
<td>0.74</td>
<td>1.7</td>
</tr>
<tr>
<td>O3</td>
<td>0.03</td>
<td>0.15</td>
<td>0.02</td>
<td>0.08</td>
<td>0.61</td>
<td>0.46</td>
<td>0.54</td>
<td>1.2</td>
</tr>
<tr>
<td>O4</td>
<td>0.13</td>
<td>-0.32</td>
<td>-0.02</td>
<td>0.17</td>
<td>0.37</td>
<td>0.25</td>
<td>0.75</td>
<td>2.7</td>
</tr>
<tr>
<td>O5</td>
<td>0.13</td>
<td>0.10</td>
<td>-0.03</td>
<td>0.04</td>
<td>-0.54</td>
<td>0.30</td>
<td>0.70</td>
<td>1.2</td>
</tr>
</tbody>
</table>

SS loadings 2.57 2.20 2.03 1.99 1.59
Proportion Var 0.10 0.09 0.08 0.08 0.06
Cumulative Var 0.10 0.19 0.27 0.35 0.41
Proportion Explained 0.25 0.21 0.20 0.19 0.15
Cumulative Proportion 0.25 0.46 0.66 0.85 1.00

With factor correlations of

<table>
<thead>
<tr>
<th></th>
<th>MR2</th>
<th>MR1</th>
<th>MR3</th>
<th>MR5</th>
<th>MR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR2</td>
<td>1.00</td>
<td>-0.21</td>
<td>-0.19</td>
<td>-0.04</td>
<td>-0.01</td>
</tr>
<tr>
<td>MR1</td>
<td>-0.21</td>
<td>1.00</td>
<td>0.23</td>
<td>0.33</td>
<td>0.17</td>
</tr>
<tr>
<td>MR3</td>
<td>-0.19</td>
<td>0.23</td>
<td>1.00</td>
<td>0.20</td>
<td>0.19</td>
</tr>
<tr>
<td>MR5</td>
<td>-0.04</td>
<td>0.33</td>
<td>0.20</td>
<td>1.00</td>
<td>0.19</td>
</tr>
<tr>
<td>MR4</td>
<td>-0.01</td>
<td>0.17</td>
<td>0.19</td>
<td>0.19</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Mean item complexity = 1.5
More important output

R code

```r
f5
diagram(f5, main="5 factors of the bfi")
plot(f5)  #an alternative way to show the results
biplot(f5 )  #show a biplot
```

Mean item complexity = 1.5
Test of the hypothesis that 5 factors are sufficient.

df null model = 300 with the objective function = 7.23 with Chi Square = 20163.79
df of the model are 185 and the objective function was 0.65

The root mean square of the residuals (RMSR) is 0.03
The df corrected root mean square of the residuals is 0.04

The harmonic n.obs is 2762 with the empirical chi square 1392.16 with prob < 5.6e-184
The total n.obs was 2800 with Likelihood Chi Square = 1808.94 with prob < 4.3e-264

Tucker Lewis Index of factoring reliability = 0.867
RMSEA index = 0.056 and the 90 % confidence intervals are 0.054 0.058
BIC = 340.53
Fit based upon off diagonal values = 0.98

Measures of factor score adequacy

<table>
<thead>
<tr>
<th></th>
<th>MR2</th>
<th>MR1</th>
<th>MR3</th>
<th>MR5</th>
<th>MR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correlation of (regression) scores with factors</td>
<td>0.92</td>
<td>0.89</td>
<td>0.88</td>
<td>0.88</td>
<td>0.84</td>
</tr>
<tr>
<td>Multiple R square of scores with factors</td>
<td>0.85</td>
<td>0.79</td>
<td>0.77</td>
<td>0.77</td>
<td>0.71</td>
</tr>
<tr>
<td>Minimum correlation of possible factor scores</td>
<td>0.70</td>
<td>0.59</td>
<td>0.54</td>
<td>0.54</td>
<td>0.42</td>
</tr>
</tbody>
</table>
Use the diagram to show the structure

5 factors of the bfi
Use the plot to show the loadings in a different fashion.
Use the biplot to show the loadings in a different fashion.
Use the biplot with the choose option for just some factors
Factor Extension

1. What if you have more variables to fit into a factor model?

2. Dwyer (1937); Mosier (1938) and then elaborated by Gorsuch (1997); Horn (1973)

3. Specify the data set, the original variables, and the extension variables.

```
f5.e <- fa.extend(bfi,nf=5,ov=c(1:4,6:9,11:14,16:19, 21:24),
                 ev=c(20,15,10,5,25))  #notice the ordering for better graphic
```

```
diagram(f5.e,e.cut=.3)
```
Factor extension of the bfi

Factor analysis and extension

Diagram of factor analysis and extension.
Cluster analysis as an alternative to factor analysis

The ICLUST algorithm was developed for the construction of internally consistent scales from items Revelle (1979)

1. Form the proximity (correlation) matrix
2. Find the most similar pair of items
3. Combine them into a new scale,
4. Recalculate the correlation matrix
5. Repeat steps 2-4 until various criteria are met

- **alpha** Coefficient $\alpha$ of the composite fails to increase (rarely happens)
- **beta** Coefficient $beta$ (the worst split half reliability) fails to increase

iclust is meant for forming scales from items and is a useful guide to the structure of a test.
iclust Revelle (1979)

R code

```r
ic <- iclust(bfi[1:25])
summary(ic)
```

ICLUST (Item Cluster Analysis) Call: iclust(r.mat = bfi[1:25])

Purified Alpha:

<table>
<thead>
<tr>
<th></th>
<th>C20</th>
<th>C16</th>
<th>C15</th>
<th>C21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alpha</td>
<td>0.80</td>
<td>0.81</td>
<td>0.73</td>
<td>0.61</td>
</tr>
</tbody>
</table>

Guttman Lambda6*

<table>
<thead>
<tr>
<th></th>
<th>C20</th>
<th>C16</th>
<th>C15</th>
<th>C21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lambda</td>
<td>0.82</td>
<td>0.81</td>
<td>0.72</td>
<td>0.61</td>
</tr>
</tbody>
</table>

Original Beta:

<table>
<thead>
<tr>
<th></th>
<th>C20</th>
<th>C16</th>
<th>C15</th>
<th>C21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beta</td>
<td>0.63</td>
<td>0.76</td>
<td>0.67</td>
<td>0.27</td>
</tr>
</tbody>
</table>

Cluster size:

<table>
<thead>
<tr>
<th></th>
<th>C20</th>
<th>C16</th>
<th>C15</th>
<th>C21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Purified scale intercorrelations

reliabilities on diagonal

correlations corrected for attenuation above diagonal:

<table>
<thead>
<tr>
<th></th>
<th>C20</th>
<th>C16</th>
<th>C15</th>
<th>C21</th>
</tr>
</thead>
<tbody>
<tr>
<td>C20</td>
<td>0.80</td>
<td>-0.291</td>
<td>-0.40</td>
<td>-0.33</td>
</tr>
<tr>
<td>C16</td>
<td>-0.24</td>
<td>0.815</td>
<td>0.29</td>
<td>0.11</td>
</tr>
<tr>
<td>C15</td>
<td>-0.30</td>
<td>0.221</td>
<td>0.73</td>
<td>0.30</td>
</tr>
<tr>
<td>C21</td>
<td>-0.23</td>
<td>0.074</td>
<td>0.20</td>
<td>0.61</td>
</tr>
</tbody>
</table>
Hierarchical cluster analysis of items using iclust
Multiple levels of factors

1. Hierarchical/higher order models (Jensen & Weng, 1994)
   - Simulated data to match Jensen & Weng (1994)
   - Ability data from the ICAR Condon & Revelle (2014)
   - Size data from the United States Airforce
   - Hierarchical solution of the SAPA Personality Inventory (spi) (Condon, 2018)

2. Bifactor models (Holzinger & Swineford, 1937; Reise, 2012; Rodriguez, Reise & Haviland, 2016)
   - Schmid & Leiman (1957) introduced a transformation of a higher order model into a bifactor model.
   - Constraints on the factor loadings given the structure.

3. Comparing factors at different levels of n factors using the bassAckward algorithm (Goldberg, 2006; Waller, 2007)
   - Two levels of factors from the SAPA Personality Inventory (spi) (Condon, 2018)

4. The SAPA Personality Inventory (spi) (Condon, 2018) data set has 135 items plus 10 criteria variables for 4,000 participants.
   - It was developed from 696 IPIP items to represent 200 broad and narrow public domain measures of personality.
Simulating 9 variables from Jensen & Weng (1994)

```r
jensen <- sim.hierarchical() # the default values are Jensen-Weng
f3 <- fa(jensen, 3)
om <- omega(jensen)
diagram(om, sl=FALSE); diagram(om) # default is to do Schmid-Leiman
```

Factor Analysis using method = minres
Call: fa(r = jensen, nfactors = 3)
Standardized loadings (pattern matrix) based upon correlation matrix

```
        MR1  MR3  MR2      h2     u2   com
V1  0.8  0.0  0.0  0.64  0.36   1
V2  0.7  0.0  0.0  0.49  0.51   1
V3  0.6  0.0  0.0  0.36  0.64   1
V4  0.0  0.7  0.0  0.49  0.51   1
V5  0.0  0.6  0.0  0.36  0.64   1
V6  0.0  0.5  0.0  0.25  0.75   1
V7  0.0  0.0  0.6  0.36  0.64   1
V8  0.0  0.0  0.5  0.25  0.75   1
V9  0.0  0.0  0.4  0.16  0.84   1
```

SS loadings

```
        MR1  MR3  MR2
V1     1.49 1.10 0.77
V2     0.17 0.12 0.09
V3     0.17 0.29 0.37
V4     0.44 0.33 0.23
V5     0.44 0.77 1.00
```

With factor correlations of

```
        MR1  MR3  MR2
MR1    1.00 0.72 0.63
MR3    0.72 1.00 0.56
MR2    0.63 0.56 1.00
```
A higher order factor representation

Hierarchical (multilevel) Structure

Hierarchical (multilevel) Structure

V1
V2
V3
V4
V5
V6
V7
V8
V9
F1
F2
F3
g

0.8
0.7
0.6
0.7
0.6
0.5
0.6
0.5
0.4
0.9
0.8
0.7
0.7
0.6
0.5
0.7
Schmid & Leiman (1957) transformation to a bifactor model

Hierarchical (multilevel) Structure

Omega with Schmid Leiman Transformation
Unfortunately, the bifactor rotation does not capture the right structure

R code

```r
f4 <- fa(jensen,4, rotate="bifactor")
```

Factor Analysis using method = minres
Call: fa(r = jensen, nfactors = 4, rotate = "bifactor")
Standardized loadings (pattern matrix) based upon correlation matrix

<table>
<thead>
<tr>
<th></th>
<th>MR1</th>
<th>MR3</th>
<th>MR2</th>
<th>MR4</th>
<th>h2</th>
<th>u2</th>
<th>com</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td>0.79</td>
<td>-0.03</td>
<td>-0.09</td>
<td>0.02</td>
<td>0.63</td>
<td>0.37</td>
<td>1.0</td>
</tr>
<tr>
<td>V2</td>
<td>0.70</td>
<td>-0.05</td>
<td>-0.09</td>
<td>-0.06</td>
<td>0.51</td>
<td>0.49</td>
<td>1.1</td>
</tr>
<tr>
<td>V3</td>
<td>0.60</td>
<td>-0.03</td>
<td>-0.07</td>
<td>0.12</td>
<td>0.38</td>
<td>0.62</td>
<td>1.1</td>
</tr>
<tr>
<td>V4</td>
<td>0.53</td>
<td>0.45</td>
<td>0.01</td>
<td>0.00</td>
<td>0.49</td>
<td>0.51</td>
<td>2.0</td>
</tr>
<tr>
<td>V5</td>
<td>0.46</td>
<td>0.39</td>
<td>0.01</td>
<td>0.00</td>
<td>0.36</td>
<td>0.64</td>
<td>2.0</td>
</tr>
<tr>
<td>V6</td>
<td>0.38</td>
<td>0.32</td>
<td>0.00</td>
<td>0.00</td>
<td>0.25</td>
<td>0.75</td>
<td>2.0</td>
</tr>
<tr>
<td>V7</td>
<td>0.43</td>
<td>0.01</td>
<td>0.42</td>
<td>0.00</td>
<td>0.36</td>
<td>0.64</td>
<td>2.0</td>
</tr>
<tr>
<td>V8</td>
<td>0.36</td>
<td>0.01</td>
<td>0.35</td>
<td>0.00</td>
<td>0.25</td>
<td>0.75</td>
<td>2.0</td>
</tr>
<tr>
<td>V9</td>
<td>0.29</td>
<td>0.01</td>
<td>0.28</td>
<td>0.00</td>
<td>0.16</td>
<td>0.84</td>
<td>2.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MR1</th>
<th>MR3</th>
<th>MR2</th>
<th>MR4</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS loadings</td>
<td>2.50</td>
<td>0.47</td>
<td>0.39</td>
<td>0.02</td>
</tr>
<tr>
<td>Proportion Var</td>
<td>0.28</td>
<td>0.05</td>
<td>0.04</td>
<td>0.00</td>
</tr>
<tr>
<td>Cumulative Var</td>
<td>0.28</td>
<td>0.33</td>
<td>0.37</td>
<td>0.38</td>
</tr>
<tr>
<td>Proportion Explained</td>
<td>0.74</td>
<td>0.14</td>
<td>0.12</td>
<td>0.01</td>
</tr>
<tr>
<td>Cumulative Proportion</td>
<td>0.74</td>
<td>0.88</td>
<td>0.99</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Another case: the ICAR 16

R code

```r
om.icar <- omega(icar,4)
```

Schmid Leiman Factor loadings greater than 0.2

<table>
<thead>
<tr>
<th>Item</th>
<th>g</th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
<th>F4*</th>
<th>h2</th>
<th>u2</th>
<th>p2</th>
</tr>
</thead>
<tbody>
<tr>
<td>reason.4</td>
<td>0.50</td>
<td>0.28</td>
<td></td>
<td></td>
<td></td>
<td>0.35</td>
<td>0.65</td>
<td>0.74</td>
</tr>
<tr>
<td>reason.16</td>
<td>0.42</td>
<td></td>
<td>0.21</td>
<td></td>
<td></td>
<td>0.23</td>
<td>0.77</td>
<td>0.76</td>
</tr>
<tr>
<td>reason.17</td>
<td>0.55</td>
<td></td>
<td></td>
<td>0.46</td>
<td></td>
<td>0.51</td>
<td>0.49</td>
<td>0.59</td>
</tr>
<tr>
<td>reason.19</td>
<td>0.44</td>
<td></td>
<td></td>
<td></td>
<td>0.21</td>
<td>0.25</td>
<td>0.75</td>
<td>0.78</td>
</tr>
<tr>
<td>letter.7</td>
<td>0.51</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.39</td>
<td>0.61</td>
<td>0.68</td>
</tr>
<tr>
<td>letter.33</td>
<td>0.46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.31</td>
<td>0.69</td>
<td>0.69</td>
</tr>
<tr>
<td>letter.34</td>
<td>0.53</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.43</td>
<td>0.57</td>
<td>0.65</td>
</tr>
<tr>
<td>letter.58</td>
<td>0.47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.28</td>
<td>0.72</td>
<td>0.78</td>
</tr>
<tr>
<td>matrix.45</td>
<td>0.40</td>
<td></td>
<td></td>
<td></td>
<td>0.64</td>
<td>0.57</td>
<td>0.43</td>
<td>0.28</td>
</tr>
<tr>
<td>matrix.46</td>
<td>0.40</td>
<td></td>
<td></td>
<td></td>
<td>0.26</td>
<td>0.24</td>
<td>0.76</td>
<td>0.65</td>
</tr>
<tr>
<td>matrix.47</td>
<td>0.43</td>
<td></td>
<td></td>
<td></td>
<td>0.23</td>
<td>0.77</td>
<td>0.79</td>
<td></td>
</tr>
<tr>
<td>matrix.55</td>
<td>0.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.13</td>
<td>0.87</td>
<td>0.66</td>
</tr>
<tr>
<td>rotate.3</td>
<td>0.36</td>
<td>0.60</td>
<td></td>
<td></td>
<td></td>
<td>0.50</td>
<td>0.50</td>
<td>0.26</td>
</tr>
<tr>
<td>rotate.4</td>
<td>0.41</td>
<td>0.60</td>
<td></td>
<td></td>
<td></td>
<td>0.53</td>
<td>0.47</td>
<td>0.32</td>
</tr>
<tr>
<td>rotate.6</td>
<td>0.40</td>
<td>0.49</td>
<td></td>
<td></td>
<td></td>
<td>0.41</td>
<td>0.59</td>
<td>0.39</td>
</tr>
<tr>
<td>rotate.8</td>
<td>0.33</td>
<td>0.54</td>
<td></td>
<td></td>
<td></td>
<td>0.41</td>
<td>0.59</td>
<td>0.27</td>
</tr>
</tbody>
</table>

With Sums of squares of:

<table>
<thead>
<tr>
<th>g</th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
<th>F4*</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.05</td>
<td>1.31</td>
<td>0.47</td>
<td>0.40</td>
<td>0.53</td>
</tr>
</tbody>
</table>
omega of the ability items

Hierarchical (multilevel) Structure

Hierarchical Scales
Reliability
β, ω
Regression
ML
Simulation
More
References

Hierarchical (multilevel) Structure

rotate.3
rotate.4
rotate.8
rotate.6
letter.34
letter.7
letter.33
letter.58
matrix.47
reason.17
reason.4
reason.16
reason.19
matrix.45
matrix.46
matrix.55

F1
F2
F3
F4
9
Pearson, polychoric and tetrachoric correlations

1. As we know, Pearson correlations are appropriate for continuous data.

2. But with categorical or dichotomous data, the correlations are attenuated.

3. The tetrachoric correlation estimates what the Pearson would be with continuous data.
   - Tetrachoric and polychoric correlations are thus estimates of the latent correlation assuming bivariate normality.
   - Appropriate for determining the structure of correlations, inappropriate for estimates of reliability.

4. The \( \alpha \), \( \omega \) functions have an option to find the tetrachoric/polychoric correlations before factoring.

5. Particularly appropriate for dichotomous variables (e.g. the ICAR example, ability)
omega of the ability items using tetrachoric correlations

Hierarchical/multilevel Structure using tetrachoric correlations

- rotate.3
- rotate.4
- rotate.8
- rotate.6
- letter.34
- letter.7
- letter.33
- letter.58
- matrix.47
- reason.17
- reason.4
- reason.19
- reason.16
- matrix.45
- matrix.46
- matrix.55

Leaders:
- F1
- F2
- F3
- F4
- g
Is there a general factor of body size? The USAF data set

g of bodysize?
fa.hierarchical solution of the spi of the spi items

Hierarchical (multilevel) Structure
The “Bass-Ackwards” algorithm

1. Goldberg (2006) described a hierarchical factor structure organization from the “top down”.
   - The original idea was to do successive factor analyses from 1 to nf factors organized by factor score correlations from one level to the next.

2. Waller (2007) discussed a simple way of doing this for components without finding the scores.

3. Using the factor correlations (from Gorsuch, 1983) to organize factors hierarchically results may be organized at many different levels.

4. The algorithm may be applied to principal components (pca) or to true factor analysis.

5. Implemented as bassAckward.

6. The solutions should not be confused with a hierarchical solution where the higher order factors are factors of the lower order factors.
bassAckward solution of the spi items for 5 and 27 factors
Scale Construction

1. *psych* was specifically designed for the problem of reading and describing sets of items and then forming unit weighed scales from these items.

2. The advantage of scales formed from unit weighted items rather than factor weights is that they are more robust to sample variation (Widaman & Revelle, 2022).

3. Although there are functions to combine a set of items into just one scale *alpha* the more typical problem is form multiple scales e.g., `scoreItems`.

4. `fastScore` will scale scores without any accompanying statistics, but the more typical case is to use `.pfunscoreItems`.

5. To find scales based upon Item Response Theory, use `scoreIrt`
Example data sets

1. The sai represents 3,032 participants on 20 state anxiety items with 1229 participants who took it twice, 1047 with three measures, and 70 with four measures.

2. The epi represents 3570 participants for the 57 items of the Eysenck Personality Inventory from the early 1990s at the PMC lab.

3. An additional data set (epiR) has test and retest information for 474 participants.

4. The Motivational State Questionnaire msqR (Revelle & Anderson, 1998) contains 75 mood items for 3032 unique participants. 2753 took it at least twice, 446 three times, and 181 four times.
Scoring scales

1. For one set of items for one scale use \texttt{alpha}
   - Will warn if item x total correlations are negative and encourage
   - Using the check.keys option to reverse score negatively keyed items

2. More typical is to specify a \texttt{keys.list} of multiple keys each with multiple items.
   - Negatively keyed items are reversed scored by subtracting from the maximum possible item score - minimum possible item score
   - Scale scores are expressed as the \textit{mean} item response, although \textit{sum} scores is also an option,
   - Missing items scores can be \textit{imputed} by means, medians, or ignored.

3. Most scoring functions return scores as well as statistics for the scales.

4. \texttt{scoreFast} and \texttt{scoreVeryFast} just return the scores.
Example keys list

```r
sai.keys <- list(sai = c("tense","regretful","upset","worrying","anxious","nervous","jittery","high.strung","worried","rattled","-calm","-secure","-at.ease","-rested","-comfortable","-confident","-relaxed","-content","-joyful","-secure","-at.ease","-rested","-comfortable","-confident","-relaxed","-content","-joyful","-pleasant" ),
sai.p = c("calm","at.ease","rested","comfortable","confident","secure","relaxed","content","joyful","pleasant" ),
sai.n = c("tense","anxious","nervous","jittery","rattled","high.strung","upset","worrying","worried","regretful" )
)

sai.keys
```

$sai

[1] "tense"   "regretful"   "upset"   "worrying"   "anxious"   "nervous"  
[7] "jittery" "high.strung" "worried"  "rattled" "-calm" "-secure"  
[13] "-at.ease" "-rested"  "-comfortable" "-confident" "-relaxed" "-content"  
[19] "-joyful" "-pleasant"

$sai.p

[1] "calm" "at.ease" "rested"  "comfortable" "confident" "secure"  
   "relaxed"  "content"  "joyful" "pleasant"

$sai.n

[1] "tense"  "anxious"  "nervous"  "jittery"  "rattled" "high.strung"  
   "upset"  "worrying"  "worried"  "regretful"
Some keys.list are part of the data set

epi.keys

epi.keys

$E
[1] "V1" "V3" "V8" "V10" "V13" "V17" "V22" "V25" "V27" "V39"
[11] "V44" "V46" "V49" "V53" "V56" "-V5" "-V15" "-V20" "-V29" "-V32"
[21] "-V34" "-V37" "-V41" "-V51"

$N
[1] "V2" "V4" "V7" "V9" "V11" "V14" "V16" "V19" "V21" "V23" "V26" "V28"
[13] "V31" "V33" "V35" "V38" "V40" "V43" "V45" "V47" "V50" "V52" "V55" "V57"

$L
[1] "V6" "V24" "V36" "-V12" "-V18" "-V30" "-V42" "-V48" "-V54"

$Imp
[1] "V1" "V3" "V8" "V10" "V13" "V22" "V39" "-V5" "-V41"

$Soc
[1] "V17" "V25" "V27" "V44" "V46" "V53" "-V11" "-V15" "-V20" "-V29"
Dictionaries

1. Referring to item numbers is not convenient for discussing results.

2. Thus, it is possible to create a dictionary of the items.

3. A dictionary can be prepared outside of R by forming a spreadsheet including at least one column labeled “content” and with rownames for the item number or name. Other columns can specify the item source, or anything interesting.

```r
headTail(epi.dictionary)
```

<table>
<thead>
<tr>
<th>V1</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>V2</td>
<td>Do you often long for excitement?</td>
</tr>
<tr>
<td>V3</td>
<td>Do you often need understanding friends to cheer you up?</td>
</tr>
<tr>
<td>V4</td>
<td>Are you usually carefree?</td>
</tr>
<tr>
<td>V54</td>
<td>Do you find it very hard to take no for an answer?</td>
</tr>
<tr>
<td>...</td>
<td>&lt;NA&gt;</td>
</tr>
<tr>
<td>V55</td>
<td>Do you worry about your health?</td>
</tr>
<tr>
<td>V56</td>
<td>Do you like playing pranks on others?</td>
</tr>
<tr>
<td>V57</td>
<td>Do you suffer from sleeplessness?</td>
</tr>
</tbody>
</table>
Using a keys list and a dictionary to show content

R code

```r
lookupFromKeys(epi.keys, epi.dictionary, n=2)
```

$E

Content

V1  Do you often long for excitement?
V3  Are you usually carefree?

$N

Content

V2  Do you often need understanding friends to cheer you up?
V4  Do you find it very hard to take no for an answer?

$L

V6  If you say you will do something do you always keep your promise, no matter how inconvenient it might be to do so?
V24 Are all your habits good and desirable ones?

$Imp

Content

V1  Do you often long for excitement?
V3  Are you usually carefree?

$Soc

Content

V17 Do you like going out a lot?
V25 Can you usually let yourself go and enjoy yourself a lot at a lively party?
Or show the items for just one scale (as a way of checking the keys)

R code

```r
lookupFromKeys(epi.keys, epi.dictionary)$Imp
```

<table>
<thead>
<tr>
<th>V1</th>
<th>Do you often long for excitement?</th>
</tr>
</thead>
<tbody>
<tr>
<td>V3</td>
<td>Are you usually carefree?</td>
</tr>
<tr>
<td>V8</td>
<td>Do you generally do and say things quickly without stopping to think?</td>
</tr>
<tr>
<td>V10</td>
<td>Would you do almost anything for a dare?</td>
</tr>
<tr>
<td>V13</td>
<td>Do you often do things on the spur of the moment?</td>
</tr>
<tr>
<td>V22</td>
<td>When people shout at you do you shout back?</td>
</tr>
<tr>
<td>V39</td>
<td>Do you like doing things in which you have to act quickly?</td>
</tr>
<tr>
<td>V5-</td>
<td>Do you stop and think things over before doing anything?</td>
</tr>
<tr>
<td>V41-</td>
<td>Are you slow and unhurried in the way you move?</td>
</tr>
</tbody>
</table>
Using `scoreItems` on the epi dataset

**R code**

```r
scales <- scoreItems(epi.keys, epi) # produces raw scores and stats
overlap <- scoreOverlap(epi.keys, epi) # finds the correlarti
```

Scale intercorrelations corrected for attenuation

raw correlations below the diagonal, (unstandardized) alpha on the diagonal

corrected correlations above the diagonal:

```
E   N   L  Imp  Soc
E  0.73 -0.228 -0.40 1.211 1.19
N -0.17 0.793 -0.28 0.025 -0.33
L -0.23 -0.165 0.44 -0.339 -0.31
Imp 0.71 0.015 -0.16 0.478 0.56 <- note that the imp and Soc scales
Soc 0.86 -0.250 -0.18 0.330 0.73 <- overlapping items with the E scale
```

Scale intercorrelations adjusted for item overlap

Scale intercorrelations corrected for attenuation

raw correlations (corrected for overlap) below the diagonal, (standardized) alpha on the diagonal

corrected (for overlap and reliability) correlations above the diagonal:

```
E   N   L  Imp  Soc
E  0.73 -0.23 -0.38 0.799 0.94
N -0.18 0.80 -0.28 0.049 -0.31
L -0.22 -0.17 0.45 -0.311 -0.30
Imp 0.47 0.03 -0.14 0.474 0.54
Soc 0.68 -0.24 -0.17 0.320 0.73
```
But what if we have overlapping scales?

1. Sometimes we are interested in how higher order scales relate to lower order scales.

2. The problem is, the items overlap.

3. Some people solve this problem by dropping the overlapping items. But this changes the meaning of the scales.

4. A fairly straightforward procedure is to estimate the overlapping variances with the best estimate of shared (common) variance, similar to what is done when finding coefficient $\alpha$.

5. Need to do this on the correlation matrix of the items, not the raw data.

6. See ?scoreOverlap
A small part of the output for scoreItems

R code

names(scales)

dim(scales$scores)

[1] 3570  5
By default, `scoreItems` imputes item medians for missing data.

```r
describe(scales$scores)
scales <- scoreItems(epi.keys, epi, impute="none")
describe(scales$scores)
```

<table>
<thead>
<tr>
<th>var</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>3570</td>
<td>1.46</td>
<td>0.17</td>
<td>1.46</td>
<td>1.46</td>
<td>0.19</td>
<td>1.04</td>
<td>2</td>
<td>0.96</td>
<td>-0.28</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>3570</td>
<td>1.54</td>
<td>0.19</td>
<td>1.54</td>
<td>1.55</td>
<td>0.19</td>
<td>1.00</td>
<td>2</td>
<td>1.00</td>
<td>-0.42</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>3570</td>
<td>1.73</td>
<td>0.18</td>
<td>1.78</td>
<td>1.74</td>
<td>0.16</td>
<td>1.00</td>
<td>2</td>
<td>1.00</td>
<td>-0.14</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Imp</td>
<td>3570</td>
<td>1.51</td>
<td>0.20</td>
<td>1.56</td>
<td>1.51</td>
<td>0.16</td>
<td>1.00</td>
<td>2</td>
<td>1.00</td>
<td>-0.57</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Soc</td>
<td>3570</td>
<td>1.45</td>
<td>0.21</td>
<td>1.38</td>
<td>1.44</td>
<td>0.23</td>
<td>1.00</td>
<td>2</td>
<td>1.00</td>
<td>-0.46</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

> scales <- scoreItems(epi.keys, epi, impute="none")
> describe(scales$scores)

<table>
<thead>
<tr>
<th>var</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>3516</td>
<td>1.46</td>
<td>0.17</td>
<td>1.46</td>
<td>1.46</td>
<td>0.19</td>
<td>1</td>
<td>2</td>
<td>0.20</td>
<td>-0.34</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>3514</td>
<td>1.54</td>
<td>0.19</td>
<td>1.54</td>
<td>1.55</td>
<td>0.19</td>
<td>1</td>
<td>2</td>
<td>-0.06</td>
<td>-0.47</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>3510</td>
<td>1.73</td>
<td>0.18</td>
<td>1.78</td>
<td>1.74</td>
<td>0.16</td>
<td>1</td>
<td>2</td>
<td>-0.56</td>
<td>-0.10</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Imp</td>
<td>3516</td>
<td>1.52</td>
<td>0.20</td>
<td>1.56</td>
<td>1.52</td>
<td>0.16</td>
<td>1</td>
<td>2</td>
<td>-0.16</td>
<td>-0.51</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Soc</td>
<td>3509</td>
<td>1.45</td>
<td>0.22</td>
<td>1.46</td>
<td>1.44</td>
<td>0.23</td>
<td>1</td>
<td>2</td>
<td>0.34</td>
<td>-0.52</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
The structure of scales can be found from correlation matrices

1. The typical use of scoring scales is from raw data.
2. But for those of us interested in large data matrices with lots of missing data, it is convenient to score from the correlation matrix level.
3. If we just care about the correlations of composite scales, the correlations are adequate.
4. Given a $N \times n$ data matrix of deviation scores, for $N$ subjects on $n$ items, $N X_n$, the covariance matrix of the $n$ items, $nC_n$, is just

$$nC_n = XX' \ast (N - 1)^{-1},$$

with variances, $\sigma_i^2$, on the diagonal of $C$ and item by item covariances, $\sigma_{ij}$, off the diagonal.
5. The covariances of scales are just

$$kCs_k = kK'_n nC_n nK_k.$$
Scales from correlations

R code

R <- lowerCor(epi, show=FALSE)
fromCors <- scoreItems(epi.keys, R)
summary(fromCors)

Call: scoreOverlap(keys = epi.keys, r = R)

Scale intercorrelations adjusted for item overlap
Scale intercorrelations corrected for attenuation
raw correlations (corrected for overlap) below the diagonal, (standardized) alpha on the diagonal,
corrected (for overlap and reliability) correlations above the diagonal:

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>N</th>
<th>L</th>
<th>Imp</th>
<th>Soc</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>0.73</td>
<td>-0.23</td>
<td>-0.38</td>
<td>0.799</td>
<td>0.94</td>
</tr>
<tr>
<td>N</td>
<td>-0.18</td>
<td>0.80</td>
<td>-0.28</td>
<td>0.049</td>
<td>-0.31</td>
</tr>
<tr>
<td>L</td>
<td>-0.22</td>
<td>-0.17</td>
<td>0.45</td>
<td>-0.311</td>
<td>-0.30</td>
</tr>
<tr>
<td>Imp</td>
<td>0.47</td>
<td>0.03</td>
<td>-0.14</td>
<td>0.474</td>
<td>0.54</td>
</tr>
<tr>
<td>Soc</td>
<td>0.68</td>
<td>-0.24</td>
<td>-0.17</td>
<td>0.320</td>
<td>0.73</td>
</tr>
</tbody>
</table>
**Item Response Theory from factor analysis**

```r
ab.irt <- irt.fa(ability)
plot(ab.irt)
plot(ab.irt,type="test")
```

Item Response Theory using factor analysis with Call: irt.fa(x = ability)

Test of the hypothesis that 1 factor is sufficient.
The degrees of freedom for the model is 104 and the objective function was 1.92
The number of observations was 1525 with Chi Square = 2906.16 with prob < 0

The root mean square of the residuals (RMSA) is 0.08
The df corrected root mean square of the residuals is 0.09

Tucker Lewis Index of factoring reliability = 0.722
RMSEA index = 0.133 and the 10 % confidence intervals are 0.129 0.137
BIC = 2143.86
Item Information for one factor of ability items

Item information from factor analysis

![Graph showing item information from factor analysis](image-url)
Test Information for one factor of ability items

Test information -- item parameters from factor analysis

![Graph showing test information and reliability](image-url)
Multiple types of reliability

1. Internal consistency estimates
   - $\alpha, \lambda_6$, use the `alpha` or `scoreItems` functions
   - $\omega_{hierarchical}$ and $\omega_{total}$ use the `omega` function

2. IntraClass coefficients
   - ICC

3. Rater agreement use `kappa` function

4. Test Retest reliability
For the next examples we will use a built in data set

1. bfi consists of 25 personality items measuring 5 factors as well as some demographics.
2. The data were collected as part of the SAPA project and have 2,800 subjects.
3. For help on this data set, ?bfi
4. To see all of the psych data sets: data(package="psych")
First, we intentionally mis-specify the data

```r
alpha(bfi[,1:5])  # score the first five items
```

Some items (A1) were negatively correlated with the total scale and probably should be reversed. To do this, run the function again with the 'check.keys=TRUE' option.

Reliability analysis

Call: alpha(x = bfi[,1:5])

```
raw_alpha          std.alpha    G6(smc) average_r  S/N      ase mean sd
0.43               0.46        0.53      0.15      0.85   0.016 4.2 0.74
```

**lower** 0.4  **alpha** 0.43 **upper** 0.46

95% confidence boundaries

Reliability if an item is dropped:

```
raw_alpha          std.alpha    G6(smc) average_r  S/N      ase mean  sd
A1     0.72           0.73        0.67      0.398     2.64 0.0087
A2     0.28           0.30        0.39      0.097     0.43 0.0219
A3     0.18           0.21        0.31      0.061     0.26 0.0249
A4     0.25           0.31        0.44      0.099     0.44 0.0229
A5     0.21           0.24        0.36      0.072     0.31 0.0238
```

Item statistics

```
n  raw.r std.r r.cor r.drop mean   sd
A1 2784 0.066 0.034  -0.39  -0.31  2.4  1.4
```
Try it again. Turn on automatic reversals. Get the scores

```r
scores <- alpha(bfi[1:5], check.keys = TRUE)

alpha(bfi[1:5], check.keys = TRUE)

Reliability analysis
Call: alpha(x = bfi[1:5], check.keys = TRUE)

<table>
<thead>
<tr>
<th>raw_alpha</th>
<th>std.alpha</th>
<th>G6(smc)</th>
<th>average_r</th>
<th>S/N</th>
<th>ase</th>
<th>mean</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>0.71</td>
<td>0.68</td>
<td>0.33</td>
<td>2.5</td>
<td>0.009</td>
<td>4.7</td>
<td>0.9</td>
</tr>
</tbody>
</table>

lower alpha upper 95% confidence boundaries
0.69 0.7 0.72

Reliability if an item is dropped:
<table>
<thead>
<tr>
<th>raw_alpha</th>
<th>std.alpha</th>
<th>G6(smc)</th>
<th>average_r</th>
<th>S/N</th>
<th>ase</th>
<th>mean</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1-</td>
<td>0.72</td>
<td>0.73</td>
<td>0.67</td>
<td>0.40</td>
<td>2.6</td>
<td>0.0087</td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>0.62</td>
<td>0.63</td>
<td>0.58</td>
<td>0.29</td>
<td>1.7</td>
<td>0.0119</td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td>0.60</td>
<td>0.61</td>
<td>0.56</td>
<td>0.28</td>
<td>1.6</td>
<td>0.0124</td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>0.69</td>
<td>0.69</td>
<td>0.65</td>
<td>0.36</td>
<td>2.3</td>
<td>0.0098</td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>0.64</td>
<td>0.66</td>
<td>0.61</td>
<td>0.32</td>
<td>1.9</td>
<td>0.0111</td>
<td></td>
</tr>
</tbody>
</table>

Warning message:
In alpha(bfi[1:5], check.keys = TRUE):
Some items were negatively correlated with total scale and were reversed.
R functions will return objects without necessarily telling you

1. The basic logic of R is that you can do lots of calculations, but you might not want all the output.

2. The output is there, to be processed by other functions if you want, but you probably don’t want to see all of it unless you ask.

3. Thus, alpha returns the scores based upon the scales you asked for, but doesn’t show them, because they are so many.

4. The str command tells you the structure of an object. The names will just list the names of the objects.
R code

```r
names(scores)
str(scores)
```

```r	names(scores)
[1] "total" "alpha.drop" "item.stats" "response.freq" "scores" "nvar" "boot.ci"
[9] "boot" "Unidim" "Fit" "call"

$ total : 'data.frame': 1 obs. of 8 variables:
  ..$ raw_alpha: num 0.703
  ..$ std.alpha: num 0.713
  ..$ G6(smc) : num 0.683
  ..$ average_r: num 0.332
  ..$ S/N : num 2.48
  ..$ ase : num 0.00895
  ..$ mean : num 4.65
  ..$ sd : num 0.898

$ alpha.drop : 'data.frame': 5 obs. of 6 variables:
  ..$ raw_alpha: num [1:5] 0.719 0.617 0.6 0.686 0.643
  ..$ std.alpha: num [1:5] 0.726 0.626 0.613 0.694 0.656
  ..$ G6(smc) : num [1:5] 0.673 0.579 0.558 0.65 0.605
  ..$ average_r: num [1:5] 0.398 0.295 0.284 0.361 0.322
  ..$ S/N : num [1:5] 2.64 1.67 1.58 2.26 1.9
  ..$ alpha se: num [1:5] 0.00873 0.0119 0.01244 0.00983 0.01115

$ item.stats : 'data.frame': 5 obs. of 7 variables:
  ..$ n : num [1:5] 2784 2773 2774 2781 2784
  ..$ raw.r : num [1:5] 0.581 0.728 0.76 0.654 0.687
  ..$ std.r : num [1:5] 0.566 0.748 0.767 0.631 0.699
  ..$ r.cor : num [1:5] 0.376 0.667 0.709 0.471 0.596
  ..$ r.drop: num [1:5] 0.308 0.564 0.587 0.394 0.489
  ..$ mean : num [1:5] 4.59 4.8 4.6 4.7 4.56
  ..$ sd : num [1:5] 1.41 1.17 1.3 1.48 1.26
```

---

names and `str` of alpha output
One of the objects of \texttt{alpha} is the scores object

\begin{verbatim}
R code
describe(scores$scores)
\end{verbatim}

But, since there are scores for all subjects, but just one score, this is not very interesting.

\begin{verbatim}
R code
describe(scores$scores)
\end{verbatim}

Note that \texttt{alpha} has the option of doing cumulative scores (adding up items, or scoring in the unit of the items (the default)).

\begin{verbatim}
R code
scores <- alpha(bfi[1:5],check.keys=TRUE,cumulative=TRUE)
#set the cumulative option to be true
describe(scores$scores)
\end{verbatim}
Better yet, score multiple scales at one time

```r
sc.bfi <- scoreItems(bfi.keys,bfi, impute="none")
sc.bfi  #lots of output
scores.bfi <- sc.bfi$scores
describe(scores.bfi)
```

<table>
<thead>
<tr>
<th>vars</th>
<th>n</th>
<th>mean</th>
<th>sd</th>
<th>median</th>
<th>trimmed</th>
<th>mad</th>
<th>min</th>
<th>max</th>
<th>range</th>
<th>skew</th>
<th>kurtosis</th>
<th>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>agree</td>
<td>1</td>
<td>2800</td>
<td>4.65</td>
<td>4.8</td>
<td>4.73</td>
<td>0.89</td>
<td>1.0</td>
<td>6</td>
<td>5.0</td>
<td>-0.76</td>
<td>0.40</td>
<td>0.02</td>
</tr>
<tr>
<td>conscientious</td>
<td>2</td>
<td>2800</td>
<td>4.27</td>
<td>4.4</td>
<td>4.30</td>
<td>0.89</td>
<td>1.0</td>
<td>6</td>
<td>5.0</td>
<td>-0.40</td>
<td>-0.19</td>
<td>0.02</td>
</tr>
<tr>
<td>extraversion</td>
<td>3</td>
<td>2800</td>
<td>4.15</td>
<td>4.2</td>
<td>4.20</td>
<td>1.19</td>
<td>1.0</td>
<td>6</td>
<td>5.0</td>
<td>-0.48</td>
<td>-0.21</td>
<td>0.02</td>
</tr>
<tr>
<td>neuroticism</td>
<td>4</td>
<td>2800</td>
<td>3.16</td>
<td>3.0</td>
<td>3.13</td>
<td>1.48</td>
<td>1.0</td>
<td>6</td>
<td>5.0</td>
<td>0.21</td>
<td>-0.67</td>
<td>0.02</td>
</tr>
<tr>
<td>openness</td>
<td>5</td>
<td>2800</td>
<td>4.59</td>
<td>4.6</td>
<td>4.62</td>
<td>0.89</td>
<td>1.2</td>
<td>6</td>
<td>4.8</td>
<td>-0.34</td>
<td>-0.29</td>
<td>0.02</td>
</tr>
</tbody>
</table>
sc.bfi #lots of ouput

Call: scoreItems(keys = bfi.keys, items = bfi, impute = "none")

(Standardized) Alpha:

<table>
<thead>
<tr>
<th>Agree</th>
<th>Conscientious</th>
<th>Extraversion</th>
<th>Neuroticism</th>
<th>Openness</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>0.7</td>
<td>0.73</td>
<td>0.76</td>
<td>0.81</td>
</tr>
</tbody>
</table>

Standard errors of unstandardized Alpha:

<table>
<thead>
<tr>
<th>Agree</th>
<th>Conscientious</th>
<th>Extraversion</th>
<th>Neuroticism</th>
<th>Openness</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASE</td>
<td>0.014</td>
<td>0.014</td>
<td>0.013</td>
<td>0.011</td>
</tr>
</tbody>
</table>

Standardized Alpha of observed scales:

<table>
<thead>
<tr>
<th>Agree</th>
<th>Conscientious</th>
<th>Extraversion</th>
<th>Neuroticism</th>
<th>Openness</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1,]</td>
<td>0.7</td>
<td>0.73</td>
<td>0.76</td>
<td>0.81</td>
</tr>
</tbody>
</table>

Average item correlation:

<table>
<thead>
<tr>
<th>Agree</th>
<th>Conscientious</th>
<th>Extraversion</th>
<th>Neuroticism</th>
<th>Openness</th>
</tr>
</thead>
<tbody>
<tr>
<td>average.r</td>
<td>0.32</td>
<td>0.35</td>
<td>0.39</td>
<td>0.47</td>
</tr>
</tbody>
</table>

Median item correlation:

<table>
<thead>
<tr>
<th>Agree</th>
<th>Conscientious</th>
<th>Extraversion</th>
<th>Neuroticism</th>
<th>Openness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.34</td>
<td>0.34</td>
<td>0.38</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Guttman 6* reliability:

<table>
<thead>
<tr>
<th>Agree</th>
<th>Conscientious</th>
<th>Extraversion</th>
<th>Neuroticism</th>
<th>Openness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lambda.6</td>
<td>0.7</td>
<td>0.72</td>
<td>0.77</td>
<td>0.81</td>
</tr>
</tbody>
</table>

Signal/Noise based upon av.r :

<table>
<thead>
<tr>
<th>Agree</th>
<th>Conscientious</th>
<th>Extraversion</th>
<th>Neuroticism</th>
<th>Openness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal/Noise</td>
<td>2.4</td>
<td>2.7</td>
<td>3.2</td>
<td>4.4</td>
</tr>
</tbody>
</table>

Scale intercorrelations corrected for attenuation

raw correlations below the diagonal, alpha on the diagonal

Note that these are the correlations of the complete scales based on the correlation matrix, not the observed scales based on the raw items.
Reliability is not just *alpha*. In fact it is more than internal consistency

See Revelle & Condon (2019) for a review.

1. Internal consistency
   - Some measures (e.g. $\alpha = \text{KR20} = \lambda_3$) were developed for desk calculators.
   - They just used item variances and total scale variances
   - Other internal consistency measures examined the factorial structure of the test (e.g. $\omega_h, \omega_t, \beta$)

2. Other measures include split-half reliabilities (there many split halves)

3. Perhaps better yet (but less common) are test-retest estimates.
\[\alpha, \omega_{\text{hierarchical}} \text{ and } \beta \text{ as alternative measures of internal consistency}\]

1. \(\alpha\) as the mean split half reliability
   - \texttt{alpha} to find \(\alpha\)
   - \texttt{splitHalf} to find all (if \(n \leq 16\)) or 10,000 random possible split half reliabilities (\(n > 16\))

2. \(\omega_{\text{hierarchical}}\) and \(\omega_{\text{total}}\) as factor based reliabilities
   - \(\omega_{\text{hierarchical}}\) estimates general factor saturation
   - Found using \texttt{omega} and \texttt{omegaSem}

3. \(\beta\) as worst split half reliability as an alternative estimate of the general factor saturation.
   - Found using a hierarchical clustering algorithm (\texttt{iclust}).
   - \texttt{iclust} is also useful for scale construction.
Using the `omega` function

```r
omega(ability, 4)
```

Omega
Call: `omega(m = ability, nfactors = 4)`
Alpha: 0.83
G.6: 0.84
Omega Hierarchical: 0.65
Omega H asymptotic: 0.76
Omega Total 0.86

Schmid Leiman Factor loadings greater than 0.2

<table>
<thead>
<tr>
<th>g</th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
<th>F4*</th>
<th>h2</th>
<th>u2</th>
<th>p2</th>
</tr>
</thead>
<tbody>
<tr>
<td>reason.4</td>
<td>0.50</td>
<td>0.27</td>
<td>0.34</td>
<td>0.66</td>
<td>0.73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reason.16</td>
<td>0.42</td>
<td>0.21</td>
<td>0.23</td>
<td>0.77</td>
<td>0.76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reason.17</td>
<td>0.55</td>
<td>0.47</td>
<td>0.52</td>
<td>0.48</td>
<td>0.57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>reason.19</td>
<td>0.44</td>
<td>0.21</td>
<td>0.25</td>
<td>0.75</td>
<td>0.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>letter.7</td>
<td>0.52</td>
<td>0.35</td>
<td>0.39</td>
<td>0.61</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>letter.33</td>
<td>0.46</td>
<td>0.30</td>
<td>0.31</td>
<td>0.69</td>
<td>0.70</td>
<td></td>
<td></td>
</tr>
<tr>
<td>letter.34</td>
<td>0.54</td>
<td>0.38</td>
<td>0.43</td>
<td>0.57</td>
<td>0.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>letter.58</td>
<td>0.47</td>
<td>0.20</td>
<td>0.28</td>
<td>0.72</td>
<td>0.78</td>
<td></td>
<td></td>
</tr>
<tr>
<td>matrix.45</td>
<td>0.40</td>
<td>0.66</td>
<td>0.59</td>
<td>0.41</td>
<td>0.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>matrix.46</td>
<td>0.40</td>
<td>0.26</td>
<td>0.24</td>
<td>0.76</td>
<td>0.65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>matrix.47</td>
<td>0.42</td>
<td>0.23</td>
<td>0.77</td>
<td>0.79</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>matrix.55</td>
<td>0.28</td>
<td>0.12</td>
<td>0.88</td>
<td>0.65</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rotate.3</td>
<td>0.36</td>
<td>0.61</td>
<td>0.50</td>
<td>0.50</td>
<td>0.26</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rotate.4</td>
<td>0.41</td>
<td>0.61</td>
<td>0.54</td>
<td>0.46</td>
<td>0.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rotate.6</td>
<td>0.40</td>
<td>0.49</td>
<td>0.41</td>
<td>0.59</td>
<td>0.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rotate.8</td>
<td>0.32</td>
<td>0.53</td>
<td>0.40</td>
<td>0.60</td>
<td>0.26</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

With eigenvalues of:

<table>
<thead>
<tr>
<th>g</th>
<th>F1*</th>
<th>F2*</th>
<th>F3*</th>
<th>F4*</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.04</td>
<td>1.32</td>
<td>0.46</td>
<td>0.42</td>
<td>0.55</td>
</tr>
</tbody>
</table>
Find $\alpha$ and all split half reliabilities of 5 Agreeableness items and 5 Conscientiousness items from the `bfi` data set included in `psych`.

```r
alpha(bfi[1:10]) #find alpha, let it automatically reverse items
sp <- splitHalf(bfi[1:10], key=c(1,9,10), raw=TRUE) #reverse 3 items
hists(sp$raw, breaks=51, main="Split Half reliabilities of bfi[1:10]"")
```

Reliability analysis
Call: `alpha(x = bfi[1:10])`

<table>
<thead>
<tr>
<th>raw_alpha</th>
<th>std.alpha</th>
<th>G6(smc)</th>
<th>average_r</th>
<th>S/N</th>
<th>ase</th>
<th>mean</th>
<th>sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.73</td>
<td>0.74</td>
<td>0.76</td>
<td>0.22</td>
<td>2.8</td>
<td>0.01</td>
<td>4.5</td>
<td>0.73</td>
</tr>
</tbody>
</table>

lower alpha upper 95% confidence boundaries
0.71 0.73 0.75

Split half reliabilities
Call: `splitHalf(r = bfi[1:10], key = c(1, 9, 10))`

Maximum split half reliability (lambda 4) = 0.81
Guttman lambda 6 = 0.76
Average split half reliability = 0.73
Guttman lambda 3 (alpha) = 0.74
Minimum split half reliability (beta) = 0.41
All possible split halves of 5 agreeableness and 5 conscientiousness items. Note the one worst one! This is not one construct.
test-retest data sets

Discussed in part by Revelle & Condon (2019)

epiR 474 participants took the Eysenck Personality Inventory twice

msqR 3032 unique participants took the MSQ at least once, 2753 at least twice, 446 three times, and 181 four times.

1. The obvious approach: score time 1 and time 2 and correlate them.

2. A somewhat more elegant approach, use testRetest
The obvious approach

```
epi1 <- selectBy(epiR, "time=1")
epi2 <- selectBy(epiR, "time=2")
scales1 <- scoreItems(epi.keys, epi1)
scales2 <- scoreItems(epi.keys, epi2)
summary(scales1)
summary(scales2)
```

```
summary(scales1)
Call: scoreItems(keys = epi.keys, items = epi1)
Scale intercorrelations corrected for attenuation
   raw correlations below the diagonal, (unstandardized) alpha on the diagonal
   corrected correlations above the diagonal:

     E   N   L  Imp  Soc
   E  0.77 -0.208 -0.36 1.219 1.14
   N -0.16  0.813 -0.39 -0.015 -0.28
   L -0.19 -0.218  0.39 -0.432 -0.15
  Imp 0.77 -0.009 -0.19  0.519  0.66
   Soc 0.87 -0.221 -0.08  0.417  0.76

> summary(scales2)
Call: scoreItems(keys = epi.keys, items = epi2)
Scale intercorrelations corrected for attenuation
   raw correlations below the diagonal, (unstandardized) alpha on the diagonal
   corrected correlations above the diagonal:

     E   N   L  Imp  Soc
   E  0.74 -0.271 -0.46 1.209 1.18
   N -0.21  0.796 -0.30 -0.074 -0.32
   L -0.25 -0.167  0.40 -0.593 -0.26
  Imp 0.73 -0.046 -0.26  0.488  0.62
   Soc 0.88 -0.251 -0.14  0.379  0.75
```
Correlate time 1 and time 2 scale scores

R code

```r
cor2(epi1.scales$scores, epi2.scales$scores)
```

<table>
<thead>
<tr>
<th></th>
<th>E</th>
<th>N</th>
<th>L</th>
<th>Imp</th>
<th>Soc</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>0.81</td>
<td>-0.16</td>
<td>-0.21</td>
<td>0.58</td>
<td>0.73</td>
</tr>
<tr>
<td>N</td>
<td>-0.14</td>
<td>0.80</td>
<td>-0.18</td>
<td>0.00</td>
<td>-0.18</td>
</tr>
<tr>
<td>L</td>
<td>-0.22</td>
<td>-0.16</td>
<td>0.65</td>
<td>-0.22</td>
<td>-0.11</td>
</tr>
<tr>
<td>Imp</td>
<td>0.60</td>
<td>-0.02</td>
<td>-0.21</td>
<td>0.70</td>
<td>0.38</td>
</tr>
<tr>
<td>Soc</td>
<td>0.73</td>
<td>-0.19</td>
<td>-0.11</td>
<td>0.35</td>
<td>0.81</td>
</tr>
</tbody>
</table>

Note that the test retest of extraversion (.81) and impulsivity (.70) are much higher than \( \alpha \) at time 1 or time 2.
```r
epi.test <- testRetest(epiR, keys=epi.keys$E)
```

**Call:** `testRetest(t1 = epiR, keys = epi.keys$E)`

**Number of subjects = 474**  **Number of items = 24**

Correlation of scale scores over time 0.82

Alpha reliability statistics for time 1 and time 2

<table>
<thead>
<tr>
<th>raw</th>
<th>G3</th>
<th>std</th>
<th>G3</th>
<th>G6</th>
<th>av.r</th>
<th>S/N</th>
<th>se</th>
<th>lower</th>
<th>upper</th>
<th>var.r</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time 1</td>
<td>0.77</td>
<td>0.77</td>
<td>0.80</td>
<td>0.12</td>
<td>3.34</td>
<td>0.11</td>
<td>0.35</td>
<td>0.98</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>Time 2</td>
<td>0.75</td>
<td>0.74</td>
<td>0.79</td>
<td>0.11</td>
<td>2.90</td>
<td>0.13</td>
<td>0.30</td>
<td>0.98</td>
<td>0.01</td>
<td></td>
</tr>
</tbody>
</table>

Mean between person, across item reliability = 0.54

Mean within person, across item reliability = 0.6
with standard deviation of 0.2

Mean within person, across item d2 = 0.18

R1F = 0.87 Reliability of average of all items for one time (Random time effects)

RkF = 0.93 Reliability of average of all items and both times (Fixed time effects)

R1R = 0.83 Generalizability of a single time point across all items (Random time effects)

Rc = 0.29 Generalizability of change (fixed time points, fixed items)

Multilevel components of variance

<table>
<thead>
<tr>
<th>variance</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>0.02</td>
</tr>
<tr>
<td>Time</td>
<td>0.00</td>
</tr>
<tr>
<td>Items</td>
<td>0.05</td>
</tr>
<tr>
<td>ID x time</td>
<td>0.00</td>
</tr>
<tr>
<td>ID x items</td>
<td>0.09</td>
</tr>
<tr>
<td>time x items</td>
<td>0.00</td>
</tr>
<tr>
<td>Residual</td>
<td>0.09</td>
</tr>
<tr>
<td>Total</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Do this again, for the Impulsivity scale

```r
imp.test <- testRetest(epiR, keys=epi.keys$Imp)
```

Test Retest reliability
Call: testRetest(t1 = epiR, keys = epi.keys$Imp)

Number of subjects = 474  Number of items = 9
Correlation of scale scores over time 0.7
Alpha reliability statistics for time 1 and time 2

<table>
<thead>
<tr>
<th>raw</th>
<th>G3</th>
<th>std G3</th>
<th>G6</th>
<th>av.r</th>
<th>S/N</th>
<th>se</th>
<th>lower</th>
<th>upper</th>
<th>var.r</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time 1</td>
<td>0.51</td>
<td>0.52</td>
<td>0.52</td>
<td>0.11</td>
<td>1.08</td>
<td>0.52</td>
<td>0.25</td>
<td>0.99</td>
<td>0.01</td>
</tr>
<tr>
<td>Time 2</td>
<td>0.51</td>
<td>0.52</td>
<td>0.52</td>
<td>0.11</td>
<td>1.07</td>
<td>0.51</td>
<td>0.25</td>
<td>0.99</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Mean between person, across item reliability = 0.52
Mean within person, across item reliability = 0.58
with standard deviation of 0.3

Mean within person, across item d2 = 0.2
R1F = 0.73 Reliability of average of all items for one time (Random time effects)
RkF = 0.85 Reliability of average of all items and both times (Fixed time effects)
R1R = 0.71 Generalizability of a single time point across all items (Random time effects)
Rc = 0.12 Generalizability of change (fixed time points, fixed items)

Multilevel components of variance

<table>
<thead>
<tr>
<th>variance</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>0.02</td>
</tr>
<tr>
<td>Time</td>
<td>0.00</td>
</tr>
<tr>
<td>Items</td>
<td>0.04</td>
</tr>
<tr>
<td>ID x time</td>
<td>0.00</td>
</tr>
<tr>
<td>ID x items</td>
<td>0.09</td>
</tr>
<tr>
<td>time x items</td>
<td>0.00</td>
</tr>
<tr>
<td>Residual</td>
<td>0.10</td>
</tr>
<tr>
<td>Total</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Item stats for the Impulsivity scale

The mean test retest correlation of the items (.52) is much higher than the mean communality of those items (.14).
Regression from correlation matrices

1. Although typically done from the raw data, multiple regression can also be done from the correlation matrix.
2. `lmCor` will take either raw data or correlation matrices as input and then do and show the multiple regressions.
3. Consider how the big 5 from the `spi` data set can predict 10 criteria.
Predicting 10 criteria from the Big 5 using the *spi* dataset

**R code**

```r
spi.scales <- scoreItems(spi.keys[1:5], spi) # score the scales
spi.scores.crit <- cbind(spi.scales$scores, spi[1:10])
spi.reg <- lmCor(x=1:5, y=6:15, data =spi.scores.crit, plot=FALSE)
spi.reg.r <- rbind(spi.reg$coefficients, R = spi.reg$R)
ord <- order(spi.reg$R) # sort the resulting regressions
df2latex(spi.reg.r[ord], big=.2) # convert to a LaTex table
```

**Table:** 10 criteria and 5 predictors from the *spi*

A table from the psych package in R

<table>
<thead>
<tr>
<th>Variable</th>
<th>p1edu</th>
<th>p2edu</th>
<th>ER</th>
<th>willns</th>
<th>smoke</th>
<th>edctn</th>
<th>exer</th>
<th>age</th>
<th>sex</th>
<th>helth</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Agree</td>
<td>0.02</td>
<td>0.01</td>
<td>-0.03</td>
<td>0.06</td>
<td>-0.08</td>
<td>0.12</td>
<td>-0.01</td>
<td>0.16</td>
<td>0.16</td>
<td>0.01</td>
</tr>
<tr>
<td>Consc</td>
<td>-0.03</td>
<td>-0.05</td>
<td>0.02</td>
<td>0.11</td>
<td>-0.08</td>
<td>0.06</td>
<td>0.16</td>
<td>0.13</td>
<td>0.10</td>
<td>0.17</td>
</tr>
<tr>
<td>Neuro</td>
<td>-0.04</td>
<td>-0.03</td>
<td>0.13</td>
<td>0.03</td>
<td>0.06</td>
<td>-0.15</td>
<td>-0.12</td>
<td>-0.14</td>
<td>0.29</td>
<td>-0.27</td>
</tr>
<tr>
<td>Extra</td>
<td>0.05</td>
<td>0.06</td>
<td>0.05</td>
<td>0.09</td>
<td>0.08</td>
<td>-0.09</td>
<td>0.09</td>
<td>-0.11</td>
<td>0.09</td>
<td>0.14</td>
</tr>
<tr>
<td>Open</td>
<td>0.06</td>
<td>0.06</td>
<td>-0.01</td>
<td>0.00</td>
<td>0.09</td>
<td>0.14</td>
<td>0.07</td>
<td>0.12</td>
<td>-0.12</td>
<td>0.01</td>
</tr>
<tr>
<td>R</td>
<td>0.10</td>
<td>0.11</td>
<td>0.13</td>
<td>0.17</td>
<td>0.18</td>
<td><strong>0.26</strong></td>
<td><strong>0.27</strong></td>
<td><strong>0.31</strong></td>
<td><strong>0.36</strong></td>
<td><strong>0.41</strong></td>
</tr>
</tbody>
</table>
Show just one prediction model

R code

```r
lmCor(health ~ Agree + Consc + Neuro + Extra + Open, 
data=spi.scores.crit)
```

multiple Regression from raw data

**DV = health**

<table>
<thead>
<tr>
<th></th>
<th>slope</th>
<th>se</th>
<th>t</th>
<th>p</th>
<th>lower.ci</th>
<th>upper.ci</th>
<th>VIF</th>
<th>Vy.x</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Intercept)</td>
<td>0.00</td>
<td>0.01</td>
<td>0.00</td>
<td>1.0e+00</td>
<td>-0.03</td>
<td>0.03</td>
<td>1.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Agree</td>
<td>0.01</td>
<td>0.02</td>
<td>0.41</td>
<td>6.8e-01</td>
<td>-0.02</td>
<td>0.04</td>
<td>1.12</td>
<td>0.00</td>
</tr>
<tr>
<td>Consc</td>
<td>0.17</td>
<td>0.02</td>
<td>11.34</td>
<td>2.3e-29</td>
<td>0.14</td>
<td>0.20</td>
<td>1.09</td>
<td>0.04</td>
</tr>
<tr>
<td>Neuro</td>
<td>-0.27</td>
<td>0.02</td>
<td>-18.00</td>
<td>9.7e-70</td>
<td>-0.30</td>
<td>-0.24</td>
<td>1.09</td>
<td>0.09</td>
</tr>
<tr>
<td>Extra</td>
<td>0.14</td>
<td>0.02</td>
<td>9.45</td>
<td>5.9e-21</td>
<td>0.11</td>
<td>0.17</td>
<td>1.10</td>
<td>0.03</td>
</tr>
<tr>
<td>Open</td>
<td>0.01</td>
<td>0.01</td>
<td>0.86</td>
<td>3.9e-01</td>
<td>-0.02</td>
<td>0.04</td>
<td>1.03</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Residual Standard Error = 0.91 with 3994 degrees of freedom

Multiple Regression

```
R   R2   Ruw  R2uw  Shrunken  R2  SE  of  R2  overall  F  df1  df2  p
health 0.41 0.16 0.35 0.12 0.16 0.01 156.95 5 3994 1.18e-152
```
Regression Models

0.24
-0.12
0.23
-0.19
0.07
-0.2
0.01
-0.12
0.13
0.01
0.01
0.17
0.14
0.27
0.12
0.13
We have known since the 1930’s the need to Cross validate

1. A lovely paper by Cureton (1950) makes this point in three pages.

2. crossValidation will apply a regression model from one sample to a second sample.

```r
N <- NROW(spi.scores.crit)
ss <- sample(N,N/2)
model <- lmCor(health ~ Agree + Consc + Neuro + Extra + Open, data=spi.scores.crit[ss,])
cv <- crossValidation(model, spi.scores.crit[-ss, ])
```

Cross Validation
Call: crossValidation(model = model, data = spi.scores.crit[-ss, ])
Validities from raw items or from the correlation matrix
Number of unique predictors used = 5

items mat
health 0.42 0.42
Correlations based upon item based regressions
[1] 0.42
Correlations based upon correlation matrix based regressions
[1] 0.42
Cross validation of multiple models

```r
ss <- sample(NROW(spi),NROW(spi)/2)
spi.reg <- lmCor(x=1:5, y=6:15, data =spi.scores.crit[ss,], plot=FALSE)
cv.reg <- crossValidation(spi.reg, data=spi.scores.crit[-ss,])
cv.reg

spi.reg <- lmCor(x=1:5, y=6:15, data =spi.scores.crit[ss,], plot=FALSE)
> cv.reg <- crossValidation(spi.reg, data=spi.scores.crit[-ss,])
> cv.reg
Cross Validation
Call: crossValidation(model = spi.reg, data = spi.scores.crit[-ss, ])

Validities from raw items or from the correlation matrix
Number of unique predictors used = 5

items  mat
age    0.33  0.33
sex    0.38  0.38
health 0.39  0.40
pledu  0.11  0.12
p2edu  0.12  0.12
education 0.26  0.26
wellness 0.15  0.15
exer   0.26  0.26
smoke  0.17  0.17
ER     0.13  0.13
```
Scale construction through “Machine Learning”

1. Supervised Learning was called item analysis in 1930

2. Need to cross validate is not a new concept.
   - Now called K-fold cross validation
     - With N = 2, this is the traditional cross validation of half the sample being derivation, half being validation.
     - Typical is 10 fold which means sample 90% validate on remaining 10% and repeat for successive slices.
   - Boot strap resampling
     - This samples N subjects (with replacement) from the N subjects.
     - This leads to 62.3 % of the subjects being ‘in the bag’ and 36.8% ‘out of bag”
     - fit the in bag subjects, validate on out of bag subjects.
### k.fold folds=10

**R code**

```r
bs <- bestScales(spi[11:145], criteria = spi[1:10], folds=10,
                 dictionary=spi.dictionary[c(2,6)])
```

Call = `bestScales(x = spi[11:145], criteria = spi[1:10], folds = 10,
                 dictionary = spi.dictionary[c(2, 6)])`

<table>
<thead>
<tr>
<th>Variable</th>
<th>Derivation Mean</th>
<th>Derivation SD</th>
<th>Validation Mean</th>
<th>Validation SD</th>
<th>Final Valid Mean</th>
<th>Final Wtd Mean</th>
<th>N Wtd</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>0.36</td>
<td>0.0055</td>
<td>0.352</td>
<td>0.042</td>
<td>0.35</td>
<td>0.36</td>
<td>1</td>
</tr>
<tr>
<td>sex</td>
<td>0.35</td>
<td>0.0072</td>
<td>0.351</td>
<td>0.054</td>
<td>0.35</td>
<td>0.35</td>
<td>1</td>
</tr>
<tr>
<td>health</td>
<td>0.44</td>
<td>0.0054</td>
<td>0.435</td>
<td>0.046</td>
<td>0.43</td>
<td>0.44</td>
<td>1</td>
</tr>
<tr>
<td>p1edu</td>
<td>0.13</td>
<td>0.0164</td>
<td>0.128</td>
<td>0.042</td>
<td>0.12</td>
<td>0.19</td>
<td>1</td>
</tr>
<tr>
<td>p2edu</td>
<td>0.11</td>
<td>0.0117</td>
<td>0.087</td>
<td>0.039</td>
<td>NA</td>
<td>0.19</td>
<td>1</td>
</tr>
<tr>
<td>education</td>
<td>0.30</td>
<td>0.0100</td>
<td>0.290</td>
<td>0.071</td>
<td>0.30</td>
<td>0.31</td>
<td>1</td>
</tr>
<tr>
<td>wellness</td>
<td>0.24</td>
<td>0.0052</td>
<td>0.221</td>
<td>0.045</td>
<td>0.23</td>
<td>0.24</td>
<td>1</td>
</tr>
<tr>
<td>exer</td>
<td>0.31</td>
<td>0.0088</td>
<td>0.289</td>
<td>0.038</td>
<td>0.30</td>
<td>0.32</td>
<td>1</td>
</tr>
<tr>
<td>smoke</td>
<td>0.27</td>
<td>0.0079</td>
<td>0.260</td>
<td>0.063</td>
<td>0.27</td>
<td>0.28</td>
<td>1</td>
</tr>
<tr>
<td>ER</td>
<td>0.16</td>
<td>0.0163</td>
<td>0.139</td>
<td>0.060</td>
<td>0.16</td>
<td>0.16</td>
<td>1</td>
</tr>
</tbody>
</table>
two example results

```R
bs

Criterion = age
Freq mean.r sd.r item
q_4296 10 -0.23 0.00 Tell a lot of lies. Honesty
q_501 10 -0.21 0.01 Cheat to get ahead. Honesty
q_4249 10 -0.21 0.01 Would call myself a nervous person. Anxiety
q_1024 10 -0.21 0.01 Hang around doing nothing. EasyGoingness
q_1452 10 -0.20 0.01 Neglect my duties. Industry
q_803 10 0.20 0.01 Express myself easily. EmotionalExpressiveness
q_1081 10 -0.20 0.00 Have difficulty expressing my feelings. EmotionalExpressiveness
q_808 9 -0.19 0.01 Fear for the worst. Anxiety

Criterion = health
Freq mean.r sd.r item
q_820 10 0.35 0.00 Feel comfortable with myself. WellBeing
q_2765 10 0.34 0.01 Am happy with my life. WellBeing
q_811 10 -0.34 0.01 Feel a sense of worthlessness or hopelessness. WellBeing
q_578 10 -0.34 0.01 Dislike myself. WellBeing
q_1371 10 0.31 0.01 Love life. WellBeing
q_56 10 0.28 0.01 Am able to control my cravings. SelfControl
q_1505 10 -0.27 0.00 Panic easily. Anxiety
q_4249 10 -0.26 0.00 Would call myself a nervous person. Anxiety
q_808 10 -0.26 0.00 Fear for the worst. Anxiety
```
Try 100 bootstrap resamplings

```r
bs <- bestScales(spi[11:145], criteria = spi[1:10], n.iter = 100,
                 dictionary = spi.dictionary[c(2, 6)])
```

<table>
<thead>
<tr>
<th></th>
<th>derivation.mean</th>
<th>derivation.sd</th>
<th>validation.m</th>
<th>validation.sd</th>
<th>final.valid</th>
<th>final.wtd</th>
<th>N.wtd</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td>0.37</td>
<td>0.018</td>
<td>0.354</td>
<td>0.026</td>
<td>0.33</td>
<td>0.36</td>
<td>10</td>
</tr>
<tr>
<td>sex</td>
<td>0.36</td>
<td>0.014</td>
<td>0.353</td>
<td>0.021</td>
<td>0.35</td>
<td>0.35</td>
<td>10</td>
</tr>
<tr>
<td>health</td>
<td>0.44</td>
<td>0.014</td>
<td>0.430</td>
<td>0.022</td>
<td>0.43</td>
<td>0.44</td>
<td>10</td>
</tr>
<tr>
<td>p1edu</td>
<td>0.16</td>
<td>0.032</td>
<td>0.116</td>
<td>0.027</td>
<td>NA</td>
<td>0.19</td>
<td>1</td>
</tr>
<tr>
<td>p2edu</td>
<td>0.15</td>
<td>0.032</td>
<td>0.085</td>
<td>0.028</td>
<td>NA</td>
<td>0.19</td>
<td>1</td>
</tr>
<tr>
<td>education</td>
<td>0.32</td>
<td>0.018</td>
<td>0.279</td>
<td>0.027</td>
<td>0.18</td>
<td>0.30</td>
<td>1</td>
</tr>
<tr>
<td>wellness</td>
<td>0.24</td>
<td>0.016</td>
<td>0.219</td>
<td>0.023</td>
<td>0.23</td>
<td>0.24</td>
<td>1</td>
</tr>
<tr>
<td>exer</td>
<td>0.32</td>
<td>0.016</td>
<td>0.300</td>
<td>0.027</td>
<td>0.30</td>
<td>0.31</td>
<td>1</td>
</tr>
<tr>
<td>smoke</td>
<td>0.28</td>
<td>0.019</td>
<td>0.258</td>
<td>0.027</td>
<td>0.28</td>
<td>0.28</td>
<td>1</td>
</tr>
<tr>
<td>ER</td>
<td>0.17</td>
<td>0.024</td>
<td>0.128</td>
<td>0.024</td>
<td>NA</td>
<td>0.16</td>
<td>1</td>
</tr>
</tbody>
</table>
With items and the number of bootstrap samples they were included

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Freq</th>
<th>mean.r</th>
<th>sd.r</th>
<th>item</th>
</tr>
</thead>
<tbody>
<tr>
<td>age</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>q_4296</td>
<td>100</td>
<td>-0.23</td>
<td>0.01</td>
<td>Tell a lot of lies. Honesty</td>
</tr>
<tr>
<td>q_501</td>
<td>99</td>
<td>-0.21</td>
<td>0.01</td>
<td>Cheat to get ahead. Honesty</td>
</tr>
<tr>
<td>q_4249</td>
<td>98</td>
<td>-0.21</td>
<td>0.02</td>
<td>Would call myself a nervous person. Anxiety</td>
</tr>
<tr>
<td>q_1024</td>
<td>92</td>
<td>-0.21</td>
<td>0.02</td>
<td>Hang around doing nothing. EasyGoingness</td>
</tr>
<tr>
<td>health</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>q_820</td>
<td>100</td>
<td>0.35</td>
<td>0.01</td>
<td>Feel comfortable with myself. WellBeing</td>
</tr>
<tr>
<td>q_2765</td>
<td>100</td>
<td>0.35</td>
<td>0.01</td>
<td>Am happy with my life. WellBeing</td>
</tr>
<tr>
<td>q_811</td>
<td>100</td>
<td>-0.34</td>
<td>0.01</td>
<td>Feel a sense of worthlessness or hopelessness. WellBeing</td>
</tr>
<tr>
<td>q_578</td>
<td>100</td>
<td>-0.34</td>
<td>0.02</td>
<td>Dislike myself. WellBeing</td>
</tr>
<tr>
<td>q_1371</td>
<td>100</td>
<td>0.31</td>
<td>0.02</td>
<td>Love life. WellBeing</td>
</tr>
<tr>
<td>q_56</td>
<td>97</td>
<td>0.28</td>
<td>0.02</td>
<td>Am able to control my cravings. SelfControl</td>
</tr>
<tr>
<td>q_1505</td>
<td>100</td>
<td>-0.27</td>
<td>0.01</td>
<td>Panic easily. Anxiety</td>
</tr>
<tr>
<td>q_4249</td>
<td>91</td>
<td>-0.26</td>
<td>0.01</td>
<td>Would call myself a nervous person. Anxiety</td>
</tr>
</tbody>
</table>
Making up data: Be sure to call this simulation!

1. Multiple simulation functions

2. See ?sim for a list
   
   sim    Create a factor simplex
   sim.simplex A data simplex
   sim.hierarchical A hierarchical factor structure
   simCor Create data from a specified correlation matrix
   sim.parallel To compare alternative factor solutions
   sim.structural Simulate complex structural models

See the help pages for details.
Read the vignettes.
Results returned

1. Most (but not all) simulation functions return:
   - A raw data set
   - The model used for simulation
   - ‘True scores’ used to create the data
Miscellaneous functions that are useful

**vJoin**  Merge two files by rownames and column names

**scrub**  Clean up data

**df2latex**  One of several functions to create \LaTeX{} tables.

**read.file** and **read.clipboard** for convenient input

1. An introduction to the *psych* package: Part I.
2. Intro: Part II: Scale construction and psychometrics
3. Installing Rand the *psych* package
4. Using R and psych to find $\omega$
5. How To: Use *psych* for factor analysis and data reduction
6. Using R to score personality scales
7. Using *psych* for regression and mediation analysis


Velicer, W. (1976). Determining the number of components from
