An introduction to R
Sponsored by
The Association of Psychological Science
and
Society of Multivariate Experimental Psychology

William Revelle

Department of Psychology
Northwestern University
Evanston, Illinois USA
Outline

1. What is R?
   - Where did it come from, why use it?
   - Installing R on your computer and adding packages
   - Basic R capabilities: Calculation, Statistical tables, Graphics

2. A brief example
   - A brief example of exploratory and confirmatory data analysis

3. Basic statistics and graphics
   - 4 steps: read, explore, test, graph
   - Basic descriptive and inferential statistics
     - t-test, ANOVA, $\chi^2$
     - Linear Regression

4. Psychometrics and beyond
   - Classical Test measures of reliability
   - Multivariate Analysis and Structural Equation Modeling
   - Item Response Theory

5. Basic R commands
   - Useful functions
R: Statistics for all us

1. What is it?
2. Why use it?
3. Common (mis)perceptions of R
4. Examples for psychologists
   - graphical displays
   - basic statistics
   - advanced statistics
   - Although programming is easy in R, that is beyond the scope of today
R: What is it?

1. R: An international collaboration
2. R: The open source - public domain version of S+
3. R: Written by statistician (and all of us) for statisticians (and the rest of us)
4. R: Not just a statistics system, also an extensible language.
   - This means that as new statistics are developed they tend to appear in R far sooner than elsewhere.
   - R facilitates asking questions that have not already been asked.
Statistical Programs for Psychologists

- General purpose programs
  - R
  - S+
  - SAS
  - SPSS
  - STATA
  - Systat

- Specialized programs
  - Mx
  - EQS
  - AMOS
  - LISREL
  - MPlus
  - Your favorite program
Statistical Programs for Psychologists

- General purpose programs
  - R
  - $+$
  - $A$
  - $P$
  - $TATA$
  - $y tat$

- Specialized programs
  - Mx (OpenMx is part of R)
  - EQ$
  - AMO$
  - LI$REL$
  - MPIu$
  - Your favorite program
R: A way of thinking

- “R is the lingua franca of statistical research. Work in all other languages should be discouraged.”
- “This is R. There is no if. Only how.”
- “Overall, SAS is about 11 years behind R and S-Plus in statistical capabilities (last year it was about 10 years behind) in my estimation.”

Taken from the R.-fortunes (selections from the R.-help list serve)
R is open source, how can you trust it?

Q: “When you use it [R], since it is written by so many authors, how do you know that the results are trustable?”

A: “The R engine [...] is pretty well uniformly excellent code but you have to take my word for that. Actually, you don’t. The whole engine is open source so, if you wish, you can check every line of it. If people were out to push dodgy software, this is not the way they’d go about it.”

Q: Are R packages bug free?

A: No. But bugs are fixed rapidly when identified.

Q: How does function x work? May I adapt it for my functions.

What is R?: Technically

- R is an open source implementation of S (S-Plus is a commercial implementation)
- R is available under GNU Copy-left
- The current version of R is 2.15.0
- R is a group project run by a core group of developers (with new releases semiannually)

(Adapted from Robert Gentleman)
R: A brief history

- 1991-93: Ross Ihaka and Robert Gentleman begin work on R project at U. Auckland
- 1995: R available by ftp under the GPL
- 96-97: mailing list and R core group is formed
- 2000: John Chambers, designer of S joins the Rcore (wins a prize for best software from ACM for S)
- 2001-2013: Core team continues to improve base package with a new release every 6 months.
- Many others contribute “packages” to supplement the functionality for particular problems
  - 2003-04-01: 250 packages
  - 2004-10-01: 500 packages
  - 2007-04-12: 1,000 packages
  - 2009-10-04: 2,000 packages
  - 2011-05-12: 3,000 packages
  - 2012-05-12: 3,786 packages
Misconception: R is hard to use

1. R doesn’t have a GUI (Graphical User Interface)
   - Partly true, many use syntax.
   - Partly not true, GUIs exist (e.g., R Commander, R-Studio).
   - Quasi GUIs for Mac and PCs make syntax writing easier.

2. R syntax is hard to use
   - Not really, unless you think an iPhone is hard to use.
   - Easier to give instructions of 1-4 lines of syntax rather than pictures of what menu to pull down.
   - Keep a copy of your syntax, modify it for the next analysis.

3. R is not user friendly: A personological description of R
   - R is introverted: it will tell you what you want to know if you ask, but not if you don’t ask.
   - R is conscientious: it wants commands to be correct.
   - R is not agreeable: its error messages are at best cryptic.
   - R is stable: it does not break down under stress.
   - R is open: new ideas about statistics are easily developed.
Misconceptions: R is hard to learn – some interesting facts

1. With a brief web based tutorial
   [http://personality-project.org/r](http://personality-project.org/r), 2nd and 3rd year undergraduates in psychological methods and personality research courses are using R for descriptive and inferential statistics and producing publication quality graphics.

2. More and more psychology departments are using it for graduate and undergraduate instruction.

3. R is easy to learn, hard to master
   - R-help newsgroup is very supportive
   - Multiple web based and pdf tutorials see (e.g., [http://www.r-project.org/](http://www.r-project.org/))
   - Short courses using R for many applications

4. Books and websites for SPSS and SAS users trying to learn R (e.g., [http://r4stats.com/](http://r4stats.com/)) by Bob Muenchen (look for link to free version).
Ok, how do I get it: Getting started with R

- Download from R Cran (http://cran.r-project.org/)
  - Choose appropriate operating system and download compiled R
- Install R (current version is 2.15.0)
- Start R
- Add useful packages (just need to do this once)
  - install.packages("ctv") #this downloads the task view package
  - library(ctv) #this activates the cvt package
  - install.views("Psychometrics") #among others
  - Take a 5 minute break
- Activate the package(s) you want to use today (e.g., psych)
  - library(psyh) #necessary for most of today’s examples
- Use R
Go to the R.project.org

Getting Started:

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To download R, please choose your preferred CRAN mirror.
- If you have questions about R like how to download and install the software, or which license terms are, please read our answers to frequently asked questions before you send an email.

News:

- R version 2.15.0 (Easter Beagle) has been released on 2012-03-30.
- R version 2.14.2 (Gift-Giving Season) has been released on 2012-02-29.
- The R Journal Vol.3/2 is available.
- useR! 2012, will take place at Vanderbilt University, Nashville Tennessee, USA, June 12-15, 2012.
Installing R on your computer and adding packages

Go to the Comprehensive R Archive Network (CRAN)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages. Windows and Mac users most likely want one of these versions of R:

- Download R for Linux
- Download R for MacOS X
- Download R for Windows

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.
Download and install the appropriate version – PC

Subdirectories:

- **base**: Binaries for base distribution (managed by Duncan Murdoch). This is what you want to install R for the first time.
- **contrib**: Binaries of contributed packages (managed by Uwe Ligges). There is also information on third party software available for CRAN Windows services and corresponding environment and make variables.
- **Rtools**: Tools to build R and R packages (managed by Duncan Murdoch). This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Duncan Murdoch or Uwe Ligges directly in case of questions/suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.
Installing R on your computer and adding packages

Download and install the appropriate version – PC

R-2.15.0 for Windows (32/64 bit)

Download R 2.15.0 for Windows (47 megabytes, 32/64 bit)

Installation and other instructions
New features in this version: Windows specific, all platforms.

If you want to double-check that the package you have downloaded exactly matches the package distributed by R, you can compare the md5sum of the .exe to the true fingerprint. You will need a version of md5sum for windows; both graphical and command line versions are available.

Frequently asked questions

- How do I install R when using Windows Vista?
- How do I update packages in my previous version of R?
- Should I run 32-bit or 64-bit R?

Please see the R FAQ for general information about R and the R Windows FAQ for Windows-specific information.

Other builds

- Patches to this release are incorporated in the r-patched snapshot build.
- A build of the development version (which will eventually become the next major release of R) is available in the r-devel snapshot build.
- Previous releases

Note to webmasters: A stable link which will redirect to the current Windows binary release is <CRAN MIRROR>/bin/windows/base/release.htm.

Last change: 2012-03-30, by Duncan Murdoch
Download and install the appropriate version – Mac

R for Mac OS X

This directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.5 and above). Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but you can find the last supported release of R for these systems (which is R 1.1.7) here. Releases for old Mac OS X systems (through Mac OS X 10.4) can be found in the old directory.

Note: CRAN does not support Mac OS X systems and cannot check these binaries for viruses. Although we take precautions when assembling binaries, use the normal precautions with downloaded executables.

Universal R 2.15.0 released on 2012/03/30

This binary distribution of R and the GUI supports PowerPC (32-bit) and Intel (32-bit and 64-bit) based Macs on Mac OS X 10.5 (Leopard), 10.6 (Snow Leopard) and 10.7 (Lion). It is possibly the last distribution supporting Mac OS X 10.5 (Leopard) and PowerPC architecture.

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example type

```
find . -name R-2.15.0.pkg
```

in the Terminal application to print the MD5 checksum for the R-2.15.0.pkg image.

Files:

- **R-2.15.0.pkg** (latest version)
  - MD5-hash: 287c225e0d1d0f358f296d6a923c6092
  - (ca. 64MB)

Three-way universal binary of R 2.15.0 for Mac OS X 10.5 (Leopard) and higher. Contains R 2.15.0 framework, R.app GUI 1.51 in 32-bit and 64-bit.

The above file is an Installer package which can be installed by double-clicking. Depending on your browser, you may need to press the control key and click on this link to download the file.

This package only contains the R framework, 32-bit GUI (R.app) and 64-bit GUI (R64.app). For Tcl/Tk libraries (needed if you want to use tcltk) and GNU Fortran (needed if you want to compile packages from sources that contain FORTRAN code) please see the tools directory.

- **Mac-GUI-1.51.tar.gz**
  - MD5-hashes: 506089a2b3a2165439b660302501584

Sources for the R.app GUI 1.51 for Mac OS X. This file is only needed if you want to join the development of the GUI, it is not intended for regular users. Read the INSTALL file for further instructions.

- **NEWS (for Mac GUI)**

News features and changes in the R app Mac GUI

The new R.app Cocoa GUI has been written by Simon Urbanek and Stefano Iacus with contributions from many developers and translators worldwide, see "About R" in the GUI.
Starting R on a PC

R version 2.15.0 (2012-03-30)
Copyright (C) 2012 The R Foundation for Statistical Computing
ISEN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
Start up R and get ready to play (Mac version)

R version 2.15.0 Patched (2012-03-30 r58887) -- "Easter Beagle"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.43 (5920) x86_64-apple-darwin9.8.0]
Annotated installation guide: don’t type the >

> `install.packages("ctv")`

> `library(ctv)`

> `install.views("Psychometrics")`

#or just install a few packages

> `install.packages("psych")`

> `install.packages("GPArotation")`

> `install.packages("MASS")`

> `install.packages("mvtnorm")`

- Install the task view installer package. You might have to choose a “mirror” site.
- Make it active
- Install all the packages in the “Psychometrics” task view. This will take a few minutes.
- Or, just install one package (e.g., psych)
  as well as a few suggested packages that add functionality for factor rotation, multivariate normal distributions, etc.
Installing just the psych package

R version 2.13.0 (2011-04-13)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: i386-pc-mingw32/i386 (32-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and 'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("psych")
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.stat.ucla.edu/bin/windows/contrib/2.13/psych_1.0-97.zip'
Content type 'application/zip' length 1952216 bytes (1.9 Mb)
opened URL
downloaded 1.9 Mb
Or, install and use `ctv` package to load a task view on a PC

```r
> install.packages("ctv")
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.stat.ucla.edu/bin/windows/contrib/2.13/ctv_0.7-2.zip'
Content type 'application/zip' length 298753 bytes (291 Kb)
opened URL
downloaded 291 Kb

package 'ctv' successfully unpacked and MD5 sums checked

The downloaded packages are in
   C:\users\revelle\Temp\RtmpWnZUt\downloaded_packages

> library(ctv)
>
```
Check the version number for R (should be $\geq 2.15.0$) and for psych ($\geq 1.2.5$)

```r
> library(psych)
> sessionInfo()

R version 2.15.0 Patched (2012-03-30 r58887)
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)

locale:

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

other attached packages:
[1] MASS_7.3-17 GPArotation_2010.07-1 psych_1.2.5
R is extensible: The use of “packages”

- More than 3750 packages are available for R (and growing daily)
- Can search all packages that do a particular operation by using the sos package
  - `install.packages("sos")` #if you haven’t already
  - `library(sos)` # make it active once you have it
  - `findFn("X")` #will search a web data base for all packages/functions that have "X"
  - `findFn("factor analysis")` #will return 9881 matches and reports the top 400
  - `findFn("Item Response Theory")` # will return 199 matches
  - `findFn("INDSCAL ")` # will return 8 matches.
- `install.packages("X")` will install a particular package (add it to your R library – you need to do this just once)
- `library(X)` #will make the package X available to use if it has been installed (and thus in your library)
A small subset of very useful packages

- **General use**
  - core R
  - MASS
  - lattice
  - lme4 (core)
  - psych
  - Zelig

- **Special use**
  - ltm
  - sem
  - lavaan
  - OpenMx
  - GPArotation
  - mvtnorm
  - > 3750 known
  - + ?

- **General applications**
  - most descriptive and inferential stats
  - Modern Applied Statistics with S
  - Lattice or Trellis graphics
  - Linear mixed-effects models
  - Personality and psychometrics
  - General purpose toolkit

- **More specialized packages**
  - Latent Trait Model (IRT)
  - SEM and CFA (one group)
  - SEM and CFA (multiple groups)
  - SEM and CFA (multiple groups+)
  - Jennrich rotations
  - Multivariate distributions
  - Thousands of more packages on CRAN
  - Code on webpages/journal articles
Questions?
Basic R commands – remember don’t enter the >

R is just a fancy calculator. Add, subtract, sum, products, group

> 2 + 2

[1] 4

> 3^4

[1] 81

> sum(1:10)

[1] 55

> prod(c(1, 2, 3, 5, 7))

[1] 210

It is also a statistics table (the normal distribution, the t distribution)

> pnorm(q = 1)

[1] 0.8413447

> pt(q = 2, df = 20)

[1] 0.9703672
R is a set of distributions. Don’t buy a stats book with tables!

**Table:** To obtain the density, prefix with *d*, probability with *p*, quantiles with *q* and to generate random values with *r*. (e.g., the normal distribution may be chosen by using *dnorm*, *pnorm*, *qnorm*, or *rnorm.*)

<table>
<thead>
<tr>
<th>Distribution</th>
<th>base name</th>
<th>P 1</th>
<th>P 2</th>
<th>P 3</th>
<th>example application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>norm</td>
<td>mean</td>
<td>sigma</td>
<td></td>
<td>Most data</td>
</tr>
<tr>
<td>Multivariate normal</td>
<td>mvnorm</td>
<td>mean</td>
<td>r</td>
<td>sigma</td>
<td>Most data</td>
</tr>
<tr>
<td>Log Normal</td>
<td>lnorm</td>
<td>log mean</td>
<td>log sigma</td>
<td></td>
<td>income or reaction time</td>
</tr>
<tr>
<td>Uniform</td>
<td>unif</td>
<td>min</td>
<td>max</td>
<td></td>
<td>rectangular distributions</td>
</tr>
<tr>
<td>Binomial</td>
<td>binom</td>
<td>size</td>
<td>prob</td>
<td></td>
<td>Bernoulli trials (e.g., coin flips)</td>
</tr>
<tr>
<td>Student’s <em>t</em></td>
<td><em>t</em></td>
<td>df</td>
<td></td>
<td></td>
<td>Finding significance of a <em>t</em>-test</td>
</tr>
<tr>
<td>Multivariate <em>t</em></td>
<td>mvt</td>
<td>df</td>
<td>corr</td>
<td>nc</td>
<td>Multivariate applications</td>
</tr>
<tr>
<td>Fisher’s <em>F</em></td>
<td><em>f</em></td>
<td>df1</td>
<td>df2</td>
<td>nc</td>
<td>Testing for significance of <em>F</em> test</td>
</tr>
<tr>
<td><em>χ²</em></td>
<td>chisq</td>
<td>df</td>
<td></td>
<td>nc</td>
<td>Testing for significance of <em>χ²</em></td>
</tr>
<tr>
<td>Exponential</td>
<td>exp</td>
<td>rate</td>
<td></td>
<td></td>
<td>Exponential decay</td>
</tr>
<tr>
<td>Gamma</td>
<td>gamma</td>
<td>shape</td>
<td>rate</td>
<td>scale</td>
<td>distribution theory</td>
</tr>
<tr>
<td>Hypergeometric</td>
<td>hyper</td>
<td>m</td>
<td>n</td>
<td>k</td>
<td></td>
</tr>
<tr>
<td>Logistic</td>
<td>logis</td>
<td>location</td>
<td>scale</td>
<td></td>
<td>Item Response Theory</td>
</tr>
<tr>
<td>Poisson</td>
<td>pois</td>
<td>lambda</td>
<td></td>
<td></td>
<td>Count data</td>
</tr>
<tr>
<td>Weibull</td>
<td>weibull</td>
<td>shape</td>
<td>scale</td>
<td></td>
<td>Reaction time distributions</td>
</tr>
</tbody>
</table>
A very small list of the many data sets available

> data()

This opens up a separate text window and lists all of the data sets in the currently loaded packages.

> data(package="psych")

Show the data sets available in a particular package (e.g., psych).

> data(Titanic)

> ? Titanic

Gets the particular data set with its help file (e.g., the survival rates on the Titanic cross classified by age, gender and class).

> data(cushny)

> ? cushney

Another original data set used by “student” (Gossett) for the t-test.
R can draw distributions

curve(dnormal(x), -3, 3, ylab = "probability of x", main = "A normal curve")

A normal curve
R can draw more interesting distributions

- **The normal curve**
- **Log normal**
- **Chi Square distribution**
- **Normal and t with 4 df**
The first line draws the normal curve, the second prints the title, the next lines draw the cross hatching.

```r
op <- par(mfrow=c(2,2))  # set up a 2 x 2 graph
curve(dnorm(x),-3,3,xlab="",ylab="Probability of z")
title(main="The normal curve",outer=FALSE)
xvals <- seq(-3,-2,length=100)
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=2,angle=-45)
xvals <- seq(-2,-1,length=100)
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=14,angle=45)
xvals <- seq(-1,-0,length=100)
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=34,angle=-45)
xvals <- seq(2,3,length=100)
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=2,angle=45)
xvals <- seq(1,2,length=100)
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=14,angle=-45)
xvals <- seq(0,1,length=100)
dvals <- dnorm(xvals)
polygon(c(xvals,rev(xvals)),c(rep(0,100),rev(dvals)),density=34,angle=45)
curve(dnorm(x),0,5,ylab='Probability of log(x)',main='Log normal')
curve(dchisq(x,1),0,5,ylab='Probability of Chi Sq',xlab='Chi Sq',main='Chi Square distribution')
curve(dnorm(x),-4,4,ylab='Probability of z or t',xlab='z or t',main='Normal and t with 4 df')
curve(dt(x,4),add=TRUE)
op <- par(mfrow=c(1,1))
```
A simple scatter plot using `plot`

```r
plot(iris[1:2], xlab="Sepal.Length", ylab="Sepal.Width", main="Fisher Iris data")
```
A simple scatter plot using `plot` with some colors

```r
```

1. Set parameters
2. bg for background colors
3. pch chooses the plot character
A scatter plot matrix plot with loess regressions using `pairs.panels`

1. Correlations above the diagonal
2. Diagonal shows histograms and densities
3. Scatter plots below the diagonal with correlation ellipse
4. Locally smoothed (loess) regressions for each pair
5. Optional color coding of grouping variables.

```r
pairs.panels(iris[1:4], bg=c("red", "yellow", "blue"), iris$Species, pch=21, main="Fisher Iris data by Species")
```
A brief example of exploratory and confirmatory data analysis

A brief example with real data

1. Get the data
2. Descriptive statistics
   - Graphic
   - Numerical
3. Inferential statistics using the linear model
   - regressions
4. More graphic displays
Get the data and describe it

1. First read the data, either from a built-in data set, a local file, a remote file, or from the clipboard.
2. Describe the data using the `describe` function from `psych`

```r
> my.data <- sat.act  # an example data file that is part of psych
# or
> file.name <- file.choose()  # look for it on your hard drive
# or
> file.name <- "http://personality-project.org/r/aps/sat.act.txt"
# now read it
> my.data <- read.table(file.name, header=TRUE)
# or
> my.data <- read.clipboard()  # if you have copied the data to the clipboard
> describe(my.data)  # report basic descriptive statistics
```

<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>gender</td>
<td>1</td>
<td>1.65</td>
<td>0.48</td>
<td>2</td>
<td>1.68</td>
<td>0.00</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>-0.61</td>
<td>-1.62</td>
<td>0.02</td>
</tr>
<tr>
<td>education</td>
<td>2</td>
<td>3.16</td>
<td>1.43</td>
<td>3</td>
<td>3.31</td>
<td>1.48</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>-0.68</td>
<td>-0.06</td>
<td>0.05</td>
</tr>
<tr>
<td>age</td>
<td>3</td>
<td>25.59</td>
<td>9.50</td>
<td>22</td>
<td>23.86</td>
<td>5.93</td>
<td>13</td>
<td>65</td>
<td>52</td>
<td>1.64</td>
<td>2</td>
<td>0.36</td>
</tr>
<tr>
<td>ACT</td>
<td>4</td>
<td>28.55</td>
<td>4.82</td>
<td>29</td>
<td>28.84</td>
<td>4.45</td>
<td>3</td>
<td>36</td>
<td>33</td>
<td>-0.66</td>
<td>0</td>
<td>0.18</td>
</tr>
<tr>
<td>SATV</td>
<td>5</td>
<td>612.23</td>
<td>112.90</td>
<td>620</td>
<td>619.45</td>
<td>118.61</td>
<td>200</td>
<td>800</td>
<td>600</td>
<td>-0.64</td>
<td>0</td>
<td>4.27</td>
</tr>
<tr>
<td>SATQ</td>
<td>6</td>
<td>610.22</td>
<td>115.64</td>
<td>620</td>
<td>617.25</td>
<td>118.61</td>
<td>200</td>
<td>800</td>
<td>600</td>
<td>-0.59</td>
<td>0</td>
<td>4.41</td>
</tr>
</tbody>
</table>
A brief example of exploratory and confirmatory data analysis.

Graphic display of data using `pairs.panels`

```r
pairs.panels(my.data)  # Note the outlier for ACT
```

![Graphical display of data using `pairs.panels`]

- **gender**: 0.09 -0.02 -0.04 -0.02 -0.17
- **education**: 0.55 0.15 0.05 0.03
- **age**: 0.11 -0.04 -0.03
- **ACT**: 0.56 0.59
- **SATV**: 0.64
- **SATQ**: 1.0 1.4 1.8

- **education**
- **age**
- **ACT**
- **SATV**
- **SATQ**
**Clean up the data using scrub**

```r
> cleaned <- scrub(my.data,"ACT",min=4)
> describe(cleaned)
```

<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>gender</td>
<td>1</td>
<td>700</td>
<td>1.65</td>
<td>2</td>
<td>1.68</td>
<td>0.00</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>-0.61</td>
<td>-1.62</td>
<td>0.02</td>
</tr>
<tr>
<td>education</td>
<td>2</td>
<td>700</td>
<td>3.16</td>
<td>3</td>
<td>3.31</td>
<td>1.48</td>
<td>5</td>
<td>5</td>
<td>-0.68</td>
<td>-0.06</td>
<td>0.05</td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>3</td>
<td>700</td>
<td>25.59</td>
<td>22</td>
<td>23.86</td>
<td>5.93</td>
<td>13</td>
<td>65</td>
<td>52</td>
<td>1.64</td>
<td>2.47</td>
<td>0.36</td>
</tr>
<tr>
<td>ACT</td>
<td>4</td>
<td>699</td>
<td>28.58</td>
<td>29</td>
<td>28.85</td>
<td>4.45</td>
<td>15</td>
<td>36</td>
<td>21</td>
<td>-0.50</td>
<td>-0.36</td>
<td>0.18</td>
</tr>
<tr>
<td>SATV</td>
<td>5</td>
<td>700</td>
<td>612.23</td>
<td>620</td>
<td>619.45</td>
<td>118.61</td>
<td>200</td>
<td>800</td>
<td>600</td>
<td>-0.64</td>
<td>0.35</td>
<td>4.27</td>
</tr>
<tr>
<td>SATQ</td>
<td>6</td>
<td>687</td>
<td>610.22</td>
<td>620</td>
<td>617.25</td>
<td>118.61</td>
<td>200</td>
<td>800</td>
<td>600</td>
<td>-0.59</td>
<td>0.00</td>
<td>4.41</td>
</tr>
</tbody>
</table>
Find the pairwise correlations, round to 2 decimals

```r
> round(cor(cleaned, use="pairwise"), 2)
```

<table>
<thead>
<tr>
<th></th>
<th>gender</th>
<th>education</th>
<th>age</th>
<th>ACT</th>
<th>SATV</th>
<th>SATQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>1.00</td>
<td>0.09</td>
<td>-0.02</td>
<td>-0.05</td>
<td>-0.02</td>
<td>-0.17</td>
</tr>
<tr>
<td>education</td>
<td>0.09</td>
<td>1.00</td>
<td>0.55</td>
<td>0.15</td>
<td>0.05</td>
<td>0.03</td>
</tr>
<tr>
<td>age</td>
<td>-0.02</td>
<td>0.55</td>
<td>1.00</td>
<td>0.11</td>
<td>-0.04</td>
<td>-0.03</td>
</tr>
<tr>
<td>ACT</td>
<td>-0.05</td>
<td>0.15</td>
<td>0.11</td>
<td>1.00</td>
<td>0.55</td>
<td>0.59</td>
</tr>
<tr>
<td>SATV</td>
<td>-0.02</td>
<td>0.05</td>
<td>-0.04</td>
<td>0.55</td>
<td>1.00</td>
<td>0.64</td>
</tr>
<tr>
<td>SATQ</td>
<td>-0.17</td>
<td>0.03</td>
<td>-0.03</td>
<td>0.59</td>
<td>0.64</td>
<td>1.00</td>
</tr>
</tbody>
</table>
### Display it differently using the `lowerCor` function

`lowerCor` finds the pairwise correlations, rounds to 2 decimals, and displays the lower half of the correlation matrix.

```r
> lowerCor(sat.act)

<table>
<thead>
<tr>
<th></th>
<th>gendr</th>
<th>edctn</th>
<th>age</th>
<th>ACT</th>
<th>SATV</th>
<th>SATQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>education</td>
<td>0.09</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>-0.02</td>
<td>0.55</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACT</td>
<td>-0.04</td>
<td>0.15</td>
<td>0.11</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SATV</td>
<td>-0.02</td>
<td>0.05</td>
<td>-0.04</td>
<td>0.56</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>SATQ</td>
<td>-0.17</td>
<td>0.03</td>
<td>-0.03</td>
<td>0.59</td>
<td>0.64</td>
<td>1.00</td>
</tr>
</tbody>
</table>
```
A brief example of exploratory and confirmatory data analysis.

Test the correlations for significance using `corr.test`

```r
> corr.test(cleaned)
Call: corr.test(x = sat.act)
Correlation matrix

<table>
<thead>
<tr>
<th></th>
<th>gender</th>
<th>education</th>
<th>age</th>
<th>ACT</th>
<th>SATV</th>
<th>SATQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>1.00</td>
<td>0.09</td>
<td>-0.02</td>
<td>-0.04</td>
<td>-0.02</td>
<td>-0.17</td>
</tr>
<tr>
<td>education</td>
<td>0.09</td>
<td>1.00</td>
<td>0.55</td>
<td>0.15</td>
<td>0.05</td>
<td>0.03</td>
</tr>
<tr>
<td>age</td>
<td>-0.02</td>
<td>0.55</td>
<td>1.00</td>
<td>0.11</td>
<td>-0.04</td>
<td>-0.03</td>
</tr>
<tr>
<td>ACT</td>
<td>-0.04</td>
<td>0.15</td>
<td>0.11</td>
<td>1.00</td>
<td>0.56</td>
<td>0.59</td>
</tr>
<tr>
<td>SATV</td>
<td>-0.02</td>
<td>0.05</td>
<td>-0.04</td>
<td>0.56</td>
<td>1.00</td>
<td>0.64</td>
</tr>
<tr>
<td>SATQ</td>
<td>-0.17</td>
<td>0.03</td>
<td>-0.03</td>
<td>0.59</td>
<td>0.64</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Sample Size

<table>
<thead>
<tr>
<th></th>
<th>gender</th>
<th>education</th>
<th>age</th>
<th>ACT</th>
<th>SATV</th>
<th>SATQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>700</td>
<td>687</td>
</tr>
<tr>
<td>SATQ</td>
<td>687</td>
<td>687</td>
<td>687</td>
<td>687</td>
<td>687</td>
<td>687</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>gender</th>
<th>education</th>
<th>age</th>
<th>ACT</th>
<th>SATV</th>
<th>SATQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>gender</td>
<td>0.00</td>
<td>0.17</td>
<td>1.00</td>
<td>1.00</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>education</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>age</td>
<td>0.58</td>
<td>0.00</td>
<td>0.00</td>
<td>0.03</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ACT</td>
<td>0.33</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SATV</td>
<td>0.62</td>
<td>0.22</td>
<td>0.26</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SATQ</td>
<td>0.00</td>
<td>0.36</td>
<td>0.37</td>
<td>0.00</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```
Are education and gender independent? \( \chi^2 \) Test of association

\[
T \leftarrow \text{with}(\text{my.data}, \text{table}(\text{gender}, \text{education}))
\]

> T

<table>
<thead>
<tr>
<th>gender</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>27</td>
<td>20</td>
<td>23</td>
<td>80</td>
<td>51</td>
<td>46</td>
</tr>
<tr>
<td>2</td>
<td>30</td>
<td>25</td>
<td>21</td>
<td>195</td>
<td>87</td>
<td>95</td>
</tr>
</tbody>
</table>

> chisq.test(T)

Pearson's Chi-squared test

data:  T
X-squared = 16.0851, df = 5, p-value = 0.006605

1. First create a table of associations
   - Do this on our data (my.data)
   - Use the “with” command to specify the data set

2. Show the table

3. Apply \( \chi^2 \) test
Multiple regression

1. Use the sat.act data example
2. Do the linear model
3. Summarize the results

```r
mod1 <- lm(SATV ~ education + gender + SATQ, data=my.data)
> summary(mod1, digits=2)
```

Call:
`lm(formula = SATV ~ education + gender + SATQ, data = my.data)`

Residuals:
```
     Min       1Q   Median       3Q      Max
-372.91  -49.08    2.30   53.68  251.93
```

Coefficients:
```
            Estimate Std. Error   t value  Pr(>|t|)  
(Intercept) 180.87348   23.41019    7.726   3.96e-14 ***
education   1.24043    2.32361     0.534     0.59363
gender    20.69271    6.99651     2.958     0.00321 **
SATQ        0.64489     0.02891   22.309     < 2e-16 ***
```

Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 86.24 on 683 degrees of freedom
(13 observations deleted due to missingness)

Multiple R-squared: 0.4231, Adjusted R-squared: 0.4205
F-statistic: 167 on 3 and 683 DF,  p-value: < 2.2e-16
Zero center the data before examining interactions

In order to examine interactions using multiple regression, we must first “zero center” the data. This may be done using the `scale` function. By default, `scale` will standardize the variables. So to keep the original metric, we make the scaling parameter `FALSE`.

```r
zsat <- data.frame(scale(my.data,scale=FALSE))
describe(zsat)
```

<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>gender</td>
<td>1</td>
<td>700</td>
<td>0</td>
<td>0.48</td>
<td>0.35</td>
<td>0.04</td>
<td>0.00</td>
<td>-0.65</td>
<td>0.35</td>
<td>1</td>
<td>-0.61</td>
<td>0.02</td>
</tr>
<tr>
<td>education</td>
<td>2</td>
<td>700</td>
<td>0</td>
<td>1.43</td>
<td>-0.16</td>
<td>0.14</td>
<td>1.48</td>
<td>-3.16</td>
<td>1.84</td>
<td>5</td>
<td>-0.68</td>
<td></td>
</tr>
<tr>
<td>age</td>
<td>3</td>
<td>700</td>
<td>0</td>
<td>9.50</td>
<td>-3.59</td>
<td>-1.73</td>
<td>5.93</td>
<td>-12.59</td>
<td>39.41</td>
<td>52</td>
<td>1.64</td>
<td></td>
</tr>
<tr>
<td>ACT</td>
<td>4</td>
<td>700</td>
<td>0</td>
<td>4.82</td>
<td>0.45</td>
<td>0.30</td>
<td>4.45</td>
<td>-25.55</td>
<td>7.45</td>
<td>33</td>
<td>-0.66</td>
<td></td>
</tr>
<tr>
<td>SATV</td>
<td>5</td>
<td>700</td>
<td>0</td>
<td>112.90</td>
<td>7.77</td>
<td>7.22</td>
<td>118.61</td>
<td>-412.23</td>
<td>187.77</td>
<td>600</td>
<td>-0.64</td>
<td></td>
</tr>
<tr>
<td>SATQ</td>
<td>6</td>
<td>687</td>
<td>0</td>
<td>115.64</td>
<td>9.78</td>
<td>7.04</td>
<td>118.61</td>
<td>-410.22</td>
<td>189.78</td>
<td>600</td>
<td>-0.59</td>
<td></td>
</tr>
</tbody>
</table>

Note that we need to take the output of `scale` (which comes back as a matrix) and make it into a dataframe if we want to use the linear model on it.
Zero center the data before examining interactions

```r
> zsat <- data.frame(scale(my.data,scale=FALSE))
> mod2 <- lm(SATV ~ education * gender * SATQ,data=zsat)
> summary(mod2)

Call:
lm(formula = SATV ~ education * gender * SATQ, data = zsat)

Residuals:
   Min     1Q Median     3Q    Max
-372.53  -48.76    3.33   51.24  238.50

Coefficients:                                    Estimate     Std. Error t value Pr(>|t|)
(Intercept)                                      0.773576     3.304938  0.234  0.81500
education                                       2.517314     2.337889  1.077  0.28198
gender                                          18.485906     6.964694  2.654  0.00814 **
SATQ                                            0.620527     0.028925 21.453  < 2e-16 ***
education:gender                                 1.249926     4.759374  0.263  0.79292
education:SATQ                                  -0.101444     0.020100 -5.047  5.77e-07 ***
gender:SATQ                                      0.007339     0.060850  0.121  0.90404
education:gender:SATQ                          0.035822     0.041192  0.870  0.38481
---
Signif. codes:  0 ^O***~O 0.001 ^O**~O 0.01 ^O*~O 0.05 ^O.~O 0.1 ^O ~O 1
```
Compare model 1 and model 2

Test the difference between the two linear models

```r
> anova(mod1, mod2)

Analysis of Variance Table

Model 1: SATV ~ education + gender + SATQ
Model 2: SATV ~ education * gender * SATQ

             Res.Df RSS Df Sum of Sq      F    Pr(>F)
Model 1: SATV ~ education + gender + SATQ
1   683 5079984
Model 2: SATV ~ education * gender * SATQ
2   679 4870243  4    209742 7.3104 9.115e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```
Show the regression lines by gender

```r
> with(my.data, plot(SATV ~ SATQ, 
| col=c("blue","red")[gender]))
> by(my.data, my.data$gender, 
| function(x) abline 
| (lm(SATV ~ SATQ, data=x), 
| lty=c("solid","dashed")
> title("Verbal varies by Quant 
| and gender")
```
A brief example of exploratory and confirmatory data analysis

Show the regression lines by education

```r
> with(my.data, plot(SATV~SATQ, col=c("blue","red")[gender]))

by(my.data, my.data$education, function(x) abline (lm(SATV~SATQ, data=x), lty=c("solid", "dashed", "dott", "dotdash", "longdash", "twodash")[x$education+1]))

> title("Verbal varies by Quant and education")
```
Questions?
Using R for psychological statistics: Basic statistics

1. Writing syntax
   - For a single line, just type it
   - Mistakes can be redone by using the up arrow key
   - For longer code, use a text editor (built into some GUIs)

2. Data entry
   - Using built in data sets for examples
   - Copying from another program
   - Reading a text or csv file
   - Importing from SPSS or SAS
   - Simulate it (using various simulation routines)

3. Descriptives
   - Graphical displays
   - Descriptive statistics
   - Correlation

4. Inferential
   - the t test
   - the F test
   - the linear model
Data entry overview

1. Using built in data sets for examples
   - `data()` will list > 100 data sets in the datasets package as well as all sets in loaded packages.
   - Most packages have associated data sets used as examples
   - `psych` has > 40 example data sets

2. Copying from another program
   - use copy and paste into R using `read.clipboard` and its variations

3. Reading a text or csv file
   - read a local or remote file

4. Importing from SPSS or SAS

5. Simulate it (using various simulation routines)
Examples of built in data sets from the psych package

```r
> data(package="psych")

Bechtoldt
Seven data sets showing a bifactor solution.
Dwyer
8 cognitive variables used by Dwyer for an example.
Reise
Seven data sets showing a bifactor solution.
all.income (income)
US family income from US census 2008
bfi
25 Personality items representing 5 factors
blot
Bond's Logical Operations Test - BLOT
burt
11 emotional variables from Burt (1915)
cities
Distances between 11 US cities
epi.bfi
13 personality scales from the Eysenck Personality
and Big 5 inventory
flat (affect)
Two data sets of affect and arousal scores as a function
personality and movie conditions
galton
Galton's Mid parent child height data
income
US family income from US census 2008
iqitems
14 multiple choice IQ items
msq
75 mood items from the Motivational State Questionnaire
3896 participants
neo
NEO correlation matrix from the NEO_PI_R manual
sat.act
3 Measures of ability: SATV, SATQ, ACT
Thurstone
Seven data sets showing a bifactor solution.
veg (vegetables)
Paired comparison of preferences for 9 vegetables
```
Reading data from another program – using the clipboard

1. Read the data in your favorite spreadsheet or text editor
2. Copy to the clipboard
3. Execute the appropriate `read.clipboard()` function with or without various options specified
   
   ```r
   my.data <- read.clipboard()  # assumes headers and tab or space delimited
   my.data <- read.clipboard.csv()  # assumes headers and comma delimited
   my.data <- read.clipboard.tab()  # assumes headers and tab delimited
   # (e.g., from Excel)
   my.data <- read.clipboard.lower()  # read in a matrix given the lower
   my.data <- read.clipboard.upper()  # or upper off diagonal
   my.data <- read.clipboard.fwf()  # read in data using a fixed format width
   # (see `read.fwf` for instructions)
   ```

4. `read.clipboard()` has default values for the most common cases and these do not need to be specified. Consult `?read.clipboard` for details.
Reading from a local or remote file

1. Perhaps the standard way of reading in data is using the `read` command.
   - First must specify the location of the file
   - Can either type this in directly or use the `file.choose` function
   - The file name/location can be a remote URL

2. Two examples of reading data

   ```r
   file.name <- file.choose() #this opens a window to allow you find the file
   my.data <- read.table(file.name)
   datafilename="http://personality-project.org/r/datasets/R.appendix1.data"
   data.ex1=read.table(datafilename,header=TRUE) #read the data into a table
   > dim(data.ex1) #what are the dimensions of what we read?
   [1] 18 2
   > describe(data.ex1) #do the data look right?
   var  n  mean   sd  median trimmed  mad  min  max  range  skew  kurtosis
    Dosage* 1 18  1.89 0.76   2  1.88 1.48  1  3  2  0.16 -1.12  0.18
    Alertness 2 18 27.67 6.82  27 27.50 8.15 17 41 24 0.25 -0.68  1.61
   ```
read a “foreign” file e.g., an SPSS sav file

read.spss reads a file stored by the SPSS save or export commands.

```r
read.spss(file, use.value.labels = TRUE, to.data.frame = FALSE, max.value.labels = Inf, trim.factor.names = FALSE, trim_values = TRUE, reencode = NA, use.missings = to.data.frame)
```

- **file** Character string: the name of the file or URL to read.
- **use.value.labels** Convert variables with value labels into R factors with those levels?
- **to.data.frame** return a data frame? Defaults to FALSE, probably should be TRUE in most cases.
- **max.value.labels** Only variables with value labels and at most this many unique values will be converted to factors if use.value.labels = TRUE.
- **trim.factor.names** Logical: trim trailing spaces from factor levels?
- **trim_values** logical: should values and value labels have trailing spaces ignored when matching for use.value.labels = TRUE?
- **use.missings** logical: should information on user-defined missing values be used to set the corresponding values to NA?
An example of reading from an SPSS file

```r
> library(foreign)
>
> datafilename <- "http://personality-project.org/r/datasets/finkel.sav"
>
> eli <- read.spss(datafilename,to.data.frame=TRUE,
use.value.labels=FALSE)

> describe(eli,skew=FALSE)
```

<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>se</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER*</td>
<td>1</td>
<td>69</td>
<td>35.00</td>
<td>35.00</td>
<td>25.20</td>
<td>1.00</td>
<td>1</td>
<td>69</td>
<td>68</td>
<td>2.42</td>
</tr>
<tr>
<td>HAPPY</td>
<td>2</td>
<td>69</td>
<td>5.71</td>
<td>6.00</td>
<td>0.00</td>
<td>2.00</td>
<td>2</td>
<td>7</td>
<td>5</td>
<td>0.13</td>
</tr>
<tr>
<td>SOULMATE</td>
<td>3</td>
<td>69</td>
<td>5.09</td>
<td>5.32</td>
<td>1.48</td>
<td>1.00</td>
<td>1</td>
<td>7</td>
<td>6</td>
<td>0.22</td>
</tr>
<tr>
<td>ENJOYDEX</td>
<td>4</td>
<td>68</td>
<td>6.47</td>
<td>6.70</td>
<td>0.00</td>
<td>2.00</td>
<td>2</td>
<td>7</td>
<td>5</td>
<td>0.12</td>
</tr>
<tr>
<td>UPSET</td>
<td>5</td>
<td>69</td>
<td>0.41</td>
<td>0.39</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.06</td>
</tr>
</tbody>
</table>

1. Make the `foreign` package active
2. Specify the name (and location) of the file to read
3. Read from a SPSS file
4. Describe it to make sure it is right
Simulate data

For many demonstration purposes, it is convenient to generate simulated data with a certain defined structure. The *psych* package has a number of built in simulation functions. Here are a few of them.

1. **Simulate various item structures**
   - `sim.congeneric` A one factor congeneric measure model
   - `sim.items` A two factor structure with either simple structure or a circumplex structure.
   - `sim.rasch` Generate items for a one parameter IRT model.
   - `sim.irt` Generate items for a one-four parameter IRT Model

2. **Simulate various factor structures**
   - `sim.simplex` Default is a four factor structure with a three time point simplex structure.
   - `sim.hierarchical` Default is 9 variables with three correlated factors.
Get the data and look at it

Read in some data, look at the first and last few cases (using `headtail`), and then get basic descriptive statistics. For this example, we will use a built in data set.

```r
> my.data <- epi.bfi
> headtail(my.data)

epiE epiS epiImp epilie epiNeur bfagree bfcon bfext bfneur bfopen bdi traitanx stateanx
1 18 10 7 3 9 138 96 141 51 138 1 24 22
2 16 8 5 1 12 101 99 107 116 132 7 41 40
3 6 1 3 2 5 143 118 38 68 90 4 37 44
4 12 6 4 3 15 104 106 64 114 101 8 54 40
... ... ... ... ... ... ... ... ... ... ... ... ...
228 12 7 4 3 15 155 129 127 88 110 9 35 34
229 19 10 7 2 11 162 152 163 104 164 1 29 47
230 4 1 1 2 10 95 111 75 123 138 5 39 58
231 8 6 3 2 15 85 62 90 131 96 24 58 58
```

`epi.bfi` has 231 cases from two personality measures.
### Now find the descriptive statistics for this data set

```r
> describe(my.data)

<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>epiE</td>
<td>1</td>
<td>13.33</td>
<td>4.14</td>
<td>14</td>
<td>13.49</td>
<td>4.45</td>
<td>1</td>
<td>22</td>
<td>21</td>
<td>-0.33</td>
<td>-0.01</td>
<td>0.27</td>
</tr>
<tr>
<td>epiS</td>
<td>2</td>
<td>7.58</td>
<td>2.69</td>
<td>8</td>
<td>7.77</td>
<td>2.97</td>
<td>0</td>
<td>13</td>
<td>13</td>
<td>-0.57</td>
<td>0.04</td>
<td>0.18</td>
</tr>
<tr>
<td>epiImp</td>
<td>3</td>
<td>4.37</td>
<td>1.88</td>
<td>4</td>
<td>4.36</td>
<td>1.48</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td>0.06</td>
<td>-0.59</td>
<td>0.12</td>
</tr>
<tr>
<td>epilie</td>
<td>4</td>
<td>2.38</td>
<td>1.50</td>
<td>2</td>
<td>2.27</td>
<td>1.48</td>
<td>0</td>
<td>7</td>
<td>7</td>
<td>0.66</td>
<td>0.30</td>
<td>0.10</td>
</tr>
<tr>
<td>epiNeur</td>
<td>5</td>
<td>10.41</td>
<td>4.90</td>
<td>10</td>
<td>10.39</td>
<td>4.45</td>
<td>0</td>
<td>23</td>
<td>23</td>
<td>0.06</td>
<td>-0.46</td>
<td>0.32</td>
</tr>
<tr>
<td>bfagree</td>
<td>6</td>
<td>125.00</td>
<td>18.14</td>
<td>126</td>
<td>125.26</td>
<td>17.79</td>
<td>74</td>
<td>167</td>
<td>93</td>
<td>-0.21</td>
<td>-0.22</td>
<td>1.19</td>
</tr>
<tr>
<td>bfcon</td>
<td>7</td>
<td>113.25</td>
<td>21.88</td>
<td>114</td>
<td>113.42</td>
<td>22.24</td>
<td>53</td>
<td>178</td>
<td>125</td>
<td>-0.02</td>
<td>0.29</td>
<td>1.44</td>
</tr>
<tr>
<td>bfext</td>
<td>8</td>
<td>102.18</td>
<td>26.45</td>
<td>104</td>
<td>102.99</td>
<td>22.24</td>
<td>8</td>
<td>168</td>
<td>160</td>
<td>-0.41</td>
<td>0.58</td>
<td>1.74</td>
</tr>
<tr>
<td>bfneur</td>
<td>9</td>
<td>87.97</td>
<td>23.34</td>
<td>90</td>
<td>87.70</td>
<td>23.72</td>
<td>34</td>
<td>152</td>
<td>118</td>
<td>0.07</td>
<td>-0.51</td>
<td>1.54</td>
</tr>
<tr>
<td>bfopen</td>
<td>10</td>
<td>123.43</td>
<td>20.51</td>
<td>125</td>
<td>123.78</td>
<td>20.76</td>
<td>73</td>
<td>173</td>
<td>100</td>
<td>-0.16</td>
<td>-0.11</td>
<td>1.35</td>
</tr>
<tr>
<td>bdi</td>
<td>11</td>
<td>6.78</td>
<td>5.78</td>
<td>6</td>
<td>5.97</td>
<td>4.45</td>
<td>0</td>
<td>27</td>
<td>27</td>
<td>1.29</td>
<td>1.60</td>
<td>0.38</td>
</tr>
<tr>
<td>traitanx</td>
<td>12</td>
<td>39.01</td>
<td>9.52</td>
<td>38</td>
<td>38.36</td>
<td>8.90</td>
<td>22</td>
<td>71</td>
<td>49</td>
<td>0.67</td>
<td>0.54</td>
<td>0.63</td>
</tr>
<tr>
<td>stateanx</td>
<td>13</td>
<td>39.85</td>
<td>11.48</td>
<td>38</td>
<td>38.92</td>
<td>10.38</td>
<td>21</td>
<td>79</td>
<td>58</td>
<td>0.72</td>
<td>0.04</td>
<td>0.76</td>
</tr>
</tbody>
</table>
```
Boxplots are a convenient descriptive device

Show the Tukey “boxplot” for the Eysenck Personality Inventory

Boxplots of EPI scales
Plot the scatter plot matrix (SPLOM) of the first 5 variables using the `pairs.panels` function

Use the `pairs.panels` function from `psych`

`pairs.panels(my.data[1:5])`
Plot the scatter plot matrix (SPLOM) of the first 5 variables using the `pairs.panels` function but with smaller `pch` and jittering the points.

```
use the `pairs.panels` function from `psych`
```

```
pairs.panels(my.data[1:5], pch='.', jiggle=TRUE)
```
### Find the correlations for this data set, round off to 2 decimal places

```r
> round(cor(my.data, use = "pairwise"), 2)

<table>
<thead>
<tr>
<th></th>
<th>epiE</th>
<th>epiS</th>
<th>epiImp</th>
<th>epilie</th>
<th>epiNeur</th>
<th>bfagree</th>
<th>bfcon</th>
<th>bfext</th>
<th>bfneur</th>
<th>bfopen</th>
<th>bdi</th>
<th>traitanx</th>
<th>stateanx</th>
</tr>
</thead>
<tbody>
<tr>
<td>epiE</td>
<td>1.00</td>
<td>0.85</td>
<td>0.80</td>
<td>-0.22</td>
<td>-0.18</td>
<td>0.18</td>
<td>-0.11</td>
<td>0.54</td>
<td>-0.09</td>
<td>0.14</td>
<td>-0.16</td>
<td>-0.23</td>
<td>-0.13</td>
</tr>
<tr>
<td>epiS</td>
<td>0.85</td>
<td>1.00</td>
<td>0.43</td>
<td>-0.05</td>
<td>-0.22</td>
<td>0.20</td>
<td>0.05</td>
<td>0.58</td>
<td>-0.07</td>
<td>0.15</td>
<td>-0.13</td>
<td>-0.26</td>
<td>-0.12</td>
</tr>
<tr>
<td>epiImp</td>
<td>0.80</td>
<td>0.43</td>
<td>1.00</td>
<td>-0.24</td>
<td>-0.07</td>
<td>0.08</td>
<td>-0.24</td>
<td>0.35</td>
<td>-0.09</td>
<td>0.07</td>
<td>-0.11</td>
<td>-0.12</td>
<td>-0.09</td>
</tr>
<tr>
<td>epilie</td>
<td>-0.22</td>
<td>-0.05</td>
<td>-0.24</td>
<td>1.00</td>
<td>-0.25</td>
<td>0.17</td>
<td>0.23</td>
<td>-0.04</td>
<td>-0.22</td>
<td>-0.03</td>
<td>-0.20</td>
<td>-0.23</td>
<td>-0.15</td>
</tr>
<tr>
<td>epiNeur</td>
<td>-0.18</td>
<td>-0.22</td>
<td>-0.07</td>
<td>-0.25</td>
<td>1.00</td>
<td>-0.08</td>
<td>-0.13</td>
<td>-0.17</td>
<td>0.63</td>
<td>0.09</td>
<td>0.58</td>
<td>0.73</td>
<td>0.49</td>
</tr>
<tr>
<td>bfagree</td>
<td>0.18</td>
<td>0.20</td>
<td>0.08</td>
<td>0.17</td>
<td>-0.08</td>
<td>1.00</td>
<td>0.45</td>
<td>0.48</td>
<td>-0.04</td>
<td>0.39</td>
<td>-0.14</td>
<td>-0.31</td>
<td>-0.19</td>
</tr>
<tr>
<td>bfcon</td>
<td>-0.11</td>
<td>0.05</td>
<td>-0.24</td>
<td>0.23</td>
<td>-0.13</td>
<td>0.45</td>
<td>1.00</td>
<td>0.27</td>
<td>0.04</td>
<td>0.31</td>
<td>-0.18</td>
<td>-0.29</td>
<td>-0.14</td>
</tr>
<tr>
<td>bfext</td>
<td>0.54</td>
<td>0.58</td>
<td>0.35</td>
<td>-0.04</td>
<td>-0.17</td>
<td>0.48</td>
<td>0.27</td>
<td>1.00</td>
<td>0.04</td>
<td>0.46</td>
<td>-0.14</td>
<td>-0.39</td>
<td>-0.15</td>
</tr>
<tr>
<td>bfneur</td>
<td>-0.09</td>
<td>-0.07</td>
<td>-0.09</td>
<td>-0.22</td>
<td>0.63</td>
<td>-0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>1.00</td>
<td>0.29</td>
<td>0.47</td>
<td>0.59</td>
<td>0.49</td>
</tr>
<tr>
<td>bfopen</td>
<td>0.14</td>
<td>0.15</td>
<td>0.07</td>
<td>-0.03</td>
<td>0.09</td>
<td>0.39</td>
<td>0.31</td>
<td>0.46</td>
<td>0.29</td>
<td>1.00</td>
<td>-0.08</td>
<td>-0.11</td>
<td>-0.04</td>
</tr>
<tr>
<td>bdi</td>
<td>-0.16</td>
<td>-0.13</td>
<td>-0.11</td>
<td>-0.20</td>
<td>0.58</td>
<td>-0.14</td>
<td>-0.18</td>
<td>-0.14</td>
<td>0.47</td>
<td>-0.08</td>
<td>1.00</td>
<td>0.65</td>
<td>0.61</td>
</tr>
<tr>
<td>traitanx</td>
<td>-0.23</td>
<td>-0.26</td>
<td>-0.12</td>
<td>-0.23</td>
<td>0.73</td>
<td>-0.31</td>
<td>-0.29</td>
<td>-0.39</td>
<td>0.59</td>
<td>0.59</td>
<td>0.11</td>
<td>0.65</td>
<td>1.00</td>
</tr>
<tr>
<td>stateanx</td>
<td>-0.13</td>
<td>-0.12</td>
<td>-0.09</td>
<td>-0.15</td>
<td>0.49</td>
<td>-0.19</td>
<td>-0.14</td>
<td>-0.15</td>
<td>0.49</td>
<td>-0.04</td>
<td>0.61</td>
<td>0.57</td>
<td>1.00</td>
</tr>
</tbody>
</table>
Find the correlations for this data set, round off to 2 decimal places using lowerCor

```r
> lowerCor(my.data)

   epiE  epiS epili epiNr bfagr bfcon bfext bfner bfopn bdi  trtnx  sttnx
epiE  1.00
epiS  0.85  1.00
epiImp 0.80  0.43  1.00
epilie -0.22 -0.05 -0.24  1.00
epiNeur -0.18 -0.22 -0.07 -0.25  1.00
bfagree  0.18  0.20  0.08  0.17 -0.08  1.00
bfcon -0.11  0.05 -0.24  0.23 -0.13  0.45  1.00
bfext  0.54  0.58  0.35 -0.04 -0.17  0.48  0.27  1.00
bfneur -0.09 -0.07 -0.09 -0.22  0.63 -0.04  0.04  0.04  1.00
bfopen  0.14  0.15  0.07 -0.03  0.09  0.39  0.31  0.46  0.29  1.00
bdi -0.16 -0.13 -0.11 -0.20  0.58 -0.14 -0.18 -0.14  0.47 -0.08  1.00
traitanx -0.23 -0.26 -0.12 -0.23  0.73 -0.31 -0.29 -0.39  0.59 -0.11  0.65  1.00
stateanx -0.13 -0.12 -0.09 -0.15  0.49 -0.19 -0.14 -0.15  0.49 -0.04  0.61  0.57  1.00
```
### Test the significance and use Holm correction for multiple tests

```r
> corr.test(my.data)
Call: corr.test(x = my.data)
Correlation matrix

<table>
<thead>
<tr>
<th></th>
<th>epiE</th>
<th>epiS</th>
<th>epiImp</th>
<th>epilie</th>
<th>epiNeur</th>
<th>bfagree</th>
<th>bfcon</th>
<th>bfext</th>
<th>bfneur</th>
<th>bfopen</th>
<th>bdi</th>
<th>traitanx</th>
<th>stateanx</th>
</tr>
</thead>
<tbody>
<tr>
<td>epiE</td>
<td>1.00</td>
<td>0.85</td>
<td>0.80</td>
<td>-0.22</td>
<td>-0.18</td>
<td>0.18</td>
<td>-0.11</td>
<td>0.54</td>
<td>-0.09</td>
<td>0.14</td>
<td>-0.16</td>
<td>-0.23</td>
<td>-0.13</td>
</tr>
<tr>
<td>epiS</td>
<td>0.85</td>
<td>1.00</td>
<td>0.43</td>
<td>-0.05</td>
<td>-0.22</td>
<td>0.20</td>
<td>0.05</td>
<td>0.58</td>
<td>-0.07</td>
<td>0.15</td>
<td>-0.13</td>
<td>-0.26</td>
<td>-0.12</td>
</tr>
<tr>
<td>epiImp</td>
<td>0.80</td>
<td>0.43</td>
<td>1.00</td>
<td>-0.24</td>
<td>-0.07</td>
<td>0.08</td>
<td>-0.24</td>
<td>0.35</td>
<td>-0.09</td>
<td>0.07</td>
<td>-0.11</td>
<td>-0.12</td>
<td>-0.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stateanx</td>
<td>-0.13</td>
<td>-0.12</td>
<td>-0.09</td>
<td>-0.15</td>
<td>0.49</td>
<td>-0.19</td>
<td>-0.14</td>
<td>-0.15</td>
<td>0.49</td>
<td>-0.04</td>
<td>0.61</td>
<td>0.57</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Sample Size

<table>
<thead>
<tr>
<th></th>
<th>epiE</th>
<th>epiS</th>
<th>epiImp</th>
<th>epilie</th>
<th>epiNeur</th>
<th>bfagree</th>
<th>bfcon</th>
<th>bfext</th>
<th>bfneur</th>
<th>bfopen</th>
<th>bdi</th>
<th>traitanx</th>
<th>stateanx</th>
</tr>
</thead>
<tbody>
<tr>
<td>epiE</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>stateanx</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
<td>231</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>epiE</th>
<th>epiS</th>
<th>epiImp</th>
<th>epilie</th>
<th>epiNeur</th>
<th>bfagree</th>
<th>bfcon</th>
<th>bfext</th>
<th>bfneur</th>
<th>bfopen</th>
<th>bdi</th>
<th>traitanx</th>
<th>stateanx</th>
</tr>
</thead>
<tbody>
<tr>
<td>epiE</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.03</td>
<td>0.27</td>
<td>0.27</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.59</td>
<td>0.02</td>
<td>1.00</td>
</tr>
<tr>
<td>epiS</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.04</td>
<td>0.08</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.62</td>
<td>1.00</td>
</tr>
<tr>
<td>epiImp</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>1.00</td>
<td>1.00</td>
<td>0.01</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>epilie</td>
<td>0.00</td>
<td>0.43</td>
<td>0.00</td>
<td>0.00</td>
<td>0.01</td>
<td>0.32</td>
<td>0.03</td>
<td>1.00</td>
<td>0.03</td>
<td>1.00</td>
<td>0.08</td>
<td>0.02</td>
<td>0.61</td>
</tr>
<tr>
<td>epiNeur</td>
<td>0.01</td>
<td>0.00</td>
<td>0.26</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.33</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>bfagree</td>
<td>0.01</td>
<td>0.00</td>
<td>0.23</td>
<td>0.01</td>
<td>0.21</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.95</td>
<td>0.00</td>
<td>0.12</td>
</tr>
<tr>
<td>bfcon</td>
<td>0.08</td>
<td>0.48</td>
<td>0.00</td>
<td>0.00</td>
<td>0.04</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.25</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>bfext</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.01</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.99</td>
<td>0.00</td>
<td>0.76</td>
</tr>
<tr>
<td>bfneur</td>
<td>0.15</td>
<td>0.30</td>
<td>0.18</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.50</td>
<td>0.57</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>bfopen</td>
<td>0.04</td>
<td>0.02</td>
<td>0.30</td>
<td>0.70</td>
<td>0.19</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>bdi</td>
<td>0.02</td>
<td>0.04</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
<td>0.03</td>
<td>0.01</td>
<td>0.03</td>
<td>0.00</td>
<td>0.25</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>traitanx</td>
<td>0.00</td>
<td>0.00</td>
<td>0.07</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>0.11</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>stateanx</td>
<td>0.05</td>
<td>0.07</td>
<td>0.18</td>
<td>0.02</td>
<td>0.00</td>
<td>0.00</td>
<td>0.04</td>
<td>0.02</td>
<td>0.00</td>
<td>0.52</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
>
## t.test demonstration with Student’s data (from the sleep dataset)

```r
> with(sleep, t.test(extra ~ group))

Welch Two Sample t-test
data: extra by group
t = -1.8608, df = 17.776, p-value = 0.07939
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-3.3654832 0.2054832
sample estimates:
mean in group 1 mean in group 2
0.75 2.33
```

But the data were actually paired. Do it for a paired t-test

```r
> with(sleep, t.test(extra ~ group, paired = TRUE))

Paired t-test
data: extra by group
t = -4.0621, df = 9, p-value = 0.002833
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-2.4598858 -0.7001142
sample estimates:
mean of the differences
-1.58
```
Two ways of showing Student’s t test data
Two ways of showing Student’s t test data

Use the `error.bars.by` and `error.bars` functions. Note that we need to change the data structure a little bit to get the within subject error bars.

```r
> error.bars.by(sleep$extra, sleep$group, by.var=TRUE, lines=FALSE, ylab="Change in Sleep", xlab="Drug condition", main="Student's sleep data")
```

```r
> error.bars(data.frame(drug1=sleep[1:10,1], drug2=sleep[11:20,1]), within=TRUE, ylab="Change in Sleep", xlab="Drug Condition", main="Student's paired sleep data")
```
Analysis of Variance

1. aov is designed for balanced designs, and the results can be hard to interpret without balance: beware that missing values in the response(s) will likely lose the balance.

2. If there are two or more error strata, the methods used are statistically inefficient without balance, and it may be better to use lme in package nlme.

```r
# read the data into a table
data.ex2 = read.table("http://personality-project.org/R/datasets/R.appendix2.data", header = TRUE)  
# show the data
data.ex2
```

<table>
<thead>
<tr>
<th>Observation</th>
<th>Gender</th>
<th>Dosage</th>
<th>Alertness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>m</td>
<td>a</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>m</td>
<td>a</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>m</td>
<td>a</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>m</td>
<td>a</td>
<td>12</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>f</td>
<td>b</td>
<td>12</td>
</tr>
<tr>
<td>15</td>
<td>f</td>
<td>b</td>
<td>18</td>
</tr>
<tr>
<td>16</td>
<td>f</td>
<td>b</td>
<td>22</td>
</tr>
</tbody>
</table>
Analysis of Variance

Do the analysis of variances and the show the table of results.

```r
aov.ex2 = aov(Alertness~Gender*Dosage,data=data.ex2) #do the analysis of variance
summary(aov.ex2) #show the summary table

> aov.ex2 = aov(Alertness~Gender*Dosage,data=data.ex2) #do the analysis of variance
> summary(aov.ex2) #show the summary table

Df  Sum Sq Mean Sq  F value    Pr(>F)
Gender 1  76.562  76.562 2.9518     0.1115
Dosage 1   5.062  5.062  0.1952     0.6665
Gender:Dosage 1  0.063  0.063 0.0024     0.9617
```
Show the results table

```r
> print(model.tables(aov.ex2, "means"), digits=3)

Residuals    12 311.250  25.938

Tables of means
Grand mean

14.0625

Gender
Gender
   f  m
16.25 11.88

Dosage
Dosage
   a  b
13.50 14.62

Gender:Dosage
Dosage
Gender  a  b
   f 15.75 16.75
   m 11.25 12.50
```
Analysis of Variance: Within subjects

1. Somewhat more complicated because we need to convert “wide” data.frames to “long” or “narrow” data.frame.

2. This can be done by using the *stack* function. Some data sets are already in the long format.

3. A detailed discussion of how to work with repeated measures designs is at [http://personality-project.org/r/r.anova.html](http://personality-project.org/r/r.anova.html) and at [http://personality-project.org/r](http://personality-project.org/r)
Analysis of variance within subjects

```r
> datafilename="http://personality-project.org/r/datasets/R.appendix5.data"
> data.ex5=read.table(datafilename,header=T)  #read the data into a table
> #data.ex5                                         #show the data
> aov.ex5 =
+ aov(Recall~(Task*Valence*Gender*Dosage)+Error(Subject/(Task*Valence)))+
+ (Gender*Dosage),data.ex5)
> summary(aov.ex5)

Error: Subject

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>1</td>
<td>542.26</td>
<td>542.26</td>
<td>5.6853</td>
<td>0.03449 *</td>
</tr>
<tr>
<td>Dosage</td>
<td>2</td>
<td>694.91</td>
<td>347.45</td>
<td>3.6429</td>
<td>0.05803 .</td>
</tr>
<tr>
<td>Gender:Dosage</td>
<td>2</td>
<td>70.80</td>
<td>35.40</td>
<td>0.3711</td>
<td>0.69760</td>
</tr>
<tr>
<td>Residuals</td>
<td>12</td>
<td>1144.56</td>
<td>95.38</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Signif. codes: 0 ^O***~O 0.001 ^O**~O 0.01 ^O*~O 0.05 ^O . ~O 1

Error: Subject:Task

<table>
<thead>
<tr>
<th></th>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>1</td>
<td>96.333</td>
<td>96.333</td>
<td>39.8621</td>
<td>0.0001 ***</td>
</tr>
<tr>
<td>Task:Gender</td>
<td>1</td>
<td>1.333</td>
<td>1.333</td>
<td>0.5517</td>
<td>0.4719</td>
</tr>
<tr>
<td>Task:Dosage</td>
<td>2</td>
<td>8.167</td>
<td>4.083</td>
<td>1.6897</td>
<td>0.2257</td>
</tr>
<tr>
<td>Task:Gender:Dosage</td>
<td>2</td>
<td>3.167</td>
<td>1.583</td>
<td>0.6552</td>
<td>0.5370</td>
</tr>
<tr>
<td>Residuals</td>
<td>12</td>
<td>29.000</td>
<td>2.417</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
... (lots more)
Multiple regression

1. Use the sat.act data set from psych
2. Do the linear model
3. Summarize the results

```r
mod1 <- lm(SATV ~ education + gender + SATQ, data=sat.act)
> summary(mod1, digits=2)
```

```
Call:
lm(formula = SATV ~ education + gender + SATQ, data = sat.act)
Residuals:
     Min      1Q  Median      3Q     Max
-372.91  -49.08   2.30  53.68  251.93
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept) 180.87348   23.41019  7.726  3.96e-14 ***
education    1.24043    2.32361  0.534   0.59363
gender      20.69271    6.99651  2.958   0.00321 **
SATQ         0.64489    0.02891 22.309  < 2e-16 ***
```

Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1  1

Residual standard error: 86.24 on 683 degrees of freedom
(13 observations deleted due to missingness)

Multiple R-squared: 0.4231, Adjusted R-squared: 0.4205
F-statistic: 167 on 3 and 683 DF,  p-value: < 2.2e-16
Zero center the data before examining interactions

```R
> zsat <- data.frame(scale(sat.act, scale=FALSE))
> mod2 <- lm(SATV ~ education * gender * SATQ, data=zsat)
> summary(mod2)

Call:
  lm(formula = SATV ~ education * gender * SATQ, data = zsat)

Residuals:
   Min      1Q  Median      3Q     Max
-372.53  -48.76   3.33   51.24  238.50

Coefficients:
                           Estimate Std. Error  t value Pr(>|t|)
(Intercept)               0.773576   3.304938  0.234   0.8150
education                2.517314   2.337889  1.077   0.2819
gender                   18.485906   6.964694  2.654  0.00814 **
SATQ                     0.620527   0.028925 21.453  < 2e-16 ***
education:gender        1.249926   4.759374  0.263   0.7929
education:SATQ          -0.101444   0.020100 -5.047 5.77e-07 ***
gender:SATQ              0.007339   0.060850  0.121   0.9040
education:gender:SATQ   0.035822   0.041192  0.870   0.3848

---
Signif. codes:  0 ^O***~O 0.001 ^O**~O 0.01 ^O*~O 0.05 ^O.~O 0.1 ^O ~O 1
```
Test the difference between the two linear models

> anova(mod1, mod2)

Analysis of Variance Table

Model 1: SATV ~ education + gender + SATQ  
Model 2: SATV ~ education * gender * SATQ

<table>
<thead>
<tr>
<th>Res.Df</th>
<th>RSS</th>
<th>Df</th>
<th>Sum of Sq</th>
<th>F</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>683</td>
<td>4</td>
<td>5079984</td>
<td>7.3104</td>
<td>9.115e-06 ***</td>
</tr>
<tr>
<td>2</td>
<td>679</td>
<td>4</td>
<td>4870243</td>
<td>9.115e-06</td>
<td>***</td>
</tr>
</tbody>
</table>

---

Signif. codes:  0 ^O***Ô  0.001 ^O**Ô  0.01 ^O*Ô  0.05 ^O.Ô  0.1 ^O
Show the regression lines by gender

```r
> with(sat.act, plot(SATV ~ SATQ, col = c("blue", "red") [gender]))
> by(sat.act, sat.act$gender, function(x) abline(
>   lm(SATV ~ SATQ, data = x),
>   lty = c("solid", "dashed"))
> title("Verbal varies by Quant and gender")
```
Psychometrics

1. Classical test theory measures of reliability
   - Scoring tests
   - Reliability (alpha, beta, omega)

2. Multivariate Analysis
   - Factor Analysis
   - Components analysis
   - Multidimensional scaling
   - Structural Equation Modeling

3. Item Response Theory
   - One parameter (Rasch) models
   - 2PL and 2PN models
What is R?  A brief example  Basic statistics and graphics  Psychometrics and beyond  Basic R commands

Classical Test measures of reliability

Classic theory estimates of reliability

1. Scoring tests

   score.items  Score 1-n scales using a set of keys and finding the simple sum or average of items. Reversed items are indicated by -1

   score.multiple.choice  Score multiple choice items by first converting to 0 or 1 and then proceeding to score the items.

2. Alternative estimates of reliability

   alpha $\alpha$  reliability of a single scale finds the average split half reliability. (some items may be reversed keyed).

   omega $\omega$  reliability of a single scale estimates the general factor saturation of the test.

   guttman  Find the 6 Guttman reliability estimates
Using `score.items` to score 25 Big 5 items (taken from the bfi example)

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

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

(Unstandardized) Alpha:

<table>
<thead>
<tr>
<th></th>
<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>0.72</td>
<td>0.76</td>
<td>0.81</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Average item correlation:

<table>
<thead>
<tr>
<th></th>
<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>0.34</td>
<td>0.39</td>
<td>0.46</td>
<td>0.23</td>
</tr>
</tbody>
</table>

Guttman 6* reliability:

<table>
<thead>
<tr>
<th></th>
<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>0.76</td>
<td>0.81</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Scale intercorrelations corrected for attenuation

raw correlations below the diagonal, alpha on the diagonal
corrected correlations above the diagonal:

<table>
<thead>
<tr>
<th></th>
<th>Agree</th>
<th>Conscientious</th>
<th>Extraversion</th>
<th>Neuroticism</th>
<th>Openness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agree</td>
<td>0.70</td>
<td>0.36</td>
<td>0.63</td>
<td>-0.245</td>
<td>0.23</td>
</tr>
<tr>
<td>Conscientious</td>
<td>0.26</td>
<td>0.72</td>
<td>0.35</td>
<td>-0.305</td>
<td>0.30</td>
</tr>
<tr>
<td>Extraversion</td>
<td>0.46</td>
<td>0.26</td>
<td>0.76</td>
<td>-0.284</td>
<td>0.32</td>
</tr>
<tr>
<td>Neuroticism</td>
<td>-0.18</td>
<td>-0.23</td>
<td>-0.22</td>
<td>0.812</td>
<td>-0.12</td>
</tr>
<tr>
<td>Openness</td>
<td>0.15</td>
<td>0.19</td>
<td>0.22</td>
<td>-0.086</td>
<td>0.60</td>
</tr>
</tbody>
</table>

...
Item by scale correlations:
corrected for item overlap and scale reliability

<table>
<thead>
<tr>
<th></th>
<th>Agree</th>
<th>Conscientious</th>
<th>Extraversion</th>
<th>Neuroticism</th>
<th>Openness</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>-0.40</td>
<td>-0.06</td>
<td>-0.11</td>
<td>0.14</td>
<td>-0.14</td>
</tr>
<tr>
<td>A2</td>
<td>0.67</td>
<td>0.23</td>
<td>0.40</td>
<td>-0.07</td>
<td>0.17</td>
</tr>
<tr>
<td>A3</td>
<td>0.70</td>
<td>0.22</td>
<td>0.48</td>
<td>-0.11</td>
<td>0.17</td>
</tr>
<tr>
<td>A4</td>
<td>0.49</td>
<td>0.29</td>
<td>0.30</td>
<td>-0.14</td>
<td>0.01</td>
</tr>
<tr>
<td>A5</td>
<td>0.62</td>
<td>0.23</td>
<td>0.55</td>
<td>-0.23</td>
<td>0.18</td>
</tr>
<tr>
<td>C1</td>
<td>0.13</td>
<td>0.53</td>
<td>0.19</td>
<td>-0.08</td>
<td>0.28</td>
</tr>
<tr>
<td>C2</td>
<td>0.21</td>
<td>0.61</td>
<td>0.17</td>
<td>0.00</td>
<td>0.20</td>
</tr>
<tr>
<td>C3</td>
<td>0.21</td>
<td>0.54</td>
<td>0.14</td>
<td>-0.09</td>
<td>0.08</td>
</tr>
<tr>
<td>C4</td>
<td>-0.24</td>
<td>-0.66</td>
<td>-0.23</td>
<td>0.31</td>
<td>-0.23</td>
</tr>
<tr>
<td>C5</td>
<td>-0.26</td>
<td>-0.59</td>
<td>-0.29</td>
<td>0.36</td>
<td>-0.10</td>
</tr>
<tr>
<td>E1</td>
<td>-0.30</td>
<td>-0.06</td>
<td>-0.59</td>
<td>0.11</td>
<td>-0.16</td>
</tr>
<tr>
<td>E2</td>
<td>-0.39</td>
<td>-0.25</td>
<td>-0.70</td>
<td>0.34</td>
<td>-0.15</td>
</tr>
<tr>
<td>E3</td>
<td>0.44</td>
<td>0.20</td>
<td>0.60</td>
<td>-0.10</td>
<td>0.37</td>
</tr>
<tr>
<td>E4</td>
<td>0.51</td>
<td>0.23</td>
<td>0.68</td>
<td>-0.22</td>
<td>0.04</td>
</tr>
<tr>
<td>E5</td>
<td>0.34</td>
<td>0.40</td>
<td>0.55</td>
<td>-0.10</td>
<td>0.31</td>
</tr>
<tr>
<td>N1</td>
<td>-0.22</td>
<td>-0.21</td>
<td>-0.11</td>
<td>0.76</td>
<td>-0.12</td>
</tr>
<tr>
<td>N2</td>
<td>-0.22</td>
<td>-0.19</td>
<td>-0.12</td>
<td>0.74</td>
<td>-0.06</td>
</tr>
<tr>
<td>N3</td>
<td>-0.14</td>
<td>-0.20</td>
<td>-0.14</td>
<td>0.74</td>
<td>-0.03</td>
</tr>
<tr>
<td>N4</td>
<td>-0.22</td>
<td>-0.30</td>
<td>-0.39</td>
<td>0.62</td>
<td>-0.02</td>
</tr>
<tr>
<td>N5</td>
<td>-0.04</td>
<td>-0.14</td>
<td>-0.19</td>
<td>0.55</td>
<td>-0.18</td>
</tr>
<tr>
<td>O1</td>
<td>0.16</td>
<td>0.20</td>
<td>0.31</td>
<td>-0.09</td>
<td>0.52</td>
</tr>
<tr>
<td>O2</td>
<td>-0.01</td>
<td>0.20</td>
<td>0.18</td>
<td>0.19</td>
<td>-0.45</td>
</tr>
<tr>
<td>O3</td>
<td>0.26</td>
<td>0.20</td>
<td>0.42</td>
<td>0.07</td>
<td>0.61</td>
</tr>
<tr>
<td>O4</td>
<td>0.06</td>
<td>0.02</td>
<td>0.10</td>
<td>0.21</td>
<td>0.32</td>
</tr>
<tr>
<td>O5</td>
<td>-0.09</td>
<td>-0.14</td>
<td>-0.11</td>
<td>0.11</td>
<td>-0.53</td>
</tr>
<tr>
<td>gender</td>
<td>0.25</td>
<td>0.11</td>
<td>0.12</td>
<td>0.14</td>
<td>-0.07</td>
</tr>
<tr>
<td>education</td>
<td>0.06</td>
<td>0.03</td>
<td>0.01</td>
<td>-0.06</td>
<td>0.13</td>
</tr>
<tr>
<td>age</td>
<td>0.22</td>
<td>0.14</td>
<td>0.07</td>
<td>-0.13</td>
<td>0.10</td>
</tr>
</tbody>
</table>
Factor analysis of Thurstone 9 variable problem

> f3 <- fa(Thurstone,3) #use this built in dataset
> f3

Factor Analysis using method = minres
Call: fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate = rotate,
    scores = scores, residuals = residuals, SMC = SMC, missing = FALSE,
    impute = impute, min.err = min.err, max.iter = max.iter,
    symmetric = symmetric, warnings = warnings, fm = fm, alpha = alpha)
Standardized loadings based upon correlation matrix

<table>
<thead>
<tr>
<th></th>
<th>MR1</th>
<th>MR2</th>
<th>MR3</th>
<th>h2</th>
<th>u2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentences</td>
<td>0.91</td>
<td>-0.04</td>
<td>0.04</td>
<td>0.82</td>
<td>0.18</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>0.89</td>
<td>0.06</td>
<td>-0.03</td>
<td>0.84</td>
<td>0.16</td>
</tr>
<tr>
<td>Sent.Completion</td>
<td>0.83</td>
<td>0.04</td>
<td>0.00</td>
<td>0.73</td>
<td>0.27</td>
</tr>
<tr>
<td>First.Letters</td>
<td>0.00</td>
<td>0.86</td>
<td>0.00</td>
<td>0.73</td>
<td>0.27</td>
</tr>
<tr>
<td>4.Letter.Words</td>
<td>-0.01</td>
<td>0.74</td>
<td>0.10</td>
<td>0.63</td>
<td>0.37</td>
</tr>
<tr>
<td>Suffixes</td>
<td>0.18</td>
<td>0.63</td>
<td>-0.08</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>Letter.Series</td>
<td>0.03</td>
<td>-0.01</td>
<td>0.84</td>
<td>0.72</td>
<td>0.28</td>
</tr>
<tr>
<td>Pedigrees</td>
<td>0.37</td>
<td>-0.05</td>
<td>0.47</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>Letter.Group</td>
<td>-0.06</td>
<td>0.21</td>
<td>0.64</td>
<td>0.53</td>
<td>0.47</td>
</tr>
</tbody>
</table>

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

With factor correlations of

<table>
<thead>
<tr>
<th></th>
<th>MR1</th>
<th>MR2</th>
<th>MR3</th>
</tr>
</thead>
<tbody>
<tr>
<td>MR1</td>
<td>1.00</td>
<td>0.59</td>
<td>0.54</td>
</tr>
<tr>
<td>MR2</td>
<td>0.59</td>
<td>1.00</td>
<td>0.52</td>
</tr>
<tr>
<td>MR3</td>
<td>0.54</td>
<td>0.52</td>
<td>1.00</td>
</tr>
</tbody>
</table>

...
Factor analysis output, continued

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the null model are 36 and the objective function was 5.2 with Chi Square of
The degrees of freedom for the model are 12 and the objective function was 0.01

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

Tucker Lewis Index of factoring reliability = 1.027
RMSEA index = 0 and the 90 % confidence intervals are 0 0.023
BIC = -61.51
Fit based upon off diagonal values = 1
Measures of factor score adequacy

<table>
<thead>
<tr>
<th>MR1</th>
<th>MR2</th>
<th>MR3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.96</td>
<td>0.92</td>
<td>0.90</td>
</tr>
<tr>
<td>0.93</td>
<td>0.85</td>
<td>0.81</td>
</tr>
<tr>
<td>0.86</td>
<td>0.71</td>
<td>0.63</td>
</tr>
</tbody>
</table>
> f3 <- fa(Thurstone,3,n.obs=213,n.iter=20)  #to do bootstrapping

Coefficients and bootstrapped confidence intervals

<table>
<thead>
<tr>
<th></th>
<th>low</th>
<th>MR1</th>
<th>upper</th>
<th>low</th>
<th>MR2</th>
<th>upper</th>
<th>low</th>
<th>MR3</th>
<th>upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentences</td>
<td>0.77</td>
<td>0.91</td>
<td>0.96</td>
<td>-0.12</td>
<td>-0.04</td>
<td>0.07</td>
<td>-0.03</td>
<td>0.04</td>
<td>0.14</td>
</tr>
<tr>
<td>Vocabulary</td>
<td>0.85</td>
<td>0.89</td>
<td>0.95</td>
<td>-0.01</td>
<td>0.06</td>
<td>0.10</td>
<td>-0.12</td>
<td>-0.03</td>
<td>0.04</td>
</tr>
<tr>
<td>Sent.Completion</td>
<td>0.73</td>
<td>0.83</td>
<td>0.87</td>
<td>-0.04</td>
<td>0.04</td>
<td>0.13</td>
<td>-0.08</td>
<td>0.00</td>
<td>0.12</td>
</tr>
<tr>
<td>First.Letters</td>
<td>-0.06</td>
<td>0.00</td>
<td>0.10</td>
<td>0.68</td>
<td>0.86</td>
<td>0.93</td>
<td>-0.13</td>
<td>0.00</td>
<td>0.13</td>
</tr>
<tr>
<td>4.Letter.Words</td>
<td>-0.14</td>
<td>-0.01</td>
<td>0.07</td>
<td>0.58</td>
<td>0.74</td>
<td>0.86</td>
<td>0.01</td>
<td>0.10</td>
<td>0.25</td>
</tr>
<tr>
<td>Suffixes</td>
<td>0.07</td>
<td>0.18</td>
<td>0.27</td>
<td>0.46</td>
<td>0.63</td>
<td>0.76</td>
<td>-0.20</td>
<td>-0.08</td>
<td>0.06</td>
</tr>
<tr>
<td>Letter.Series</td>
<td>-0.04</td>
<td>0.03</td>
<td>0.13</td>
<td>-0.10</td>
<td>-0.01</td>
<td>0.10</td>
<td>0.56</td>
<td>0.84</td>
<td>0.93</td>
</tr>
<tr>
<td>Pedigrees</td>
<td>0.25</td>
<td>0.37</td>
<td>0.46</td>
<td>-0.16</td>
<td>-0.05</td>
<td>0.08</td>
<td>0.27</td>
<td>0.47</td>
<td>0.66</td>
</tr>
<tr>
<td>Letter.Group</td>
<td>-0.16</td>
<td>-0.06</td>
<td>0.06</td>
<td>0.09</td>
<td>0.21</td>
<td>0.31</td>
<td>0.44</td>
<td>0.64</td>
<td>0.79</td>
</tr>
</tbody>
</table>

Interfactor correlations and bootstrapped confidence intervals

<table>
<thead>
<tr>
<th></th>
<th>lower estimate</th>
<th>upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.40</td>
<td>0.64</td>
</tr>
<tr>
<td>2</td>
<td>0.29</td>
<td>0.63</td>
</tr>
<tr>
<td>3</td>
<td>0.29</td>
<td>0.61</td>
</tr>
</tbody>
</table>
The simple factor structure

factor.diagram(f3) # show the diagram

Factor Analysis


MR1

0.9

0.9

0.8

MR2

0.9

0.7

0.6

MR3

0.8

0.6

0.5

MR4

0.5

0.5

0.6
Two ways of viewing the higher order structure

```r
om <- omega(Thurstone)
omega.diagram(om, sl = FALSE)
```
A hierarchical cluster structure found by `iclust`

`iclust(Thurstone)`

```
Letter.Group          0.77
Letter.Series         0.77
Pedigrees             0.8
Sent.Completion       0.93
Vocabulary            0.91
Sentences             0.96
Suffixes              0.84
4.Letter.Words        0.82
First.Letters         0.82
```

```
C3: α = 0.75
C6: α = 0.78
C8: α = 0.89
C7: α = 0.87
C4: α = 0.92
C5: α = 0.82
C2: α = 0.81
```
Structural Equation modeling packages

1. **sem** (by John Fox and others)
   - uses RAM notation

2. **lavaan** (by Yves Rosseel and others)
   - Mimics as much as possible MPLUS output
   - Allows for multiple groups
   - Easy syntax

3. **OpenMx**
   - Open source and R version of Mx
   - Allows for multiple groups (and almost anything else)
   - Complicated syntax
Multiple packages to do Item Response Theory analysis

1. *psych* uses a factor analytic procedure to estimate item discriminations and locations
   - `irt.fa` finds either tetrachoric or polychoric correlation matrices
   - Converts factor loadings to discriminations
   - `plot.irt` plots item information and item characteristic functions
   - Look at examples for `irt.fa`
   - Two example data sets: `iqitems` and `bfi`

2. Other packages include *ltm*, *eRm*, *mirt*, + others
Item Response Information curves for 14 iq items

Item information from factor analysis

Latent Trait (normal scale)

Item Information

0.0 0.2 0.4 0.6

-3 -2 -1 0 1 2 3

0.0 0.2 0.4 0.6

Item Information

1
3
5 6
7
9
10
11
13
A brief technical interlude

1. Data structures
   - The basic: scalars, vectors, matrices
   - More advanced data frames and lists
   - Showing the data

2. Getting the length, dimensions and structure of a data structure
   - length(x), dim(x), str(x)

3. Objects and Functions
   - Functions act upon objects
   - Functions actually are objects themselves
   - Getting help for a function or a package
The basic types of data structures

1. **Scalers (characters, integers, reals, complex)**
   ```r
   A <- 1
   B <- 2
   ```

2. **Vectors (of scalers, all of one type) have `length`**
   ```r
   C <- month.name[1:5]
   D <- 12:24
   length(D)
   ```
   ```r
   [1] 13
   ```

3. **Matrices (all of one type) have dimensions**
   ```r
   E <- matrix(1:20, ncol = 4)
   dim(E)
   ```
   ```r
   [1] 5 4
   ```
Show values by entering the variable name

> A
[1] 1

> B
[1] 2

> C
[1] "January" "February" "March" "April" "May"

> D

> E
[1,] 1  6  11  16
[2,] 2  7  12  17
[3,] 3  8  13  18
[4,] 4  9  14  19
[5,] 5 10  15  20
More complicated (and useful) types: Data frames and Lists

1. Data frames are collections of vectors and may be of different type. They have two dimensions.
   ```r
   > E.df <- data.frame(names = C, values = c(31, 28, 31, 30, 31))
   > dim(E.df)
   [1] 5 2
   ```

2. Lists are collections of whatever you want. They have length, but do not have dimensions.
   ```r
   > F <- list(first = A, a.vector = C, a.matrix = E)
   > length(F)
   [1] 3
   ```
Show values by entering the variable name

> E.df

    names values
 1 January    31
 2 February   28
 3 March     31
 4 April     30
 5 May       31

> F

$first
[1] 1

$a.vector
[1] "January"  "February"  "March"   "April"   "May"

$a.matrix

[1,]  1   6  11  16
[2,]  2   7  12  17
[3,]  3   8  13  18
[4,]  4   9  14  19
[5,]  5  10  15  20
To show the structure of a list, use `str`

```r
> str(F)
List of 3
  $ first   : num 1
  $ a.vector: chr [1:5] "January" "February" "March" "April" ...
  $ a.matrix: int [1:5, 1:4] 1 2 3 4 5 6 7 8 9 10 ...
```

To address an element of a list, call it by name or number, to get a row or column of a matrix specify the row, column or both.

```r
> F[[2]]
[1] "January"  "February"  "March"   "April"   "May"
> F[["a.matrix"]][, 2]
[1] 6 7 8 9 10
> F[["a.matrix"]][2, ]
[1] 2 7 12 17
```
Setting row and column names using `paste`

```r
E <- matrix(1:20, ncol = 4)
colnames(E) <- paste("C", 1:ncol(E), sep = "")
rownames(E) <- paste("R", 1:nrow(E), sep = "")
E

C1 C2 C3 C4
R1 1 6 11 16
R2 2 7 12 17
R3 3 8 13 18
R4 4 9 14 19
R5 5 10 15 20

E["R2", ]

C1 C2 C3 C4
2 7 12 17

E[, 3:4]

C3 C4
R1 11 16
R2 12 17
R3 13 18
R4 14 19
R5 15 20
```
Objects and Functions

1. **R is a collection of Functions that act upon and return Objects**
2. **Although most functions can act on an object and return an object** \((a = f(b))\), some are binary operators
   - primitive arithmetic functions \(+, -, *, /, \%\%\%
   - logical functions \(<, >, \leq, \geq, !=
3. **Some functions do not return values**
   - `print(x,digits=3)
   - `summary(some object)
4. **But most useful functions act on an object and return a resulting object**
   - this allows for extraordinary power because you can combine functions by making the output of one the input of the next.
   - The number of R functions is very large, for each package has introduced more functions, but for any one task, not many functions need to be learned.
Getting help

1. All functions have a help menu
   - `help(the function)`
   - `? the function`
   - most function help pages have examples to show how to use the function

2. Most packages have “vignettes” that give overviews of all the functions in the package and are somewhat more readable than the help for a specific function.
   - The examples are longer, somewhat more readable. (e.g., the vignette for `psych` is available either from the menu (Mac) or from [http://cran.r-project.org/web/packages/psych/vignettes/overview.pdf](http://cran.r-project.org/web/packages/psych/vignettes/overview.pdf)

3. To find a function in the entire R space, use `findFn` in the `sos` package.

4. Online tutorials (e.g., [http://Rpad.org](http://Rpad.org) for a list of important commands, [http://personality-project.org/r](http://personality-project.org/r)) for a tutorial for psychologists.

5. Online and hard copy books
Useful functions

A few of the most useful data manipulations functions (adapted from Rpad-refcard). Use ? for details

- `file.choose()` find a file
- `file.choose(new=TRUE)` create a new file
- `read.table(filename)`
- `read.csv(filename)` reads a comma separated file
- `read.delim(filename)` reads a tab delimited file
- `c(...)` combine arguments
- `from:to` e.g., 4:8
- `seq(from,to,by)`
- `rep(x,times)` repeat x
- `gl(n,k,...)` generate factor levels
- `matrix(x,nrow=,ncol=)` create a matrix
- `data.frame(...)` create a data frame
- `dim(x)` dimensions of x
- `str(x)` Structure of an object
- `list(...)` create a list
- `colnames(x)` (x) set or find column names
- `rownames(x)` (x) set or find row names
- `nrow(z)` number of row, columns
- `rbind(...)` combine by rows
- `cbind(...)` combine by columns
- `is.na(x)` also is.null(x), is...
- `na.omit(x)` ignore missing data
- `table(x)`
- `merge(x,y)`
- `apply(x,rc,FUNCTION)`
- `ls()` show workspace
- `rm()` remove variables from workspace
### Useful functions

**More useful statistical functions, Use `?` for details**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>mean(x)</code></td>
<td>Average of the values in <code>x</code></td>
</tr>
<tr>
<td><code>is.na(x)</code></td>
<td>Check if <code>x</code> contains missing values, also <code>is.null(x)</code></td>
</tr>
<tr>
<td><code>na.omit(x)</code></td>
<td>Ignore missing data in <code>x</code></td>
</tr>
<tr>
<td><code>sum(x)</code></td>
<td>Sum of the values in <code>x</code></td>
</tr>
<tr>
<td><code>rowSums(x)</code></td>
<td>Sum of each row in <code>x</code></td>
</tr>
<tr>
<td><code>min(x)</code></td>
<td>Minimum value in <code>x</code></td>
</tr>
<tr>
<td><code>max(x)</code></td>
<td>Maximum value in <code>x</code></td>
</tr>
<tr>
<td><code>range(x)</code></td>
<td>Range of the values in <code>x</code></td>
</tr>
<tr>
<td><code>table(x)</code></td>
<td>Frequency table of <code>x</code></td>
</tr>
<tr>
<td><code>summary(x)</code></td>
<td>Summary statistics of <code>x</code></td>
</tr>
<tr>
<td><code>sd(x)</code></td>
<td>Standard deviation of <code>x</code></td>
</tr>
<tr>
<td><code>cor(x)</code></td>
<td>Correlation coefficient between <code>x</code> and another vector</td>
</tr>
<tr>
<td><code>cov(x)</code></td>
<td>Covariance between <code>x</code> and another vector</td>
</tr>
<tr>
<td><code>solve(x)</code></td>
<td>Inverse of <code>x</code></td>
</tr>
<tr>
<td><code>lm(y~x)</code></td>
<td>Linear model with <code>y</code> as the dependent variable and <code>x</code> as the independent</td>
</tr>
<tr>
<td><code>aov(y~x)</code></td>
<td>ANOVA model with <code>y</code> as the dependent variable and <code>x</code> as the factor</td>
</tr>
</tbody>
</table>

**Selected functions from `psych` package**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>describe(x)</code></td>
<td>Descriptive statistics of <code>x</code></td>
</tr>
<tr>
<td><code>describe.by(x,y)</code></td>
<td>Descriptives by group of <code>x</code> and <code>y</code></td>
</tr>
<tr>
<td><code>pairs.panels(x)</code></td>
<td>SPLOM plot for <code>x</code></td>
</tr>
<tr>
<td><code>error.bars(x)</code></td>
<td>Means + error bars of <code>x</code></td>
</tr>
<tr>
<td><code>error.bars.by(x)</code></td>
<td>Error bars by groups of <code>x</code></td>
</tr>
<tr>
<td><code>fa(x,n)</code></td>
<td>Factor analysis with <code>n</code> variables</td>
</tr>
<tr>
<td><code>principal(x,n)</code></td>
<td>Principal components with <code>n</code> variables</td>
</tr>
<tr>
<td><code>iclust(x)</code></td>
<td>Item cluster analysis</td>
</tr>
<tr>
<td><code>score.items(x)</code></td>
<td>Score multiple scales of <code>x</code></td>
</tr>
<tr>
<td><code>score.multiple.choice(x)</code></td>
<td>Score multiple choice scales of <code>x</code></td>
</tr>
<tr>
<td><code>alpha(x)</code></td>
<td>Cronbach’s alpha</td>
</tr>
<tr>
<td><code>omega(x)</code></td>
<td>MacDonald’s omega</td>
</tr>
<tr>
<td><code>irt.fa(x)</code></td>
<td>Item response theory through factor analysis of <code>x</code></td>
</tr>
</tbody>
</table>
Questions?