

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

William Revelle, David M. Condon & Sara Weston*
Northwestern University
Evanston, Illinois USA
*Washington University, St. Louis, USA

<https://personality-project.org/r/aps/aps-short.pdf>
Partially supported by a grant from the National Science Foundation: SMA-1419324



Outline

Part I: What is R, where did it come from, why use it

- Installing R and adding packages

Part II: A brief introduction – an overview

- R is just a fancy (very fancy) calculator
- Descriptive data analysis
- Some inferential analysis

Part III R is a powerful statistical system

- Data entry
- Descriptive
- Inferential (t and F)
- Regression
- Basic R commands

Part IV: Psychometrics

- Reliability and its discontents
- Exploratory Factor Analysis, Confirmatory Factor Analysis, SEM

Part V: Help and More Help

- List of useful commands



Outline of Part I

What is R?

Where did it come from, why use it?

Misconceptions

Installing R on your computer and adding packages

Installing for your operating system

R-Applications

Installing and using packages

What are packages

Installing packages

-> Part II



Where did it come from, why use it?

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
5. List of major commands and packages

Although programming is easy in R, that is beyond the scope of today



Where did it come from, why use it?

R: What is it?

1. R: An international collaboration
2. R: The open source - public domain version of S+
3. R: Written by statisticians (and some 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.



Where did it come from, why use it?

Statistical Programs for Psychologists

- General purpose programs
 - R
 - S+
 - SAS
 - SPSS
 - STATA
 - Systat
- Specialized programs
 - Mx
 - EQS
 - AMOS
 - LISREL
 - MPlus
 - Your favorite program



Where did it come from, why use it?

Statistical Programs for Psychologists

- General purpose programs
 - R
 - \$+
 - \$\$
 - \$\$\$
 - \$TATA
 - \$y\$at
- Specialized programs
 - Mx (OpenMx is part of R)
 - EQ\$
 - AMO\$
 - LI\$REL
 - MPlu\$
 - Your favorite program



Where did it come from, why use it?

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.”
- Q: My institute has been heavily dependent on SAS for the past while, and SAS is starting to charge us a very deep amount for license renewal.... The team is [considering] switching to R, ... I am talking about the entire institute with considerable number of analysts using SAS their entire career. ... What kind of problems and challenges have you faced?
A: “One of your challenges will be that with the increased productivity of the team you will have time for more intellectually challenging problems. That frustrates some people.”



Where did it come from, why use it?

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.
- A: Look at the code. Borrow what you need.



Where did it come from, why use it?

What is R?: Technically

- R is an open source implementation of S (The statistical language developed at Bell Labs). (S-Plus is a commercial implementation)
- R is a language and environment for statistical computing and graphics. R is available under GNU Copy-left
- R is a group project run by a core group of developers (with new releases semiannually). The current version of R is 3.3.0
- R is an integrated suite of software facilities for data manipulation, calculation and graphical display.

(Adapted from Robert Gentleman and the r-project.org web page)



Where did it come from, why use it?

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It is:

1. an effective data handling and storage facility,
2. a suite of operators for calculations on arrays, in particular matrices,
3. a large, coherent, integrated collection of intermediate tools for data analysis,
4. graphical facilities for data analysis and display either on-screen or on hardcopy, and
5. a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

“Many users think of R as a statistics system. We prefer to think of it as an environment within which statistical techniques are implemented. R can be extended (easily) via packages ... available through the CRAN family of Internet sites covering a very wide range of modern statistics.” (Adapted from r-project.org web page)



Where did it come from, why use it?

R: A brief history

- 1991-93: Ross Ihaka and Robert Gentleman begin work on R project for Macs at U. Auckland (S for Macs).
- 1995: R available by ftp under the General Public License.
- 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)
- 2001-2016: Core team continues to improve base package with a new release every 6 months (now more like yearly).
- 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
 - 2012-08-27: 4,000 packages
 - 2014-05-16: 5,547 packages (on CRAN) + 824 bioinformatic packages on BioConductor
 - 2015-05-20 6,678 packages (on CRAN) + 1024 bioinformatic packages + ?,000s on GitHub
 - 2016-03-31 8,427 packages (on CRAN) + 1,104 bioinformatic packages + ?,000s on

GitHub/R-Forge (increased by 245 in last 30 days)



○○○○○○○○●○○○
○○

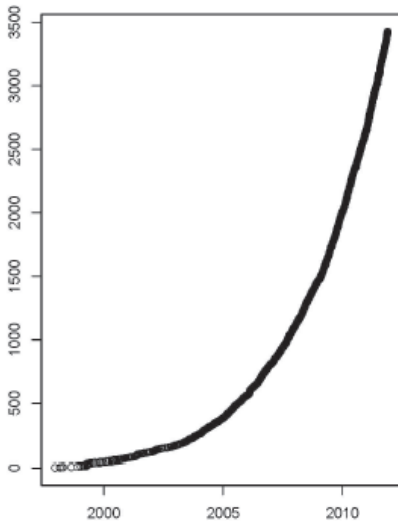
○○○○

○○○
○○

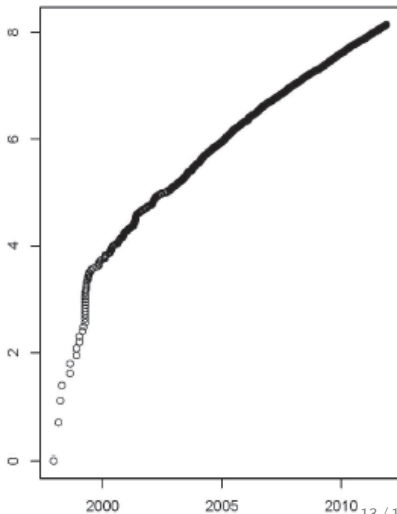
Where did it come from, why use it?

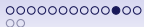
Rapid and consistent growth in packages contributed to R

Number of Active CRAN Packages



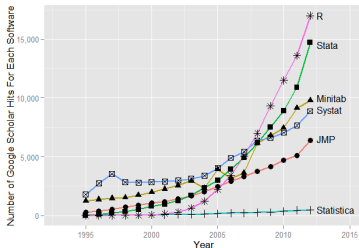
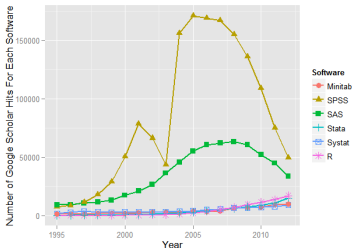
Log Number of Active CRAN Packages





Where did it come from, why use it?

Popularity compared to other statistical packages



<http://r4stats.com/articles/popularity/> considers various measures of popularity

1. discussion groups
2. blogs
3. Google Scholar citations ($> 27,000$ citations, $\approx 1,800/\text{year}$)
4. Google Page rank



Where did it come from, why use it?

R as a way of facilitating replicable science

1. R is not just for statisticians, it is for all research oriented psychologists.
2. R scripts are published in psychology journals to show new methods:
 - *Psychological Methods*
 - *Psychological Science*
 - *Journal of Research in Personality*
3. R based data sets are now accompanying journal articles:
 - The *Journal of Research in Personality* now accepts R code and data sets.
 - JRP special issue in R,
4. By sharing our code and data the field can increase the possibility of doing replicable science.



Where did it come from, why use it?

Reproducible Research: Sweave and KnitR

Sweave is a tool that allows to embed the R code for complete data analyses in \LaTeX documents. The purpose is to create dynamic reports, which can be updated automatically if data or analysis change. Instead of inserting a prefabricated graph or table into the report, the master document contains the R code necessary to obtain it. When run through R, all data analysis output (tables, graphs, etc.) is created on the fly and inserted into a final \LaTeX document. The report can be automatically updated if data or analysis change, which allows for truly reproducible research.

Friedrich Leisch (2002). Sweave: Dynamic generation of statistical reports using literate data analysis. I

Supplementary material for journals can be written in Sweave/KnitR.



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 menu after 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>, 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 (usually)
 - Multiple web based and pdf tutorials see (e.g., <http://www.r-project.org/>)
 - Short courses using R for many applications. (Look at APS program).
4. Books and websites for SPSS and SAS users trying to learn R (e.g., <http://r4stats.com/>) by Bob Muenchen (look for link to free version).



Go to the R.project.org



[Home]

Download

[CRAN](#)

R Project

[About R](#)
[Contributors](#)
[What's New?](#)
[Mailing Lists](#)
[Bug Tracking](#)
[Conferences](#)
[Search](#)

R Foundation

[Foundation](#)
[Board](#)
[Members](#)
[Donors](#)
[Donate](#)

The R Project for Statistical Computing

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](#).

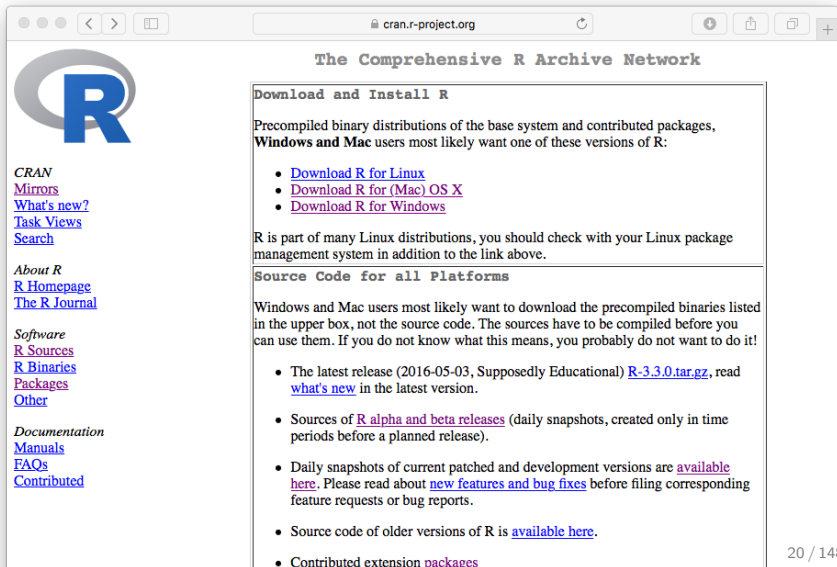
If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- **R version 3.2.0** (Full of Ingredients) has been released on 2015-04-16.
- **R version 3.1.3** (Smooth Sidewalk) has been released on 2015-03-09.
- **The R Journal Volume 6/2** is available.
- **useR! 2015**, will take place at the University of Aalborg, Denmark, June 30 - July 3, 2015.
- **useR! 2014**, took place at the University of California, Los Angeles, USA June 30 - July 3, 2014.



Go to the Comprehensive R Archive Network (CRAN)



The screenshot shows a web browser window displaying the CRAN website. The browser's address bar contains 'cran.r-project.org'. The page title is 'The Comprehensive R Archive Network'. On the left side, there is a navigation menu with links for 'CRAN', 'Mirrors', 'What's new?', 'Task Views', 'Search', 'About R', 'R Homepage', 'The R Journal', 'Software', 'R Sources', 'R Binaries', 'Packages', 'Other', 'Documentation', 'Manuals', 'FAQs', and 'Contributed'. The main content area is titled 'The Comprehensive R Archive Network' and contains two main sections: 'Download and Install R' and 'Source Code for all Platforms'. The 'Download and Install R' section provides precompiled binary distributions for Windows and Mac users, with links to 'Download R for Linux', 'Download R for (Mac) OS X', and 'Download R for Windows'. The 'Source Code for all Platforms' section explains that Windows and Mac users should download precompiled binaries rather than source code, and lists several bullet points regarding the latest release (2016-05-03), daily snapshots, and source code availability.

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 \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

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!

- The latest release (2016-05-03, Supposedly Educational) [R-3.3.0.tar.gz](#), read [what's new](#) in the latest version.
- 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](#)

Download and install the appropriate version – PC

The screenshot shows a web browser window at cran.r-project.org. The page title is "R-3.3.0 for Windows (32/64 bit)". On the left sidebar, there are links for "CRAN", "Mirrors", "What's new?", "Task Views", "Search", "About R", "R Homepage", "The R Journal", "Software", "R Sources", "R Binaries", "Packages", "Other", "Documentation", "Manuals", and "FAQs". The main content area features a large "Download R 3.3.0 for Windows (62 megabytes, 32/64 bit)" button, followed by links for "Installation and other instructions" and "New features in this version". A paragraph explains how to verify the downloaded package using md5sum. Below this is a "Frequently asked questions" section with three bullet points: "Does R run under my version of Windows?", "How do I update packages in my previous version of R?", and "Should I run 32-bit or 64-bit R?". At the bottom, there is a link to the "R FAQ" and "R Windows FAQ", and an "Other builds" section with two bullet points: "Patches to this release are incorporated in the r-patched snapshot build." and "A build of the development version (which will eventually become the next major release)".

R-3.3.0 for Windows (32/64 bit)

[Download R 3.3.0 for Windows](#) (62 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

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

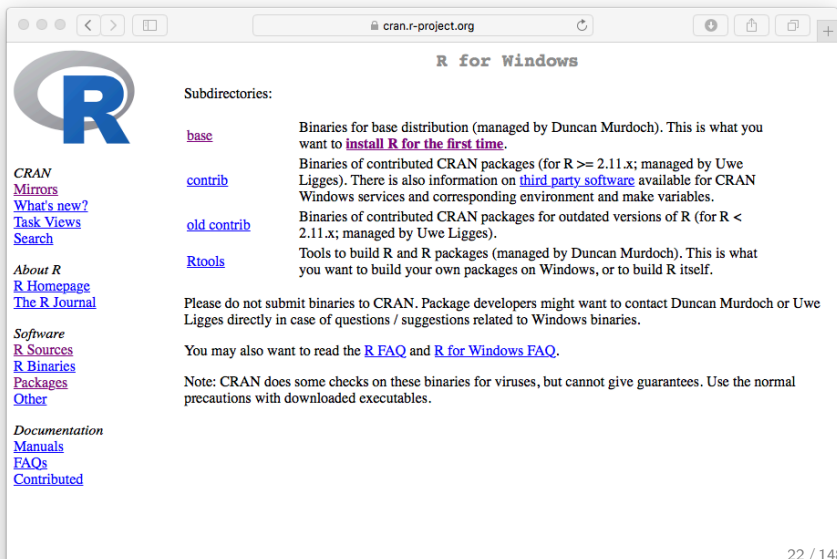
- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)
- [Should I run 32-bit or 64-bit R?](#)

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)

Download and install the appropriate version – PC



The screenshot shows a web browser window with the address bar set to `cran.r-project.org`. The page title is "R for Windows". On the left side, there is a navigation menu with the following links: [CRAN](#), [Mirrors](#), [What's new?](#), [Task Views](#), [Search](#), [About R](#), [R Homepage](#), [The R Journal](#), [Software](#), [R Sources](#), [R Binaries](#), [Packages](#), [Other](#), [Documentation](#), [Manuals](#), [FAQs](#), and [Contributed](#). The main content area is titled "Subdirectories:" and lists three categories: [base](#), [contrib](#), and [old_contrib](#). Each category has a brief description of the binaries it contains. At the bottom, there is a note about submitting binaries to CRAN and a link to the [R FAQ](#) and [R for Windows FAQ](#).

R for Windows

Subdirectories:

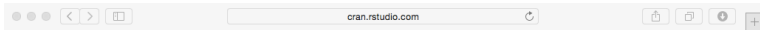
- [base](#) Binaries for base distribution (managed by Duncan Murdoch). This is what you want to [install R for the first time](#).
- [contrib](#) Binaries of contributed CRAN packages (for R \geq 2.11.x; managed by Uwe Ligges). There is also information on [third party software](#) available for CRAN Windows services and corresponding environment and make variables.
- [old_contrib](#) Binaries of contributed CRAN packages for outdated versions of R (for R $<$ 2.11.x; managed by Uwe Ligges).
- [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.

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 10.6 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 1.7.1) [here](#). Releases for old Mac OS X systems (through Mac OS X 10.5) and PowerPC Macs can be found in the [old](#) directory.

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

R 3.2.0 "Full of Ingredients" released on 2015/04/18

This binary distribution of R and the GUI supports 64-bit Intel based Macs on Mac OS X 10.9 (Mavericks) or higher.

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

```
md5 R-3.2.0.pkg
```

in the *Terminal* application to print the MD5 checksum for the R-3.2.0.pkg image. On Mac OS X 10.7 and later you can also validate the signature using `pkgutil --check-signature R-3.2.0.pkg`

Files:

R-3.2.0.pkg

```
MD5:hash: e864e66b37d3bb4030ac21c9e8797b24
SHA1:
hash: 67316440d7ab531012b3a6873b11c4a8e71e7194
(ca. 70MB)
```

R 3.2.0 binary for Mac OS X 10.9 (Mavericks) and higher, signed package. Contains R 3.2.0 framework, R.app GUI 1.65 in 64-bit for Intel Macs, Tcl/Tk 8.6.0 X11 libraries and Texinfo 5.2. The latter two components are optional and can be omitted when choosing "custom install", it is only needed if you want to use the `tcltk` R package or build package documentation from sources.

Note: the use of X11 (including `tcltk`) requires [XQuartz](#) to be installed since it is no longer part of OS X. Always re-install XQuartz when upgrading your OS X to a new major version.

(If you are using legacy OS X 10.6 through 10.8 and are interested in R 3.2.0, please see the [R for Mac development page](#).)

R-3.1.3-snowleopard.pkg

```
MD5:hash: 8bb33518236635460990409f115ec7
```

R 3.1.3 binary for Mac OS X 10.6 (Snow Leopard) and higher, signed package. Contains R 3.1.3 framework, R.app GUI 1.65 in 64-bit for Intel

- CRAN
- [Mirrors](#)
- [What's new?](#)
- [Task Views](#)
- [Search](#)

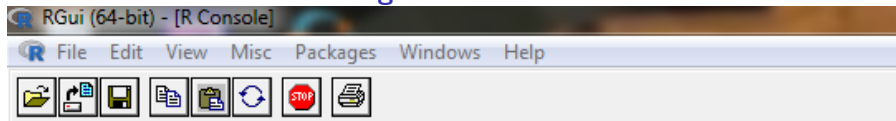
- About R
- [R Homepage](#)
- [The R Journal](#)

- Software
- [R Sources](#)
- [R Binaries](#)
- [Packages](#)
- [Other](#)

- Documentation
- [Manuals](#)
- [FAQs](#)
- [Contributed](#)



Starting R on a PC



```
R version 3.3.0 (2016-05-03) -- "Supposedly Educational"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.
```


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

```
R Under development (unstable) (2016-05-10 r70594) -- "Unsuffered Cons  
Copyright (C) 2016 The R Foundation for Statistical Computing  
Platform: x86_64-apple-darwin13.4.0 (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 1.68 (7213) x86_64-apple-darwin13.4.0]
```

```
[Workspace restored from /Users WR/.RData]  
[History restored from /Users WR/.Rapp.history]
```



Check the version number for R \geq 3.3.0) and for psych (\geq 1.6.4)

R code

```
library(psych) #make the psych package active
sessionInfo() #what packages are active
```

R Under development (unstable) (2016-05-10 r70594)

Platform: x86_64-apple-darwin13.4.0 (64-bit)

Running under: OS X 10.11.4 (El Capitan)

locale:

```
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
```

attached base packages:

```
[1] stats      graphics  grDevices  utils      datasets  methods   base
```

other attached packages:

```
[1] psych_1.6.4
```

loaded via a namespace (and not attached):

```
[1] parallel_3.4.0 mnormt_1.5-4
```



Various ways to run R

1. UNIX (and *NIX like) environments
 - Can be scripted for use on remote servers
 - Particularly fast if on remote processors with many cores
 - RStudio Server as “Integrated Development Environment” (IDE)
2. PC
 - quasi GUI + text editor of choice
 - RStudio as “Integrated Development Environment” (IDE) (recommended by Sara)
3. Mac
 - R.app + text editor of choice (preferred by Bill)
 - RStudio as “Integrated Development Environment” (IDE) (preferred by David)
 - allows for multiple cores for parallel processing
4. From the web
 - [R Fiddle](#)





R-Applications

R Studio is a useful “Integrated Development Environment” (IDE)

The screenshot displays the RStudio IDE with the following components:

- Script Editor:** Contains R code for loading the 'psych' package, creating a data frame 'myData' from 'sat.act', and cleaning it to create 'cleaned'.
- Console:** Shows the execution of the code, including summary statistics for 'myData' and 'cleaned', and an error message: "Error in `[.data.frame`(x, , 1) : undefined columns selected".
- Environment Pane:** Lists the objects 'cleaned' and 'myData' with their dimensions and variable types.
- Diagnostic Plot:** A grid of plots for each variable (gender, education, age, ACT, SATV, SATQ) showing density, histogram, and boxplot views, along with correlation coefficients.

```

1 library(psych)
2 myData <- sat.act
3 pairs.panels(myData)
4 describe(myData)
5 cleaned <- scrub(myData,"ACT",min=5)
6 describe(cleaned)

```

```

6:18 (Top Level) : R Script :

```

```

Console ~/
ACT      4 700 28.55 4.82 29 28.84 4.45 3 36 33 -0.66 0.53 0.18
SATV     5 700 61.23 112.90 620 619.45 118.61 200 800 600 -0.64 0.33 4.27
SATQ     6 687 610.22 115.64 620 617.25 118.61 200 800 600 -0.59 -0.02 4.41
> cleaned <- scrub(myData,"act",min=5)
Error in `[.data.frame`(x, , 1) : undefined columns selected
> describe(myData)
  vars  n  mean  sd median trimmed  mod min max range skew kurtosis se
gender 1 700 1.65 0.48 2 1.68 0.00 1 2 1 -0.61 -1.62 0.02
education 2 700 3.16 1.43 3 3.31 1.48 0 5 5 -0.68 -0.07 0.05
age 3 700 25.59 9.50 22 23.86 5.93 13 65 52 1.64 2.42 0.36
ACT 4 700 28.55 4.82 29 28.84 4.45 3 36 33 -0.66 0.53 0.18
SATV 5 700 61.23 112.90 620 619.45 118.61 200 800 600 -0.64 0.33 4.27
SATQ 6 687 610.22 115.64 620 617.25 118.61 200 800 600 -0.59 -0.02 4.41
> cleaned <- scrub(myData,"ACT",min=5)
> describe(cleaned)
  vars  n  mean  sd median trimmed  mod min max range skew kurtosis se
gender 1 700 1.65 0.48 2 1.68 0.00 1 2 1 -0.61 -1.62 0.02
education 2 700 3.16 1.43 3 3.31 1.48 0 5 5 -0.68 -0.07 0.05
age 3 700 25.59 9.50 22 23.86 5.93 13 65 52 1.64 2.42 0.36
ACT 4 699 28.58 4.73 29 28.85 4.45 15 36 21 -0.50 -0.37 0.18
SATV 5 700 61.23 112.90 620 619.45 118.61 200 800 600 -0.64 0.33 4.27
SATQ 6 687 610.22 115.64 620 617.25 118.61 200 800 600 -0.59 -0.02 4.41
>

```

Environment History

Global Environment

cleaned 700 obs. of 6 variables

```

gender : int 2 2 2 1 1 1 2 1 2 2 ...
education: int 3 3 3 4 2 5 5 3 4 5 ...
age : int 19 23 20 27 33 26 30 19 23 40 ...
ACT : int 24 35 21 26 31 28 36 22 22 35 ...
SATV : int 500 600 480 550 600 640 610 520 400 730 ...
SATQ : int 500 500 470 520 550 640 500 560 600 800 ...

```

myData 700 obs. of 6 variables

```

gender : int 2 2 2 1 1 1 2 1 2 2 ...

```

Files Plots Packages Help Viewer

Zoom Export Clear All

gender 0 2 4 5 20 35 200 500 800 1.0 2.0

education 0 3 1.0 1.6

age 5 30 20 40 60

ACT 200 800 200 500 800

SATV 200 800

SATQ 200 800

Correlation Matrix:

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					

R Studio may be run on a remote server

The screenshot shows the RStudio interface running on a remote server. The browser address bar indicates the URL `https://revelle.ci.northwestern.edu/rstudio/`. The console window on the left contains the following R code and output:

```

> library(psych)
> sessionInfo()
R version 3.0.2 (2013-09-25)
Platform: x86_64-redhat-linux-gnu (64-bit)

locale:
[1] C

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

other attached packages:
[1] psych_1.4.5

loaded via a namespace (and not attached):
[1] tools_3.0.2
> ?cor.ci
> keys.list <-
+ list(agree=c("A1", "A2", "A3", "A4", "A5"), conscientious=c("C1", "C2", "C3", "C4", "C5"),
+ extraversion=c("E1", "E2", "E3", "E4", "E5"), neuroticism=c("N1", "N2", "N3", "N4", "N5"),
+ openness = c("O1", "O2", "O3", "O4", "O5"))
> keys <- make.keys(bfi, keys.list)
> rci <- cor.ci(bfi[1:200], keys, n.iter=10) #also shows the graphic
Loading required package: parallel
> cor.plot.upperLowerCi(rci) #to show the upper and lower confidence intervals
>
|
  
```

The main window displays a heatmap titled "Upper and lower confidence intervals of correlations". The heatmap shows the correlation coefficients between five personality traits: agree, conscientious, extraversion, neuroticism, and openness. The diagonal elements are all 1.0. The off-diagonal elements are: agree-conscientious (0.38), agree-extraversion (0.51), agree-neuroticism (-0.26), agree-openness (0.39), conscientious-extraversion (0.41), conscientious-neuroticism (-0.18), conscientious-openness (0.35), extraversion-neuroticism (-0.39), extraversion-openness (0.39), neuroticism-openness (-0.24), and openness-neuroticism (0.01). The heatmap uses a color scale from -1 (red) to 1 (blue).

	agree	conscientious	extraversion	neuroticism	openness
agree	1	0.38	0.51	-0.26	0.39
conscientious	0.1	1	0.41	-0.18	0.35
extraversion	0.27	0.19	1	-0.39	0.39
neuroticism	-0.1	0.06	-0.12	1	-0.24
openness	-0.03	0.14	0.15	0.01	1

R-Applications

Fiddle allows to run on a remote server hosted by datacamp (but R = 3.1.2 and psych = 1.3.12)

The screenshot shows the R-Fiddle web interface. On the left, there is a code editor with the following R code:

```

1 library(psych)
2 pairs.panels(sat.act)
3 f3 <- fa(Thurstone,3)
4 fa.diagram(f3)
5 describe(sat.act)
6 |
    
```

On the right, a 'Factor Analysis' diagram is displayed, showing three latent variables (FA1, FA2, FA3) and their associated observed variables with loadings. For example, 'Generalized' and 'Vocabulary' both load on FA1 with a coefficient of 0.9. 'Self-Confidence' loads on FA1 with a coefficient of -0.8. 'Form Language' and 'Aptitude' both load on FA2 with a coefficient of 0.9. 'Vocabulary' and 'Form Language' both load on FA3 with a coefficient of 0.6. The diagram also shows correlations between the latent variables: FA1 and FA2 (0.6), FA1 and FA3 (0.5), and FA2 and FA3 (0.5).

Below the code editor, there are two buttons: 'Graphs' and 'Run Code'. Below the diagram, there are two buttons: 'Graphs' and 'Run Code'.

Below the 'Run Code' button, the output of the R code is displayed:

```

Loading required package: MASS

Loading required package: GPArotation

NULL

      var  n mean   sd median trimmed   mad min max range skew
gender 1 700  1.65  0.48    2   1.68  0.00  1  2    1 -0.61
education 2 700  3.16  1.43    3   3.31  1.48  0  5    5 -0.68
age 3 700 25.59  9.50   22  23.86  5.93 13 65   52  1.64
ACT 4 700 28.55  4.82   29  28.84  4.45  3 36   33 -0.66
SATV 5 700 612.23 112.90  620 619.45 118.61 200 800  600 -0.64
SATQ 6 687 610.22 115.64  620 617.25 118.61 200 800  600 -0.59

      kurtosis  se
gender -1.62 0.02
education -0.07 0.05
age 2.42 0.36
ACT 0.53 0.18
SATV 0.33 4.27
SATQ -0.02 4.41
    
```



What are packages

R is extensible: The use of “packages”

1. More than 8,427 packages are available for R (and growing daily. It was 6,652 last year).
2. 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("principal components")` #will return 2,675 matches from 173 packages and reports the top 400
 - `findFn("Item Response Theory")` # will return 510 matches in 77 packages
 - `findFn("INDSCAL ")` # will return 18 matches in 5 packages.
3. `install.packages("X")` will install a particular package (add it to your R library – you need to do this just once)
4. `library(X)` #will make the package X available to use if it has been installed (and thus in your library)



What are packages

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
 - > 8,427 known
 - + ?
- General applications
 - most descriptive and inferential stats
 - Modern Applied Statistics with S
 - Lattice or Trellis graphics
 - Linear mixed-effects models
 - Personality/psychometrics general purpose
 - General purpose toolkit
- More specialized packages
 - Latent Trait Model (IRT)
 - SEM and CFA (RAM path notation)
 - 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



What are packages

Even more very useful packages (see also Computer World list)

- General use
 - devtools
 - readxl
 - foreign
 - RMySQL
 - readr
 - rio
- Special use
 - dplyr
 - plyr
 - data.table
 - knitr
 - sweave
 - ggplot2
 - > 8,427 known
 - + ?
- General applications
 - Get packages from GitHub
 - input from excel
 - input from SPSS, etc.
 - input from MySQL
 - fast input for very large csv files
 - simple to use integrated input/output
- More specialized packages
 - reshape from wide to long etc.
 - reshape
 - faster data handling for large data sets
 - integrate markdown documentation with R
 - integrate \LaTeX documentation with R
 - powerful grammar of graphics
 - Thousands of more packages on CRAN
 - Code on webpages/journal articles



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 3.3.0) (See a tutorial on how to install R and various packages at <http://personality-project.org/r/psych>)
- 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 ctv package
 - `install.views("Psychometrics")` #among others
 - Take a 5 minute break
- Activate the package(s) you want to use today (e.g., *psych*)
 - `library(psych)` #necessary for most of today's examples
- Use R



Installing packages

Annotated installation guide: don't type the >

```

> install.packages("ctv")

> library(ctv)

> install.views("Psychometrics")

#or just install a few packages
> install.packages("psych",
                  dependencies=TRUE)
#which installs psych and its
  required packages

> install.packages("GPARotation")
> install.packages("mnormt")

```

- 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.



Questions?



Outline

1. Part I: What is R, where did it come from, why use it
 - Installing R and adding packages
2. Part II: A brief introduction – an overview
 - R is just a fancy (very fancy) calculator
 - Descriptive data analysis
 - Some inferential analysis
3. Part III: Using R
 - Data entry
 - Descriptive
 - Inferential (t and F)
 - Regression, partial correlation, mediation
 - Basic R commands
4. Part IV: Psychometrics
 - Reliability and its discontents (α , ω_h , ω_t , λ_6)
 - EFA, CFA, and SEM
5. Part V: Help and More Help
 - List of useful commands



Outline of Part II

-> Part I: What is R

Basic R: A brief example

Basic R capabilities: Calculation, Statistical tables

Basic Graphics

A brief example of exploratory and confirmatory data analysis

Data preparation, descriptive statistics, data cleaning,
correlation plots

Inferential statistics

Multiple regression modeling and graphics

-> Part III: Basic statistics and graphics



Basic R commands – remember don't enter the >

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

```
> 2 + 2          #sum two numbers
```

```
[1] 4           #show the output
```

```
> 3^4           #3 raised to the 4th
```

```
[1] 81          #that was easy
```

```
> sum(1:10)     #find the sum of the first 10 numbers
```

```
[1] 55          #the answer
```

```
> prod(c(1, 2, 3, 5, 7)) #the product of the concatenated (c) numbers
```

```
[1] 210         #Note how we combined product with concatenate
```

It is also a statistics table (the normal distribution, the t, the F, the χ^2 distribution, the xyz distribution)

```
> pnorm(q = 1)  #the probability of a normal with value of 1 sd
```

```
[1] 0.8413447   #
```

```
> pt(q = 2, df = 20) #what about the probability of a t-test value of
```

```
[1] 0.9703672   #this is the upper tail
```



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`.) Each function can be modified with various parameters.

Distribution	base name	P 1	P 2	P 3	example application
<i>Normal</i>	norm	mean	sigma		Most data
<i>Multivariate normal</i>	mvnorm	mean	r	sigma	Most data
<i>Log Normal</i>	lnorm	log mean	log sigma		income or reaction time
<i>Uniform</i>	unif	min	max		rectangular distributions
<i>Binomial</i>	binom	size	prob		Bernuilli trials (e.g. coin flips)
<i>Student's t</i>	t	df		nc	Finding significance of a t-test
<i>Multivariate t</i>	mvt	df	corr	nc	Multivariate applications
<i>Fisher's F</i>	f	df1	df2	nc	Testing for significance of F test
χ^2	chisq	df		nc	Testing for significance of χ^2
<i>Exponential</i>	exp	rate			Exponential decay
<i>Gamma</i>	gamma	shape	rate	scale	distribution theoryh
<i>Hypergeometric</i>	hyper	m	n	k	
<i>Logistic</i>	logis	location	scale		Item Response Theory
<i>Poisson</i>	pois	lambda			Count data
<i>Weibull</i>	weibull	shape	scale		Reaction time distributions



An example of using r, p, and q for a distributions

R code

```
set.seed(42) #set the random seed to get the same sequence
x <- rnorm(5) #find 5 randomly distributed normals
round(x,2) #show them, rounded to 2 decimals
round(pnorm(x),2) #show their probabilities to 2 decimals
round(qnorm(pnorm(x)),2) #find the quantiles of the normal
```

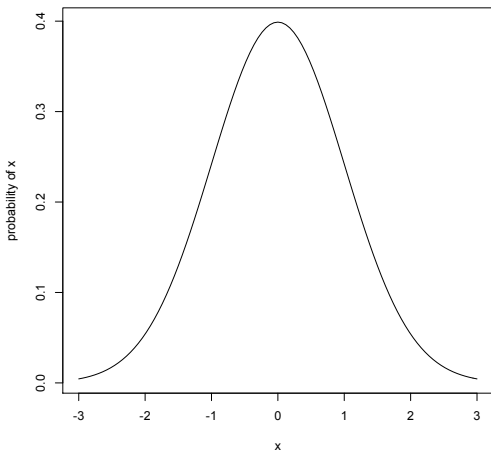
Produces this output

```
> set.seed(42) #set the random seed to get the same sequence
> x <- rnorm(5) #find 5 randomly distributed normals
> round(x,2) #show them, rounded to 2 decimals
[1] 1.37 -0.56 0.36 0.63 0.40
> round(pnorm(x),2) #show their probabilities to 2 decimals
[1] 0.91 0.29 0.64 0.74 0.66
> round(qnorm(pnorm(x)),2) #find the quantiles of the normal
[1] 1.37 -0.56 0.36 0.63 0.40
```



R can draw distributions

A normal curve



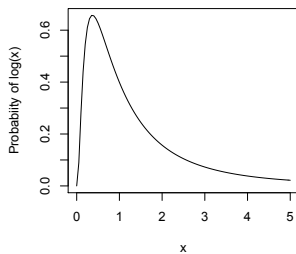
We do this by using the curve function to which we pass the values of the dnorm function.

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

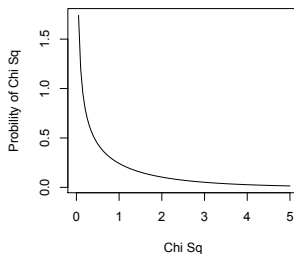
Basic Graphics

R can draw more interesting distributions

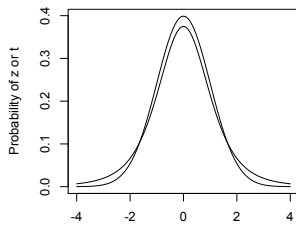
Log normal



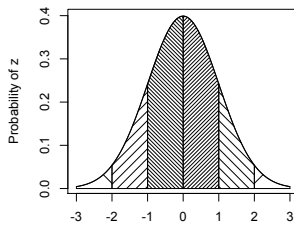
Chi Square distribution



Normal and t with 4 df



The normal curve



R is also a graphics calculator

R code

```

op <- par(mfrow=c(2,2))      #set up a 2 x 2 graph
curve(dlnorm(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)
#
#somewhat more complicated
#first draw the normal curve
curve(dnorm(x),-3,3,xlab="",ylab="Probability of z") #the range of x
title(main="The normal curve",outer=FALSE) #the title
#add the cross hatching by using polygons
xvals <- seq(-3,-2,length=100) #From -3 to 2 with 100 points
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)
op <- par(mfrow=c(1,1)) #back to a normal 1 x 1 graph

```



R can help teach with 100s of example data sets.

```
> data()
```

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

```
> data(Titanic)
```

```
> ? Titanic
```

```
> data(cushny)
```

```
> ? cushney
```

```
> data(UCBAdmissions)
```

```
> ? UCBAdmissions
```

1. This opens up a separate text window and lists all of the data sets in the currently loaded packages.
2. Show the data sets available in a particular package (e.g., *psych*).
3. Gets the particular data set with its help file (e.g., the survival rates on the Titanic cross classified by age, gender and class).
4. Another original data set used by "student" (Gossett) for the t-test.
5. The UC Berkeley example of "sex discrimination" as a Simpson paradox

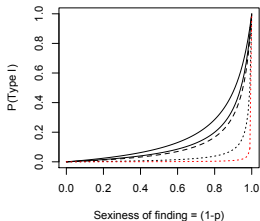


Basic Graphics

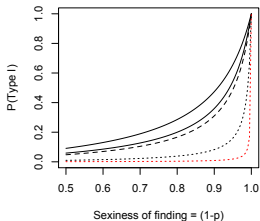
R can show current statistical concepts:

Type I Errors: It is not the power, it is the prior likelihood
dashed/dotted lines reflect alpha = .05, .01, .001 with power = 1

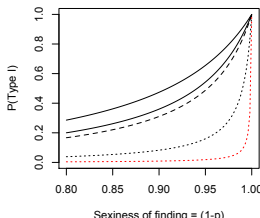
P(Type I) given alpha, power, sexiness



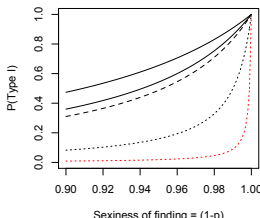
P(Type I) given alpha, power, sexiness



P(Type I) given alpha, power, sexiness



P(Type I) given alpha, power, sexiness

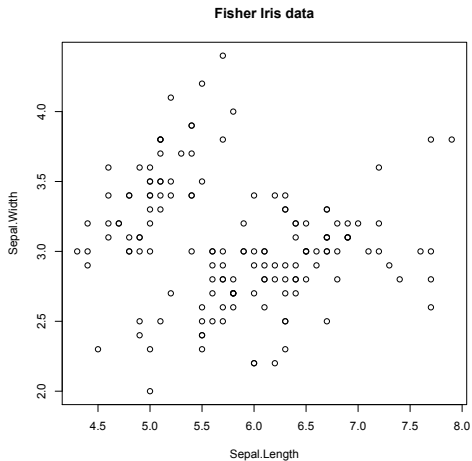


1. Extreme claims require extreme probabilities
2. Given that a finding is “significant”, what is the likelihood that it is a Type I error?
3. Depends upon the prior likelihood (the 'sexiness') of the claim.



Basic Graphics

A simple scatter plot using `plot` with Fisher's Iris data set.

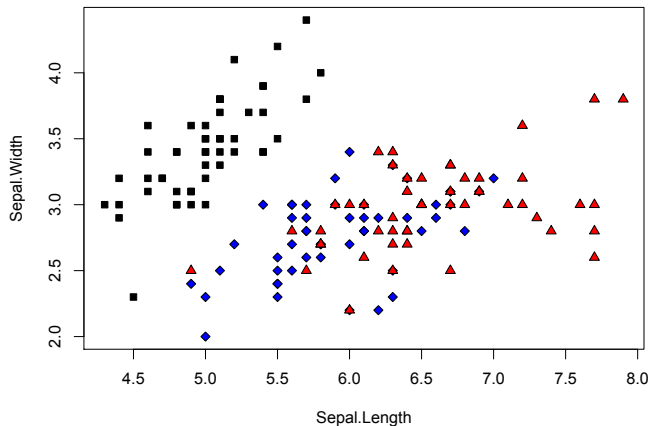


```
plot(iris[1:2], xlab="Sepal.Length", ylab="Sepal.Width",  
     ,main="Fisher Iris data")
```



A simple scatter plot using plot with some colors and shapes

Fisher Iris data with colors and shapes

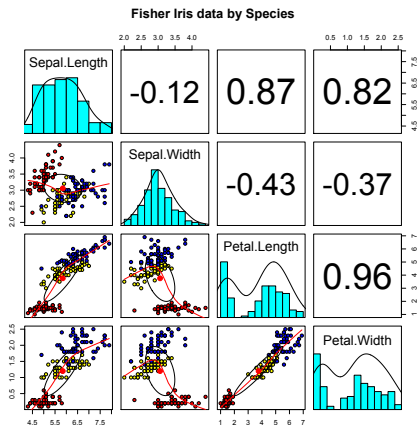


1. Set parameters
2. bg for background colors
3. pch chooses the plot character

```
plot(images/iris[1:2],xlab="Sepal.Length", ylab="Sepal.Width" ,main="Fisher Iris data with
colors and shapes", bg=c("black","blue", "red")[iris[,5]],pch=21+ as.numeric(iris[,5]))
```



Basic Graphics

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.

```

pairs.panels(iris[1:4],bg=c("red","yellow","blue")
[iris$Species],pch=21,main="Fisher Iris data by
Species")

```



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



○○○
○○○○○○○

●○○○○○
○○○○○

Data preparation, descriptive statistics, data cleaning, correlation plots

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 code

```
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 either locally or remotely
my.data <- read.table(file.name,header=TRUE)
#or if you have copied the data to the clipboard
# my.data <- read.clipboard() #you can read it from there
describe(my.data) #report basic descriptive statistics
```

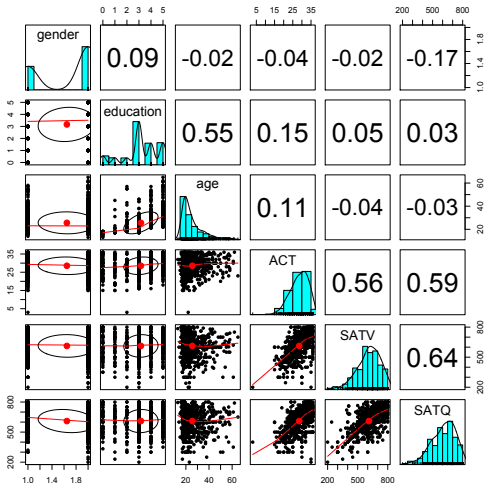
	var	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
gender	1	700	1.65	0.48	2	1.68	0.00	1	2	1	-0.61	-1.62	0.02
education	2	700	3.16	1.43	3	3.31	1.48	0	5	5	-0.68	-0.06	0.05
age	3	700	25.59	9.50	22	23.86	5.93	13	65	52	1.64	2.47	0.36
ACT	4	700	28.55	4.82	29	28.84	4.45	3	36	33	-0.66	0.56	0.18
SATV	5	700	612.23	112.90	620	619.45	118.61	200	800	600	-0.64	0.35	4.27
SATQ	6	687	610.22	115.64	620	617.25	118.61	200	800	600	-0.59	0.00	4.41



Data preparation, descriptive statistics, data cleaning, correlation plots

Graphic display of data using `pairs.panels`

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



Clean up the data using scrub. Use ?scrub for help on the parameters.

We noticed an outlier in the ACT data in the previous graph (you always graph your data, don't you).

We also noticed that the minimum value for ACT was unlikely (of course, you always describe your data).

So we change any case below 4 on the ACT to be missing (NA).

R code

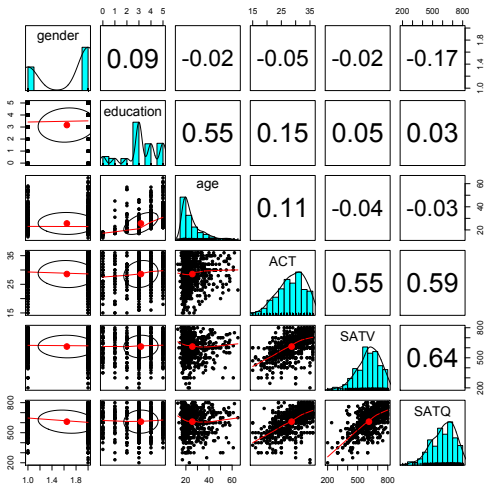
```
cleaned <- scrub(my.data, "ACT", min=4) #what data set,
#which variable, what value to fix
describe(cleaned) #look at the data again
pairs.panels(cleaned)
```

	var	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
gender	1	700	1.65	0.48	2	1.68	0.00	1	2	1	-0.61	-1.62	0.02
education	2	700	3.16	1.43	3	3.31	1.48	0	5	5	-0.68	-0.06	0.05
age	3	700	25.59	9.50	22	23.86	5.93	13	65	52	1.64	2.47	0.36
ACT	4	699	28.58	4.73	29	28.85	4.45	15	36	21	-0.50	-0.36	0.18
SATV	5	700	612.23	112.90	620	619.45	118.61	200	800	600	-0.64	0.35	4.27
SATQ	6	687	610.22	115.64	620	617.25	118.61	200	800	600	-0.59	0.00	4.41



Data preparation, descriptive statistics, data cleaning, correlation plots

Graphic display of cleaned data using pairs.panels



Find the pairwise correlations, round to 2 decimals

This also shows how two functions can be nested. We are rounding the output of the cor function.

R code

```
#specify all the parameters being passed
round(cor(x=sat.act,use="pairwise"),digits=2)
#the short way to specify the rounding parameter
round(cor(cleaned,use="pairwise"),2)
```

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



Display it differently using the lowerCor function

Operations that are done a lot may be made into your own functions. Thus, `lowerCor` finds the pairwise correlations, rounds to 2 decimals, displays the lower half of the correlation matrix, and then abbreviates the column labels to make them line up nicely

R code

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

```

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

```



Testing the significance of one correlation using `cor.test`.

R code

```
cor.test(my.data$ACT, my.data$SATQ)
```

Pearson's product-moment correlation

```
data: my.data$ACT and my.data$SATQ
t = 18.9822, df = 685, p-value < 2.2e-16
alternative hypothesis: true correlation
is not equal to 0
95 percent confidence interval:
 0.5358435 0.6340672
sample estimates:
      cor
0.5871122
```

1. Specify the variables to correlate
2. Various statistics associated with the correlation.
3. But what if you want to do many tests?
Use `corr.test`



Test many correlations for significance using `corr.test`

R code

```
corr.test(cleaned)
```

```
all:corr.test(x = cleaned)
```

```
Correlation matrix
```

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

```
Sample Size
```

	gender	education	age	ACT	SATV	SATQ
gender	700	700	700	699	700	687
...						
SATQ	687	687	687	686	687	687

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

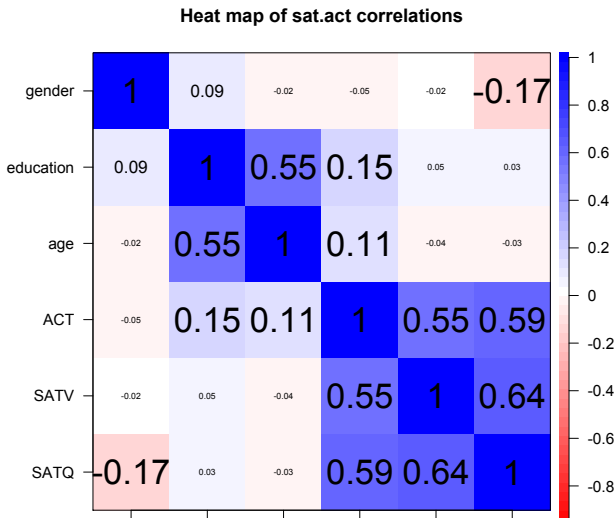
	gender	education	age	ACT	SATV	SATQ
gender	0.00	0.17	1.00	1.00	1	0
education	0.02	0.00	0.00	0.00	1	1
age	0.58	0.00	0.00	0.03	1	1
ACT	0.21	0.00	0.00	0.00	0	0



Inferential statistics

The SAT.ACT correlations. Confidence values from resampling

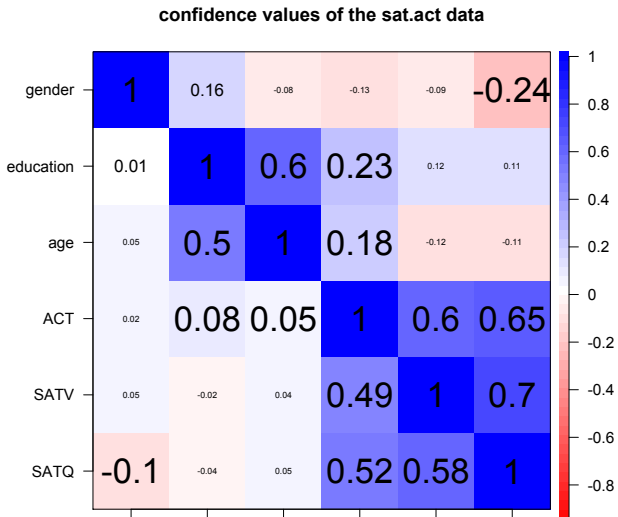
```
ci <- cor.ci(cleaned,main='Heat map of sat.act')
```



Inferential statistics

The SAT.ACT bootstrapped confidence intervals of correlation

```
cor.plot.upperLowerCi(ci,main="Heat map of sat.act")
```



Are education and gender independent? χ^2 Test of association

```
T <- with(my.data, table(gender, education))
```

```
> T
```

	education					
gender	0	1	2	3	4	5
1	27	20	23	80	51	46
2	30	25	21	195	87	95

```
> 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 χ^2 test



Finding χ^2 from a table of data

- Consider the effect of a treatment on later arrest (From Ashley Kendall, 2016)

Condition	Arrested	Not Arrested
Control	14	21
Treatment	3	23

R code

```
ak.df <- data.frame(Control=c(14,21), Treated =c(3,23))
rownames(ak.df) <- c("Arrested", "Not Arrested")
ak.df #show the data frame
chisq.test(ak.df) #Test it using the Yates continuity correction
```

```
> ak.df #show the data frame
      Control Treated
Arrested    14      3
Not Arrested 21     23
> chisq.test(ak.df) #Test it using the Yates continuity correction
      Pearson's Chi-squared test with Yates' continuity correction
data:  ak.df
X-squared = 4.6791, df = 1, p-value = 0.03053
```

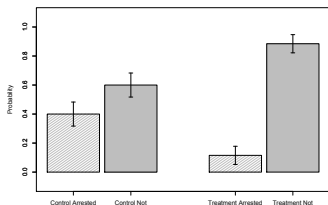


Graph the tabled data showing confidence intervals of proportions

R code

```
ak.df <- data.frame(Control=c(14,21),Treated =c(3,23))
ak.p <- t(t(ak.df)/colSums(ak.df)) #convert to probabilities
standard.error <- sqrt(ak.p[1,] * ak.p[2,]/colSums(ak.df))
stats <- data.frame(mean=as.vector(ak.p),
                    se=rep(standard.error, each=2))
rownames(stats) <- c("Control Arrested", "Control Not",
                    "Treatment Arrested", "Treatment Not")
error.bars(stats=stats, bars=TRUE, space=c(.1, .1, 1, .1),
            density=c(20, -10, 20, -10), ylab="Probability",
            xlab="Control vs Treatment",
            main="Effect of Treatment on subsequent arrest (95% confidence)")
```

Effect of Treatment on subsequent arrest (95% confidence)



```
round(stats, 2)
```

	mean	se
Control Arrested	0.40	0.08
Control Not	0.60	0.08
Treatment Arrested	0.12	0.06
Treatment Not	0.88	0.06



Multiple regression and the general linear model

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

R code

```
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 code

```
csat <- data.frame(scale(my.data, scale=FALSE))
describe(csat) #centered not standardized data
```

	vars	n	mean	sd	median	trimmed	mad	min	max	ra
gender	1	700	0	0.48	0.35	0.04	0.00	-0.65	0.35	
education	2	700	0	1.43	-0.16	0.14	1.48	-3.16	1.84	
age	3	700	0	9.50	-3.59	-1.73	5.93	-12.59	39.41	
ACT	4	700	0	4.82	0.45	0.30	4.45	-25.55	7.45	
SATV	5	700	0	112.90	7.77	7.22	118.61	-412.23	187.77	
SATQ	6	687	0	115.64	9.78	7.04	118.61	-410.22	189.78	

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



Zero center the data before examining interactions

R code

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

Call:

all:

```
lm(formula = SATV ~ education * gender * SATQ, data = csat)
```

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 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 84.69 on 679 degrees of freedom
(13 observations deleted due to missingness)
```

```
Multiple R-squared:  0.4469,    Adjusted R-squared:  0.4412
```

```
F-statistic: 78.37 on 7 and 679 DF,  p-value: < 2.2e-16
```



Compare model 1 and model 2 using anova

Test the difference between the two linear models

R code

```
anova(mod1, mod2)
```

Analysis of Variance Table

Analysis of Variance Table

Model 1: SATV ~ education + gender + SATQ

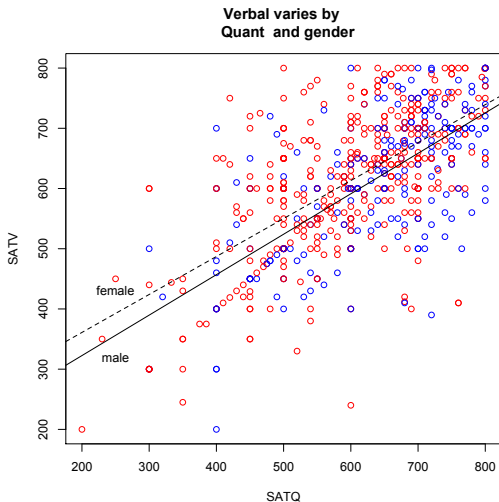
Model 2: SATV ~ education * gender * SATQ

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	683	5079984				
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



First plot all the data.

Then add the regression lines.

Then put a title on the whole thing.

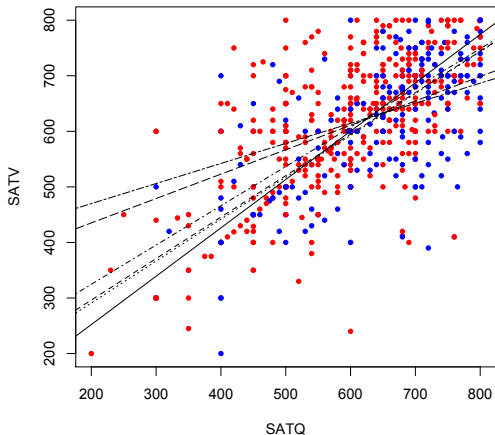
R code

```
#first plot the data points
with(my.data,plot(SATV~SATQ,
  col=c("blue","red")[gender]))
#add the regression lines
by(my.data,my.data$gender,
  function(x) abline
    (lm(SATV~SATQ,data=x),
     lty=c("solid","dashed"
           )[x$gender]))
#add a title
title("Verbal varies by
  Quant and gender")
#label the lines
text(250,320,"male")
text(250,430,"female")
```



Show the regression lines by education

Verbal varies by Quant
and education



Do this again, but for levels of education as the moderator.

R code

```
with(my.data, plot(SATV~SATQ,
  col=c("blue", "red")[gender],
  pch=20)) #plot character
by(my.data, my.data$education,
  function(x) abline
  (lm(SATV~SATQ, data=x),
  lty=c("solid", "dashed", "dotted",
  "dotdash", "longdash",
  "twodash")[x$education+1]))

title("Verbal varies by Quant
and education")
```

Questions?



Outline

Part I: What is R, where did it come from, why use it

- Installing R and adding packages

Part II: A brief introduction – an overview

- R is just a fancy (very fancy) calculator
- Descriptive data analysis
- Some inferential analysis

Part III R is a powerful statistical system

- Data entry
- Descriptive
- Inferential (t and F)
- Regression
- Basic R commands

Part IV: Psychometrics

- Reliability and its discontents
- Exploratory Factor Analysis, Confirmatory Factor Analysis, SEM

Part V: Help and More Help

- List of useful commands



Outline of Part III

-> Part II: Introduction and Overview

Basic statistics and graphics

4 steps: read, explore, test, graph

Basic descriptive statistics and graphics

Graphic displays

Correlations

Inferential statistics

The t-test

ANOVA

Linear Regression

Regression from the raw data

Regression from covariance/correlation matrices

R structure

Basic R

Objects and Functions

-> Part IV: Psychometrics



4 steps: read, explore, test, graph

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



4 steps: read, explore, test, graph

Data entry overview

- 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 > 50 example data sets
- Copying from another program
 - use copy and paste into R using `read.clipboard` and its variations
- Reading a text or csv file
 - read a local or remote file
- Importing from SPSS or SAS
 - Use either the *foreign*, *haven* or *rio* packages
- Simulate it (using various simulation routines)
- Model it using simulations (e.g., `cta` (Revelle & Condon, 2015))



4 steps: read, explore, test, graph

Examples of built in data sets from the psych package

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

ability	16 multiple choice IQ items from the ICAR project (Condon & Revelle, 2014)
Bechtoldt	Seven data sets showing a bifactor solution (Bechtoldt, 1961; Holzinger & Swineford, 1937; Thurstone & Thurstone, 1941).
Dwyer	8 cognitive variables used by Dwyer (1937) for an example.
Reise	Seven data sets showing a bifactor solution (Reise, Morizot & Hays, 2007).
affect	Data sets of affect and arousal scores as a function of personality and movie conditions (Smillie, Cooper, Wilt & Revelle, 2012)
income	US family income from US census 2008
bfi	25 Personality items representing 5 factors (N=2800)
blot	Bond's Logical Operations Test - BLOT (N=150) (Bond, 1995)
burt	11 emotional variables from Burt (1915)
cities	Distances between 11 US cities
epi.bfi	13 scales from the Eysenck Personality Inventory and Big 5 inventory
income	US family income from US census 2008
msq	75 mood items from the Motivational State Questionnaire for N=3896
neo	NEO correlation matrix from the NEOPI-R manual (Costa & McCrae, 1985)
sat.act	3 Measures of ability: SATV, SATQ, ACT (N=700)
Thurstone	Seven data sets showing a bifactor solution.
veg (vegetables)	Paired comparison of preferences for 9 vegetables (Guilford, 1954)



4 steps: read, explore, test, graph

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

```
my.data <- read.clipboard() #assumes headers and tab or space delimit
my.data <- read.clipboard.csv() #assumes headers and comma delimit
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
                                (see read.fwf for instruct.)
```

4. `read.clipboard()` has default values for the most common cases and these do not need to be specified. Consult `?read.clipboard` for details. In particular, are headers provided for each column of input?



4 steps: read, explore, test, graph

Reading from a local or remote file

- 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. This goes to your normal system file handler.
 - The file name/location can be a remote URL. (Note that `read.file` might not work on https files.)
- Two examples of reading data

R code

```
file.name <- file.choose() #this opens a window to allow you find the file
#or
file.name="http://personality-project.org/r/datasets/R.appendix1.data"
my.data <- read.table(file.name,header=TRUE) #unless it is https (see above)
#or
my.data =read.https(file.name,header=TRUE) #read an https file
dim(my.data) #find the dimensionality of our data
describe(my.data) #describe it to check the means, ranges, etc.
```

```
> dim(my.data ) #what are the dimensions of what we read?
```

```
[1] 18 2
```

```
> describe(my.data ) #do the data look right?
```

	var	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
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



4 steps: read, explore, test, graph

Put it all together: read, show, describe

R code

```
datafilename="http://personality-project.org/r/datasets/R.appendix1.data"
data.ex1<- read.table(datafilename,header=TRUE) #unless it is https (see above)
dim(data.ex1) #what are the dimensions of what we read?
data.ex1 #show the data
headTail(data.ex1) #just the top and bottom lines
describe(data.ex1) #descriptive stats
```

```
      Dosage Alertness
1         a         30
2         a         38
... (rows deleted by hand)
17        c         20
18        c         19

> headTail(data.ex1) #just the top and bottom lines
      Dosage Alertness
1         a         30
2         a         38 'head' rows
3         a         35
4         a         41
... <NA> ... (rows automatically deleted)
15        c         17
16        c         21
17        c         20 'tail' rows
18        c         19

> describe(data.ex1) #descriptive stats
      vars n mean sd median trimmed mad min max range skew kurtosis se
Dosage*  1 18  1.89 0.76      2   1.88 1.48  1  3    2 0.16   -1.35 0.18
Alertness 2 18 27.67 6.82     27  27.50 8.15 17 41   24 0.25   -1.06 1.61
```

1. Read the data from a remote file
2. Show all the cases (problematic if there are 100s – 1000s)
3. Just show the first and last (4) lines
4. Find descriptive statistics



4 steps: read, explore, test, graph

However, some might want to Import SAS or SPSS files

There are several different packages that make importing SPSS, SAS, Systat, etc. files easy to do.

- foreign** Read data stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase. Comes installed with R. Somewhat complicated syntax.
- haven** Reads/writes SPSS and Stata files. Handles SPSS labels nicely (keeps the item labels, but converts the data to factors).
- rio** A general purpose package that requires installation of many of the other packages used for data import. Easiest to use, but overkill if just reading in one type of file. Basically a front end to many import/export packages. It determines which package to use based upon the file name suffix (e.g., csv, txt, sav, ...)



4 steps: read, explore, test, graph

Read a “foreign” file e.g., an SPSS sav file, using foreign package

`read.spss` Reads a file stored by the SPSS save or export commands. (The defaults lead to problems, make sure to specify that you want `use.value.labels = FALSE`, `to.data.frame = TRUE`)

```
read.spss(file, use.value.labels = FALSE, to.data.frame = TRUE,
          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? Should be `FALSE`

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`?



4 steps: read, explore, test, graph

An example of reading from an SPSS file using foreign

```
> library(foreign)

> datafilename <- "http://personality-project.org/r/datasets/finkel.sav"

> eli <- read.spss(datafilename,to.data.frame=TRUE,
                  use.value.labels=FALSE)

> headTail(eli,2,2)

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

	USER	HAPPY	SOULMATE	ENJOYDEX	UPSET
1	"001"	4	7	7	1
2	"003"	6	5	7	0
...	<NA>
68	"076"	7	7	7	0
69	"078"	2	7	7	1

	var	n	mean	sd	median	trimmed	mad	min	max	range	skew
USER*	1	69	35.00	20.06	35	35.00	25.20	1	69	68	2.42
HAPPY	2	69	5.71	1.04	6	5.82	0.00	2	7	5	0.13
SOULMATE	3	69	5.09	1.80	5	5.32	1.48	1	7	6	0.22
ENJOYDEX	4	68	6.47	1.01	7	6.70	0.00	2	7	5	0.12
UPSET	5	69	0.41	0.49	0	0.39	0.00	0	1	1	0.06

1. Make the *foreign* package active
2. Specify the name (and location) of the file to read
3. Read from a SPSS file
4. Show the top and bottom 2 cases
5. Describe it to make sure it is right



4 steps: read, explore, test, graph

An example of reading from an SPSS file using rio

```
> library(rio)

> datafilename <- "http://personality-project.org/r/datasets/finkel.sav"

> eli <- import(datafilename) #note that it figures out what to do
> headTail(eli,2,2) #The first and last 2
> describe(eli,skew=FALSE)
```

```
      USER HAPPY SOULMATE ENJOYDEX UPSET
1  "001"      4         7         7      1
2  "003"      6         5         7      0
... <NA>    ...         ...         ...
68 "076"      7         7         7      0
69 "078"      2         7         7      1
>
```

```
      var  n  mean   sd median trimmed  mad min max range  se
USER*   1 69 35.00 20.06    35  35.00 25.20  1 69   68 2.4
HAPPY   2 69  5.71  1.04     6  5.82  0.00  2  7    5 0.13
SOULMATE 3 69  5.09  1.80     5  5.32  1.48  1  7    6 0.22
ENJOYDEX 4 68  6.47  1.01     7  6.70  0.00  2  7    5 0.12
UPSET   5 69  0.41  0.49     0  0.39  0.00  0  1    1 0.06
```

1. Make the *rio* package active
2. Specify the name (and location) of the file to read
3. Import from a SPSS file
4. Show the top and bottom 2 cases
5. Describe it to make sure it is right



4 steps: read, explore, test, graph

An example of reading from an SPSS file using haven

```
> library(haven)

> datafilename <- "http://personality-project.org/r/datasets/finkel.sav"

> eli <- read_spss(datafilename) #note that it figures out what to do
> headTail(eli,3,2) The first 3 and last 2
> describe(eli,skew=FALSE)
```

	USER	HAPPY	SOULMATE	ENJOYDEX	UPSET
1	"001"	4	7	7	1
2	"003"	6	5	7	0
3	"004"	6	7	7	0
...	<NA>
68	"076"	7	7	7	0
69	"078"	2	7	7	1>

	var	n	mean	sd	median	trimmed	mad	min	max	range	se
USER*	1	69	35.00	20.06	35	35.00	25.20	1	69	68	2.42
HAPPY	2	69	5.71	1.04	6	5.82	0.00	2	7	5	0.15
SOULMATE	3	69	5.09	1.80	5	5.32	1.48	1	7	6	0.22
ENJOYDEX	4	68	6.47	1.01	7	6.70	0.00	2	7	5	0.12
UPSET	5	69	0.41	0.49	0	0.39	0.00	0	1	1	0.06

1. Make the *haven* package active
2. Specify the name (and location) of the file to read
3. Import from a SPSS file
4. Show the top 3 and bottom 2 cases
5. Describe it to make sure it is right



4 steps: read, explore, test, graph

Simulate data (Remember to always call them simulated!)

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 code

```
headTail(epi.bfi)
```

	epiE	epiS	epiImp	epilie	epiNeur	bfragee	bfcon	bfext	bfneur	bfopen
1	18	10	7	3	9	138	96	141	51	138
2	16	8	5	1	12	101	99	107	116	132
3	6	1	3	2	5	143	118	38	68	90
4	12	6	4	3	15	104	106	64	114	101
...
228	12	7	4	3	15	155	129	127	88	110
229	19	10	7	2	11	162	152	163	104	164
230	4	1	1	2	10	95	111	75	123	138
231	8	6	3	2	15	85	62	90	131	96

`epi.bfi` has 231 cases from two personality measures.



Now find the descriptive statistics for this data set

R code

```
describe(epi.bfi)
```

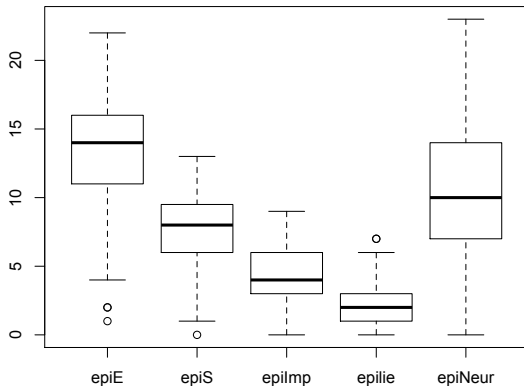
	var	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
epiE	1	231	13.33	4.14	14	13.49	4.45	1	22	21	-0.33	-0.01	0.27
epiS	2	231	7.58	2.69	8	7.77	2.97	0	13	13	-0.57	0.04	0.18
epiImp	3	231	4.37	1.88	4	4.36	1.48	0	9	9	0.06	-0.59	0.12
epilie	4	231	2.38	1.50	2	2.27	1.48	0	7	7	0.66	0.30	0.10
epiNeur	5	231	10.41	4.90	10	10.39	4.45	0	23	23	0.06	-0.46	0.32
bfragee	6	231	125.00	18.14	126	125.26	17.79	74	167	93	-0.21	-0.22	1.19
bfcon	7	231	113.25	21.88	114	113.42	22.24	53	178	125	-0.02	0.29	1.44
bfext	8	231	102.18	26.45	104	102.99	22.24	8	168	160	-0.41	0.58	1.74
bfneur	9	231	87.97	23.34	90	87.70	23.72	34	152	118	0.07	-0.51	1.54
bfopen	10	231	123.43	20.51	125	123.78	20.76	73	173	100	-0.16	-0.11	1.35
bdi	11	231	6.78	5.78	6	5.97	4.45	0	27	27	1.29	1.60	0.38
traitanx	12	231	39.01	9.52	38	38.36	8.90	22	71	49	0.67	0.54	0.63
stateanx	13	231	39.85	11.48	38	38.92	10.38	21	79	58	0.72	0.04	0.76



Boxplots are a convenient descriptive device

Show the Tukey “boxplot” for the Eysenck Personality Inventory

Boxplots of EPI scales

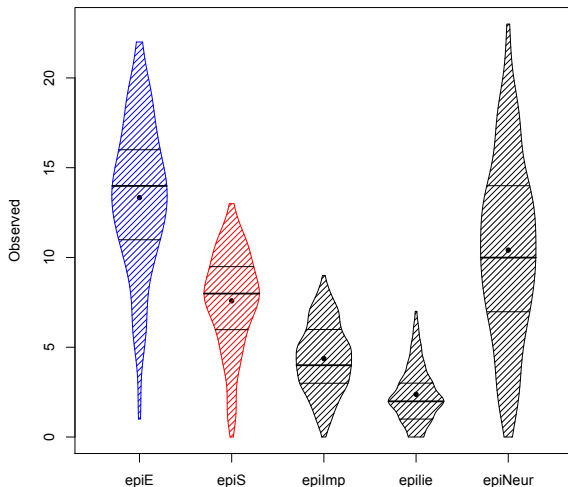


Use the box plot function and select the first five variables.

```
my.data <- epi.bfi
boxplot(my.data[1:5])
```

An alternative display is a 'violin' plot (available as `violinBy`)

Density plot

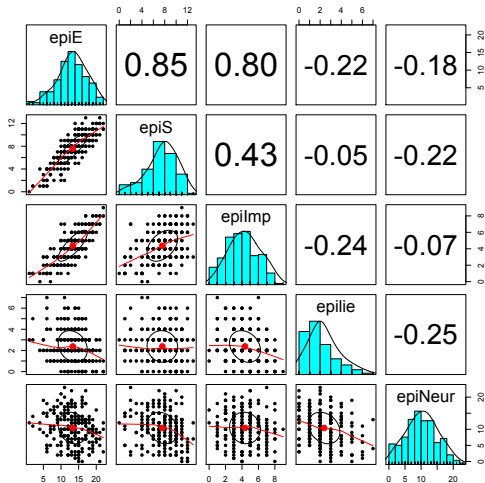


Use the `violinBy` function from *psych*

```
violinBy(my.data[1:5])
```


Graphic displays

Plot the scatter plot matrix (SPLOM) of the first 5 variables using the `pairs.panels` function. Note that the plotting points overlap because of the polytomous nature of the data.



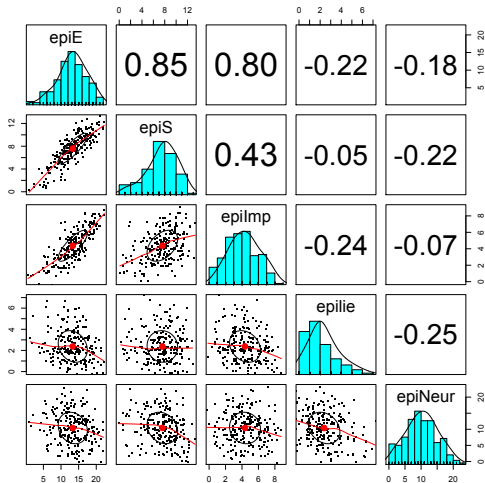
Use the `pairs.panels` function from *psych*

```
pairs.panels(my.data[1:5])
```



Graphic displays

Plot the scatter plot matrix (SPLOM) of the first 5 variables using the `pairs.panels` function but with smaller plot character (`pch`) and jittering the points in order to better show the distributions.



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

```
pairs.panels(my.data[1:5], pch='.',
             jiggle=TRUE)
```



Correlations

Find the correlations for this data set, round off to 2 decimal places.

Because we have some missing data, we use “pairwise complete” correlations. For the purists amongst us, it is irritating that the columns are not equally spaced.

R code

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

	epiE	epiS	epiImp	epilie	epiNeur	bfagree	bfcon	bfext	bfneur	bfopen	bdi	traitanx	stateanx
epiE	1.00	0.85	0.80	-0.22	-0.18	0.18	-0.11	0.54	-0.09	0.14	-0.16	-0.23	-0.13
epiS	0.85	1.00	0.43	-0.05	-0.22	0.20	0.05	0.58	-0.07	0.15	-0.13	-0.26	-0.12
epiImp	0.80	0.43	1.00	-0.24	-0.07	0.08	-0.24	0.35	-0.09	0.07	-0.11	-0.12	-0.09
epilie	-0.22	-0.05	-0.24	1.00	-0.25	0.17	0.23	-0.04	-0.22	-0.03	-0.20	-0.23	-0.15
epiNeur	-0.18	-0.22	-0.07	-0.25	1.00	-0.08	-0.13	-0.17	0.63	0.09	0.58	0.73	0.49
bfagree	0.18	0.20	0.08	0.17	-0.08	1.00	0.45	0.48	-0.04	0.39	-0.14	-0.31	-0.19
bfcon	-0.11	0.05	-0.24	0.23	-0.13	0.45	1.00	0.27	0.04	0.31	-0.18	-0.29	-0.14
bfext	0.54	0.58	0.35	-0.04	-0.17	0.48	0.27	1.00	0.04	0.46	-0.14	-0.39	-0.15
bfneur	-0.09	-0.07	-0.09	-0.22	0.63	-0.04	0.04	0.04	1.00	0.29	0.47	0.59	0.49
bfopen	0.14	0.15	0.07	-0.03	0.09	0.39	0.31	0.46	0.29	1.00	-0.08	-0.11	0.61
bdi	-0.16	-0.13	-0.11	-0.20	0.58	-0.14	-0.18	-0.14	0.47	-0.08	1.00	0.65	0.65
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	0.57
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



Correlations

Find the correlations for this data set, round off to 2 decimal places using lowerCor

This is just a wrapper for `round(cor(x,use='pairwise'),2)` that has been prettied up with `lowerMat`.

R code

```
lowerCor(my.data)
```

```

      epiE  epiS  epImp  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 code

```
corr.test(my.data)
```

```
Call:corr.test(x = my.data)
```

```
Correlation matrix
```

	epiE	epiS	epiImp	epilie	epiNeur	bfragee	bfcon	bfext	bfneur	bfopen	bdi	traitanx	stateanx
epiE	1.00	0.85	0.80	-0.22	-0.18	0.18	-0.11	0.54	-0.09	0.14	-0.16	-0.23	
epiS	0.85	1.00	0.43	-0.05	-0.22	0.20	0.05	0.58	-0.07	0.15	-0.13	-0.26	
epiImp	0.80	0.43	1.00	-0.24	-0.07	0.08	-0.24	0.35	-0.09	0.07	-0.11	-0.12	
..													
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	

```
Sample Size
```

	epiE	epiS	epiImp	epilie	epiNeur	bfragee	bfcon	bfext	bfneur	bfopen	bdi	traitanx	stateanx
epiE	231	231	231	231	231	231	231	231	231	231	231	231	
..													
stateanx	231	231	231	231	231	231	231	231	231	231	231	231	

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

	epiE	epiS	epiImp	epilie	epiNeur	bfragee	bfcon	bfext	bfneur	bfopen	bdi	traitanx	stateanx
epiE	0.00	0.00	0.00	0.03	0.27	0.27	1.00	0.00	1.00	1.00	0.59	0.02	
epiS	0.00	0.00	0.00	1.00	0.04	1.00	1.00	0.00	1.00	0.62	1.00	0.00	
epiImp	0.00	0.00	0.00	0.01	1.00	0.08	0.01	0.00	1.00	1.00	1.00	1.00	
epilie	0.00	0.43	0.00	0.00	0.01	0.32	0.03	1.00	0.03	1.00	0.08	0.02	
epiNeur	0.01	0.00	0.26	0.00	0.00	1.00	1.00	0.33	0.00	1.00	0.00	0.00	
bfragee	0.01	0.00	0.23	0.01	0.21	0.00	0.00	0.00	1.00	0.00	0.95	0.00	
bfcon	0.08	0.48	0.00	0.00	0.04	0.00	0.00	0.00	1.00	0.00	0.25	0.00	
bfext	0.00	0.00	0.00	0.50	0.01	0.00	0.00	0.00	1.00	0.00	0.99	0.00	
bfneur	0.15	0.30	0.18	0.00	0.00	0.50	0.50	0.57	0.00	0.00	0.00	0.00	
bfopen	0.04	0.02	0.30	0.70	0.19	0.00	0.00	0.00	0.00	0.00	1.00	1.00	
bdi	0.02	0.04	0.11	0.00	0.00	0.03	0.01	0.03	0.00	0.25	0.00	0.00	
traitanx	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.11	0.00	0.00	
stateanx	0.05	0.07	0.18	0.02	0.00	0.00	0.04	0.02	0.00	0.52	0.00	0.00	

>

The t-test

t.test demonstration with Student's data (from the sleep and cushny datasets)

```

> 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
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
> 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
95 percent confidence interval:
-2.4598858 -0.7001142
sample estimates:

```

extra	group	ID
1	0.7	1 1
2	-1.6	1 2
3	-0.2	1 3
4	-1.2	1 4
5	-0.1	1 5
6	3.4	1 6
7	3.7	1 7
...		
13	1.1	2 3
14	0.1	2 4
15	-0.1	2 5
16	4.4	2 6
17	5.5	2 7
18	1.6	2 8
19	4.6	2 9
20	3.4	2 10



The cushny data set organizes the data differently

R code

```
cushny
with(cushny, t.test(delta1, delta2L))
with(cushny, t.test(delta1, delta2L, paired=TRUE))
```

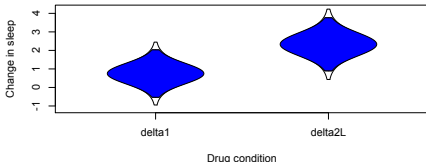
```
> cushny
  Control drug1 drug2L drug2R delta1 delta2L delta2R
1      0.6   1.3   2.5   2.1   0.7   1.9   1.5
2      3.0   1.4   3.8   4.4  -1.6   0.8   1.4
...
9      1.1   1.1   5.7   5.8   0.0   4.6   4.7
10     2.9   4.9   6.3   6.4   2.0   3.4   3.5
> with(cushny, t.test(delta1, delta2L)) #not paired (for demonstration)
...
      t = -1.8608, df = 17.776, p-value = 0.07939
...
> with(cushny, t.test(delta1, delta2L, paired=TRUE)) #paired t-test is appropriate
  Paired t-test
data:  delta1 and delta2L
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
```



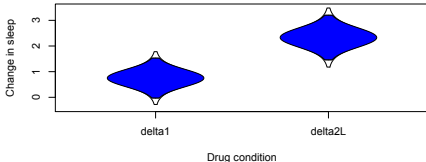
The t-test

Two ways of showing Student's t test data

Student's unpaired sleep change data



Student's paired sleep change data



Use the `error.bars` function and draw "cat's eyes". Make a two panel graph.

R code

```
op <- par(mfrow=c(2,1)) #make two rows
error.bars(cushny[c(5,6)],within=FALSE,
           ylab="Change in sleep",
           xlab="Drug condition",
           main="Student's unpaired sleep change data")
```

R code

```
error.bars(cushny[c(5,6)],within=TRUE,
           ylab="Change in sleep",
           xlab="Drug condition",
           main="Student's paired sleep change data")
```

```
op <- par(mfrow=c(1,1)) #go back the original 1 x 1 p
```

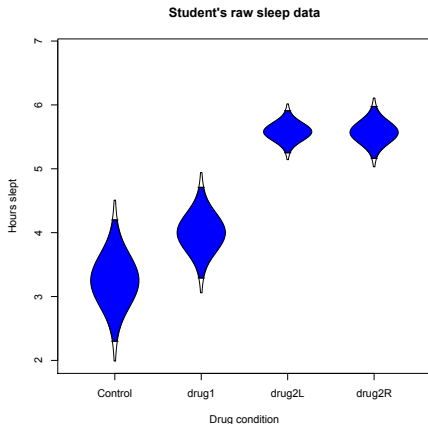


The t-test

But the actual data were repeated within subjects (see cushny)

R code

```
error.bars(cushny[1:4], within=TRUE, ylab="Hours slept",  
           xlab="Drug condition", main="Student's raw 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 code

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

```
data.ex2 #show the data
  Observation Gender Dosage Alertness
1           1      m      a           8
2           2      m      a          12
3           3      m      a          13
4           4      m      a          12
...
14          14      f      b          12
15          15      f      b          18
16          16      f      b          22
```



Analysis of Variance

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

R code

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

```
> aov.ex2 = aov(Alertness~Gender*Dosage, data=data.ex2)
> summary(aov.ex2)           #show the summary table
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Gender	1	76.56	76.56	2.952	0.111
Dosage	1	5.06	5.06	0.195	0.666
Gender: Dosage	1	0.06	0.06	0.002	0.962
Residuals	12	311.25	25.94		



Show the results table

R code

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

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

```
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
```



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> and at <http://personality-project.org/r>
4. See also the tutorial by Jason French at <http://jason-french.com/tutorials/repeatedmeasures.html>



ANOVA

Analysis of variance within subjects: Getting and showing the data

R code

```
datafilename="http://personality-project.org/r/datasets/R.appendix5.d
data.ex5=read.table(datafilename,header=T) #read the data into a table
data.ex5 #show the data
```

```
> data.ex5
  Obs Subject Gender Dosage Task Valence Recall
1    1      A      M      A    F      Neg      8
2    2      A      M      A    F      Neu      9
3    3      A      M      A    F      Pos      5
4    4      A      M      A    C      Neg      7
5    5      A      M      A    C      Neu      9
6    6      A      M      A    C      Pos     10
7    7      B      M      A    F      Neg     12
8    8      B      M      A    F      Neu     13
9    9      B      M      A    F      Pos     14
10  10      B      M      A    C      Neg     16
...
28  28      E      M      B    C      Neg      4
29  29      E      M      B    C      Neu      9
...
107 107      R      F      C    C      Neu     21
108 108      R      F      C    C      Pos     20
```



ANOVA

Analysis of variance within subjects

R code

```
filename="http://personality-project.org/r/datasets/R.appendix5.data"
data.ex5=read.table(filename,header=TRUE) #read the data into a table
#do the anova
aov.ex5 = aov(Recall~(Task*Valence*Gender*Dosage)+Error(Subject/(Task*Valence))+
  (Gender*Dosage),data.ex5)
#look at the output
summary(aov.ex5)
```

Error: Subject

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Gender	1	542.3	542.3	5.685	0.0345 *
Dosage	2	694.9	347.5	3.643	0.0580 .
Gender: Dosage	2	70.8	35.4	0.371	0.6976
Residuals	12	1144.6	95.4		

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

Error: Subject:Task

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Task	1	96.33	96.33	39.862	3.87e-05 ***
Task: Gender	1	1.33	1.33	0.552	0.472
Task: Dosage	2	8.17	4.08	1.690	0.226
Task: Gender: Dosage	2	3.17	1.58	0.655	0.537
Residuals	12	29.00	2.42		

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+ lots more



ANOVA

Analysis of variance within subjects output (continued)

Error: Subject:Valence

	Df	Sum Sq	Mean Sq	F	value	Pr(>F)
Valence	2	14.69	7.343	2.998	0.0688	.
Valence:Gender	2	3.91	1.954	0.798	0.4619	
Valence:Dosage	4	20.26	5.065	2.068	0.1166	
Valence:Gender:Dosage	4	1.04	0.259	0.106	0.9793	
Residuals	24	58.78	2.449			

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

Error: Subject:Task:Valence

	Df	Sum Sq	Mean Sq	F	value	Pr(>F)
Task:Valence	2	5.39	2.6944	1.320	0.286	
Task:Valence:Gender	2	2.17	1.0833	0.531	0.595	
Task:Valence:Dosage	4	2.78	0.6944	0.340	0.848	
Task:Valence:Gender:Dosage	4	2.67	0.6667	0.327	0.857	
Residuals	24	49.00	2.0417			



Multiple regression

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

```
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



Regression from the raw data

Zero center the data before examining interactions

```
> 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	0.81500	
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 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1



Compare model 1 and model 2

Test the difference between the two linear models

```
> anova(mod1,mod2)
```

Analysis of Variance Table

Model 1: SATV ~ education + gender + SATQ

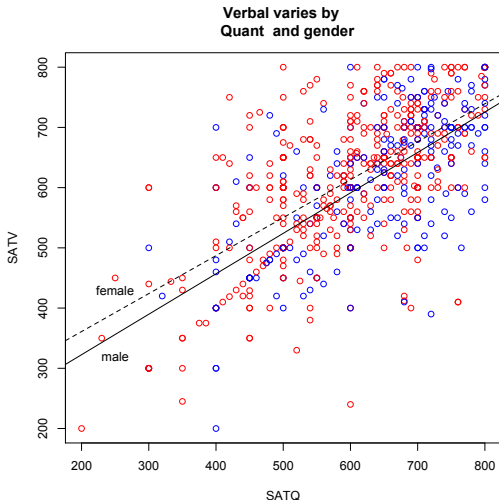
Model 2: SATV ~ education * gender * SATQ

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	683	5079984				
2	679	4870243	4	209742	7.3104	9.115e-06 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.'



Show the regression lines by gender



```
> 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")
```



Regression from covariance/correlation matrices

1. Although most regression examples use the raw data, it is also possible to do this from the correlation/covariance matrices.
2. This is particularly useful for analyzing text book examples or data sets that come from synthetic covariance matrices (SAPA data).
3. Two functions do this
 - 3.1 `setCor` will find (and draw the paths) between a set of X variables and a set of Y variables from either the raw data or from a correlation matrix.
 - 3.2 `mediate` will show path diagrams in a way to highlight “mediated” (indirect) and direct effects. The significance of the indirect effect is found by bootstrapped confidence intervals
4. Both of these functions just use the standard matrix equation

$$\beta_{xy} = \mathbf{R}^{-1} r_{xy}$$
5. The two examples are taken from the PMI example in Hayes (2013) which is saved as a covariance matrix in the `mediate` help file.



setCor finds regressions from covariances

R code

```
lowerMat(C.pmi) #show it
setCor(2:4, c(1, 5, 6), data=C.pmi)
```

```
> lowerMat(C.pmi)
      cond pmi  imprt rectn gendr age
cond   0.25
pmi    0.12  1.75
imprt  0.16  0.65  3.02
reaction 0.12  0.91  1.25  2.40
gender  0.03  0.01 -0.02 -0.01  0.23
age     0.07 -0.04  0.74 -0.75  0.88 33.65
```

Multiple Regression from matrix input

Beta weights

	pmi	imprt	reaction
cond	0.18	0.19	0.16
gender	0.00	-0.08	-0.01
age	-0.01	0.09	-0.09

Multiple R

	pmi	imprt	reaction
	0.18	0.21	0.18

multiple R2

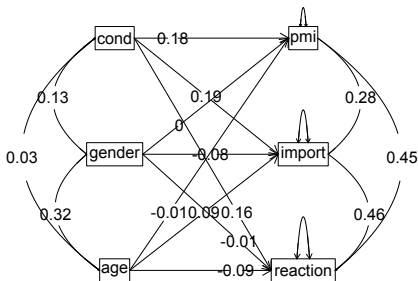
	pmi	imprt	reaction
	0.033	0.043	0.033

(Specify n.obs if you want the standard errors , t, and probabilities of the estimates.)



Regressions from a covariance matrix

Regression Models



unweighted matrix correlation = 0.11



A mediation example from Hayes (2013)

R code

```
#n.iter set to 50 (instead of default of 5000) for speed of example
mediate(y="reaction",x = "cond",m=c("pmi","import"),data=C.pmi,n.obs=123,n.iter=50)
```

```
Call: mediate(y = "reaction", x = "cond", m = c("pmi", "import"), data = C.pmi,
  n.obs = 123, n.iter = 50)
```

```
The DV (Y) was reaction . The IV (X) was cond . The mediating variable(s) = pmi import .
Total Direct effect(c) of cond on reaction = 0.5 S.E. = 0.28 t direct = 1.79 with
Direct effect (c') of cond on reaction removing pmi import = 0.1 S.E. = 0.24 t dir
Indirect effect (ab) of cond on reaction through pmi import = 0.39
Mean bootstrapped indirect effect = 0.4 with standard error = 0.13 Lower CI = 0.19 Up
R2 of model = 0.33
```

To see the longer output, specify short = FALSE in the print statement

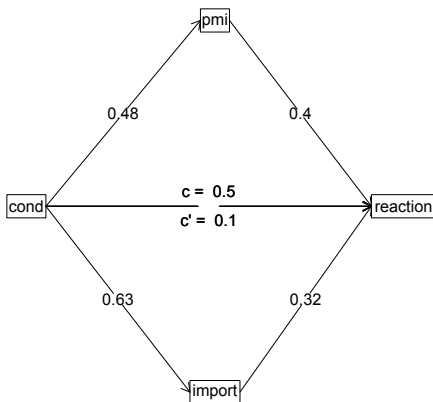
Full output

```
Total effect estimates (c)
  reaction se t Prob
cond      0.5 0.28 1.79 0.0766
Direct effect estimates (c')
  reaction se t Prob
cond      0.10 0.24 0.43 6.66e-01
pmi       0.40 0.09 4.26 4.04e-05
import    0.32 0.07 4.59 1.13e-05
'a' effect estimates
  cond se t Prob
pmi    0.48 0.24 2.02 0.0452
import 0.63 0.31 2.02 0.0452
'b' effect estimates
  reaction se t Prob
pmi      0.40 0.09 4.26 4.04e-05
import   0.32 0.07 4.59 1.13e-05
'ab' effect estimates
  reaction boot sd lower upper
```



A mediation example from Hayes (2013)

Mediation model



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 (`?function`) or `?? function`
4. Vignettes for help on the entire package (available either as part of the help file, or as a web page supplement to the package).



The basic types of data structures

1. Scalars (characters, integers, reals, complex)

```
> A <- 1      #Assign the value 1 to the object A
> B <- 2      #Assign the value 2 to the object B
```

2. Vectors (of scalars, all of one type) have length

```
> C <- month.name[1:5] #Assign the names of the first 5 months to C
> D <- 12:24          #assign the numbers 12 to 24 to D
> length(D)          #how many numbers are in D?
```

```
[1] 13
```

3. Matrices (all of one type) have dimensions

```
> E <- matrix(1:20, ncol = 4)
> dim(E) #number of rows and columns of E
```

```
[1] 5 4
```



Show values by entering the variable name

```

> A      #what is the value of A?

[1] 1

> B      #and of B?

[1] 2

> C      #and C

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

> D

[1] 12 13 14 15 16 17 18 19 20 21 22 23 24

> E

      [,1] [,2] [,3] [,4]
[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.

```
> E.df <- data.frame(names = C, values = c(31, 28, 31, 30, 31))
> dim(E.df)
```

```
[1] 5 2
```

2. Lists are collections of what ever you want. They have length, but do not have dimensions.

```
> 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] [,2] [,3] [,4]
[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
```



Basic R

1. To show the structure of a list, use `str`

```
> 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 ...
```

2. 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.

```
> F[[2]]
```

```
[1] "January" "February" "March" "April"
```

```
> F[["a.matrix"]][, 2]
```

```
[1] 6 7 8 9 10
```

```
> F[["a.matrix"]][2, ]
```

```
[1] 2 7 12 17
```



Addressing the elements of a data.frame or matrix

Setting row and column names using paste

```
> 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 $+$, $-$, $*$, $/$, $\%*\%$, \wedge
 - logical functions $<$, $>$, $==$, $!=$
3. Some functions return "invisible" values
 - e.g., `p <- print(x,digits=3)` will print out `x` to 3 digits but also returns a value to `p`.
 - Similarly, `s <- summary(some object)` will return the value of the `summary` function.
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. Keep a list of the ones you use.



Getting help

- All functions have a help menu
 - `help(the function)`
 - `? the function`
 - Most function help pages have examples to show how to use the function
- 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>
- To find a function in the entire R space, use `findFn` in the *sos* package.
- Online tutorials (e.g., <http://Rpad.org> for a list of important commands, <http://personality-project.org/r>) for a tutorial for psychologists.
- Online and hard copy books



Objects and Functions

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

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



Objects and Functions

More useful statistical functions, Use ? for details

mean (x)
is.na (x) also is.null(x), is...
na.omit (x) ignore missing data
sum (x)
rowSums (x) see also colSums(x)
min (x)
max (x)
range (x)
table (x)
summary (x) depends upon x
sd (x) standard deviation
cor (x) correlation
cov (x) covariance
solve (x) inverse of x
lm (y~x) linear model
aoa (y~x) ANOVA

Selected functions from *psych* package

describe (x) descriptive stats
describeBy (x,y) descriptives by group
pairs.panels (x) SPLOM
error.bars (x) means + error bars
error.bars.by (x) Error bars by groups
fa (x,n) Factor analysis
principal (x,n) Principal components
iclust (x) Item cluster analysis
scoreItems (x) score multiple scales
score.multiple.choice (x) score multiple choice scales
alpha (x) Cronbach's alpha
omega (x) MacDonald's omega
irt.fa (x) Item response theory through factor analysis



Outline

1. Part I: What is R, where did it come from, why use it
 - Installing R and adding packages
2. Part II: A brief introduction – an overview
 - R is just a fancy (very fancy) calculator
 - Descriptive data analysis
 - Some inferential analysis
3. Part III: Using R
 - Data entry
 - Descriptive
 - Inferential (t and F)
 - Regression, partial correlation, mediation
 - Basic R commands
4. Part IV: Psychometrics
 - Reliability and its discontents (α , ω_h , ω_t , λ_6)
 - EFA, CFA, and SEM
5. Part V: Help and More Help
 - List of useful commands



Outline of Part IV: Psychometrics

-> Part III: Basic Statistics

Classical Test Theory measures of reliability

Split Half Reliability and α

Multiple Scales

Multivariate Analysis and Structural Equation Modeling

Exploratory Factor Analysis

Confirmatory Factor Analysis and Structural Equation Modeling

Item Response Theory

Multiple programs

IRT from factor analysis: the `irt.fa` function in psych

-> Part V: More help



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



Classical Test Theory estimates of reliability

1. Alternative estimates of reliability

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

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

guttman Find the 6 Guttman reliability estimates

splitHalf Find the range of split half reliabilities

2. Scoring tests with multiple scales

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

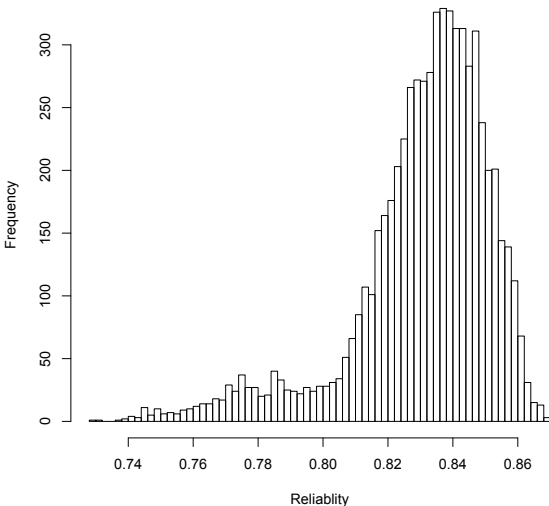
score.multiple.choice Score multiple choice items by first converting to 0 or 1 and then proceeding to score the items.



Split Half Reliability and α

6,435 split half reliabilities of a 16 item ability test

Split half reliabilities of 16 ability measures

**R code**

```
sp <- splitHalf(ability,  
  raw=TRUE, brute=TRUE)  
hist(sp$raw,breaks=50)
```



Finding coefficient α for a scale (see Revelle and Zinbarg, 2009, however, for why you should not)

R code

```
alpha(ability)
```

```
Reliability analysis
```

```
Call: alpha(x = ability)
```

```
raw_alpha std.alpha G6(smc) average_r S/N ase mean sd
0.83      0.83      0.84      0.23 4.9 0.0086 0.51 0.25
```

```
lower alpha upper      95% confidence boundaries
0.81 0.83 0.85
```

```
Reliability if an item is dropped:
```

```
raw_alpha std.alpha G6(smc) average_r S/N alpha se
reason.4   0.82      0.82      0.82      0.23 4.5 0.0093
reason.16  0.82      0.82      0.83      0.24 4.7 0.0091
...
rotate.6   0.82      0.82      0.82      0.23 4.5 0.0092
rotate.8   0.82      0.82      0.83      0.24 4.6 0.0091
```

```
Item statistics
```

```
      n      r r.cor r.drop mean  sd
reason.4 1442 0.58 0.54 0.50 0.68 0.47
reason.16 1463 0.50 0.44 0.41 0.73 0.45
r...
rotate.4  1460 0.58 0.56 0.48 0.22 0.42
rotate.6  1456 0.56 0.53 0.46 0.31 0.46
rotate.8  1460 0.51 0.47 0.41 0.19 0.39
```



Using scoreItems to score 25 Big 5 items (see bfi example)

R code

```
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(bfi,keys.list)
scores <- scoreItems(keys,bfi)
```

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

(Unstandardized) Alpha:

	Agree	Conscientious	Extraversion	Neuroticism	Openness
alpha	0.7	0.72	0.76	0.81	0.6

Average item correlation:

	Agree	Conscientious	Extraversion	Neuroticism	Openness
average.r	0.32	0.34	0.39	0.46	0.23

Guttman 6* reliability:

	Agree	Conscientious	Extraversion	Neuroticism	Openness
Lambda.6	0.7	0.72	0.76	0.81	0.6

Scale intercorrelations corrected for attenuation

raw correlations below the diagonal, alpha on the diagonal

corrected correlations above the diagonal:

	Agree	Conscientious	Extraversion	Neuroticism	Openness
Agree	0.70	0.36	0.63	-0.245	0.23
Conscientious	0.26	0.72	0.35	-0.305	0.30
Extraversion	0.46	0.26	0.76	-0.284	0.32
Neuroticism	-0.18	-0.23	-0.22	0.812	-0.12
Openness	0.15	0.19	0.22	-0.086	0.60



Multiple Scales

score.items output, continued

Item by scale correlations:

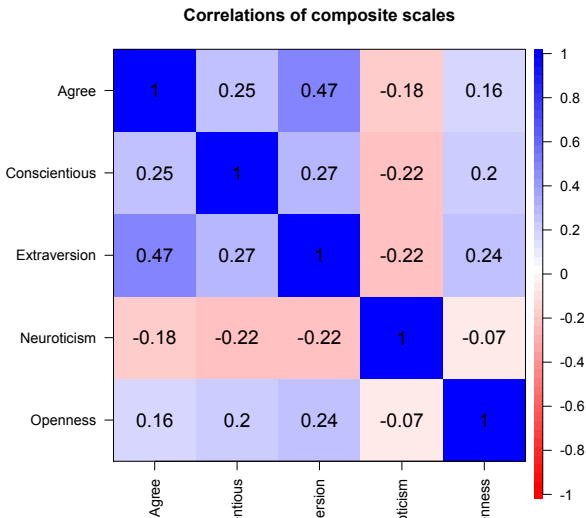
corrected for item overlap and scale reliability

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



Correlations of composite scores based upon item correlations

```
ci <- cor.ci(bfi,keys=keys,main='Correlations of composite scales')
```

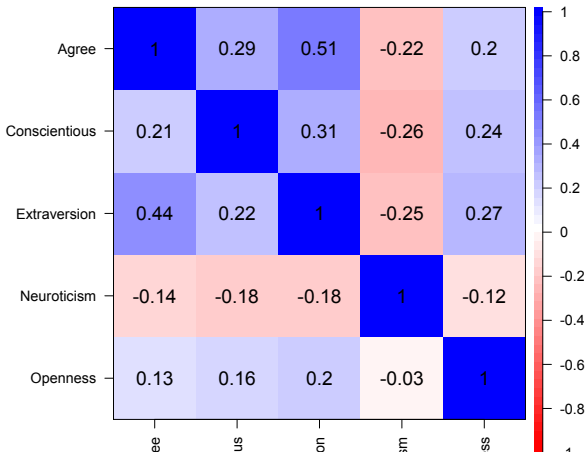


Multiple Scales

Upper and Lower bounds of Correlations of composite scores based upon item correlations and bootstrap resampling

cor.plot.upperLowerCi(ci,main='Upper and lower bounds of Big 5 correlations')

Upper and lower bounds of Big 5 correlations



Factor analysis of Thurstone 9 variable problem

R code

```
f3 <- fa(Thurstone, nfactors=3) #use this built in dataset
f3 #we keep the output as an object to use later
```

Factor Analysis using method = minres

Call: fa(r = Thurstone, nfactors = 3)

Standardized loadings (pattern matrix) based upon correlation matrix

	MR1	MR2	MR3	h2	u2	com
Sentences	0.91	-0.04	0.04	0.82	0.18	1.0
Vocabulary	0.89	0.06	-0.03	0.84	0.16	1.0
Sent.Completion	0.83	0.04	0.00	0.73	0.27	1.0
First.Letters	0.00	0.86	0.00	0.73	0.27	1.0
4.Letter.Words	-0.01	0.74	0.10	0.63	0.37	1.0
Suffixes	0.18	0.63	-0.08	0.50	0.50	1.2
Letter.Series	0.03	-0.01	0.84	0.72	0.28	1.0
Pedigrees	0.37	-0.05	0.47	0.50	0.50	1.9
Letter.Group	-0.06	0.21	0.64	0.53	0.47	1.2

	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
Proportion Explained	0.44	0.31	0.25
Cumulative Proportion	0.44	0.75	1.00

With factor correlations of

	MR1	MR2	MR3
MR1	1.00	0.59	0.54
MR2	0.59	1.00	0.52
MR3	0.54	0.52	1.00



Factor analysis output, continued

With factor correlations of

	MR1	MR2	MR3
MR1	1.00	0.59	0.54
MR2	0.59	1.00	0.52
MR3	0.54	0.52	1.00

Mean item complexity = 1.2

Test of the hypothesis that 3 factors are sufficient.

The degrees of freedom for the null model are 36 and the objective function was 5.2

The degrees of freedom for the model are 12 and the objective function was 0.01

The root mean square of the residuals (RMSR) is 0.01

The df corrected root mean square of the residuals is 0.01

Fit based upon off diagonal values = 1

Measures of factor score adequacy

	MR1	MR2	MR3
Correlation of scores with factors	0.96	0.92	0.90
Multiple R square of scores with factors	0.93	0.85	0.81
Minimum correlation of possible factor scores	0.86	0.71	0.63



Bootstrapped confidence intervals

R code

```
fa (Thurstone, 3, n.obs=213, n.iter=20) #to do bootstrapping
```

...

Coefficients and bootstrapped confidence intervals

	low	MR1	upper	low	MR2	upper	low	MR3	upper
Sentences	0.83	0.91	0.97	-0.10	-0.04	0.06	-0.02	0.04	0.12
Vocabulary	0.80	0.89	0.98	0.00	0.06	0.15	-0.12	-0.03	0.06
Sent.Completion	0.75	0.83	0.90	-0.05	0.04	0.11	-0.08	0.00	0.12
First.Letters	-0.08	0.00	0.09	0.68	0.86	0.97	-0.09	0.00	0.13
4.Letter.Words	-0.13	-0.01	0.12	0.57	0.74	0.90	-0.01	0.10	0.23
Suffixes	0.07	0.18	0.26	0.50	0.63	0.76	-0.23	-0.08	0.07
Letter.Series	-0.09	0.03	0.13	-0.06	-0.01	0.08	0.68	0.84	0.99
Pedigrees	0.27	0.37	0.52	-0.17	-0.05	0.04	0.33	0.47	0.60
Letter.Group	-0.16	-0.06	0.08	0.12	0.21	0.29	0.41	0.64	0.84

Interfactor correlations and bootstrapped confidence intervals

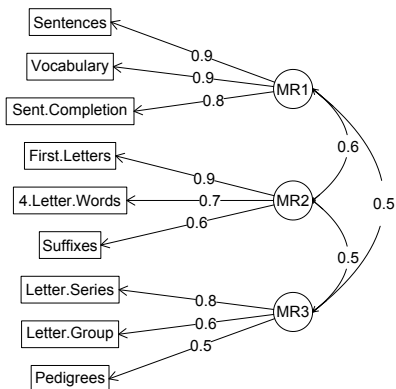
	lower	estimate	upper
MR1-MR2	0.47	0.59	0.68
MR1-MR3	0.39	0.54	0.61
MR2-MR3	0.30	0.52	0.64



The simple factor structure

`factor.diagram(f3) # show the diagram`

Factor Analysis

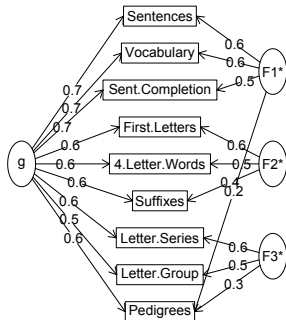


Two ways of viewing the higher order structure

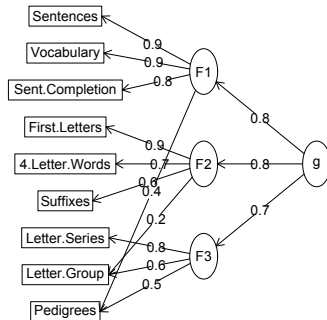
```
om <- omega(Thurstone)
```

```
omega.diagram(om,sl=FALSE)
```

Omega

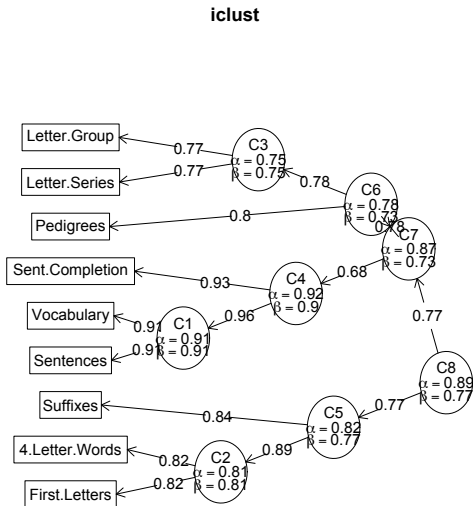


Hierarchical (multilevel) Structure



A hierarchical cluster structure found by iclust

iclust(Thurstone)



Structural Equation modeling packages

1. **sem** (Fox, Nie & Byrnes, 2013)
 - uses RAM notation
2. **lavaan** (Rosseel, 2012)
 - Mimics as much as possible MPLUS output
 - Allows for multiple groups
 - Easy syntax
3. **OpenMx** (Neale, Hunter, Pritikin, Zahery, Brick, Kickpatrick, Estabrook, Bates, Maes & Boker, 2016)
 - 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

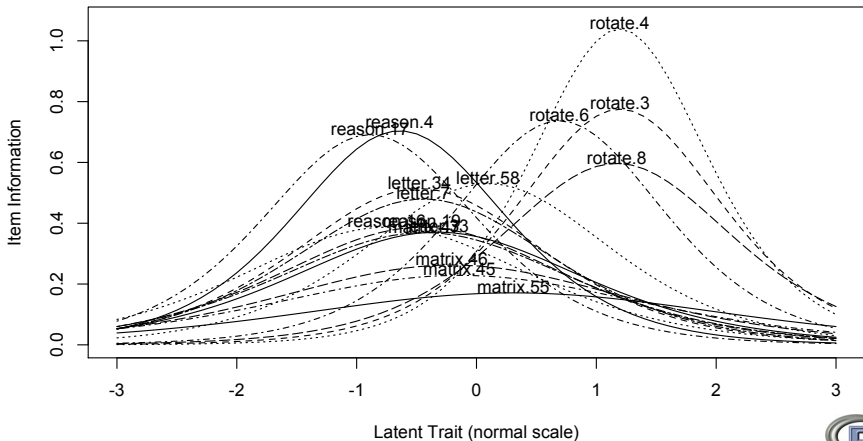
1. *psych* uses a factor analytic procedure to estimate item discriminations and locations
 - `irt.fa` finds either tetrachoric or polychoric correlation matrices
 - converts factor loadings to discriminations
 - `plot.irt` plots item information and item characteristic functions
 - look at examples for `irt.fa`
 - two example data sets: `ability` and `bfi`
2. Other packages to do more conventional IRT include *ltm*, *eRm*, *mirt*, + others



IRT from factor analysis: the irt.fa function in psych

Item Response Information curves for 16 ability items from ICAR

Item information from factor analysis



IRT from factor analysis: the `irt.fa` function in psych

Questions?



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

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



More useful statistical functions, Use ? for details

<code>mean</code>	(<code>x</code> , <code>na.rm=TRUE</code>) *	Selected functions from <i>psych</i> package	
<code>is.na</code>	(<code>x</code>) also <code>is.null(x)</code> , <code>is...</code>	<code>describe</code>	(<code>x</code>) descriptive stats
<code>na.omit</code>	(<code>x</code>) ignore missing data	<code>describeBy</code>	(<code>x</code> , <code>y</code>) descriptives by group
<code>sum</code>	(<code>x</code>)	<code>pairs.panels</code>	(<code>x</code>) SPLOM
<code>rowSums</code>	(<code>x</code>) see also <code>colSums(x)</code>	<code>error.bars</code>	(<code>x</code>) means + error bars
<code>colSums</code>	(<code>x</code>) see also <code>rowSums(x)</code>	<code>error.bars.by</code>	(<code>x</code>) Error bars by groups
<code>min</code>	(<code>x</code> , <code>na.rm=TRUE</code>)*	<code>fa</code>	(<code>x</code> , <code>n</code>) Factor analysis
<code>max</code>	(<code>x</code>) *ignores NA values	<code>principal</code>	(<code>x</code> , <code>n</code>) Principal components
<code>range</code>	(<code>x</code>)	<code>iclust</code>	(<code>x</code>) Item cluster analysis
<code>table</code>	(<code>x</code>)	<code>scoreItems</code>	(<code>x</code>) score multiple scales
<code>summary</code>	(<code>x</code>) depends upon <code>x</code>	<code>score.multiple.choice</code>	(<code>x</code>) score multiple choice scales
<code>sd</code>	(<code>x</code>) standard deviation	<code>alpha</code>	(<code>x</code>) Cronbach's alpha
<code>cor</code>	(<code>x</code> , <code>use="pairwise"</code>) correlation	<code>omega</code>	(<code>x</code>) MacDonald's omega
<code>cov</code>	(<code>x</code>) covariance	<code>irt.fa</code>	(<code>x</code>) Item response theory through factor analysis
<code>solve</code>	(<code>x</code>) inverse of <code>x</code>	<code>mediate</code>	(<code>y</code> , <code>x</code> , <code>m</code> , <code>data</code>) Mediation/moderation
<code>lm</code>	(<code>y~x</code>) linear model		
<code>aoa</code>	(<code>y~x</code>) ANOVA		



More help

1. An introduction to R as HTML, PDF or EPUB from <http://cran.r-project.org/manuals.html> (many different links on this page)
2. FAQ General and then Mac and PC specific
3. R reference card <http://cran.r-project.org/doc/contrib/Baggott-refcard-v2.pdf>
4. Various “cheat sheets” from RStudio <http://www.rstudio.com/resources/cheatsheets/>
5. Using R for psychology <http://personality-project.org/r/>
6. Package vignettes (e.g., <http://personality-project.org/r/psych/vignettes/overview.pdf>)
7. R listserv, StackOverflow, your students and colleagues



Outline

1. Part I: What is R, where did it come from, why use it
 - Installing R and adding packages
2. Part II: A brief introduction – an overview
 - R is just a fancy (very fancy) calculator
 - Descriptive data analysis
 - Some inferential analysis
3. Part III: Using R
 - Data entry
 - Descriptive
 - Inferential (t and F)
 - Regression, partial correlation, mediation
 - Basic R commands
4. Part IV: Psychometrics
 - Reliability and its discontents (α , ω_h , ω_t , λ_6)
 - EFA, CFA, and SEM
5. Part V: Help and More Help
 - List of useful commands



- Bechtoldt, H. (1961). An empirical study of the factor analysis stability hypothesis. *Psychometrika*, 26(4), 405–432.
- Bond, T. G. (1995). *BLOT: Bond's Logical Operations Test*. Townsville, Australia: James Cook University. (Original work published 1976).
- Burt, C. (1915). General and specific factors underlying the primary emotions. In *Reports of the British Association for the Advancement of Science (85th Meeting)*, (pp. 694–696)., London (retrieved from the web at <http://www.biodiversitylibrary.org/item/95822#790>)). John Murray.
- Condon, D. M. & Revelle, W. (2014). The International Cognitive Ability Resource: Development and initial validation of a public-domain measure. *Intelligence*, 43, 52–64.
- Costa, P. T. & McCrae, R. R. (1985). *NEO PI professional manual*. Odessa, FL: Psychological Assessment Resources, Inc.



- Dwyer, P. S. (1937). The determination of the factor loadings of a given test from the known factor loadings of other tests. *Psychometrika*, 2(3), 173–178.
- Fox, J., Nie, Z., & Byrnes, J. (2013). *sem: Structural Equation Models*. R package version 3.1-3.
- Guilford, J. P. (1954). *Psychometric Methods* (2nd ed.). New York: McGraw-Hill.
- Hayes, A. F. (2013). *Introduction to mediation, moderation, and conditional process analysis: A regression-based approach*. New York: Guilford Press.
- Holzinger, K. & Swineford, F. (1937). The bi-factor method. *Psychometrika*, 2(1), 41–54.
- Neale, M. C., Hunter, M. D., Pritikin, J. N., Zahery, M., Brick, T. R., Kickpatrick, R. M., Estabrook, R., Bates, T. C., Maes, H. H., & Boker, S. M. (2016). OpenMx 2.0: Extended structural equation and statistical modeling. *Psychometrika*.



- Reise, S., Morizot, J., & Hays, R. (2007). The role of the bifactor model in resolving dimensionality issues in health outcomes measures. *Quality of Life Research*, *16*(0), 19–31.
- Revelle, W. & Condon, D. M. (2015). A model for personality at three levels. *Journal of Research in Personality*, *56*, 70–81.
- Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1–36.
- Smillie, L. D., Cooper, A., Wilt, J., & Revelle, W. (2012). Do extraverts get more bang for the buck? refining the affective-reactivity hypothesis of extraversion. *Journal of Personality and Social Psychology*, *103*(2), 306–326.
- Thurstone, L. L. & Thurstone, T. G. (1941). *Factorial studies of intelligence*. Chicago, Ill.: The University of Chicago press.

