Psychology 371: Personality Research
Using R in Personality Research

William Revelle

Department of Psychology
Northwestern University
Evanston, Illinois USA

June, 2014

Outline

R: What is it?

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

Statistical Programs for Psychologists

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

Statistical Programs for Psychologists

- General purpose programs
  - R
  - $+
  - A
  - $P$$
  - $TATA
  - $y$tat
- Specialized programs
  - Mx (OpenMx is part of R)
  - EQ$
  - AMO$
  - LI$REL
  - MPIu$
  - Your favorite program

R: A way of thinking

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

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.

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 3.1.0
- 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)

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

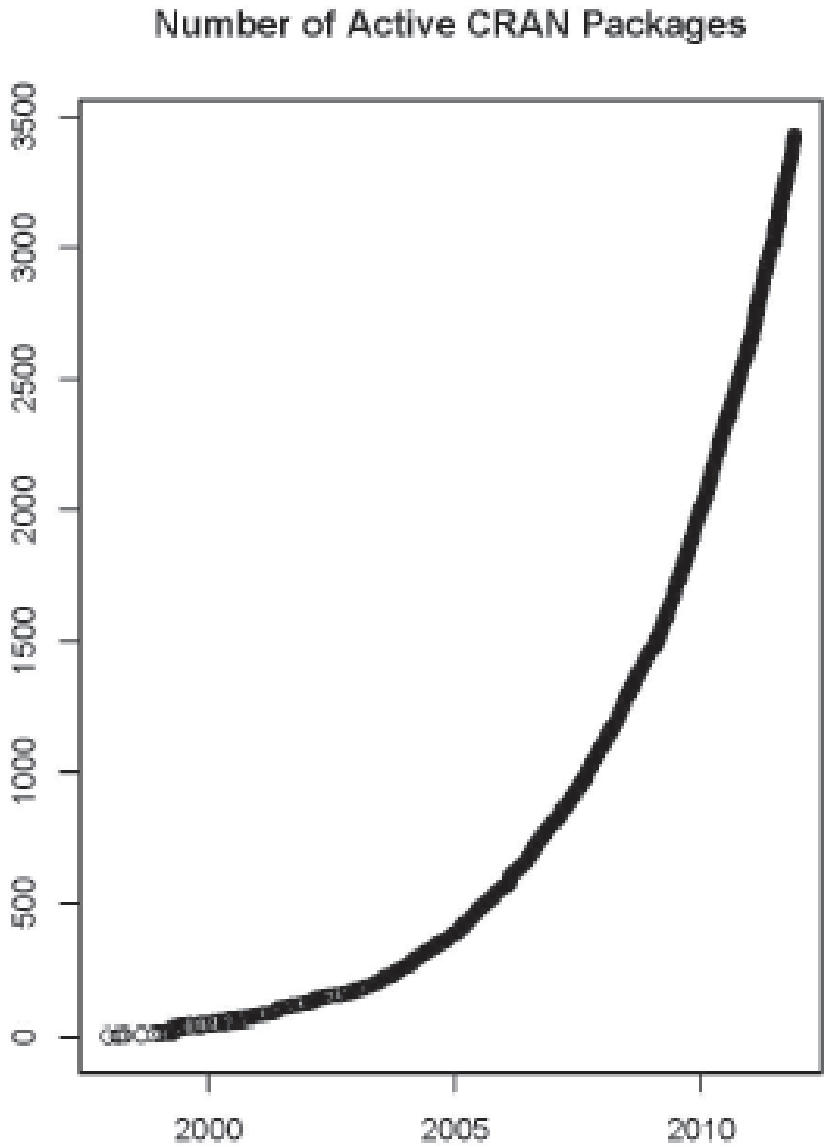
1. an effective data handling and storage facility,
2. a suite of operators for calculations on arrays, in particular matrices,
3. a large, coherent, integrated collection of intermediate tools for data analysis,
4. graphical facilities for data analysis and display either on-screen or on hardcopy, and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

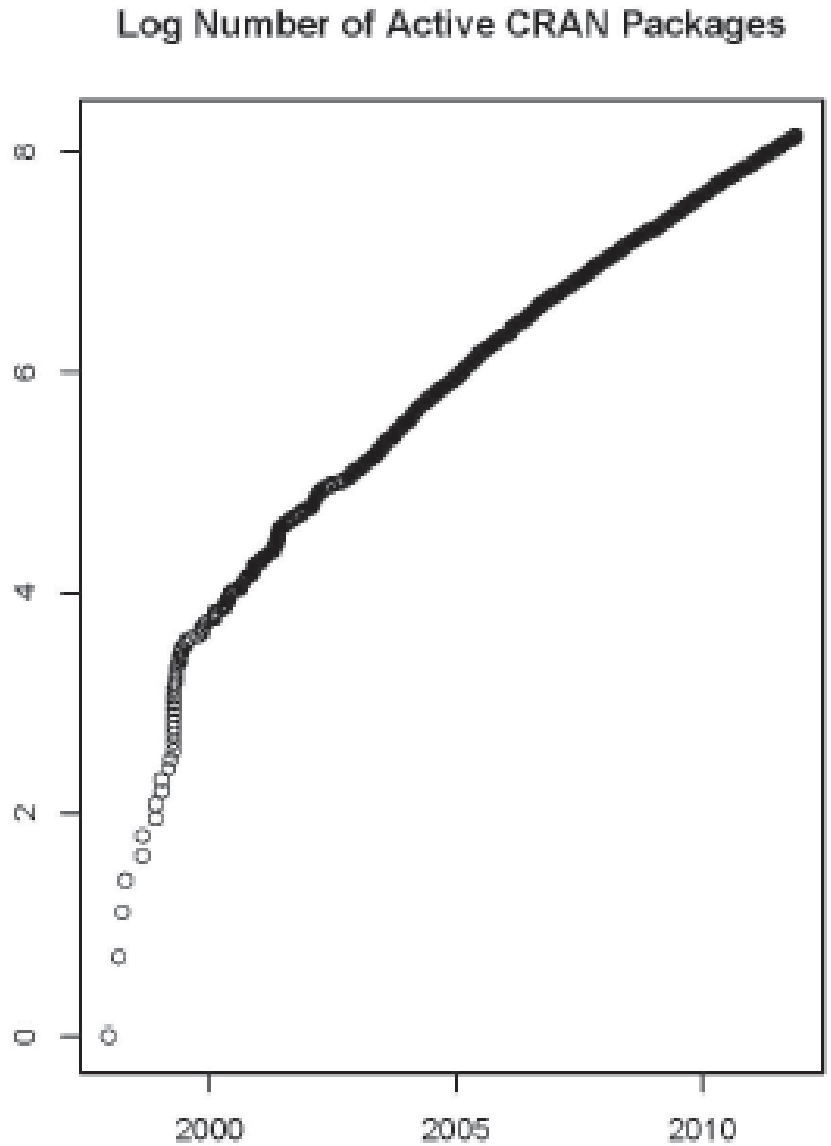
Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages ... available through the CRAN family of Internet sites covering a very wide range of modern statistics. (Adapted from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

R: A brief history

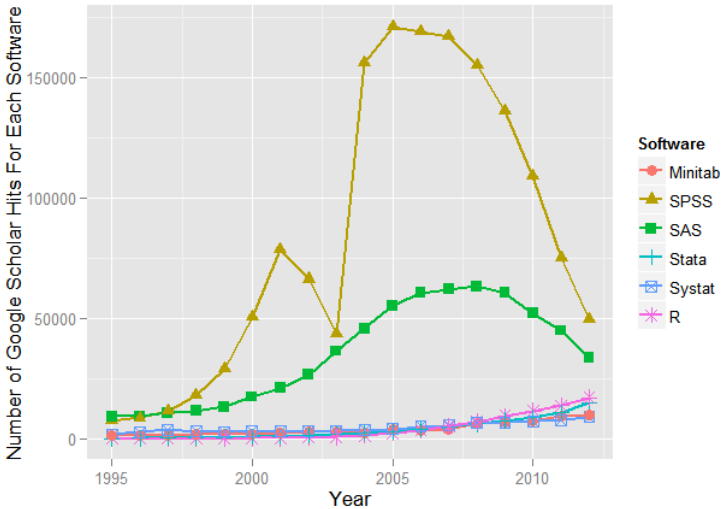
- 1991-93: Ross Dhaka and Robert Gentleman begin work on R project for Macs at U. Auckland
- 1995: R available by ftp under the GPL
- 96-97: mailing list and R core group is formed
- 2000: John Chambers, designer of S joins the Rcore (wins a prize for best software from ACM for S)
- 2001-2014: Core team continues to improve base package with a new release every 6 months.
- Many others contribute "packages" to supplement the functionality for particular problems
  - 2003-04-01: 250 packages
  - 2004-10-01: 500 packages
  - 2007-04-12: 1,000 packages
  - 2009-10-04: 2,000 packages
  - 2011-05-12: 3,000 packages
  - 2012-08-27: 4,000 packages../../../R.short.courses/R/images/
  - 2014-05-16: 5,547 packages (on CRAN) + 824 bioinformatic

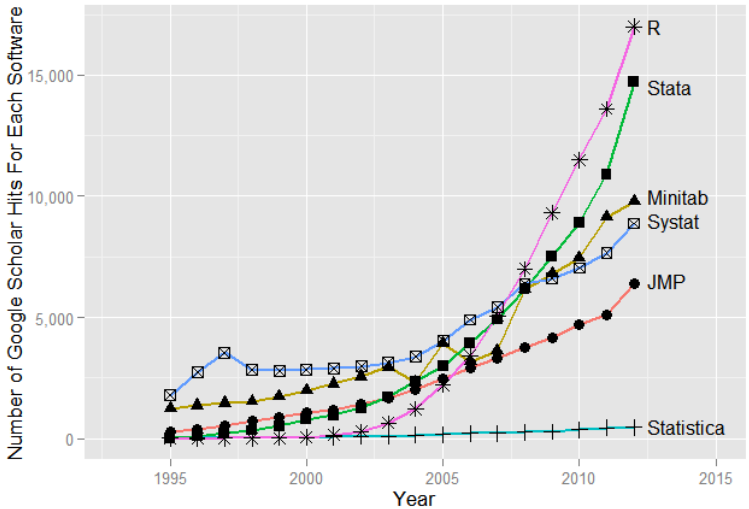
Rapid and consistent growth in packages contributed to R





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 ($> 14,000$ citations, $\approx 1,800/year$)
4. Google Page rank

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 is coming this fall.
4. By sharing our code and data the field can increase the possibility of doing replicable science.

Reproducible Research: Sweave and KnitR

Sweave is a tool that allows to embed the R code for complete data analyses in LaTeXdocuments. 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 LaTeXdocument. 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
   - 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).

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 3.1.0) (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

Annotated installation guide: don't type the >

```
> install.packages("ctv")
```

- Install the task view installer package. You might have to choose a "mirror" site.

```
> library(ctv)
```

- Make it active

```
> install.views("Psychometrics")
```

- Install all the packages in the "Psychometrics" task view. This will take a few minutes.

```
#or just install a few packages
> install.packages("psych")
```

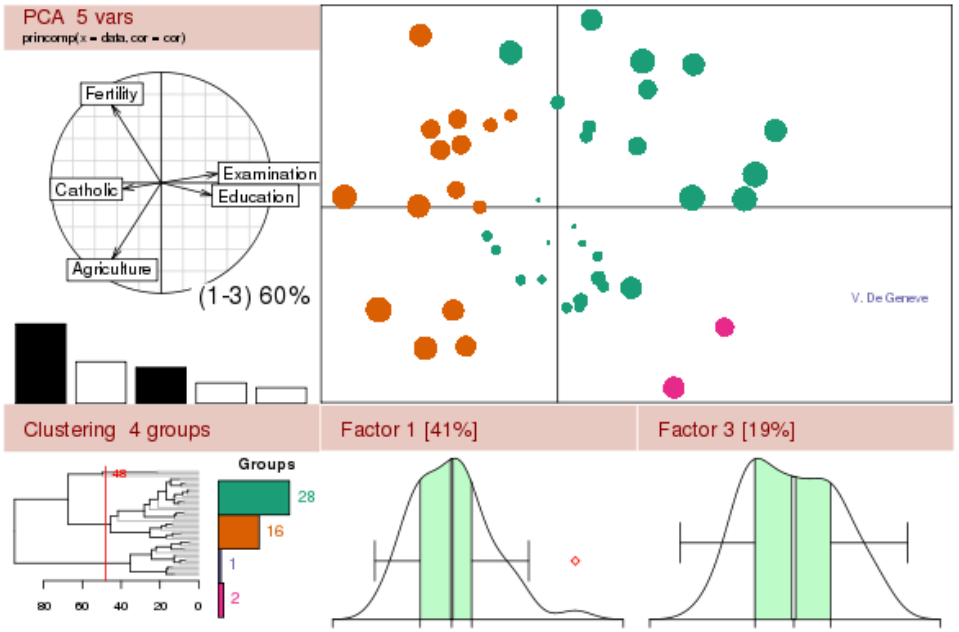
- Or, just install one package (e.g., psych)

```
> install.packages("GPArotation")
> install.packages("MASS")
> install.packages("mvtnorm")
```

- as well as a few suggested packages that add functionality for factor rotation, multivariate normal distributions, etc.

Go to the R.project.org

The R Project for Statistical Computing



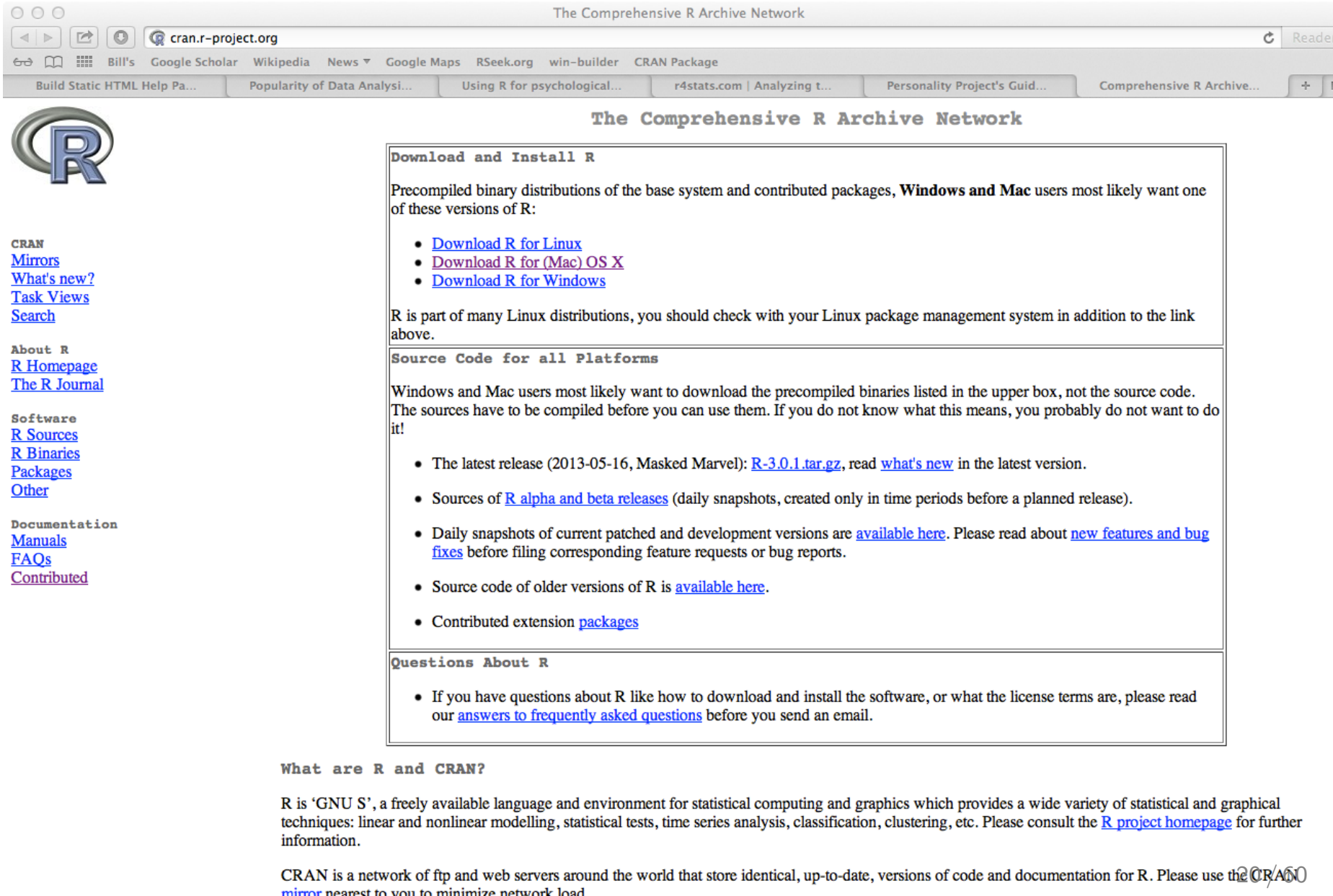
Getting Started:

- 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.

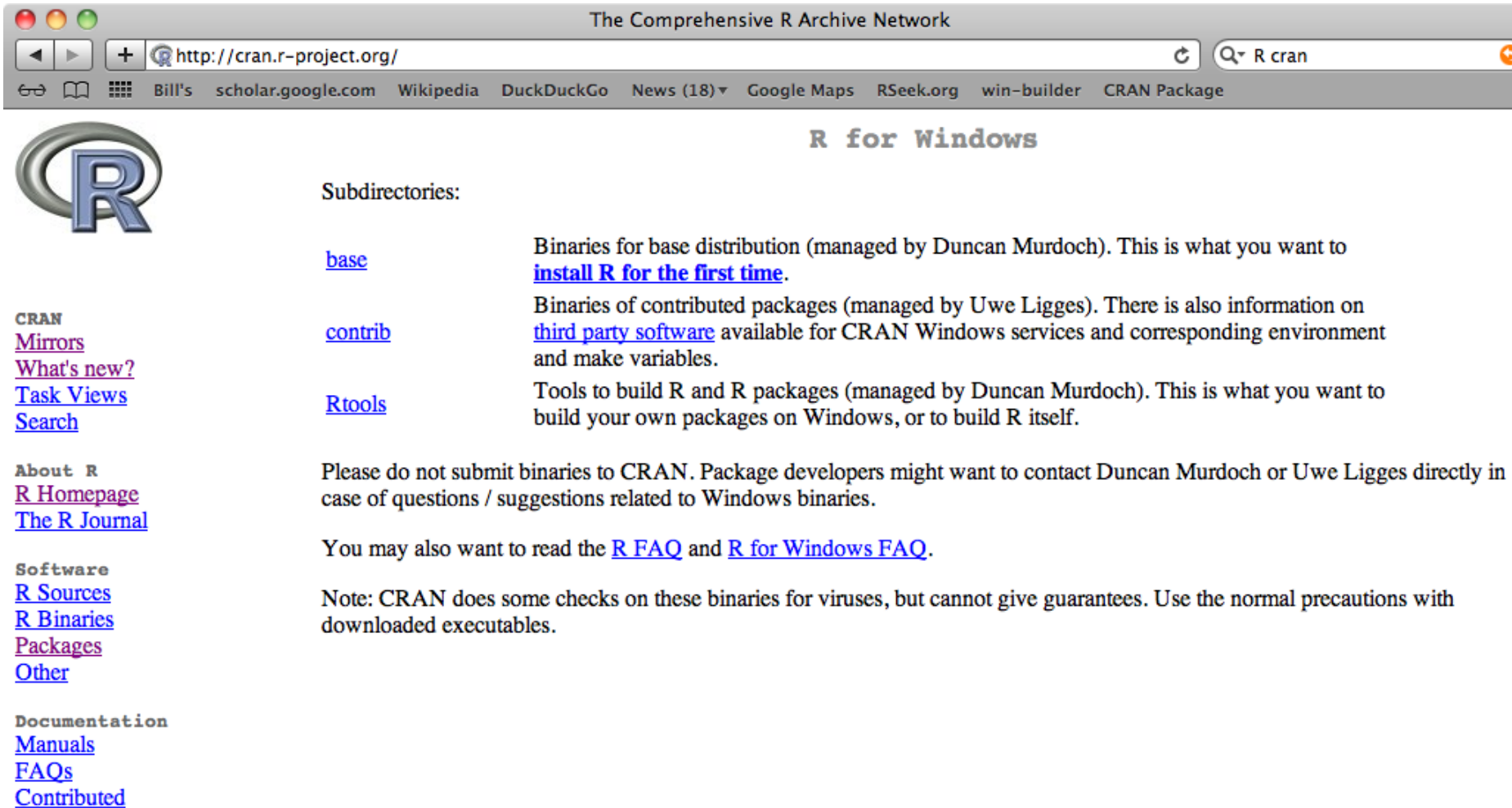
News:

- **R version 3.1.0** (Spring Dance) has been released on 2014-04-10.
- **R version 3.0.3** (Warm Puppy) has been released on 2014-03-06.
- **The R Journal Vol.5/2** is available.
- **useR! 2013**, took place at the University of Castilla-La Mancha, Albacete, Spain, July 10-12 2013.
- **R version 2.15.3** (Security Blanket) has been released on 2013-03-01.

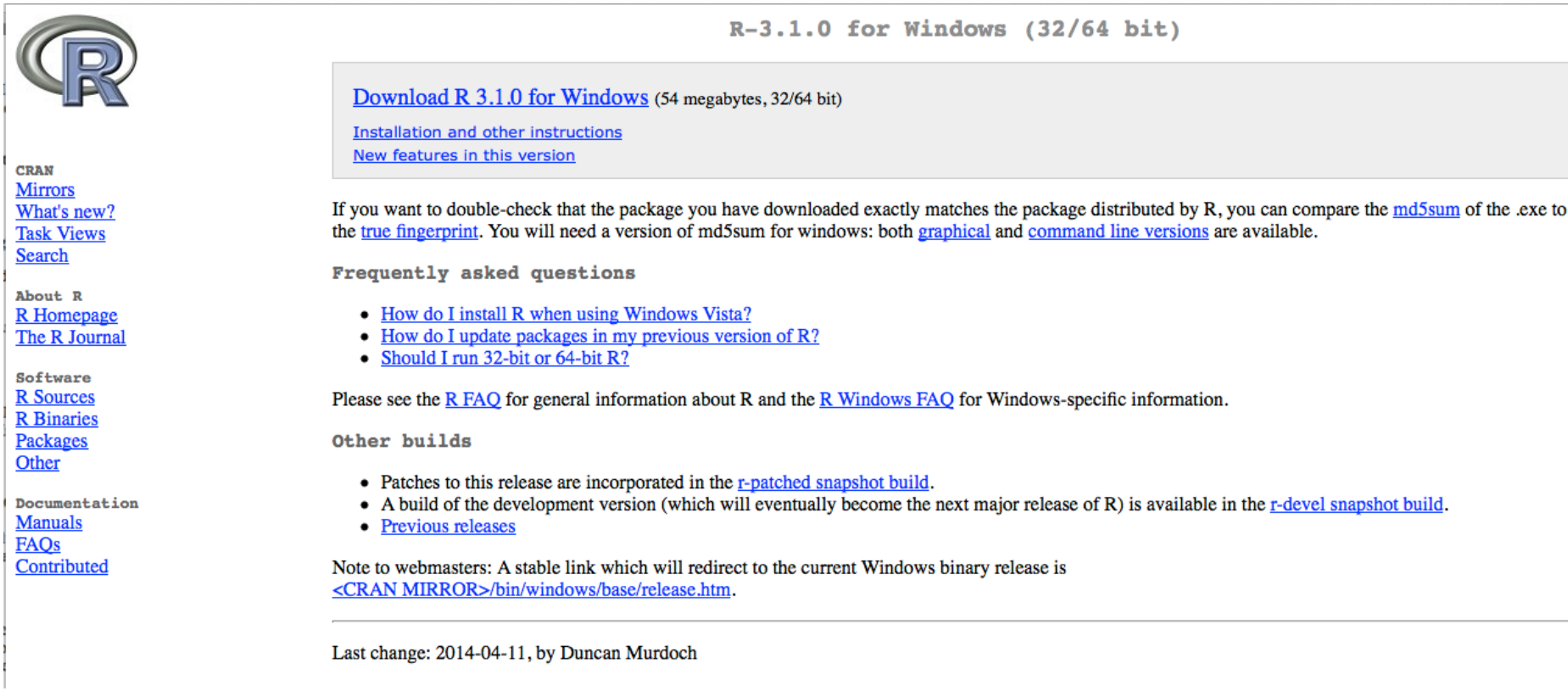
Go to the Comprehensive R Archive Network (CRAN)



Download and install the appropriate version – PC



Download and install the appropriate version – PC



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

R-3.1.0 for Windows (32/64 bit)

Download R 3.1.0 for Windows (54 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

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

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

Other builds

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

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

Last change: 2014-04-11, by Duncan Murdoch

Download and install the appropriate version – Mac

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

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 3.1.0 "Spring Dance" released on 2014/04/10

This binary distribution of R and the GUI supports 64-bit Intel based Macs on Mac OS X 10.6 (Snow Leopard) 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-3.1.0-snowleopard.pkg`
in the *Terminal* application to print the MD5 checksum for the R-3.1.0-snowleopard.pkg image. On Mac OS X 10.7 and later you can also validate the signature using `pkgutil --check-signature R-3.1.0-snowleopard.pkg`

Files:

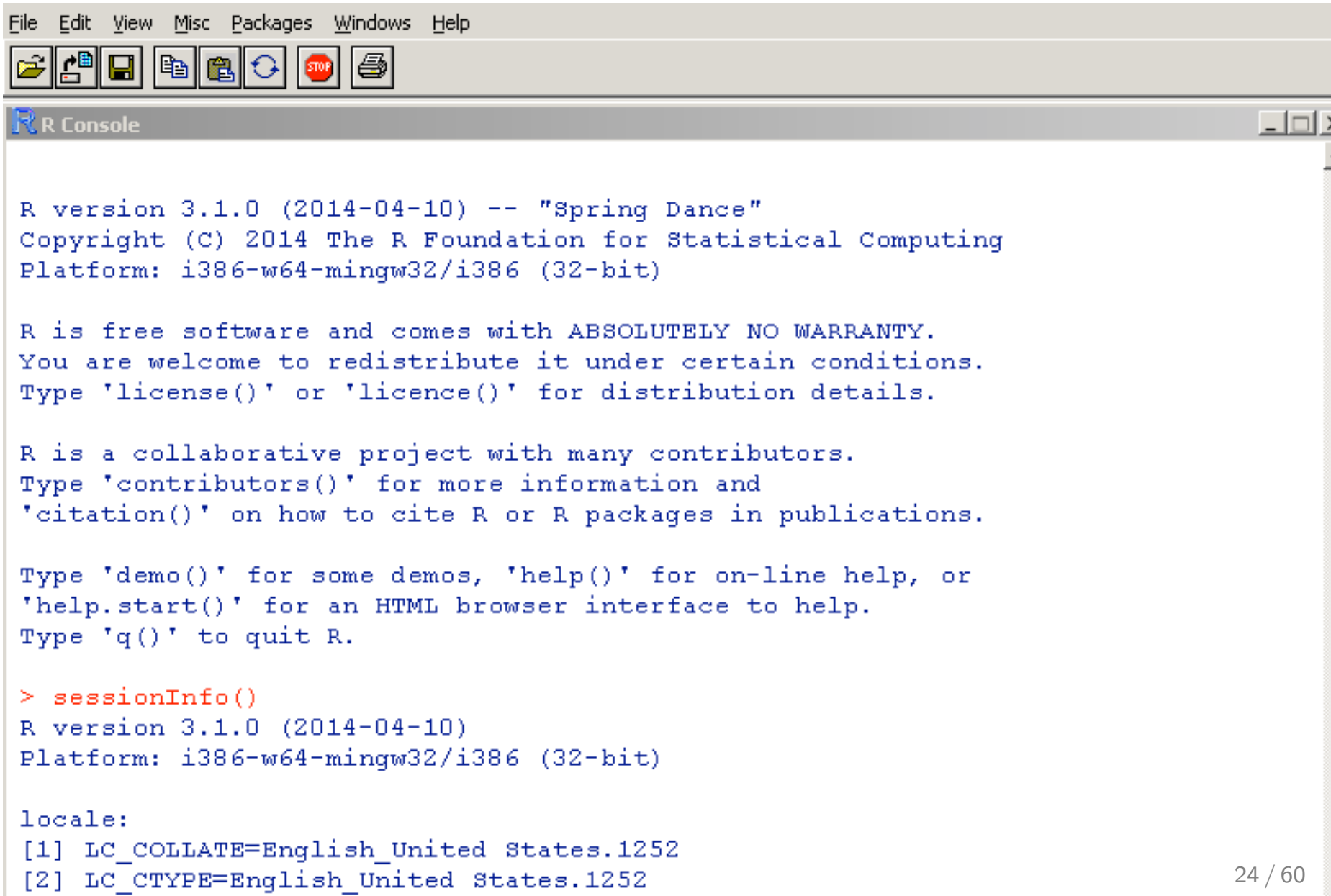
R-3.1.0-snowleopard.pkg MD5-hash: 60f9713fbb4c2868771de60ed3882009 SHA1-hash: 7943746b409897fc5c9ee5fb21e103c766b4ecfe (ca. 68MB)	**R 3.1.0** binary for Mac OS X 10.6 (Snow Leopard) and higher, signed package. Contains R 3.1.0 framework, R.app GUI 1.64 in 64-bit for Intel Macs. The above file is an Installer package which can be installed by double-clicking. Depending on your browser, you may need to press the control key and click on this link to download the file. This package contains the R framework, 64-bit GUI (R.app) and Tcl/Tk 8.6.0 X11 libraries. The latter component is optional and can be ommitted when choosing "custom install", it is only needed if you want to use the `tcltk` R package. GNU Fortran is **NOT** included (needed if you want to compile packages from sources that contain FORTRAN code) please see the tools directory.
R-3.1.0-mavericks.pkg MD5-hash: 06c5465ef290b64efe25ac32790e816e SHA1-hash: 5d1b93ab3569b7f126fb9ec7c196ef52134f9105 (ca. 55MB)	**R 3.1.0** binary for Mac OS X 10.9 (Mavericks) and higher, signed package. It contains the same software versions as above, but this R build has been built with Xcode 5 to leverage new compilers and functionalities in Mavericks not available in earlier OS X versions.
Mac-GUI-1.64.tar.gz MD5-hash: 3c33bcda08a00f8778cfc978c7f806d5	Sources for the R.app GUI 1.64 for Mac OS X. This file is only needed if you want to join the development of the GUI, it is not intended for regular users. Read the INSTALL file for further instructions.
NEWS (for Mac GUI)	News features and changes in the R.app Mac GUI

The new R.app Cocoa GUI has been written by Simon Urbanek and Stefano Iacus with contributions from many developers and translators world-wide, see "About R" in the GUI.

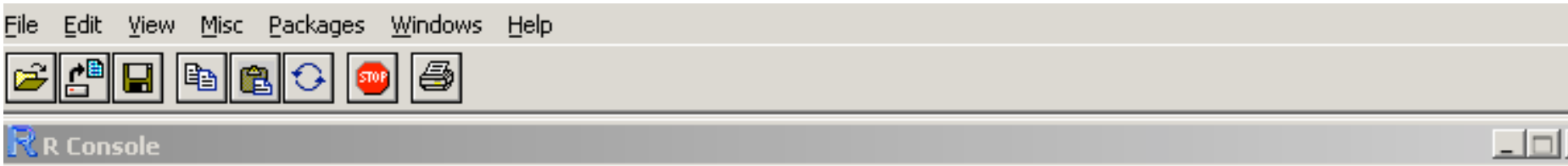
Subdirectories:

tools	Additional tools necessary for building R for Mac OS X: Universal GNU Fortran compiler for Mac OS X (see R for Mac tools page for details).
contrib	Binaries of package builds for Mac OS X 10.6 or higher (Snow Leopard build)
mavericks	Binaries of package builds for Mac OS X 10.9 or higher (Mavericks build)
leopard	Legacy binaries of universal (32-bit and 64-bit) package builds for Mac OS X 10.5 or higher (Leopard build)
universal	Legacy binaries of universal (32-bit) package builds for Mac OS X 10.4 (Tiger build)

Starting R on a PC

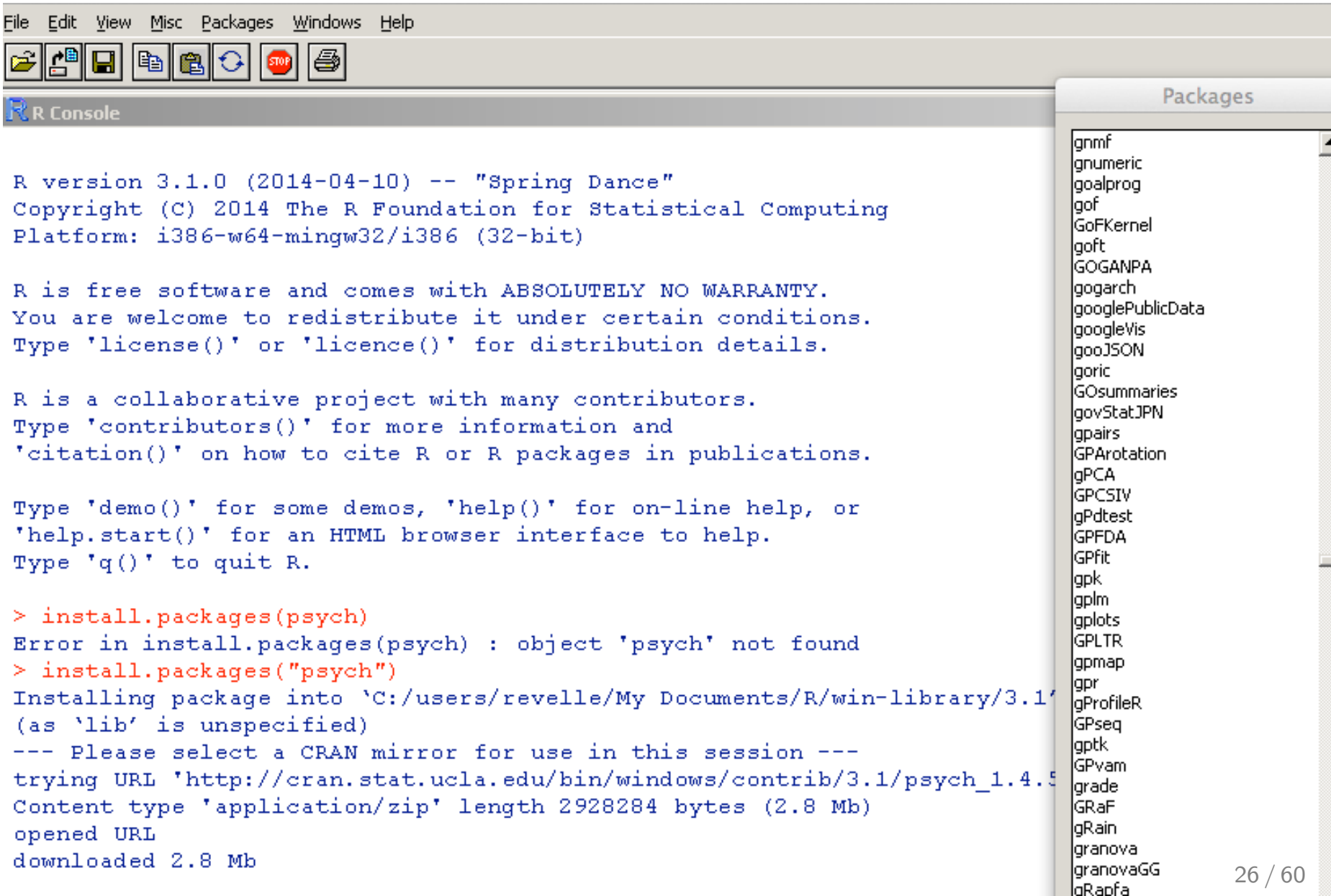


Installing a package (psych) on a PC by hand – note error




```
R version 3.1.0 (2014-04-10) -- "Spring Dance"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-bit)

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

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

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

> install.packages(psych)
Error in install.packages(psych) : object 'psych' not found
> install.packages("psych")
Installing package into 'C:/users/revelle/My Documents/R/win-library/3.1'
(as 'lib' is unspecified)
--- Please select a CRAN mirror for use in this session ---
trying URL 'http://cran.stat.ucla.edu/bin/windows/contrib/3.1/psych_1.4.5.zip'
```

Installing packages using the menu



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

```
R Under development (unstable) (2014-04-17 r65403) -- "Unsuffered Consequences"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin13.1.0 (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 1.65 (6738) x86_64-apple-darwin13.1.0]

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

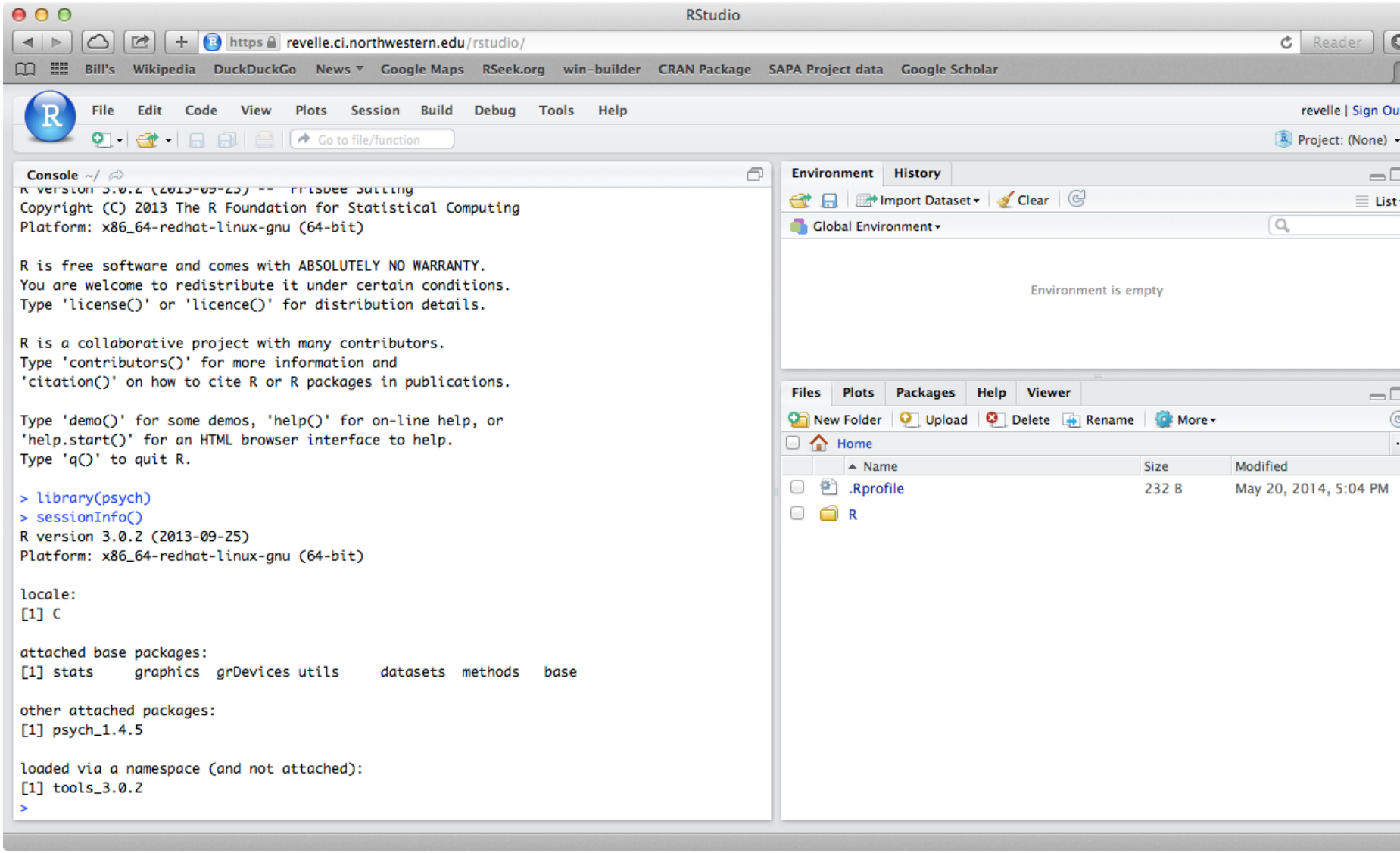
Check the version number for R (should be $\geq$ 3.1.0) and for psych ($\geq$1.4.5)

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


R Under development (unstable) (2014-04-17 r65403)
Platform: x86_64-apple-darwin13.1.0 (64-bit)

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_1.4.5
>
```

Various ways to run R

1. UNIX (and *NIX like) environments
   - Non interactive
   - Particularly fast if on remote processors
   - RStudio Server as "Integrated Development Environment" (IDE)
   - RStudiio can be run remotely with a browser (e.g., even from an IPad)
2. PC
   - quasi GUI + text editor of choice
   - RStudio as "Integrated Development Environment" (IDE)
3. Mac
   - R.app + text editor of choice
   - RStudio as "Integrated Development Environment" (IDE)

R Studio is a useful "Integrated Development Environment" (IDE)



Basic R commands – remember don't enter the >

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

```
> 2 + 2
```

```
[1] 4
```

```
> 3^4
```

```
[1] 81
```

```
> sum(1:10)
```

```
[1] 55
```

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

```
[1] 210
```

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

```
> pnorm(q = 1)
```

```
[1] 0.8413447
```

```
> pt(q = 2, df = 20)
```

```
[1] 0.9703672
```

R is a set of distributions. Don't buy a stats book with tables!

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

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

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 delimited
my.data <- read.clipboard.csv()   #assumes headers and comma delimited
my.data <- read.clipboard.tab()   #assumes headers and tab delimited
                                        (e.g., from Excel)
my.data <- read.clipboard.lower()  #read in a matrix given the lower
my.data <- read.clipboard.upper() #  or upper  off diagonal
my.data <- read.clipboard.fwf()  #read in data using a fixed format width
                                        (see read.fwf for instructions)
```

4. `read.clipboard()` has default values for the most common cases and these do not need to be specified. Consult ?read.clipboard for details.

Reading from a local or remote file

1. Perhaps the standard way of reading in data is using the `read` command.
   - First must specify the location of the file
   - Can either type this in directly or use the `file.choose` function. This goes to your normal system file handler.
   - The file name/location can be a remote URL. (Note that read.file will not work on https files.)
2. Two examples of reading data

```
file.name <- file.choose() #this opens a window to allow you find the file
#or
datafilename="http://personality-project.org/r/datasets/R.appendix1.data"
my.data <- read.table(fdatafilename,header=TRUE)    #unless it is https (se
#or
data.ex1=read.https(datafilename,header=TRUE)   #read an https file
> dim(data.ex1)  #what are the dimensions of what we read?
[1] 18  2
> describe(data.ex1) #do the data look right?
           var  n  mean   sd median trimmed  mad min max range skew kurtosi
Dosage*      1 18  1.89 0.76      2    1.88 1.48   1   3     2 0.16    -1.1
Alertness    2 18 27.67 6.82     27   27.50 8.15  17  41    24 0.25    -0.6
```

Put it all together: read, show, describe

```
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 are 100s – 1000s)
3. Just show the first and last (4) lines
4. Find descriptive statistics

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.

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

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

epi.bfi has 231 cases from two personality measures.

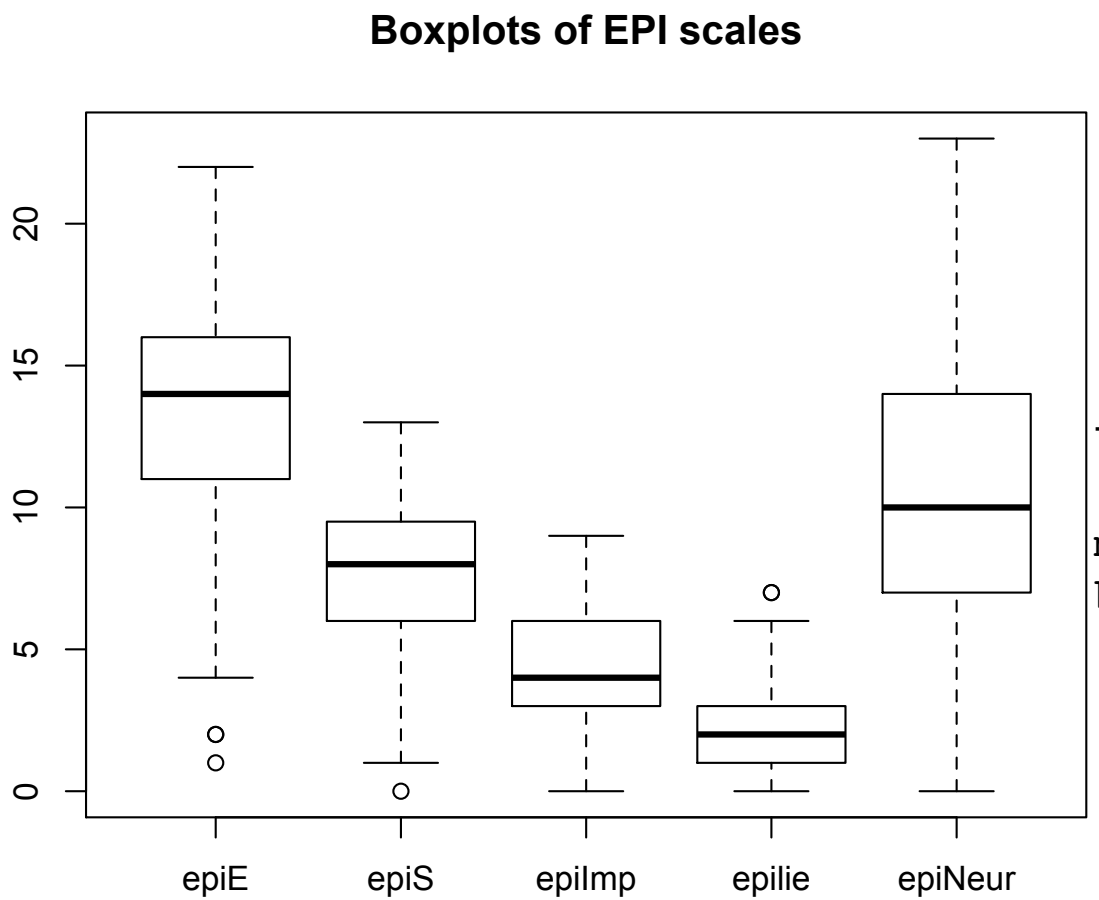
Now find the descriptive statistics for this data set

```
> 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
bfagree	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

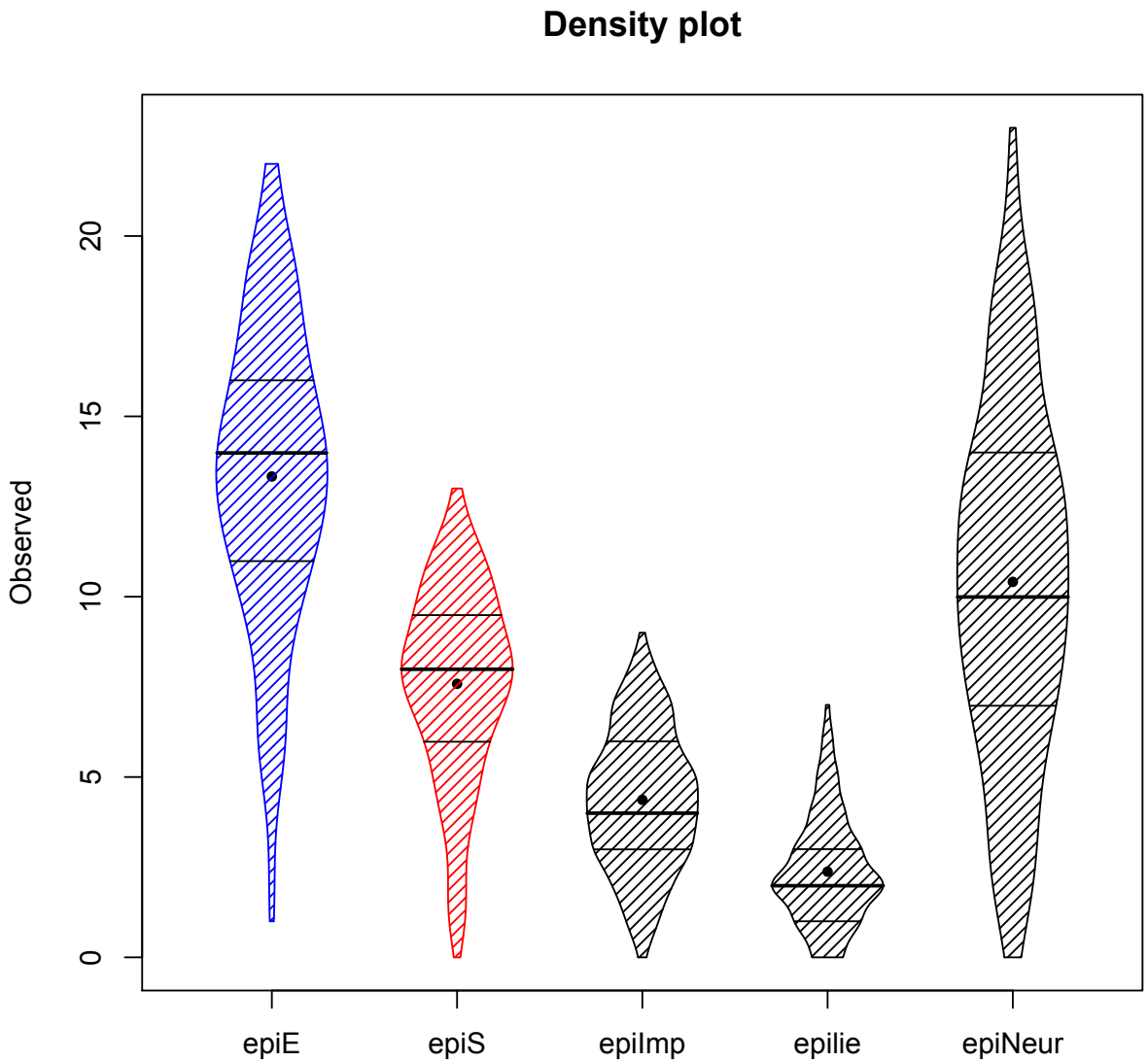
Show the Tukey "boxplot" for the Eysenck Personality Inventory



Use the `box plot` function

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

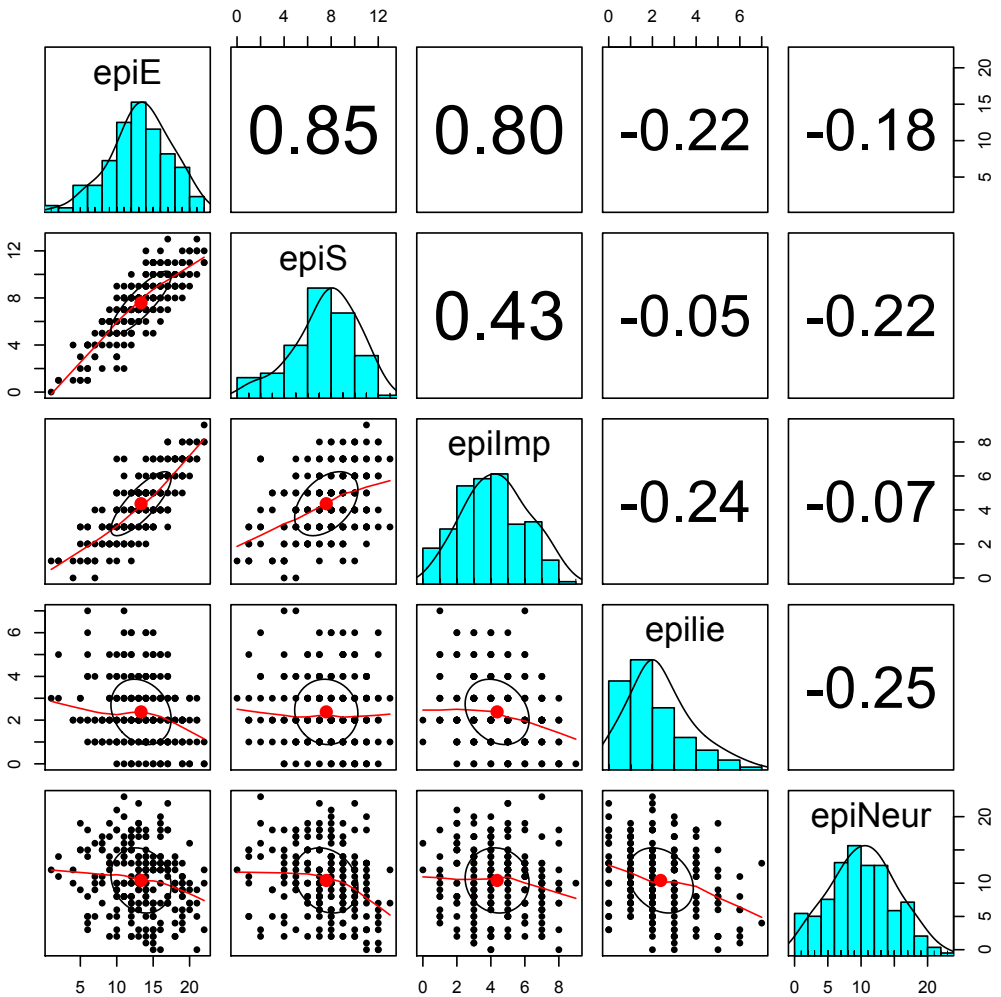
An alternative display is a 'violin' plot (available as `violinBy`)



Use the `violinBy` function from *psych*

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

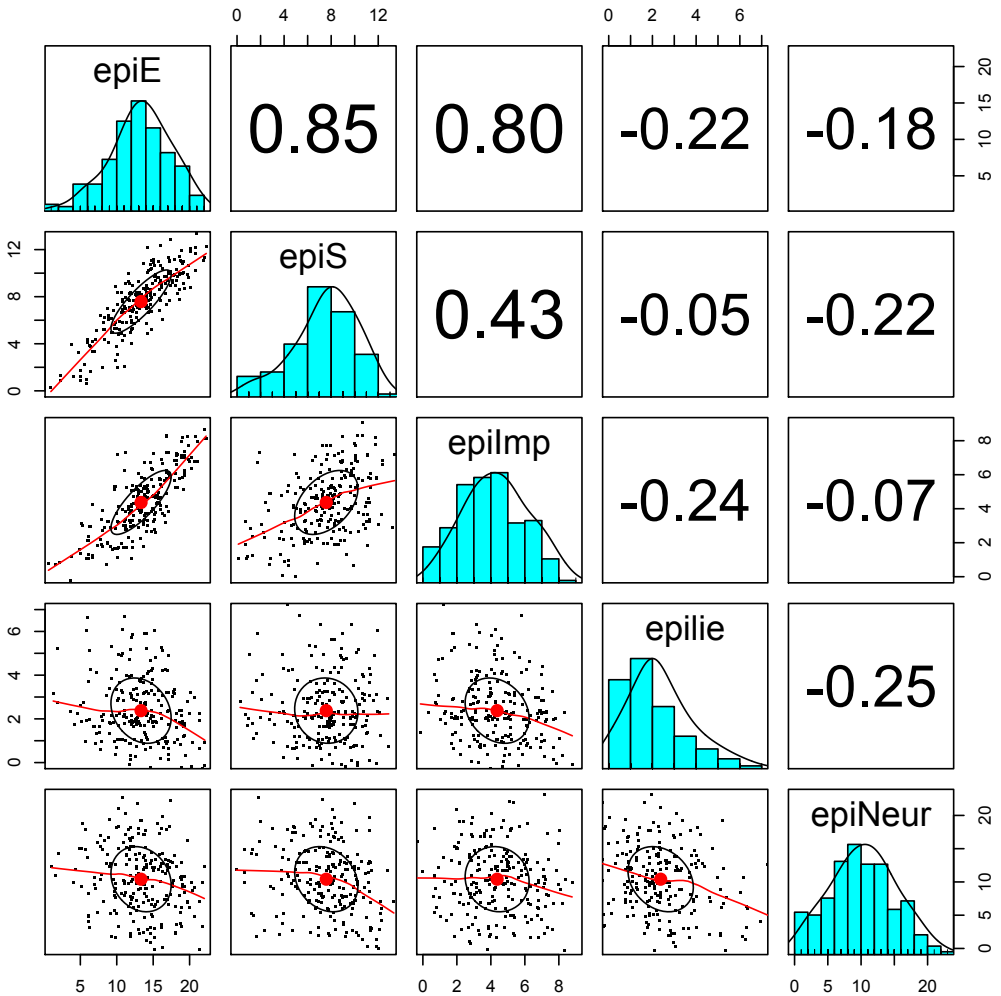
Plot the scatter plot matrix (SPLOM) of the first 5 variables using the `pairs.panels` function



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

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

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



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

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

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

```
> 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	-0.04
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	0.61
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

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

```
> 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

```
> corr.test(my.data)
Call:corr.test(x = my.data)
Correlation matrix
          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
..
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
Sample Size
          epiE epiS epiImp epilie epiNeur bfagree bfcon bfext bfneur bfopen bdi traitanx stateanx
epiE       231  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      231
Probability values (Entries above the diagonal are adjusted for multiple tests.)
          epiE epiS epiImp epilie epiNeur bfagree 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     1.00
epiS      0.00 0.00   0.00   1.00    0.04    0.08  1.00  0.00   1.00   0.62 1.00    0.00     1.00
epiImp    0.00 0.00   0.00   0.01    1.00    1.00  0.01  0.00   1.00   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     0.61
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     0.00
bfagree   0.01 0.00   0.23   0.01    0.21    0.00  0.00  0.00   1.00   0.00 0.95    0.00     0.12
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     1.00
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     0.76
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     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     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     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     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     0.00
>
```

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

Classic theory estimates of reliability

1. Scoring tests
   - `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.
2. 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

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

```
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...
```

Using `scoreItems` to score 25 Big 5 items (taken from the bfi example

```
> 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
...
```

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
```

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*.

```
datafilename="https://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.

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

```
 > aov.ex2 = aov(Alertness~Gender*Dosage,data=data.ex2)  #do the analysis of va
> summary(aov.ex2)                                       #show the summary table
             Df  Sum Sq Mean Sq F value Pr(>F)
Gender        1  76.562  76.562  2.9518 0.1115
Dosage        1   5.062   5.062  0.1952 0.6665
Gender:Dosage 1   0.063   0.063  0.0024 0.9617
```

Show the results table

```
> print(model.tables(aov.ex2,"means"),digits=3)
Residuals     12 311.250  25.938

Tables of means
Grand mean

14.0625

 Gender
Gender
    f     m
16.25 11.88

 Dosage
Dosage
    a     b
13.50 14.62

 Gender:Dosage
      Dosage
Gender a     b
     f 15.75 16.75
     m 11.25 12.50
```

Analysis of Variance: Within subjects

1. Somewhat more complicated because we need to convert "wide" data.frames to "long" or "narrow" data.frame.
2. This can be done by using the `stack` function. Some data sets are already in the long format.
3. A detailed discussion of how to work with repeated measures designs is at http://personality-project.org/r/r.anova.html and at http://personality-project.org/r
4. See also the tutorial by Jason French at http://jason-french.com/tutorials/repeatedmeasures.html

Analysis of variance within subjects

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

Error: Subject
              Df  Sum Sq Mean Sq F value  Pr(>F)
Gender         1  542.26  542.26  5.6853 0.03449 *
Dosage         2  694.91  347.45  3.6429 0.05803 .
Gender:Dosage  2   70.80   35.40  0.3711 0.69760
Residuals     12 1144.56   95.38
Signif. codes:  0 Ô***Õ 0.001 Ô**Õ 0.01 Ô*Õ 0.05 Ô.Õ 0.1 Ô Õ 1

Error: Subject:Task
                   Df Sum Sq Mean Sq F value    Pr(>F)
Task                1 96.333  96.333 39.8621 3.868e-05 ***
Task:Gender         1  1.333   1.333  0.5517    0.4719
Task:Dosage         2  8.167   4.083  1.6897    0.2257
Task:Gender:Dosage  2  3.167   1.583  0.6552    0.5370
Residuals          12 29.000   2.417
... (lots more)
```

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 Ô***Õ 0.001 Ô**Õ 0.01 Ô*Õ 0.05 Ô.Õ 0.1 Ô Õ 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

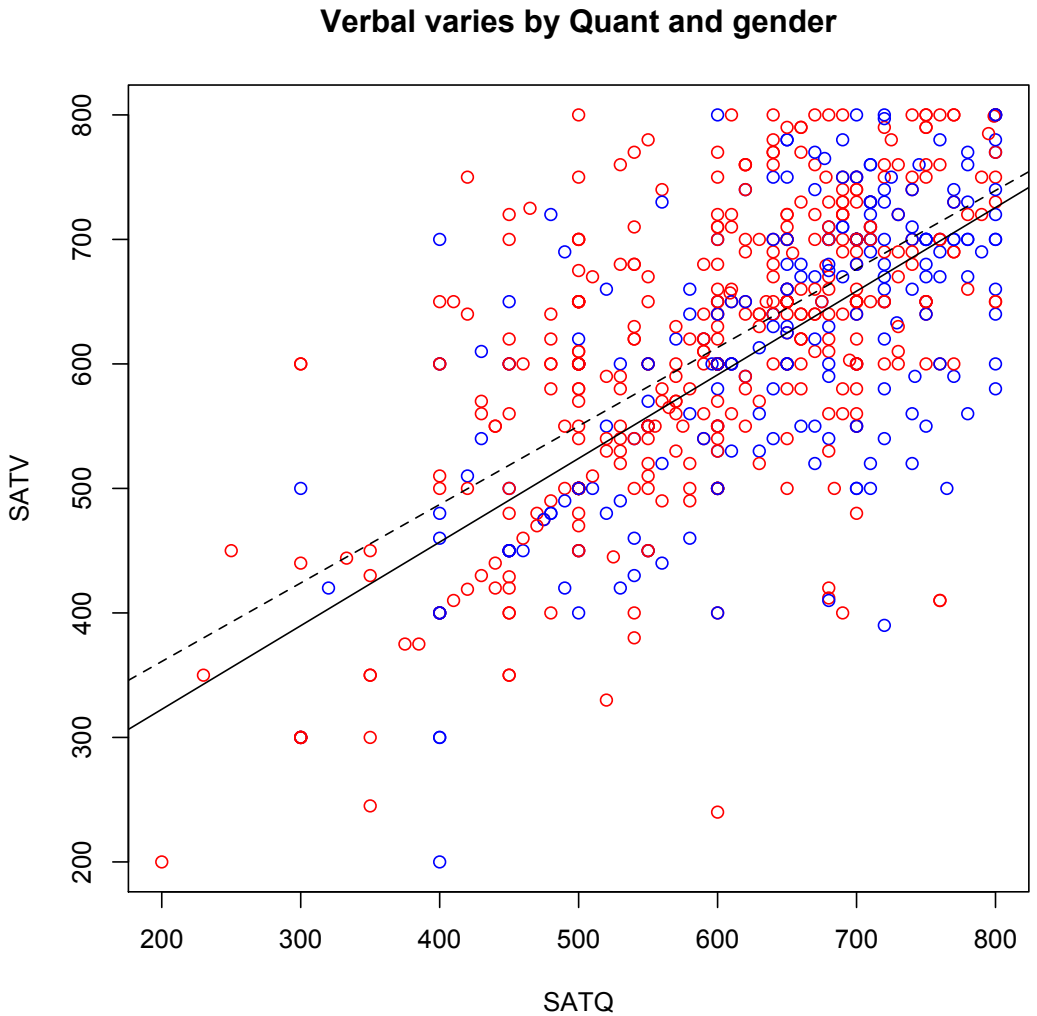
```
> zsat <- data.frame(scale(sat.act,scale=FALSE))
> mod2 <-  lm(SATV  ~ education * gender * SATQ,data=zsat)
> summary(mod2)

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 Ô***Õ 0.001 Ô**Õ 0.01 Ô*Õ 0.05 Ô.Õ 0.1 Ô Õ 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 Ô***Õ 0.001 Ô**Õ 0.01 Ô*Õ 0.05 Ô.Õ 0.1 Ô
```

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")
```

A short guide to R

1. http://personality-project.org/r/ A short guide to R
   - Examples that take you through much of what you need
2. The *psych* package
   - Most of the stats that you will need
   - Remember to make psych active by library(psych) at the start of each session.

"How to"

1. Installing R and the *psych* package
   - http://personality-project.org/r/psych/HowTo/getting_started.pdf
2. Using the *psych* package to score personality scales
   - http://personality-project.org/r/psych/HowTo/scoring.pdf
3. Using the *psych* package for factor analysis
   - http://personality-project.org/r/psych/HowTo/factor.pdf