

A

Glossary of R jargon

Below is a selection of common R terms defined first using Stata jargon (or plain English when possible) and then more formally using R jargon. Some definitions in Stata jargon are quite loose given the fact that they have no direct analog of some R terms. Definitions in R terms are often quoted (with permission) or paraphrased from *S Poetry* by Patrick Burns [3].

Apply

The process of having a command work on variables or observations. Determines whether a procedure will act as a typical command or as a function instead. Also the name of a function that controls that process. More formally, the process of targeting a function on rows or columns. Also a function that does that.

Argument

The options that control what the commands do and the arguments that control what functions do. Confusing because in R, functions do what both commands and functions do in Stata. More formally, input(s) to a function that control it. Includes data to analyze.

Array

A matrix with more than two dimensions. All variables must be only one type (e.g., all numeric or all character). More formally, a vector with a dim attribute. The dim controls the number and size of dimensions.

Assignment function

Assigns values like the equal sign in Stata. The two-key sequence, “<-”, that places data or results of procedures or transformations into a variable or data set. More formally, the two-key sequence, “<-”, that gives names to objects.

Atomic object

A variable whose values are all of one type, such as all numeric or all character. More formally, an object whose components are all of one mode. Modes allowed are numeric, character, logical, or complex.

Attach

The process of adding a data set or add-on module to your path. Attaching a data set appears to copy the variables into an area that lets you use them by a simple component name like “gender” rather than by using the \$ format name like “mydata\$gender.” Done using the `attach` function. More formally, the process of adding a database to your search list. Also a function that does this.

Attributes

Traits of a data set like its variable names and labels. More formally, traits of objects such as names, class, or dim.

Class

An attribute of a variable or data set that a command used to change its options automatically. More formally, the class attribute of an object determines which method of a generic function is used when the object is an argument in the function call.

Component

Like one data set stored in a zipped set of data sets. More formally, an item in a list. The length of a list is the number of components it has.

CRAN

The Comprehensive R Archive Network at <http://cran.r-project.org/>. An Internet archive like the Statistical Software Components (SSC) Archive. Consists of a set of sites around the world called mirrors that provide R and its add-on packages for you to download and install.

Data frame

A data set. More formally, a set of vectors bound together in a list. They can be different modes or classes (e.g., numeric and character), but they must have equal length.

Database

One data set or a set of them, or an add-on module. More formally, an item on the search list or something that might be. Can be an R data file or a package.

Dim

A variable whose values are the number of rows and columns in a data set. It is stored in the data set itself. Also, a procedure that prints or sets these values. More formally, the attribute that describes the *dimensions* of an array. Also, the function that retrieves or changes that attribute.

Element

A specific value for a variable. More formally, an item in a vector.

Extractor function

A postestimation command. More formally, a function that has methods that apply to modeling objects.

Factor

A categorical variable and its value labels. Value labels may be nothing more than “1,” “2,” . . . , if not assigned explicitly. More formally, the

type of object that represents a categorical variable. It stores its labels in its `levels` attribute.

Function

A command and/or a function. When you apply it down through cases, it is just like a Stata command. However, you can also apply it across rows like a Stata function. More formally, an R program that is stored as an object.

Generic function

A command or function that has different default options or arguments set depending on the type of data you give it. More formally, a function whose behavior is determined by the class of one or more of its arguments. The class of the relevant argument(s) determines which method the generic function will use.

Index

The order number of a variable in a data set or the subscript of a value in a variable. In our practice data set `gender` is the second variable, so its index is 2. Gender is `mydata[,2]`. The first index selects rows, the second selects columns. If empty, it refers to all rows/columns. More formally, the number of the component in a list or data frame, or of an element in a vector.

Install

You install packages just like `ado` files, just once per version of R. However, you must load it from the library every time you start R. More formally, adding a package into your library.

Label

A procedure that creates variable labels. Also, a parameter that sets value labels using the `factor` or `ordered` commands. More formally, a function from the `Hmisc` package that creates variable labels. Also an argument that sets factor labels using the `factor` or `ordered` functions.

Length

The number of observations/cases in a variable, including missing values, or the number of variables in a data set. More formally, a measure of objects. For vectors, it is the number of its elements (including NAs). For lists or data frames, it is the number of its components.

Levels

The values that a categorical variable can have. Actually stored as a part of the variable itself in what appears to be a very short character variable (even when the values themselves are numbers). More formally, an attribute to a factor object that is a character vector of the values the factor can have. Also an argument to the `factor` and `ordered` functions that can set the levels.

Library

Where a given version of R stores its base packages and the add-on modules you have installed. Also a procedure that loads a package from the library into working memory. You must do that in every R

session before using a package. More formally, a directory containing R packages that is set up so that the library function can attach it. Also a function that attaches a package from the library onto your search list.

List

Like a zipped collection of data sets that you can analyze easily without unzipping. More formally, a set of objects of any class. Can contain vectors, data frames, matrices and even other lists.

Load

Bringing a data set (or collection of data sets) from disk to memory. You must do this before you can use data in R. Also the command that performs that task, like the Stata `use` command. More formally, bringing an R data file into your workspace. Also the function that performs that task.

Matrix

A data set that must contain only one type of variable, e.g. all numeric or character. More formally, a two-dimensional array; that is, a vector with a `dim` attribute of length 2.

Method

The analyses and/or graphs that a procedure will perform by default, that is different for different types of variables. The default settings for some commands depend on the scale of the variables you provide. E.g. `summary(temperature)` provides mean temperature, `summary(gender)` counts males & females. More formally, a function that provides the calculation of a generic function for a specific class of object.

Mode

A variable's type such as numeric or character. More formally, a fundamental property of an object. Can be numeric, character, logical or complex.

Modeling function

A command that performs estimation. More formally, a function that tests association or group differences and usually accepts a formula (e.g. $y \sim x$) and a `data=` argument.

Modeling objects

A model created by a modeling function.

NA

A missing value. Stands for *Not Available*. See also `NaN`.

Names

Variable names. They are stored in a character variable that is a part of a data set or variable. Since R can use an index number instead, names are optional. Also a procedure that extracts or changes variable names. More formally, an attribute of many objects that labels the elements or components of the object. Also the function that retrieves or sets this attribute.

NaN

A missing value. Stands for *N*ot a *N*umber. Something that is undefined mathematically such as zero divided by zero.

NULL

An object you can use to drop variables or values. E.g. `mydata$x |> NULL` drops the variable `x` from the data set `mydata`. More formally, `NULL` has a zero length and no particular mode. Assigning it to an object deletes it.

Numeric

A variable that contains only numbers. More formally, the atomic mode that represents real numbers. This contains storage modes `double`, `single` and `integer`.

Object

A data set, a variable or even the equivalent of a Stata command). More formally, almost everything in R. If it has a mode, it is an object. Includes data frames, vectors, matrices, arrays, lists and functions.

Object Oriented Programming

A style of software in which the output of a procedure depends on the type of data you provide it. R has an object orientation, Stata added it in version 11.

Option

A statement that sets general parameters, such as the width of each line of output. More formally, settings that control some aspect of your R session, such as the width of each line of output. Also a function that queries or changes the settings.

Package

A set of `ado`-files, and related files, such as help, for each bundled together. Like the packages at the SSC. May come with R or be written by its users. More formally, a collection of functions and, optionally, data objects.

R

A language and environment for statistical computing and graphics. An implementation of the S language.

R-PLUS

A commercial version of R. It includes a graphical user interface, context-sensitive editor and other features.

Replacement

A way to replace values. More formally, when you use subscripts on the left side of an assignment to change the values in an object. E.g. setting 9 to missing: `x[x==9] <- NA`

S

The language from which R evolved. R can run many S programs, but S cannot use R packages.

S3, S4

Used in the R help files to refer to different versions of S. The differences between them are of importance mainly to advanced programmers.

Script

The equivalent of a do file. An R program.

Search list

Somewhat like an operating system search path for R objects. More formally, the collection of databases that R will search, in order, for objects.

S-PLUS

The commercial version of S. Mostly compatible with R but will not run R packages. It includes graphical user interface and can analyze “big data” that is larger than your computer’s main memory.

Subscript

Choosing variables or values by the order in which they appear or by their name. More formally, the extraction or replacement of an object using its index or name in square [brackets].

Vector

A variable. It can exist on its own in memory or it can be part of a data set. More formally, a set of values that have the same mode, i.e. an atomic object.

Workspace

A temporary work area in which all R computation happens. Data that exists there will vanish if not saved to your hard drive before quitting R. More formally, the area of your computer’s main memory where R does all its work. Data must be loaded into it from files, and packages must be loaded into it from the library, before you can use either.

B

Comparison of Stata commands and R functions

With over 3,000 add-on packages, many containing multiple procedures, R can do almost everything that Stata can do and quite a bit more. People are releasing new packages at a rapid pace and R can give you the latest count with the following program.

The first function sets the repositories for R to search. A dialog box will prompt you so you can select them all. Next, the `available.packages()` function searches the Internet repositories for the packages that are currently available, and stores their names in `myPackageNames`. Finally, the `unique` function counts the number of unique package names.

```
> setRepositories()  
  (select all of the repositories it offers)  
> myPackageNames <- available.packages()  
> length(unique( rownames(myPackageNames) ))  
  
[1] 3175
```

So at the time of publication, there were 3,175 add-on packages.

The table below focuses only on a small but important subset of areas. Much more detailed information about R packages is available organized in Task Views at <http://cran.r-project.org/web/views/index.html>. Another site to search by task is at <http://biostat.mc.vanderbilt.edu/s/finder/finder.html>. Detailed information about most R packages is available at <http://www.r-project.org/>, choose CRAN, then choose a mirror, then choose Packages.

Table B.1. Comparison of Stata commands and functions to R functions.

| Stata command or function | R function (or package name) |
|---------------------------|---|
| * comment | # comment |
| append | rbind.fill (reshape); rbind |
| anova; oneway | aov |
| bitest | binom.test |
| ci | confint |
| clear | rm(list=ls()) |
| correlate | cor, corr.test, corr (Hmisc) |
| describe | attributes; str; ls.str; contents (Hmisc) |
| drop | mydata["varname"]<-NULL |
| edit | fix; edit |
| findit | help.search |
| help | help.start |
| help <topic> | help(<topic>) or ?<topic> |
| include | source |
| glm | glm |
| graph | plot; qplot (Hmisc); ggplot (Hmisc) |
| infile; infix | read.table; read.csv; read.FWF |
| keep | Select the variables using any technique. |
| kwallis | kruskal.test |
| label values | factor; ordered |
| label variables | label (Hmisc) |
| list | print; head; tail; |
| logistic | glm(...family=binomial) |
| mean | mean |
| merge | merge |
| nbreg | glm.nb() |
| ologit | polr |
| outsheet | write.table |
| poisson | glm(...family=poisson) |
| predict | predict |
| qnorm | qqnorm |
| recode | recode (car) |
| rename | rename (reshape); names |
| regress | lm |
| reshape | reshape; melt/cast (reshape) |
| save | save; save.image |
| set | options |
| search | help.search |
| ssc | install.packages |
| signrank | wilcox.test |
| sort | order |
| sum | sum |
| summarize | summary |
| tab <x>, gen() | factor, ordered |
| table | CrossTable (gmodels); table |
| tabulate | table |
| tabstat | tapply; by; aggregate |
| ttest | t.test |
| type | file.show |
| update | update.packages |
| use | load |
| xtgee | gee |

Automating Your R Setup

Stata has the *profile.do* file lets you automatically set options. R has a similar file called `.Rprofile`. This file is stored in your initial working directory, which you can locate with the `getwd()` function.

We will look at some useful things to automate in an `.Rprofile`.

C.1 Setting Options

In your `.Rprofile`, you can set options just as you would in R. I usually set my console width to 64 so the output fits training examples better. I also ask for five significant digits and tell it to mark significant results with stars. The latter is the default, but since many people prefer to turn that feature off, I included it. You would turn them off with a setting of `FALSE`.

```
options(width=64, digits=5, show.signif.stars=TRUE)
```

Enter `help(options)` for a comprehensive list of parameters that you can set using the `options` function.

Setting the random number seed is a good idea if you want to generate numbers that are random but repeatable. That is handy for training examples in which you would like every student to see the same result. Here I set it to the number 1234.

```
set.seed(1234)
```

The `setwd` function sets the working directory, the place that all of your files will go if you do not specify a path.

```
setwd("/myRfolder")
```

Since I included the `"/` in the working directory path, it will go to the root level of my hard drive. That works in most operating systems. Note that it must be a forward slash, even in Windows, which usually uses backward slashes in filenames. If you leave the slash off completely, it will set it to be a folder within your normal working directory.

C.2 Creating Objects

We also like to define the set of packages that we install whenever we upgrade to a new version of R. With these stored in `myPackages`, I can install them all with a single function call. For details, see Chapter 2, “Installing and Updating R.” This is the list of some of the packages used in this book.

```
myPackages <- c("car","hexbin","ggplot2",
               "gmodels","gplots", "Hmisc",
               "reshape","Rcmdr","prettyR")
```

C.3 Loading Packages

You can have R load your favorite packages automatically too. This is particularly helpful when setting up a computer to run R with a graphical user interface like R Commander. Loading packages at startup does have some disadvantages though. It slows down your startup time, takes up memory in your workspace, and can create conflicts when different packages have functions with the same name. Therefore, you do not want to load too many this way.

Loading packages at startup requires the use of the `local` function. The `getOption` function gets the names of the original packages to load and stores them in a character vector I named `myOriginal`. I then created a second character vector, `myAutoLoads`, containing the names of the packages I want to add to the list. I then combined them into one character vector, `myBoth`. Finally, I used the `options` function to change the default packages to the combined list of both the original list and my chosen packages:

```
local({
  myOriginal <- getOption("defaultPackages")

  # edit next line to be your list of favorites.
  myAutoLoads <- c("Hmisc","ggplot2")

  myBoth <- c(myOriginal,myAutoLoads)

  options(defaultPackages = myBoth)
})
```

C.4 Running Functions

If you want R to run any functions automatically, you create your own single functions that do the required steps. To have R run a function before all

others, name it “.First.” To have it run the function after all others, name it “.Last.” Notice that utility functions require a prefix of “utils::” or R will not find them while it is starting up. The `timestamp` function is one of those. It returns the time and date. The `cat` function prints messages. Its name comes from the UNIX command “cat”. It is short for *catenate* (a synonym for concatenate). In essence, we will use it to concatenate the timestamp to your console output.

```
.First <- function()
{
  cat("\n          Welcome to R!\n")
  utils::timestamp()
  cat("\n")
}
```

You can also have R run any functions before exiting the package. I have it turn off my graphics device drivers with the `graphics.off` function to ensure that no files are left open.

I like to have it save my command history in case I later decide I should have saved some of the commands to a script file. Below I print a farewell message and then save the history to a file named `myLatest.Rhistory`.

```
.Last <- function()
{
  graphics.off() #turns off graphics devices just in case.
  cat("\n\n myCumulative.Rhistory has been saved." )
  cat("\n\n Goodbye!\n\n")
  utils::savehistory(file="myCumulative.Rhistory")
}
```

WARNING: Since the `.First` and `.Last` functions begin with a period, they are invisible to the `ls` function by default. The function call

```
ls(all.names=TRUE)
```

will show them to you. Since they are functions, if you save a workspace that contains them, they will continue to operate whenever you load that workspace, even if you delete the `.Rprofile`! This can make it *very* difficult to debug a problem until you realize what is happening. As usual, you can display them by typing their names and run them by adding empty parentheses to them:

```
.First()
```

If you need to delete them from the workspace, `rm` will do it with no added arguments:

```
rm(.First,.Last)
```

C.5 Example .Rprofile

The following is the .Rprofile with all of the above function calls combined. You do not have to type this in; it is included in the book's programs and data files at <http://r4stats.com>.

```
# Startup Settings
# Place any R commands below.

options(width=64, digits=5, show.signif.stars=TRUE)
set.seed(1234)
setwd("/myRfolder")
myPackages <- c("car", "hexbin", "ggplot2",
  "gmodels", "gplots", "Hmisc",
  "reshape", "ggplot2", "Rcmdr")
utils::loadhistory(file = "myCumulative.Rhistory")

# Load packages automatically below.

local({
  myOriginal <- getOption("defaultPackages")

  # Edit next line to include your favorites.
  myAutoLoads <- c("Hmisc", "ggplot2")
  myBoth <- c(myOriginal, myAutoLoads)
  options(defaultPackages = myBoth)
})

# Things put here are done first.
.First <- function()
{
  cat("\n          Welcome to R!\n")
  utils::timestamp()
  cat("\n")
}

# Things put here are done last.
.Last <- function()
{
  graphics.off()
  cat("\n\n myCumulative.Rhistory has been saved." )
  cat("\n\n Goodbye!\n\n")
  utils::savehistory(file="myCumulative.Rhistory")
}
```

D

Example Simulation

The following examples are fully working program files. When run, they each create the same synthetic logistic regression data set consisting of 50,000 observations and a response or dependent variable with two normally distributed continuous predictors, x_1 and x_2 . The values assigned to the predictors and intercept are

$$x_1 = 0.75 \quad x_2 = -1.25 \quad \text{intercept or constant} = 3$$

The binary response, or dependent variable, is created using a binomial random number generator, based on the linear predictor, xb , which is created from the randomly generated data.

Once the data has been created, it is estimated using the GLM functions of the two software applications. For details, see Joseph Hilbe's article, Creation of Synthetic Discrete Response Regression Models [22].

D.1 Stata Example Simulation

```
* Filename: GenerateLogit.do

clear
set obs 50000
set seed 13579
gen x1 = invnorm(runiform())
gen x2 = invnorm(runiform())
gen xb = 2 + 0.75*x1 - 1.25*x2
gen exb = 1/(1+exp(-xb))
gen by = rbinomial(1, exb)
glm by x1 x2, nolog fam(bin 1)
```

D.2 R Example Simulation

```
# Filename: GenerateLogit.R

library(MASS)
x1 <- runif(50000)
x2 <- runif(50000)
xb <- 2 + .75*x1 - 1.25*x2
exb <- 1/(1+exp(-xb))
by <- rbinom(50000, size = 1, prob = exp)

lry <- glm(by ~ x1 + x2, family=binomial(link="logit"))
summary(lry)
```

References

- [1] Carlos Alzola and Jr. Frank E. Harrell. *An Introduction to S and the Hmisc and Design Libraries*. Available from <http://biostat.mc.vanderbilt.edu/RS/sintro.pdf>, 2006.
- [2] Richard A. Becker, Allan R. Wilks, Ray Brownrigg, and Thomas P Minka. *maps: Draw Geographical Maps*. Available from <http://cran.r-project.org>, 2009. R package version 2.1-0.
- [3] Burns. *S poetry*. Available from <http://www.burns-stat.com/pages/spoetry.html>, 1998.
- [4] Dan Carr, ported by Nicholas Lewin-Koh, and Martin Maechler. *hexbin: Hexagonal Binning Routines*. Available from <http://cran.r-project.org>, 2008.
- [5] John M. Chambers. *Software for Data Analysis: Programming with R*. Springer, 2008.
- [6] W. S. Cleveland. *Visualizing Data*. Hobart Press, 1993.
- [7] R core members, Saikat DebRoy, Roger Iyengar, et al. *foreign: Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...*. Available from <http://cran.r-project.org>, 2009. R package version 0.8-33.
- [8] David B. Dahl. *xtable: Export tables to LaTeX or HTML*. Available from <http://cran.r-project.org>, 2009. R package version 1.5-5.
- [9] Peter Dalgaard. *Introductory Statistics with R (Statistics and Computing)*. Springer, 2008.
- [10] Martin Elff. *memisc: Tools for Management of Survey Data, Graphics, Programming, Statistics, and Simulation*. Available from <http://cran.r-project.org>, 2009. R package version 0.95-22.
- [11] The R Foundation for Statistical Computing. R: Regulatory compliance and validation issues a guidance document for the use of r in regulated clinical trial environments, 2008. Available from <http://www.r-project.org/doc/R-FDA.pdf>.
- [12] John Fox. *car: Companion to Applied Regression*. Available from <http://cran.r-project.org>, 2009. R package version 1.2-12.

- [13] John Fox and with contributions from many others. *Rcmdr: R Commander*. Available from <http://CRAN.R-project.org>, 2009. R package version 1.4-7.
- [14] Jr. Frank E. Harrell and with contributions from many other users. *Hmisc: Harrell Miscellaneous*. Available from <http://cran.r-project.org>, 2008. R package version 3.5-2.
- [15] Michael Friendly. *Visualizing Categorical Data: Data, Stories, and Pictures*. SAS Publishing, 2000.
- [16] Robert C. Gentleman, Vincent J. Carey, Douglas M. Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, Kurt Hornik, Torsten Hothorn, Wolfgang Huber, Stefano Iacus, Rafael Irizarry, Friedrich Leisch, Cheng Li, Martin Maechler, Anthony J. Rossini, Gunther Sawitzki, Colin Smith, Gordon Smyth, Luke Tierney, Jean Y. H. Yang, and Jianhua Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.
- [17] Kenneth W. Haemer. Double scales are dangerous. *The American Statistician*, 2(3):24, 1948.
- [18] James W. Hardin and Joseph Hilbe. *Generalized Linear Models and Extensions*. Stata Press, 2001.
- [19] James W. Hardin and Joseph Hilbe. *Generalized Estimating Equations*. Chapman & Hall/CRC, 2003.
- [20] Markus Helbig and Simon Urbanek. *JGR—Java Gui for R*. Available from <http://www.rosuda.org/JGR>, 2009. R package version 1.6-3.
- [21] Joeseeph Hilbe. *Logistic Regression Models*. Chapman & Hall/CRC Press, 2009.
- [22] Joeseeph Hilbe. *Creation of Synthetic Discrete Response Regression Models*. Stata Journal, 2010. forthcoming.
- [23] Joseph Hilbe. *Negative Binomial Regression*. Cambridge University Press, 2007.
- [24] Roger Koenker. *quantreg: Quantile Regression*. Available from <http://cran.r-project.org>, 2009. R package version 4.38.
- [25] Ulrich Kohler and Frauke Kreuter. *Data Analysis using Stata, 2nd ed.* Stata Press books. Stata Press, 2009.
- [26] Max Kuhn and Steve Weaston. *odfWeave: Sweave processing of Open Document Format (ODF) files*. Available from <http://cran.r-project.org>, 2009.
- [27] Duncan Temple Lang, Debby Swayne, Hadley Wickham, and Michael Lawrence. *rggobi: Interface between R and GGobi*. Available from <http://www.ggobi.org/rggobi>, 2008. R package version 2.1.10.
- [28] Eric Lecoutre. The R2HTML package. *R News*, 3(3):33–36, 2003. Available from <http://cran.r-project.org>.

- [29] Friedrich Leisch. Sweave: Dynamic generation of statistical reports using literate data analysis. In Wolfgang Härdle and Bernd Rönz, editors, *Compstat 2002 — Proceedings in Computational Statistics*, pages 575–580. Physica Verlag, 2002.
- [30] Jim Lemon and Philippe Grosjean. *prettyR: Pretty descriptive stats*. Available from <http://cran.r-project.org>, 2009.
- [31] Thomas Lumley. *biglm: Bounded Memory Linear and Generalized Linear Models*. Available from <http://CRAN.R-project.org>, 2009. R package version 0.7.
- [32] David Meyer, Achim Zeileis, and Kurt Hornik. *vcd: Visualizing Categorical Data*. Available from <http://www.jstatsoft.org/v17/i03/>, 2009. R package version 1.2-3.
- [33] Michael N. Mitchell. *A Visual Guide to Stata Graphics, 2nd ed.* Stata Press, 2008.
- [34] Robert A. Muenchen. *R for SAS and SPSS Users*. Springer, 2008.
- [35] Paul Murrell. *R Graphics*. Chapman & Hall/CRC, 2005.
- [36] Roger B. Newson. Rsource: Stata module to run r from inside stata using an r source file, 2008. Available from <http://ideas.repec.org/c/boc/bocode/s456847.html>.
- [37] Jose C. Pinheiro and Douglas M. Bates. *Mixed Effects Models in S and S-Plus*. Springer, 2002.
- [38] R Development Core Team. *R: A Language and Environment for Statistical Computing*. Available from <http://www.R-project.org>, 2008.
- [39] R Development Core Team. *R Data Import/Export*. Available from <http://www.R-project.org>, 2008.
- [40] Deepayan Sarkar. *Lattice Multivariate Data Visualization with R*. Springer, 2007.
- [41] Deepayan Sarkar. *lattice: Lattice Graphics*. Available from <http://CRAN.R-project.org>, 2009. R package version 0.17-22.
- [42] Gregory R. Warnes. Includes R source code and/or documentation contributed by Ben Bolker, Thomas Lumley, , and Randall C Johnson. *gmodels: Various R Programming Tools for Model Fitting*. Available from <http://CRAN.R-project.org>, 2009. R package version 2.15.0.
- [43] Gregory R. Warnes. Includes R source code and/or documentation contributed by (in alphabetical order) Ben Bolker, Lodewijk Bonebakker, Robert Gentleman, Wolfgang Huber Andy Liaw, Thomas Lumley, Martin Maechler, Arni Magnusson, Steffen Moeller, Marc Schwartz, and Bill Venables. *gplots: Various R Programming Tools for Plotting Data*. Available from <http://CRAN.R-project.org>, 2009. R package version 2.7.1.

- [44] Phil Spector. *Data Manipulation with R (Use R)*. Springer, 2008.
- [45] StataCorp. *Base Reference Manual*. Stata Press, 2009.
- [46] StataCorp. *Data-Management Reference Manual*. Stata Press, 2009.
- [47] Deborah F. Swayne, Duncan Temple Lang, Andreas Buja, and Dianne Cook. GGobi: Evolving from XGobi into an extensible framework for interactive data visualization. *Computational Statistics & Data Analysis*, 43:423–444, 2003.
- [48] Duncan Temple Lang. The omegahat environment: New possibilities for statistical computing. *Journal of Computational and Graphical Statistics*, 9(3), 2000.
- [49] S. Urbanek and M Theus. High interaction graphics for r, 2003. Available from <http://rosuda.org/iPlots/iplots.html>.
- [50] W. N. Venables. Exegeses on linear models, 1998. Available from <http://www.stats.ox.ac.uk/pub/MASS3/Exegeses.pdf>.
- [51] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S, 4th ed.* Springer, 2002.
- [52] W. N. Venables, B. D. Ripley, and the R Core Development Team. *An Introduction to R*. Springer, 2007.
- [53] Hadley Wickham. Reshaping data with the reshape package. *Journal of Statistical Software*, 21(12), 2007.
- [54] Hadley Wickham. *ggplot2: An Implementation of the Grammar of Graphics*. Available from <http://cran.r-project.org>, 2008. R package version 0.8.1.
- [55] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer, 2009.
- [56] Leland Wilkinson, Graham Wills, Graham Wills, and Anonymous. *The Grammar of Graphics*. Springer, 2005.
- [57] Graham Williams. *rattle: A Graphical User Interface for Data Mining in R Using GTK*. Available from <http://rattle.togaware.com>, 2008. R package version 2.3.128.

Index

- + R prompt, 6
- + R operator, 168
- + Stata operator, 168
- .GlobalEnv environment, 295
- .RData files, 275
- .Rprofile, 15, 32
- .Rprofile R file, 507
- \$ notation, 59, 67, 128
- \$ prefixing by dataframe\$, 128
- %in% R operator, 126, 127, 149, 249
- & R operator, 180
- & Stata operator, 180
- ** R operator, 168
- ** Stata operator, 168
- * R operator, 168
- * Stata operator, 168
- * Stata operator wildcard, 119
- Stata operator (dash), 119
- .First R function, 509
- .Last R function, 509
- .Last.value. R object, 93
- / R operator, 168
- / Stata operator, 168
- : R operator, 46, 121, 278
- < R operator, 180
- < Stata operator, 180
- <- R operator, getting help on, 38
- <= R operator, 180
- <= Stata operator, 180
- = Stata operator, 180
- == R operator, 50, 125, 180
- == Stata operator, 50
- > R operator, 180
- > R prompt, 6
- > Stata operator, 180
- >= R operator, 180
- ??, searching help with, 39
- ^ R operator, 127, 168
- ^ Stata operator, 168
- 9, 99, 999 as missing values, 188, 190

- abbreviating argument names in R, 72
- accuracy, of R, 4
- adding variables to data sets, 214
- add1 R function, 474
- addition, 168
- addmargins R function, 464
- advantages, R vs. Stata, 3
- aesthetics in ggplot2 package, 386
- aggregated data sets
 - creating, 219
 - merging with original data, 222
- aggregate R function, 219
 - compared to others, 232
 - getting group means for ANOVA, 482
 - getting group variances for ANOVA, 482
- aggregation methods compared, 232
- aggregation, tabular, 224
- all keyword in Stata, 120
- analysis of variance, 481
- ANOVA, 481
- anova R function, 77
- anova Stata command, 481
- antilogarithm, 168
- AOV, 481
- aov R function, 483

- appending data sets, 210
- append** Stata command, 210
- apply** R function, 174
- ARFF, *see* Attribute-Relation File Format
- arguments, of a Stata command, 48
- arguments, of an R function, 47, 48, 70
- arrays, 63
- as.character** R function
 - converting factors, 260
- as.data.frame** R function, 249
 - for converting tables to dataframes, 462
- as.data.frame** R function, applied to
 - by objects, 230
- as.factor** R function, 256
- as.list** R function, 249
- as.logical** R function, 125
- as.matrix** R function, 173, 249
- as.numeric** R function, 180
 - converting factors, 260
- as.vector** R function, 249
- attaching data frames, how it works, 296
- attach** R function, 129
 - when creating new variables, 170
 - used with data frames, 296
 - used with files, 298
 - how it works, 296
- attribute-relation file format, 29
- attributes** R function, 292
- automating settings, 507

- bar plots
 - of means, using **barplot** R function, 326
- bartlett.test** R function, 478
- batch submission
 - Linux or UNIX, 26
 - in R, 48
 - Windows, 26
- Benford's law, 29
- biglm** package, 305
- binary version of R, 9
- binding rows, 210
- breaking a function call across two lines, 45
- Buja, Andreas, 311

- by objects, converting to data frames, 230
- by processing, 228
- by R function, 229
 - finding first or last observation per group, 238
 - compared to others, 232
- bysort** Stata command, 243
- by Stata operator, 228

- calculations, simple, 46
- car** package, 18, 205, 477
- car** package, used with ANOVA, 486
- Carr, Dan, 352
- cast** R function, 240
- cbind** R function, 60, 249
 - when creating new variable, 169
- cd** Stata command, 302
- Chambers, John, vi
- changing object class, 249
- character factors, 254
- character vector, 48
- check.names** argument, checking names
 - for validity, 269
- chisq.test** R function, 463
- chop** R function, 423
- class of an R object, 73
- class, converting, 249
- class** R function, 73, 76, 292
 - regarding factors, 254
- Cleveland, W. S., 313
- coersion
 - forcing mode change, 48
 - preventing in **data.frame** call, 58
- colClasses** argument, 99
- colMeans** R function, 174
- colSums** R function, 174
- collapse** Stata command, 219, 225
- collapsing data sets, 219
- colon R operator, 46
- columns, of a data frame, 56
- comma
 - importance when selecting observations, 141
 - importance when selecting variables, 124
- commands, compared to functions, 172
- commands, in Stata, 48
- comment attribute, 266

- `comment` R function, 266
- comments, to document programs, 69
- `complete.cases` R function, 189
- components of a list, 78
- Comprehensive R Archive Networks, 9
- concatenating data sets, 210
- conditional transformations, 180
 - multiple, 183
- `contents` R function, 40, 294
- Contents SAS procedure, 294
- continuing a function call on another line, 45
- controlling functions, 248
- converting data structures, 247
- converting factors into numeric or character variables, 260
- converting logical vector to numeric, 250
- converting numeric vector to logical, 250
- converting object class, 249
- Cook, Dianne, 311
- coordinate systems, defined in `ggplot2` package, 386
- `correlate` Stata command, 465
- correlation, 465
- `cor` R function
 - applied to a matrix, 62
- `cor.test` R function, 469
- CRAN, *see* Comprehensive R Archive Network
- CRAN R archive, 9
- `c` R function, 249
- `CrossTable` R function, 460
- cross-tabulation, 460
- crosstabular aggregation, 224
- `cumsum` R function, 458
- `cut` R function, 423
- `cut2` R function, 204
- cumulative proportions, 458
- cumulative sums, 458

- Dahl, David, 272
- dash, use of in Stata, 119
- data
 - accessing in packages, 17
 - how R stores it, 305
- data editor, 91
- Data export to CSV, 114

- `data.frame` R function, 57, 249
 - its impact on spaces in variable names, 269
- data frames, 56
 - creating, 56
 - selecting elements, 58
- data generation, 277
- data mining, 29
- data, retrieving lost, 93
- `data` R function, 17
- data set, Stata, 56
- data structures, 47
- data structures, converting, 247
- data transformations, 167
- date, displaying in output, 509
- deleting objects, *see* removing objects, 299
- `describe` R function, 13
- `describe` Stata command, 456
- `detach` R function, 13, 129
- `detail` Stata command, 456
- `digits` setting significant digits, 468
- directory
 - working directory, 68
 - getting or extracting, 68
 - setting, 68
- `display` Stata command, 46
- division, 168
- do loops, 167
- dollar notation, 59, 67, 128
- `do.call` R function, 249
 - converting by object to data frame, 232, 238
- `do` Stata command, 25
- dropping factor levels, 262
- dropping variables, 209
- `drop1` R function, 474
- `drop` Stata command, 209
- `duplicated` R function, 235
- duplicate observations, finding and removing, 234
- `duplicates` Stata command, 234

- `edit` R function
 - when renaming variables, 196
- `egen` Stata command, 167
- `egen` Stata function, 176
- elements, of a vector, 47
- Elff, Martin, 270

- Emacs text editor, 24
- environment, 295
- escape
 - unrecognized, 68
- example programs
 - for applying statistical functions, 178
 - for assigning missing values, 192
 - for by or split-file processing, 233
 - for collapsing/ aggregating data, 226
 - for collapsing/aggregating data, 226
 - for conditional transformations, 182
 - for exporting data from R to Stata, 117
 - for formatting output, 275
 - for generating data, 285
 - for graphics, traditional, 374
 - for joining/merging data sets, 217
 - for keeping and dropping variables, 210
 - for multiple conditional transformations, 185
 - for reading delimited text files, 100
 - for reading fixed-width text files
 - one record per case, 110
 - two or more records per case, 112
 - for reading text data within a program, 104
 - for recoding variables, 208
 - for removing duplicate observations, 236
 - for renaming variables, 200
 - for reshaping data, 242
 - for selecting last observation per group, 239
 - for selecting observations, 152
 - for selecting variables and observations, 162
 - for sorting data sets, 246
 - for stacking/Appending data sets, 213
 - for statistical tests, 489
 - for transforming variables, 171
 - for value labels or formats, 263
 - for variable labels, 270
 - for variable selection, 134
 - for writing comma-delimited files, 115
- ggplot2, 436
- exponentiation, 168
- Exporting data to CSV, 114
- exp** R function, 168
- exp** Stata function, 168
- extractor functions, advantages of, 77
- facets, defined in **ggplot2** package, 386
- factor levels, dropping, 262
- factor** R function, 93, 253
- factors, 51
 - character, 254
 - converting into numeric or character variables, 260
 - converting variables to, 30
 - creating
 - from character vectors, 54
 - from numeric vectors, 51
- file management, 291
- fillin** Stata command, 108
- findit** Stata command, 38
- first observation per group, selecting, 237
- Fisher's exact test, 460
- fisher.test** R function, 460
- fix** R function
 - when renaming variables, 194
- foreign** package, 113
- forest models, 30
- format** Stata command, 253
- formulas, using short variable names with, 130
- Fox, John, 18, 205, 477, 486
- Friendly, Michael, 315
- functions
 - controlling with an object's class, 73
 - controlling with arguments, 70
 - controlling with extractor functions, 75
 - controlling with formulas, 72
 - to comment out blocks of code, 70
- functions, compared to commands, 172
- functions, controlling, 248
- functions, in R, 48
- functions, writing your own, 81
- generate** Stata command, 277
- generating a data frame, 285
- generating continuous random numbers, 283
- generating data, 277

- generating integer random numbers, 281
- gen Stata command, 167
- geoms, defined in `ggplot2` package, 386
- `getOption` R function, 508
- `getwd`, 301
- `getwd` R function, 68
- GGobi, 29
- GGobi software, 311
- `ggplot` arguments and functions table, 389
- `ggplot2` package, 385
- `ggplot` R function, 386
- glob, 128
- glob, global wildcard, 150
- `glob2rx` R function, 150
- global, 128
- global, searching with, 150
- GlobalEnv environment, 295
- `glob2rx` R function, 128
- `gl` R function for generating levels of factors, 279
- Gouberman, Alex, 311
- `gplots` package, 370
- grammar of graphics, 385
- Graphical User Interfaces, 26
 - Java GUI for R, 30
 - R Commander, 26
 - Rattle, 29
- graphics
 - arguments and functions, for `ggplot2` package, 389
 - aspect ratio
 - using `ggplot` R function, 430
 - using `qplot` R function, 430
 - bar charts
 - for groups, using `ggplot` R function, 393
 - for groups, using `qplot` R function, 393
 - bar plot
 - horizontal, using `barplot` R function, 322
 - bar plots
 - of counts, using `barplot` function, 319
 - of counts, using `ggplot` R function, 389
 - of counts, using `qplot` R function, 389
 - horizontal, using `ggplot` R function, 390
 - horizontal, using `qplot` R function, 390
 - presummarized data, using `ggplot` R function, 396
 - presummarized data, using `qplot` R function, 396
 - stacked, using `barplot` R function, 322
 - stacked, using `ggplot` R function, 391
 - stacked, using `qplot` R function, 391
 - for subgroups, using `barplot` function, 324
 - using `barplot` R function, 319
 - using `ggplot` R function, 389
 - using `qplot` R function, 389
 - box plots
 - using `boxplot` R function, 368
 - using `ggplot` R function, 425
 - using `plot` R function, 368
 - using `qplot` R function, 425
 - colors, in traditional graphics, 327
 - density plots
 - using `ggplot` R function, 401
 - using `qplot` R function, 401
 - devices, 316
 - dot charts
 - using `dotchart` R function, 333
 - using `ggplot` R function, 397
 - using `qplot` R function, 397
 - dual-axes plots
 - using `plot` R function, 366
 - equations, adding to, 371
 - error bar plots
 - using `ggplot` R function, 428
 - using `plotmeans` R function, 370
 - using `qplot` R function, 428
 - functions
 - to add elements to existing traditional plots, 347
 - GGobi software, 311
 - `ggplot2` example programs, 436
 - `grid` graphics system, 316
 - hexbin plots

- using `ggplot` R function, 414
- using `hexbin` R function, 352
- using `qplot` R function, 414
- histograms
 - using `ggplot` R function, 400
 - using `hist` R function, 333, 334
 - using `qplot` R function, 400
- histograms, with density overlaid
 - using `ggplot` R function, 401
 - using `qplot` R function, 401
- histograms overlaid
 - using `ggplot` R function, 404
 - using `hist` R function, 337
 - using `qplot` R function, 404
- histograms, stacked
 - using `ggplot` R function, 403
 - using `hist` R function, 336
 - using `qplot` R function, 403
- interaction plots
 - using `interaction.plot` R function, 370
- `iplots` package, 311
- `jpeg` device driver, 317
- labels, in `ggplot2` graphics, 399
- `lattice` package, 312, 316
- legends, in traditional graphics, 327
- line plots
 - using `ggplot` R function, 408
 - using `plot` R function, 347
 - using `qplot` R function, 408
- logarithmic axes
 - using `ggplot` R function, 430
 - using `qplot` R function, 430
- missing values' effect on, 387
- mosaic plots
 - using `mosaicplot` function, 324
 - using `plot` function, 324
- multiple plots on a page
 - in `ggplot2`, 431
 - in traditional graphics, 330
- normal QQ plot
 - using `ggplot` R function, 405
 - using `qplot` R function, 405
 - using `qqnorm` R function, 342
 - using `qq.plot` R function, 341
- overview, 311
- parameters
 - for both `par` and traditional graphics functions, 347
 - demo plot in traditional graphics, 373
 - in `ggplot2` graphics, 435
 - in `ggplot2` package, 433
 - to set or query using only the `par` function in traditional graphics, 345
 - in traditional graphics, 330, 372
 - for traditional high-level graphics functions, 343
- `pdf` device driver, 317
- `pictex` device driver, 317
- pie charts
 - using `ggplot` R function, 392
 - using `pie` R function, 331
 - using `qplot` R function, 392
- plots by group or level, using `ggplot` R function, 394
- plots by group or level, using `qplot` R function, 394
- `png` device driver, 317
- `quartz` device driver, 317
- `rattle` package link to GGobi, 311
- `rggobi` package, 311
- saving plots to a file
 - in `ggplot2` package, 433
- scatter plot for correlation, 465
- scatter plot matrices
 - using `ggplot` R function, 424
 - using `plot` R function, 364
 - using `qplot` R function, 424
- scatter plots
 - using `ggplot` R function, 408
 - using `plot` R function, 347
 - using `qplot` R function, 408
- scatter plots with confidence ellipse
 - using `plot` R function, 356
- scatter plots with confidence and prediction intervals
 - using `plot` R function, 357
- scatter plots, with density contours
 - using `ggplot` R function, 413
 - using `qplot` R function, 413
- scatter plots, faceted by group
 - using `ggplot` R function, 422
 - using `qplot` R function, 422
- scatter plots, with fit lines
 - using `ggplot` R function, 415
 - using `qplot` R function, 415

- scatter plots by group or level
 - using `plot` R function, 354
- scatter plots, with jitter
 - using `ggplot` R function, 410
 - using `plot` R function, 350
 - using `qplot` R function, 410
- scatter plots, with labels as points
 - using `ggplot` R function, 420
 - using `plot` R function, 362
 - using `qplot` R function, 420
- scatter plots, for large data sets
 - using `ggplot` R function, 411
 - using `plot` R function, 350
 - using `qplot` R function, 411
- scatter plots, with linear fits by group
 - using `ggplot` R function, 422
 - using `plot` R function, 353
 - using `qplot` R function, 422
- scatter plots, with reference lines
 - using `ggplot` R function, 416
 - using `qplot` R function, 416
- scatter plots, with reference lines added
 - using `plot` R function, 352
- scatter plots, with symbols by group
 - using `ggplot` R function, 421
 - using `qplot` R function, 421
- strip charts
 - using `stripchart` R function, 342
- strip plots
 - using `qplot` R function, 405
 - using `ggplot` R function, 405
- systems, compared to procedures, 315
- titles, in `ggplot2` graphics, 399
- titles, in traditional graphics, 327
- traditional, 312
- `windows` device driver, 317
- `win.meta` device driver, 317
- `x11` device driver, 317
- `grid` graphics system, 316
- Grosjean, Philippe, 12, 177, 270
- `gsort` Stata command, 243
- GUI, *see* Graphical User Interfaces

- Harrell, Frank, 10, 40, 113, 204, 266, 294, 454
- `head` R function, 17, 292
- help
 - for R data sets, 42
 - for R functions, 37
 - for R functions that call other functions, 40
 - for R packages, 41
 - via mailing lists, 42
- `help.search` R function, 38
- `help.start` R function, 38
- `hexbin` package, 352
- `hexbin` R function, 352
- history file in R, 305
- history, of R work, 21, 23, 24
- `hist` R function, 466
- `Hmisc` package, 266
- `Hmisc` package, 10, 13, 40, 113, 204, 294, 454
- Hornik, Kurt, 315
- HTML output, 272

- `ifelse` R function, 180
- `if` Stata command, 119
- IF Stata option, 139
- importing data
 - from Stata, 113
- imputation, 30
- `include` Stata command, 104
- index values, generating from variable names, 132
- indexing, 120
 - data frames, 58
 - lists, 66
 - matrices, 62
 - vectors, 50
- indices, *see* indexing
- `input` Stata command, 102
- `installed.packages` R function, 12
- `install.packages` R function, 15
- `in` Stata command, 119
- `in` Stata operator, 127, 145
- `in` Stata option, 149
- `interaction.plot` R function, 370
- interactive mode in R, 48
- `iplots` package, 311
- `I()` R function, 387
- `is.data.frame` R function, 248
- `is.na` R function, 177
- `is.vector` R function, 248

- Java GUI for R, 30
- `jpeg` R function, 317

- JGR, 30
- join, full outer, 217
- join, inner, 217
- joining data sets, 214
- journal, Stata file, 23
- keeping variables, 209
- keep** Stata command, 209
- Koenker, Roger, 275
- kruskal.test** R function, 487
- Kuhn, Max, 275
- kwallis** Stata command, 487
- label data** Stata command, 42
- label** Stata command, 266
- lappy** R function, 175
- last observation per group, selecting, 237
- LaTeX output, 272
- latex.table** R function, 275
- lattice** package, 312, 316
- Lawrence, Michael, 311
- Lecoutre, Eric, 275
- Leisch, Friedrich, 275
- Lemon, Jim, 12, 177, 270
- length
 - of data frame components, 56
- length** R function, 176
- length** Stata function, 176
- levene.test** R function, 482
- Levine test, 477
- Lewin-Koh, Nicholas, 352
- library**
 - loading packages with, 12
 - showing installed packages, 12
- linear regression, 470
- linesize, controlling in R, 46
- linestyle** Stata option, 315
- Linux, R versions for, 9
- list** R function, 249
- lists, 63
 - creating, 63
 - related to data frames, 63
 - selecting elements, 66
- list** Stata command, 37, 48
- lm** R function, 470
- loadhistory** R function, 21, 23, 24, 305
- loading data subsets, 162
- load** R function, 21, 22, 24, 162, 302
- local** R function, 508
- log file in Stata, 305
- logarithm, 167
- logical operators, 180
- log** R function, 168
- log10** R function, 168
- log** Stata function, 168
- log10** Stata function, 168
- long format data sets, 240
- ls** R function, 67, 291
 - applied to search path, 295
- Lumley, Thomas, 305
- Mac OS, R versions for, 9
- macro substitution, 109
- macros, writing in R, 81
- Maechler, Martin, 352
- managing files and workspace, 291
- Mann-Whitney U test for independent groups, 479
- mapply** R function, 178
- margins, adding to cross-tabulations, 464
- masked objects, 13
- masking objects, 297
- MASS** package, 474
- matching data sets, 214
- mathematical operators, 168
- matrices, *see* matrix
- matrix, 56, 60
 - creating, 60
 - selecting elements, 62
 - stored in a data frame or list, 62
- matrix** R function, 60, 249
- McNemar test, 460
- mcnemar.test** R function, 460
- mean** R function, 71
 - applied to a matrix, 62
 - getting group means for ANOVA, 482
 - help file, 39
- median** R function, 176
- melt** R function, 240
- memisc** package, 270
- merge_all** R function, 217
- merge** R function, 216
- merge** Stata command, 214
- merging data sets, 214
- merging, data with aggregates, 222

- merging more than two data frames at once, 217
- methods, 248
- Meyer, David, 315
- Microsoft Word, 275
- Minard, Charles Joseph, 313
- missing values, 48, 49, 186
 - 9, 99, 999 as missing codes, 190
 - finding observations with none, 189
 - substituting means, 188
- mode, 48
 - character, 48
- mode of an R object, 73
- model selection, stepwise... , 474
- multiplication, 168
- Murrell, Paul, 312, 316

- NA, not available or missing, 48, 49, 186
- na.rm argument, 49
- names R function, 123, 127, 197, 292
 - used with `which`, 133
- names, rules for in R, 45
- na.omit R function, 189
- na.strings argument for reading
 - missing values, 187
- ncol R function, 123
- nesting function calls, 167
- Newson, Roger, 3
- nonparametric analysis of variance, 487
- NULL R object, 209

- objects, 69
 - changing class, 249
- objects R function, 67, 291
- observations
 - advanced selection methods, 151
 - saving selections to data frame, 152
 - selecting all, 140
 - selecting in Stata, 139
 - selecting using `subset` function, 150
 - selecting using index number, 140
 - selecting using logic, 145
 - selecting using row names, 143
 - selecting using string search, 148
 - selection example R program, 153
 - selection example Stata program, 153
- observations, renaming, 200
- observations, selecting, 139
- observations, selecting first or last per group, 237
- observations, StataS, 56
- ODBC, *see* Open Database Connectivity
- odfWeave package, 275
- ODS, *see* Output Delivery System
- OMS, *see* Output Management System
- oneway Stata command, 481
- Open Database Connectivity, 29
- Open Document Format, 275
- OpenOffice, 275
- operators, logical, 180
- operators, mathematical, 168
- options R function
 - at startup, 507
 - setting significant digits, 468
- options, setting automatically, 507
- options Stata command, 46
- ordered R function, 257
- order R function, 243
- outliers, 30
- outreg2 Stata command, 272
- outtex Stata command, 272

- packages
 - accessing data in, 17
 - conflicts among, 13
 - detaching, 15
 - installing, 10
 - uninstalling, 15
- pairwise.t.test R function, 484
- pairwise.wilcox.test R function, 488
- paste R function, 124, 199
- path, 295
- pdf R function, 317
- pictex R function, 317
- pie charts
 - using `ggplot` R function, 399
 - using `qplot` R function, 398
- plot R function, 77
 - for correlations, 465
 - testing linearity, 470
 - used to plot ANOVA diagnostics, 486
 - used to plot multiple comparisons, 485
- png R function, 317
- postestimation Stata commands, 470
- predict R function, 77, 475

- prettyR** package, 12, 13, 177, 270
- printing
 - components of a list, 81
 - printing the contents of **lm** objects, 78
- print** R function, 37, 48, 292
- profile.do** Stata file, 507
- prompt characters, R, 6
- proportions, row, column, total, 463
- prop.table** R function, 458, 463
- pwcorr** Stata command, 465

- qplot** arguments and functions table, 389
- qplot** R function, 385
- qqnorm** R function, 466
- quantreg** package, 275
- quartz** R function, 317
- quickplot** R function, 386

- R Commander, 26
- R-help mailing list, 42
- random number generation
 - continuous, 283
 - data frames, 285
 - integer, 281
- range** R function, 222, 230
- ranksum** Stata command, 479
- rattle** package
 - for data mining, 311
 - link to GGobi, 311
- rbind** R function, 210, 249
 - converting by object to data frame, 231
- rbind.fill** R function, 212
- rcorr** R function, 466
- reading data within a program, 102
- reading delimited text files, 93
- reading Stata data sets, 113
- reading text files
 - skipping columns, 99
 - two or more records per case, 111
- reading text files, tab-delimited, 96
- read.table** R function, 94, 109
 - reading factors, 254
 - setting missing values, 187
- recode** R function, 205
- recode** Stata command, 204
- records, 56
- regression, 30
 - linear, 470
- regress** Stata command, 470
- regular expression, when searching for
 - object names, 292
- reliability, of R, 4
- remove.packages** R function, 15
- remove** R function, 69, 299
- removing objects, 69, 299
- rename** R function, 194
- rename** Stata command, 194
- renaming observations, 200
- renaming variables, 194
 - by column name, 198
 - by index, 197
 - many sequentially numbered names, 199
- repeated measures data sets, 240
- repositories, selecting, 15
- rep** R function, 280
- reshape** package, 194, 212, 217, 240
- reshape** Stata command, 240
- reshaping data sets, 240
- resid** R function, 77
- retrieving lost data, 93
- rggobi** package, 311
- R2HTML** package, 275
- Ripley, B.D., 474
- rm** R function, 69, 299
 - in relation to dropping variables, 209
- rnorm** R function, 283
- rolling up data, 219
- rounding off, 168
- rounding off decimals, 458
- round** R function, 168, 458
 - applied to cross-tabulations, 464
- round** Stata function, 168
- rowMeans** R function, 174
- row names, 58
 - changing, 200
 - setting in data editor, 92
- row.names** R function, 58, 92, 149, 151, 200
- rownonmiss** Stata function, 176, 177
- RSiteSearch** R function, 43
- rows, of a data frame, 56
- rowSums** R function, 174
- rpart** R function, 204
- runif** R function, 283
- running R

- in batch mode, 25
- interactively, 19
- sapply** R function, 176
- Sarkar, Deepayan, 312, 313
- savehistory, 305
- savehistory** R function, 21, 23, 24
- save.image** R function, 21, 22, 24, 69, 302
 - when creating new variables, 171
- saveold** Stata command, 113
- save** R function, 69, 162, 302
 - when creating new variables, 171
- saving data subsets, 162
- saving selected variables to a data frame, 133
- saving your work, 67
- scales, defined in **ggplot2** package, 386
- scientific notation, 453
- scipen** R option, 453
- sd** R standard deviation function, 176
- sd** Stata function, 176
- search path, 295
- searching for R packages or functions, 43
- search** R function, 295
- selecting
 - data frame components, 58
 - list components, 66
 - matrix elements, 62
 - vector elements, 50
- selecting observations, 139
 - all, 140
 - example R program, 153
 - example Stata program, 153
 - saving to data frame, 152
 - in Stata, 139
 - using **subset** function, 150
 - using advanced methods, 151
 - using index numbers, 140
 - using logic, 145
 - using row names, 143
 - using string search, 148
- selecting variables
 - all variables, 120
 - saving to a data frame, 133
 - in Stata, 119
 - using \$ notation, 128
 - using **attach** R function, 129
 - using **subset** R function, 131
 - using **with** R function, 130
 - using column name, 123
 - using formulas, 130
 - using index number, 120
 - using list index, 132
 - using logic, 124
 - using R's **with** function, 130
 - using simple name, 129
 - using string search, 126
- selecting variables and observations, 157
- seq** R function, 278
- sessionInfo** R function, 43
- setRepositories** R function, 16
- set.seed** R function, 278, 281
- setwd** R function, 68, 302
- shortcuts, use in Windows, 304
- signrank** Stata command, 480
- signtest** Stata command, 480
- sink** R function, 24
- sorting
 - controlling order, 245
 - dangers of, 56
 - data frames, 243
 - on more than one variable, 245
- sort** Stata command, 243
- source code for R, 9
- source** R function, 104
- split-file processing, 228
- split** R function, 210
- sqrt** R function, 168
- sqrt** Stata function, 168
- square root, 168
- SSC Stata archive, 9
- stacking data sets, 210
- star wildcard operator in Stata, 119
- stata.get** R function, 113, 114
- Statistical Software Components
 - Archive, 9, 10
- statistics
 - analysis of variance, 481
 - aov** R function, 483
 - Bartlett test for equality of variance, 478
 - chisq.test** R function, 463
 - correlation, 465
 - cor.test** R function, 469
 - cross-tabulation, 460
 - descriptive, 454

- equality of variance, 477
- Fisher's exact test, 460
- frequencies, 454
- group means, 482
- group variances, 482
- homogeneity of variance for ANOVA, 482
- Kruskal-Wallis test, 487
- Levine test for homogeneity of variance, 477
- Mann-Whitney U test for independent groups, 479
- McNemar's test, 460
- model selection, stepwise... , 474
- `pairwise.t.test` R function for multiple comparisons, 484
- `pairwise.wilcox.test` R function, 488
- percentages, 458
- postestimation in Stata, 470
- proportions, 458, 463
- `prop.table` R function, 458
- `rcorr` R function, 466
- regression
 - linear, 470
 - predicting with new data, 475
- rounding off decimals, 458
- `summary` R function, compared to `describe` R function, 456
- sums of squares, different types, 486
- t-test
 - for independent groups, 476
 - for paired groups, 478
- `TukeyHSD` R function for multiple comparisons, 484
- Wilcoxon rank sum test for independent groups, 479
- Wilcoxon Signed-Rank Test for paired groups, 480
- statistics, defined in `ggplot2` package, 386
- `stepAIC` R function, 474
- `step` R function, 474
- stepwise model selection, 474
- `str` R function, 292
- `stringsAsFactors` argument, 58
- `stringsAsFactors` argument to `data.frame` R function, 254
- `stringsAsFactors` argument to `read.fwf` R function, 109
- subscripting, 120
- `subset` R function, 133, 157, 477
 - selecting both variables and observations, 157
 - selecting observations, 150
 - selecting variables, 131
- subtraction, 168
- summarization methods compared, 232
- summarized data sets
 - creating, 219
 - merging with original data, 222
- `summarize` Stata command, 120, 454, 456
- `sum` R function
 - when used for counting, 177
- sums of squares, different types, 486
- Support Vector Machines, 30
- SVM, *see* Support Vector Machines
- `Sweave` R function, 275
- Swayne, Deborah F., 311
- `symbolstyle` Stata option, 315
- t-test
 - for independent groups, 476
 - for paired groups, 478
- `table` R function, 49, 225
 - applied to a matrix, 62
 - compared to others, 232
 - for cross-tabulation, 462
- tables
 - adding row/column totals, 464
 - calculating chi-squared on, 463
 - converting to percents, 464
 - converting to proportions, 463
- tabular aggregation, 224
- `tabulate` Stata command, 49, 224, 454, 456, 459, 460, 462
- `tail` R function, 17, 292
- `tapply` R function, 221
 - compared to others, 232
- technical support, 42
- technical support, for R, 4
- Temple Lang, Duncan, 311
- `textConnection` R function, 104
- text files, tab-delimited, 96
- Theus, Martin, 311
- time, displaying in output, 509

- `timestamp` R function, 509
- transcript of R work, 24
- transformations, 167
 - multiple conditional, 183
 - single conditional, 180
- `transform` R function, 168, 297
- tree models, 30
- `t.test` R function
 - for independent groups, 476
 - for paired groups, 478
- `tttest` Stata command, 476, 478
- `TukeyHSD` R function, 484
- Type III sums of squares, 486
- type, of a variable, 48

- `unclass` R function, 78, 249
- undeleting data using `.Last.value` R function, 93
- uninstalling
 - R, 15
 - an R package, 15
- `unique` R function, 234
- `unlist` R function, 249
- `update.packages` R function, 14
- updating R installation, 14
- Urbanek, Simon, 311

- `valid.n` R function, 177
- value labels, 253
- `value label` Stata command, 253
- variables
 - saving selection to a data frame, 133
 - selecting all variables, 120
 - selecting in Stata, 119
 - selecting using `$` notation, 128
 - selecting using `attach` R function, 129
 - selecting using `subset` R function, 131
 - selecting using `with` R function, 130
 - selecting using column name, 123
 - selecting using formulas, 130
 - selecting using index number, 120
 - selecting using list index, 132
 - selecting using logic, 124
 - selecting using simple name, 129
 - selecting using string search, 126
- variables, Stata, 56
- `var` R function, 176
 - getting group variances for ANOVA, 482
- `var.test` R function, 478
- `vignette` R function, 44
- vector arithmetic, 167
- vectors, 47
 - analyzing, 49
 - creating, 47
 - selecting elements, 50
- Venable, W.N., 474
- Venables, W.N., 486
- visualization, 29

- Warnes, Gregory, 370
- Weaston, Steve, 275
- `which` R function, 151, 159, 184, 249
 - used with `names`, 133
- Wichtrey, Tobias, 311
- Wickham, Hadley, 194, 212, 240, 311, 313, 387
- wide format data sets, 240
- width, of output in R, 46
- Wilcoxon test
 - for paired groups, 480
- Wilcoxon test for independent groups, 479
- wildcard, 128
 - searching with, 150
- Williams, Graham, 29, 311
- `windows` R function, 317
- `win.meta` R function, 317
- Windows, R versions for, 9
- `within` R function, 168
- `with` R function, 130
 - compared to `within`, 168
- working directory, 68
- working directory, getting and setting, 301
- workspace, 67
- workspace management, 291
- `write.csv` R function, 114
- `write.foreign` R function, 116

- `x11` R function, 317
- `xtable` function, 272
- `xtable` package, 272

- Zeileis, Achim, 315