eference

nson@gci.net)

ference card in-

thin emacs, use
SS properly in-
ands have many
Help section for

q()
q("yes")
<-
d) _
->
LL packageX
m1[,2]
m1[,2:5]
c(2,3,4,5)]
m1$a1
ibrary(mva)

l with R, or
, for an ex-
earch site at
al , a FAQ
nd a long list of
Rtips.html.

Getting Help

Getting help with command xyz `help(xyz)`

Get help with command xyx `?xyz`

Start browser help `help.start()`

Get help for a specific package `help(package=mva)`

Unix – apropos – like search of the help files for "biplot" `help.search("biplot")`

Search for object with name slime. `apropos("slime")`

Run examples of legend command `example(legend)`

Input and Output

Run the commands in file1 `source(file1)`

Read data from file1 `read.table("file1"`

Open a spreadsheet to inspect or modify variable z. The variable z must already exist. `data.entry(z)`

Read a vector from file1 `scan("file1")`

Send all output to file1, until the command sink() is entered. `sink("file1")`

Write x to file1 `write(x,"file1")`

Write a table to file f1. The variable x must be a data.frame. `write.table(x,"f1")`

Managing Variables and Objects

Most of these functions, such as dimnames(), can be assigned to. This is particularly useful when assigning a class to an object e.g., `class(x1) <- "newclass"`.

Edit a data frame df `df <- edit(df)`

The edit command will open a spreadsheet–like window in which you may view and edit many (not quite all) data frames. It is spreadsheet–*like*, but is not a full–featured spreadsheet. **Notice:** Do *not* close the spreadsheet window via the "kill-button" on the window frame. This will cause R to segfault. You *must* use the quit button.

Put the variable x1 in the search path. `attach(x1)`

Remove the variable x1 from the search path. `detach(x1)`

List all active objects. `ls()`

Remove object x1. `rm(x1)`

Remove all active objects. `rm(list=ls())`

© 2004 Nels Tomlinson Permissions on back. v1.0d

Return dimensions of x1 (often a matrix). `dim(x1)`

Return the names of the dimensions of x1. `dimnames(x1)`

Get the length of x1 (usually a vector). `length(x1)`

Make the vector 1,2,3 `1:3`

Make the vector 1,2,3 `c(1,2,3)`

Make the vector 1,5,9 `seq(1,10,by=4)`

Repeat the vector x1 n times. `rep(x1,n)`

Bind columns x1,x2,x3 into a matrix. `cbind(x1,x2,x3)`

Bind rows x1,x2,x3 into a matrix. `rbind(x1,x2,x3)`

Let x1,x2,x3 be vectors of length n (note, that's not the same as nx1 matrices). Then `dim(cbind(x1,x1,x1))` returns n by 3, and `dim(rbind(x1,x1,x1))` returns 3 by n.

Convert the vectors x1,x2,x3 to a data.frame. Notice that they do not need to be of the same type. `data.frame(x1,x2,x3)`

Convert to factor `as.factor()`

Convert to matrix `as.matrix()`

Convert to vector `as.vector()`

Returns logical TRUE if x1 is a factor `is.factor()`

Returns logical TRUE if x1 is a matrix `is.matrix()`

Returns logical TRUE if x1 is a vector `is.vector()`

Transpose the rows and columns of a matrix. `t()`

Returns vector with indices of x1 where x1=a1. `which(x1==a1)`

Plotting

Plot y1 against x1, as line. `plot(x1,y1,type='l')`

Draw line on existing plot. Many arguments. `abline()`

specify titles `title(main="",xlab="",ylab="")`

Place legend `legend(x,y,legend,bty)`

Open a new plot device `x11()`

Switch to device 3 `dev.set(3)`

Turn off current device. `dev.off()`

Open a postscript device `postscript()`

© 2004 Nels Tomlinson Permissions on back. v1.0d

`par` can be used to set or query graphical parameters. Some useful ones include `mfrow`, `new`. `par()` returns all the graphical parameters.

Control flow

For loop. x1 may be any vector. `for(i1 in x1)`

if-else `if(condition1)consequence else alternate`

While Loop. `while(condition1){expressions}`

Switch, like case. `switch(expression,...)`

Break out of a loop `break`

Notice that == is a logical operator, = is used to pass arguments by name, and <– is used as assignment operator. Using the single `&` or | does element–wise operations. Using double `&&` or || evaluates only the first element of each vector. Use this double form for program flow control.

Logical AND `&, &&`

Logical OR `|, ||`

Logical negation `!`

Exclusive or, element–wise `xor`

Arithmetic, etc.

The operators +,–,* and / provide the four basic arithmatic operations. Quoting from R-Help: *They return numeric vectors containing the result of the element by element operations. The elements of shorter vectors are recycled as necessary (with a warning when they are recycled only fractionally).* This recycling is standard behavior for R, and makes for very intuitive programming.

Matrix multiplication `%*%`

Integer division `%/%`

Raise y1 to x1 power `y1 ^ x1`

Modulus: x1 mod y1 `x1 %% y1`

Square root `sqrt()`

Compute derivatives of simple expressions, symbolically. `deriv(expr,name)`

General-purpose optimization based on the Nelder Mead, quasi-Newton and conjugate-gradient algorithms. It includes an option for box-constrained optimization. `optim()`

Statistics

`max()`, `min()`, `median()`, `sum()` and `var()` do what their names indicate.

© 2004 Nels Tomlinson Permissions on back. v1.0d

data frame x1.

sort(x1)
x1. Rank retuns
rank(x1)
a vector, as the

4,59,26)))

s, and can be
and. They may
mmand R CMD
s, and you must
ters in the pack-
ackages marked
ory, and should

ins all the com-
base
accompany the
lication boot
Found in VR.

usseeuw et al.)

ctest
AS, SPSS, Stata,
foreign
us functions.

tice package.

(and density es-
Wand, M.P. and
KernSmooth
S-Plus trellis.

s to accompany
d Statistics with
MASS
estimation and
mgcv

on back. v1.0d

Classical multivariate analysis `mva`

** Linear and nonlinear mixed effects models `nlme`

** Feed-forward neural networks and multinomial log-linear models. Found in VR. `nnet`

Numerical solvers for solving ordinary differential equations. `odesolve`

** Recursive partitioning and regression trees `rpart`

Binary file stream support functions. A set of routines for binary access to files. `Rstreams`

Make a 3–D scatterplot. `scatterplot3d`

** Functions for kriging and point pattern analysis. Found in VR. `spatial`

** survival analysis: descriptive statistics, two-sample tests, parametric accelerated failure models, Cox model. Delayed entry (truncation) allowed for all models; interval censoring for parametric models. `survival`

Package of time series functions. `ts`

Package for time series analysis. `tseries`

The sections below contain partial listings of the functions to be found in some of these packages.

boot

Bootstrap resampling. `boot`

Bootstrap resampling of arrays `boot.array`

Empirical influence values, via jackknife `empinf`

Confidence envelopes for curves. `envelope`

Importance sampling estimates of the central moment. `imp.moments`

Importance sampling estimates of tail probability. `imp.prob`

Importance sampling estimates of quantiles. `imp.quantile`

Simplex method for linear programming. `simplex`

ctest

Exact Binomial test. `binom.test`

Pearson's Chi–squared test for count data `chisq.test`

Test for zero correlation `cor.test`

Fisher's Exact Test for Count Data `fisher.test`

Kruskal-Wallis Rank Sum Test `kruskal.test`

Pairwise t tests `pairwise.t.test`

Power calculations for one and two sample t tests `power.t.test`

© 2004 Nels Tomlinson Permissions on back. v1.0d

Shapiro-Wilk Normality Test `shapiro.test`

Student's t–test `t.test`

F Test to Compare Two Variances `var.test`

Wilcoxon Rank Sum and Signed Rank Tests `wilcox.test`

foreign

Lookup information about a SAS XPORT format library `lookup.xport`

Read Stata binary files `read.dta`

Reading a Minitab Portable Worksheet `read.mtp`

Read an SPSS data file `read.spss`

Read a SAS XPORT format library `read.xport`

Write files in Stata binary format `write.dta`

gregmisc

Produce a boxplot annotated with the number of observations. `boxplot.n`

Compute and display confidence intervals for model estimates. `ci`

Plot error bars. `plotCI`

KernSmooth

Compute a Binned Kernel Density Estimate `bkde`

Compute a 2D Binned Kernel Density Estimate `bkde2D`

Select a Bandwidth for Kernel Density Estimation `dpik`

lattice

From the manual: *Grid (and hence Lattice) graphics do not mix with conventional R Graphics. The way to use both is to enclose any Grid graphics commands by calls to `grid.start` and `grid.stop`. Note that `library(lattice)` automatically calls grid.start.*

Draw bar plots `barchart`

Draw box and whisker plots `bwplot`

Draw kernel density plots `densityplot`

Draw histograms `histogram`

Draw scatterplot matrices `splom`

Draw strip plots (One dimensional scatter plots) `stripplot`

Tukey mean–difference plot `tmd`

Draw conditioning plots `xyplot`

MASS

Adaptive numerical integration `area`

Box-Cox transformations for linear models `boxcox`

© 2004 Nels Tomlinson Permissions on back. v1.0d

Try all one-term deletions from a model `dropterm`

Performs stepwise model selection by exact AIC. `stepAIC`

Calculate the variance-covariance matrix for a fitted model object. `vcov`

Rstreams

Read an S-PLUS Binary File `data.restore`

Open or Close a Stream `openstream`

To read binary data from a stream, use `readint()`, `readfloat()`, `readcomplex()`, `readchar()`, `readlines()` for the appropriate data type. To write binary data to a stream, use `writeint()`, etc.

tseries

Augmented Dickey–Fuller test `adf.test`

Fit ARMA Models to Time Series `arma`

Computes and prints the BDS test statistic for the null that x is a series of i.i.d. random variables. `bds.test`

Fit GARCH Models to Time Series `garch`

Read Time Series Data `read.ts`

Copyright © 2004 Nels Tomlinson
Maintainer: Nels Tomlinson nelstomlinson@gci.net
v1.0d for R Version 1.3.1 2004
Permission is granted to make and distribute copies or derivatives of this card provided the copyright notice and this permission notice are preserved on all copies.