R Reference Card

by Tom Short, EPRI PEAC, tshort@epri-peac.com 2004-11-07
Granted to the public domain. See www.Rpad.org for the source and latest
version. Includes material from R for Beginners by Emmanuel Paradis (with
permission).

Getting help
Most R functions have online documentation.
help(topic) documentation on topic
?topic id.
help.search("topic") search the help system
apropos("topic") the names of all objects in the search list matching
the regular expression "topic"
help.start() start the HTML version of help
str(x) display the internal "str"ucture of an R object
summary(a) gives a "summary" of a, usually a statistical summary but it is
generic meaning it has different operations for different classes of a
ls() show objects in the search path; specify pat="pat" to search on a
pattern
ls.str() str() for each variable in the search path
dir() show files in the current directory
methods(a) shows S3 methods of a
methods(class=class(a)) lists all the methods to handle objects of
class a

Input and output
load() load the datasets written with save
data(x) loads specified data sets
library(x) load add-on packages
read.table(file) reads a file in table format and creates a data
frame from it; the default separator sep="" is any whitespace; use
header=TRUE to read the first line as a header of column names; use
as.is=TRUE to prevent character vectors from being converted to fac-
tors; use comment.char="" to prevent "#" from being interpreted as a
comment; use skip=n to skip n lines before reading data; see the
help for options on row naming, NA treatment, and others
read.csv("filename",header=TRUE) id. but with defaults set for
reading comma-delimited files
read.delim("filename",header=TRUE) id. but with defaults set for
reading tab-delimited files
read.fwf file, widths, header=FALSE, sep="", as.is=FALSE
read a table of fixed width/formatted data into a 'data.frame'; widths
is an integer vector, giving the widths of the fixed-width fields
save(file,...) saves the specified objects (...) in the XDR platform-
save.image(file) saves the specified objects (...) in the XDR platform-

Data creation
c(...) generic function to combine arguments with the default forming a
vector; with recursive=TRUE descends through lists combining all
elements into one vector
from:to generates a sequence: "2:5" has operator priority; 1:4+1 is "2,3,4,5"
seq(from,to) generates a sequence by specifying increment; length-
specifies desired length
seq(along=x) generates 1, 2, ..., length(along); useful for for
loops
rep(x, times) replicate x times; use each= to repeat "each" element
of x each times; rep(c(1,2,3),2) is 1 2 2 1 2 2 3 3
data.frame(...) create a data frame of the named or unnamed
arguments: data.frame(x=1:4,y=c("a","b","c","d"),n=10); shorter
vectors are recycled to the length of the longest
list(...) create a list of the named or unnamed arguments:
list(a=c(1,2),b="hi",c=c(1:3));
array(x,dim=) array with data x; specify dimensions like
dim=c(3,2): elements of x recycle if x is not long enough
matrix(x,nrow=,ncol=) matrix; elements of x recycle
factor(x,levels=) encodes x as a vector as a factor
gl(k,nrow=n,ncol=) generate levels (factors) by specifying
the pattern of their levels; k is the number of levels, and n is the
number of replications
expand.grid(...) a data frame from all combinations of the supplied vec-
tors or factors
rbind(...) combine arguments by rows for matrices, data frames, and others
cbind(...) id. by columns

Slicing and extracting data
Indexing vectors
x[1] first element
x[2:5] all but the 1st element
first n elements
elements from a to b
first n elements
all elements greater than 3
all elements between 3 and 5
all elements between 3 and 5

Variable conversion
as.array(x), as.data.frame(x), as.numeric(x),
as.logical(x), as.complex(x), as.character(x),
... convert type; for a complete list, use methods(as)

Variable information
is.na(x), is.null(x), is.array(x), is.data.frame(x),
is.numeric(x), is.complex(x), is.character(x),
... test for type; for a complete list, use methods(is)

Data selection and manipulation
which.max(x) returns the index of the greatest element of x
which.min(x) returns the index of the smallest element of x
cut(x) reverses the elements of x
sort(x) sorts the elements of x in increasing order; to sort in decreasing
order: rev(sort(x))
cut(x,breaks) divides x into intervals (breaks); breaks is the number of cut
intervals or a vector of cut points
match(x,y) returns a vector of the same length than x with the elements
of x which are in y (NA otherwise)
which(x == a) returns a vector of the indices of x if the comparison
operation is true (TRUE), in this example the values of 1 for which x[1]
is a (the argument of this function must be a variable of mode logical)
choose(n,k) computes the combinations of k elements among n repetitions
na.omit(x) suppresses the observations with missing data (NA) (sup-
presses the corresponding line if x is a matrix or a data frame)
na.fail(x) returns an error message if x contains at least one NA

Indexing lists
x[n] list with elements n
x[[n]] nth element of the list
x[['name']] element of the list named "name"

na.name id.

Indexing matrices
x[i,j] element at row i, column j
x[i,] row i
x[,j] column j
x[,c(1,3)] columns 1 and 3
x[['name']]; row named "name"

Indexing data frames (matrix indexing plus the following)
x[['name']] column named "name"
na.name id.

Indexing functions
save(file,...) saves the specified objects (...) in the XDR platform-
read.fwf(file,widths,header=FALSE,sep=" ") to read a table copied from Excel, use
result <- read.delim("clipboard")
write to a table to the clipboard for Excel, use
write.table(x,clipboard,"sep="",col.names=NA)

For database interaction, see packages RODBC, DBI, MySQL, PostgreSQL, and
Oracle. See packages XML,hdf5, netCDF for reading other file formats.
unique(x) if x is a vector or a data frame, returns a similar object but with the duplicate elements suppressed

table(x) returns a table with the numbers of the different values of x (typically for integers or factors)

subset(x, ...) returns a selection of x with respect to criteria (...), typically comparisons: x>10; if x is a data frame, the option select gives the variables to be kept or dropped using a minus sign

sample(x, size) resample randomly and without replacement size elements in the vector x, the option replace = TRUE allows to resample with replacement

prop.table(x, margin=) table entries as fraction of marginal table

Math

sin, cos, tan, asin, acos, atan, atan2, log, log10, exp

max(x) maximum of the elements of x

min(x) minimum of the elements of x

range(x) id. then c(min(x), max(x))

sum(x) sum of the elements of x

diff(x) lagged and iterated differences of vector x

prod(x) product of the elements of x

mean(x) mean of the elements of x

median(x) median of the elements of x

quantile(x, probs=) a vector of the positions of first matches for the elements of x

log(x, base) logarithm of the elements of x with base

cov(x) variance-covariance matrix of the elements of x (calculated on x)

cov(x,y) sum of rows for a matrix-like object; rowSums(x) is a faster version

colsum(x). colSums(x) id. for columns

rowMeans(x) fast version of row means

weighted.mean(x, w) weighted mean of the elements of x

var(x, y) variance of the elements of x

sd(x) standard deviation of x

cor(x) correlation matrix of x if it is a matrix or a data frame (1 if x is a vector)

var(x, y) or cov(x,y) variance between x and y, or between the columns of x and those of y if they are matrices or data frames

cor(x, y) linear correlation between x and y, or correlation matrix if they are matrices or data frames

round(x, n) rounds the elements of x to n decimals

log(x, base) computes the logarithm of x with base base

collapse= an optional string to separate "collapsed" results (a single space is the default); value of FUN (output only.) Time zone as a character string (empty if not available).

Where leading zeros are shown they will be used on output but are optional on input. See strftime.

Plotting

plot(x) plot of the values of x (on the y-axis) ordered on the x-axis

plot(x, y) bivariate plot of x (on the x-axis) and y (on the y-axis)

hist(x) histogram of the frequencies of x

barplot(x) histogram of the values of x; use horiz=FALSE for horizontal bars

dotchart(x) if x is a data frame, plots a Cleveland dot plot (stacked plots line-by-line and column-by-column)

table(x) circular pie-chart

pie(x) bivariate dot plot (stacked plots line-by-line and column-by-column)

boxplot(x) "box-and-whiskers" plot

sunflowerplot(x, y) id. than plot() but the points with similar coordinates are drawn as flowers which petal number represents the number of points

interaction.plot (f1, f2, y) if f1 and f2 are factors, plots the means of y (on the y-axis) with respect to the values of f1 (on the x-axis) and of f2 (different curves); the option fun allows to choose the summary statistic of y (by default mean=mean

Date and Time

The class Date has dates without times. POSIXct has dates and times, including time zones. Comparisons (e.g. >, seq(), and diff() are useful. Date also allows + and -. DateTimeClasses gives more information. See also package chron.

as.Date(s) and as.POSIXct(s) convert to the respective class; format(dt) converts to a string representation. The default string format is "2001-02-21". These accept a second argument to specify a format for conversion. Some common formats are:

%a, %A Abbreviated and full weekday name.
%b, %B Abbreviated and full month name.
%d Day of the month (01–31).
%m Hours (00–23).
%h Hours (01–12).
%j Day of year (001–366).
%m Month (01–12).
%w Minutes (0–59).
%p AM/PM indicator.
%y Second as decimal number (00–61).
%Y Week (00–53); the first Sunday as day 1 of week 1.
%W Weekday (0–6, Sunday is 0).
%Y Year without century (00–99). Don’t use.
%y Year with century.
%Z Offset from Greenwich; %Z is 8 hours west of.
%Z Time zone as a character string (empty if not available).

Strings

paste(…) concatenate vectors after converting to character; sep= is the string to separate terms (a single space is the default); collapse= is an optional string to separate "collapsed" results.

substitute(x, start, stop) substrings in a character vector; can also as assign, as substitute(x, start, stop) <- value

grepl(pattern, x) searches for matches to pattern within x; see grep

gsub(pattern, replacement, x) replacement of matches determined by regular expression matching sub() is the same but only replaces the first occurrence.

toupper(x) convert to lowercase

toupper(x) convert to uppercase

match(x, table) a vector of the positions of first matches for the elements of x among table

% in table id. but returns a logical vector

pmatch(x, table) partial matches for the elements of x among table

nchar(x) number of characters

Dates and Times

The class Date has dates without times. POSIXct has dates and times, including time zones. Comparisons (e.g. >, seq(), and diff() are useful. Date also allows + and -. DateTimeClasses gives more information. See also package chron.

as.Date(s) and as.POSIXct(s) convert to the respective class; format(dt) converts to a string representation. The default string format is "2001-02-21". These accept a second argument to specify a format for conversion. Some common formats are:
matplot(x, y) bivariate plot of the first column of x vs. the first one of y, the second one of x vs. the second one of y, etc.

fourfoldplot(x) visualizes, with quarters of circles, the association between two dichotomous variables for different populations (x must be an array with dim=c(2,2, k), or a matrix with dim=c(2, 2) if k = 1)

assocplot(x) Cohen-Friendly graph showing the deviations from independence of rows and columns in a two dimensional contingency table

mosaicplot(x) ‘mosaic’ graph of the residuals from a log-linear regression of a contingency table

pairs(x) if x is a matrix or a data frame, draws all possible bivariate plots between the columns of x

plot.ts(x) if x is an object of class "ts", plot of x with respect to time, x may be multivariate but the series must have the same frequency and dates

ts.plot(x) id. but if x is multivariate the series may have different dates and must have the same frequency

qqnorm(x) quantiles of x with respect to the values expected under a normal law

qqplot(x, y) quantiles of y with respect to the quantiles of x

contour(x, y, z) contour plot (data are interpolated to draw the curves), x and y must be vectors and z must be a matrix so that dim(z)=c(length(x), length(y)) (x and y may be omitted)

filled.contour(x, y, z) id. but the areas between the contours are coloured, and a legend of the colours is drawn as well

image(x, y, z) id. but with colours (actual data are plotted)

persp(x, y, z) id. but in perspective (actual data are plotted)

stars(x) if x is a matrix or a data frame, draws a graph with segments or a star where each row of x is represented by a star and the columns are symbols

circles, squares, rectangles, stars, thermometers or ‘boxplots’ which sizes, colours… are specified by supplementary arguments

termplot(mod.obj) plot of the (partial) effects of a regression model

The following parameters are common to many plotting functions:

add=TRUE if FALSE does not draw the axes and the box

type=“p” specifies the type of plot, “p”: points, “l”: lines, “b”: points connected by lines, “o”: id. but the lines are over the points, “h”: vertical lines, “s”: steps, the data are represented by the top of the vertical lines, “i”: id. but the data are represented by the bottom of the vertical lines

xlim=, ylim= specifies the lower and upper limits of the axes, for example with xlim=c(1, 10) or ylim=range(x)

labx=, laby= annotates the axes, must be variables of mode character

main= main title, must be a variable of mode character

sub= sub-title (written in a smaller font)

Low-level plotting commands

points(x, y) adds points (the option type= can be used)

lines(x, y) id. but with lines

text(x, y, labels, ...) adds text given by labels at coordinates (x,y); a typical use is: plot(x, y, type=“n”); text(x, y, names)

mtext(text, side=3, line=0, ...) adds text given by text in the margin specified by side (see axis[] below); line specifies the line from the plotting area

segments(x0, y0, x1, y1) draws lines from points (x0,y0) to points (x1,y1)

arrows(x0, y0, x1, y1, angle= 30, code=2) id. with arrows at points (x0,y0) if code=2, at points (x1,y1) if code=1, or both if code=2; angle controls the angle from the shaft of the arrow to the edge of the arrow, x0 and y0 are the lengths of the segments

abline(a,b) draws a line of slope b and intercept a

abline(h=yy) draws a horizontal line at ordinate y

abline(v=xx) draws a vertical line at abscissa x

abline(lm.obj) draws the regression line given by lm.obj

rect(x1, y1, x2, y2) draws a rectangle which left, right, bottom, and top limits are x1,x2, y1, and y2, respectively

polygon(x, y) draws the polygon joining the points with coordinates given by x and y

legend(x, y, legend) adds the legend at the point (x,y) with the symbols given by legend

title() adds a title and optionally a sub-title

axis(side, vect) adds an axis at the bottom (side=1), on the left (2), at the top (3), or on the right (4); vect (optional) gives the abscissa (or ordinates) where tick-marks are drawn

rug(x) draws the data on the x-axis as small vertical lines

locator(n, type=“n”, ...) returns the coordinates (x,y) after the user has clicked n times on the plot with the mouse; also draws symbols (type=“p”) or lines (type=“l”) with respect to optional graphic parameters (…). by default nothing is drawn (type=“n”)

Graphical parameters

These can be set globally with par(…); many can be passed as parameters to plotting commands.

dat= controls text justification (0 left-justified, 0.5 centred, 1 right-justified)

cex= the size in points of texts and symbols

cex.axis=, cex.lab=, cex.main=, cex.sub= control the sizes of the axes, titles, labels, and the sub-title

cex= controls the size of tick-marks on the axes as a fraction of the smallest of the width or height of the plot; if cex=1 a grid is drawn

tcl= a value which specifies the length of tick-marks on the axes as a fraction of the height of a line of text (by default tcl=0.5)

xaxt= if xaxt=“n” the x-axis is set but not drawn (useful in conjunction with axis(side=1, …))
yaxt= if yaxt=“n” the y-axis is set but not drawn (useful in conjunction with axis(side=2, …))

Lattice (Trellis) graphics

xyplot(y˜x) bivariate plots (with many functionalities)
barchart(y˜x) histogram of the values of y with respect to those of x
dotplot(y˜x) Cleveland dot plot (stacked plots line-by-line and column-by-column)
densityplot(y˜x) density functions plot

histogram(x˜) histogram of the frequencies of x

bwplot(y˜x) “box-and-whiskers” plot

qqmath(y˜x) quantiles of x with respect to the values expected under a theoretical distribution

stripplot(y˜x) single dimension plot, x must be numeric, y may be a factor

trellis() boxplot for multiple comparisons

parallel(y˜x) parallel coordinates plot

levelplot(z˜x*y|g1*g2) coloured plot of the values of z at the coordinates given by x and y (x, y and z are all of the same length)

gamplot(z˜x*y|g1*g2) 3d surface plot
cloud(z˜x*y|g1*g2) 3d scatter plot
In the normal Lattice formula, \( y \times |g1 \times g2 \) has combinations of optional conditioning variables \( g1 \) and \( g2 \) plotted on separate panels. Lattice functions take many of the same arguments as base graphics plus also data- the data frame for the formula variables and subset- for subsetting. Use panel- to define a custom panel function (see apropos("panel") and ?alllines). Lattice functions return an object of class trellis and have to be print-ed to produce the graph. Use print(xyplot(...)) inside functions where automatic printing doesn’t work. Use lattice.theme and lset to change Lattice defaults.

### Optimization and model fitting

- **optim(par, fn, method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN") general-purpose optimization; par is initial values, fn is function to optimize (normally minimize) nlm(f,p) minimize function f using a Newton-type algorithm with starting values p**
- **lm(formula) fit linear models; formula is typically of the form response termA + termB + ...; use I(x^y) + I(x^2) for terms made of nonlinear components glm(formula, family=) fit generalized linear models, specified by giving a symbolic description of the linear predictor and a description of the error distribution; family is a description of the error distribution and link function to be used in the model; see ?family**
- **nls(formula) nonlinear least-squares estimates of the nonlinear model parameters**
- **approx(x,y=) linearly interpolate given data points; x can be an xy plotting structure spline(x,y=) cubic spline interpolation loess(formula) fit a polynomial surface using local fitting**

Many of the formula-based modeling functions have several common arguments: data- the data frame for the formula variables, subset- a subset of variables used in the fit, na.action- action for missing values: "na.fail", "na.omit", or a function. The following generics often apply to model fitting functions:

- **predict(formula) predictions from fit based on input data**
- **df.residual(formula) returns the number of residual degrees of freedom**
- **coef(formula) returns the estimated coefficients (sometimes with their standard-errors)**

### Statistics

- **aov(formula)** analysis of variance model
- **anova(formula) analysis of variance (or deviance) tables for one or more fitted model objects**
- **density(x) kernel density estimates of x**
- **binom.test(), pairwise.t.test(), power.t.test(), prop.test(), t.test()... use help.search("test")**

### Distributions

- **rnorm(n, mean=0, sd=1) Gaussian (normal)**
- **rexp(n, rate=1) exponential**
- **rgamma(n, shape, scale=1) gamma**