# R Reference Card

by Bent Jørgensen, adapted from the R/Rpad Reference Card by Tom Short. Granted to the public domain. 2011-01-12

# 1 Getting started

http://cran.r-project.org/ main R site (CRAN) for download Click the R icon to start R, or type the command R Files are stored by default the R root directory Change working directory by right-clicking the R icon, select *Properties*, and edit the *Start in* path Make an R icon for each project, with its own working directory > prompt for new command on console + prompt for completing a command

Ctrl c or Esc break out from a pending +

; or  ${\bf Enter}$  separate two commands

† bring back previous commands

# begin comment (until end of line)

= or <- assignment operator, also ->

<-- global assignment (to overrule scope rules)

func(arg) call function func with arguments arg

func show body of function func

**q("yes")** or **q("no")** quit R with/without saving workspace Strings are enclosed by quotes, e.g. "alphabet"

Variable names are case sensitive; can have letters, digits and period; must not start with a number

 $\mathbf{x}[\mathbf{i}]$  access element i of vector  $\mathbf{x}$ 

names(list) show names of the elements in list

list\$element access element of list

T=TRUE, F=FALSE logical values

Inf infinity, e.g. 1/0

**NaN** not a number, e.g. 0/0

**NA** missing value, when no value has been assigned

**NULL** the null object, e.g. an empty list

#### 2 Help functions, workspace and packages

Most R functions have online help
Many functions have optional arguments, see help
help.start() online access to R manuals
help(command) or ?command help on command
help.search("command") or ??command
search the help system

RsiteSearch("command") general search in R site demo() run R demo, e.g. demo(graphics) example (function) show example of function str(a) display the internal structure of an object a summary(a) display a summary of object a dir() show files in the current directory **setwd(path)** set windows directory to path ls() or objects() show objects in the search path ls.str() apply str() for each variable in the search path rm(x,y,...) remove object x,y,... from workspace rm(list=objects()) remove everything created so far options(...) set global options, e.g. width, digits, error methods(a) show methods of function a methods(class=class(a)) methods for objects of class a search() show packages currently loaded library(package) load package for current session apropos("command") or find("command") find library containing "command"

library(help=package) list datasets and functions in package install.packages("suplib") install suplib from CRAN

update.packages() update packages from CRAN Pre-loaded packages: base, datasets, graphics, grDevices, methods, stats, utils

## 3 Input and output

file denotes file name like "path/rscript.r" or "ascii.txt" file="" refers to the standard input or output save.image(file) save current workspace in file load(file) load previously saved workspace file data(x) load specified data set (data frame) x source(file) execute the R-script file sink(file) send output to file instead of screen, until sink() scan(file) read file into a vector or list read.table(file,header=TRUE) read file into data frame; first line defines column names; space is separator; see help for options on row naming, NA treatment etc. read.csv(file,header=TRUE) read comma-delimited file read.delim(file,header=TRUE) read tab-delimited file read.fwf(file,widths) read fixed width formatted file; integer vector widths define column widths save(file, x, y, ...) save x, y, ... to file in portable format  $\mathbf{print}(\mathbf{x},\mathbf{y},\dots)$  print objects  $\mathbf{x},\mathbf{y},\dots$  in default format format(x,y,...) format x,y,... for pretty printing, e.g. print(format(x)) cat(x,y,...,file="", sep="") concatenate and print x,y,... to file (default console), using separator sep write.table(x,file="",row.names=T,col.names=T, sep="") print x as data frame, using separator sep; eol is end-of-line separator, na is missing values string  $write.table(x,"clipboard",sep="\t",col.names=NA)$ write a table to the clipboard for Excel in Windows read.delim("clipboard") read a table copied from Excel foreign read data stored by Minitab, SAS, SPSS, Stata, e.g.

## 4 Simple patterned vectors

 $\mathbf{x}$ =numeric(n) create numeric n-vector with all elements 0  $\mathbf{c}(\mathbf{x},\mathbf{y},\ldots)$  stack (concatenate) values or vectors  $\mathbf{x},\mathbf{y},\ldots$  to form a long vector

read.dta("statafile"), read.ssd("sasfile"), read.spss("spssfile")

**a:b** generate sequence of integers  $a, \ldots, b;$  ":" has priority; e.g. 1:4+1 is 2 3 4 5

seq(from,to,by=) generate a sequence with increment by; length= specifies length

seq(along=x) generate 1, 2, ..., length(x)

rep(x,n) replicate x n times, e.g. rep(c(1,2,3),2) is 1 2 3 1 2 3; each= repeat each element of x each times, e.g. rep(c(1,2,3),each=2) is 1 1 2 2 3 3

list(...) create list of the named or unnamed arguments; e.g. list(a=c(1,2),b="hi",c=3i)

factor(x,levels=) encode vector x as factor (define groups)
gl(n,k,length=n\*k,labels=1:n) regular pattern of factor
levels; n is the number of levels, and k is the number
of replications

 $\begin{array}{llll} \mathbf{expand.grid}(\mathbf{x},\mathbf{y},\dots) & \mathrm{create} & \mathrm{data} & \mathrm{frame} & \mathrm{with} & \mathrm{all} \\ & \mathrm{combinations} & \mathrm{of} & \mathrm{the} & \mathrm{values} & \mathrm{of} & \mathsf{x},\mathsf{y},\dots, & \mathrm{e.g.} & \mathsf{expand.grid}(\mathsf{c}(1,2,3),\mathsf{c}(10,20)) \end{array}$ 

## 5 Arithmetic operations

+, -, \*, / arithmetic operators, e.g. x\*y+z/3-u x^y or x\*\*y raise x to the power y, e.g.. x^2 or 5\*\*y %/% and %% integer division and remainder trunc(x) integer part of x, e.g. trunc(2.5)=2, trunc(-2.5)=-2 ceiling(x) round up to nearest integer, e.g. ceiling(2.5)=3

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

#### 6 Math functions

sin cos tan asin acos atan atan2 log log10 log2 exp sqrt sinh cosh tanh asinh acosh atanh pi =  $\pi$  abs(x) = |x| sign(x) log(x,b) log to base b log1p(x) = log(1+x), expm1(x) = exp(x)-1 beta(a,b) and lbeta(a, b) B(a,b) and log B(a,b) gamma(x) and lgamma(x)  $\Gamma(x)$  and log  $\Gamma(x)$  digamma(x) digamma function  $\psi(x) = \Gamma'(x)/\Gamma(x)$  trigamma(x) trigamma function  $\psi'(x)$  psigamma(x, deriv = 0) derivative of  $\psi(x)$  of order deriv factorial(x) and lfactorial(x) x factorial x! and log x! choose(n, k) and lchoose(n, k)  $\binom{n}{k}$  and log  $\binom{n}{k}$  Option na.rm=FALSE gives missing data (NA) removal

## 7 Logical operations

x=logical(n) create logical n-vector with all values FALSE
& logical 'and' operator, | logical 'or' operator
== logical 'equal' operator
!= logical 'not equal' operator
! logical negation operator
<, <=, >, >= inequality operators
any(x) returns TRUE if at least one component of x is TRUE
all(x) returns TRUE if all component of x are TRUE

## 8 Complex numbers

x=complex(n) create complex n-vector with all elements 0
a + bi complex number, e.g. 3+4i; 1i is the imaginary unit
Re(x) and lm(x) real and imaginary parts of x
Mod(x) or abs(x) modulus of x

Arg(x) angle of x in radians

 $\mathbf{Conj}(\mathbf{x})$  complex conjugate of  $\mathbf{x}$ 

# 9 Matrices and arrays

X=matrix(x,nrow=r,ncol=c) r×c matrix; elements of X are recycled

rbind(x,y,...) form a matrix with rows x,y,...
cbind(x,y,...) form a matrix with columns x,y,...

 $\mathbf{t}(\mathbf{X})$  transpose of X

det(X) and trace(X) determinant and trace of square matrix
norm(as.matrix(x),"F") Euclidean norm of vector x

kappa(X) condition number of X

 $\operatorname{diag}(X)$  returns the diagonal of matrix X

diag(X) creates diagonal matrix from vector X

diag(n) creates  $n \times n$  identity matrix

X%\*%Y matrix multiplication of X and Y

solve(X) matrix inverse of X

solve(A,b) solves  $A \times = b$  for  $\times$ 

forwardsolve(A,b) id. for A lower triangular

**backsolve**( $\mathbf{A}$ , $\mathbf{b}$ ) solves  $A \times = \mathbf{b}$  for  $\times$  when A is upper triangular  $\mathbf{qr}$ .**solve**( $\mathbf{X}$ ) similar to  $\mathbf{solve}(\mathbf{X})$ , but uses QR decomposition  $\mathbf{qr}(\mathbf{X})$  QR decomposition of  $\mathbf{X}$ , see help

chol(X) upper triangular Choleski factor of X

chol12inv(X) matrix inverse of X by Choleski decomposition
eigen(X) eigenvalue decomosition of X (list),

eigenvalues in \$values and eigenvectors in \$vectors  $\mathbf{svd}(\mathbf{X})$  singular value decomposition of X (list), \$u \$d \$v  $\mathbf{crossprod}(\mathbf{X},\mathbf{Y})$  matrix (inner) product  $\mathbf{X}^{\top}\mathbf{Y}$  outer  $(\mathbf{x},\mathbf{y})$  outer product  $\mathbf{xy}^{\top}$  for vectors

kronecker(X,Y) Kronecker product of X and Y
lower.tri(X,diag=TRUE) logical matrix,

TRUE below diagonal;

diag=TRUE causes diagonal elements to be included

upper.tri(X,diag=TRUE) TRUE above diagonal;

diag=TRUE causes diagonal elements to be included
rowSums(X) and rowMeans(X) form row sums or means of X
colSums(X) and colMeans(X) form column sums/means of X
rowsum(X,group) column sums within each level for group;
is equivalent to colSums(X) for a single group

array(X,dim) create array from X with dimensions from vector dim; elements of X recycle if necessary

dim(X) dimension of array X

Matrix package for sparse and dense matrix operations

## 10 Strings

x=character(n) create n-vector of empty strings
nchar(x) number of characters in x

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

substring(x,start) and substr(x,start,stop) access substring of
x; from position start (to stop)

strsplit(x,split) split x according to the substring split
grep(pattern,x) searches for matches to pattern within x;
see ?regex for information about regular expressions

gsub(pattern,replacement,x) replace matches to regular expression pattern; sub() only replaces first occurrence

tolower(x) and toupper(x) convert to lowercase/uppercase
match(x,table) a vector of the positions of first matches for
the elements of x in table

x %in% table id. but returns a logical vector pmatch(x,table) partial matches for the elements of x in table

#### 11 Data frames and time series

X=data.frame(x,y,...) create data frame with named colums; colums may have different types;

shorter vectors are recycled to the length of the longest

X\$v access variable v of data frame X

transform(X, var1=expr1,var2=expr2,...) create or modify variables of X without using \$

 $with(X,{commands})$  apply commands to variables of X without using \$

 $within(X,{commands})$  apply commands to possibly transform variables of X without using \$

attach(X) attach data frame X to the current session, to access
variables of X without using \$

detach(X) detach previously attached data frame X
ts(data,a,b,n) create time series object from data vector
 on interval [a,b] with n observations per time unit;
 multiple time series if data is matrix or data frame

# 12 Elements and subsets of vectors and matrices

 $\mathbf{x}[\mathbf{n}] \mathbf{n}^{th}$  element of vector  $\mathbf{x}$ 

 $\mathbf{x}[-\mathbf{n}]$  all but the  $\mathbf{n}^{th}$  element of vector  $\mathbf{x}$ 

 $\mathbf{x}[1:n]$  first n elements of vector x

x[-(1:n)] elements from n+1 to the end of vector x

x[c(1,4,2)] specific elements of vector x

x["name"] element named "name" of vector x

 $\mathbf{x}[\mathbf{x} > \mathbf{3}]$  all elements greater than 3 of vector  $\mathbf{x}$ 

 $\mathbf{x}[\mathbf{x} > \mathbf{3} \& \mathbf{x} < \mathbf{5}]$  all elements between 3 and 5 of vector  $\mathbf{x}$ 

x[x %in% list] elements of x in list

X[i,j] element at (i,j) of matrix X

X[i,] and X[j] row i and column j of matrix X

X[,c(1,3)] columns 1 and 3 of matrix X

X["name",] row named "name" of matrix X
X[["name"]] or X\$name column "name" of data frame X
x[["name"]] or x\$name element "name" of list x
x[[n]] or x[n] n<sup>th</sup> element of list x

## 13 Descriptive statistics

max(x) maximum of the elements of x

min(x) minimum of the elements of x

range(x) gives c(min(x), max(x))

IQR(x) interquartile range of x

**fivenum(x)** Tukey's five number summary (minimum, lower-quartile, median, upper-quartile, maximum) for 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=) sample quantiles corresponding to the given probabilities (defaults to 0,.25,.5,.75,1)

 $\mathbf{weighted.mean}(\mathbf{x}, \mathbf{w}) \text{ mean of } \mathbf{x} \text{ with weights } \mathbf{w}$ 

rank(x) ranks of the elements of x

 $\operatorname{var}(\mathbf{x})$  or  $\operatorname{cov}(\mathbf{x})$  variance of the elements of  $\mathbf{x}$  (calculated on n-1); if  $\mathbf{x}$  is a matrix or a data frame, the variance-covariance matrix is calculated

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) covariance 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

log(x, base) computes the logarithm of x with base base
scale(x) if x is a matrix, centers and scales the data; to center only use the option scale=FALSE, to scale only center=FALSE (by default center=TRUE, scale=TRUE)

pmin(x,y,...) a vector which *i*th element is the minimum of x[i], y[i], ...

 $\mathbf{pmax}(\mathbf{x},\mathbf{y},...)$  id. for the maximum

**cumsum**(**x**) a vector which *i*th element is the sum from x[1] to x[i]

 $\operatorname{cumprod}(\mathbf{x})$  id. for the product

**cummin(x)** id. for the minimum

**cummax(x)** id. for the maximum

 $\mathbf{union}(\mathbf{x}, \mathbf{y})$ , intersect(x,y), setdiff(x,y), setequal(x,y), is.element(el,set) "set" functions

# 14 Distributions and simulation

rnorm(n, mean=0, sd=1) Gaussian (normal)

rexp(n, rate=1) exponential

rgamma(n, shape, scale=1) gamma

rbinom(n, size, prob) binomial

rpois(n, lambda) Poisson

rgeom(n, prob) geometric

rhyper(nn, m, n, k) hypergeometric

rweibull(n, shape, scale=1) Weibull

reauchy(n, location=0, scale=1) Cauchy

rbeta(n, shape1, shape2) beta

rt(n, df) 'Student' (t)

 $\mathbf{rf}(\mathbf{n}, \mathbf{df1}, \mathbf{df2})$  Fisher-Snedecor (F)

rchisq(n, df) Pearson

rlogis(n, location=0, scale=1) logistic rlnorm(n, meanlog=0, sdlog=1) lognormal rnbinom(n, size, prob) negative binomial runif(n, min=0, max=1) uniform rwilcox(nn, m, n), rsignrank(nn, n) Wilcoxon's statistics All these functions can be used by replacing the letter r with d, p or q to get, respectively, the probability density (dfunc(x, ...)), the cumulative probability density (pfunc(x, ...)), and the quantile function (qfunc(p, ...), with 0 < p < 1).

#### 15 Tests and ANOVA

Do we need arrays for this. Combine with categorical data. **aov(formula)** analysis of variance model

anova(fit,...) analysis of variance (or deviance) tables for one or more fitted model objects

density(x) kernel density estimates of x

## 16 Regression and model fitting

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
summary(fit) display a summary of fitted model fit
Many of the formula-based modeling functions have several
common arguments: data= the data frame for the formula vari-

ables, 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(fit,...) predictions from fit based on input data
df.residual(fit) returns the number of residual degrees of freedom.

residuals(fit) returns the residuals

deviance(fit) returns the deviance

fitted(fit) returns the fitted values

summary(fit)\$sigma extracts residual variance for fit etc.
logLik(fit) computes the logarithm of the likelihood and the

number of parameters

AIC(fit) computes the Akaike information criterion or AIC

## 17 Statistical packages

survival, extremes, multivariate analysis, GEE, GLMM

## 18 Simple plotting commands

plot(x) plot the values of x agains index

plot(x, y) scatterplot of points (x,y)

 $\operatorname{curve}(\exp(\mathbf{x}), \mathbf{a}, \mathbf{b})$  makes plot of expression  $\exp(\mathbf{x})$  on [a, b]

hist(x) histogram of the frequencies of x

stem(x) stem-and-leaf plot of x

 $\mathbf{barplot}(\mathbf{x})$  draw bar chart for 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)

 $\mathbf{pie}(\mathbf{x})$  circular pie-chart

boxplot(x,y,...) draw boxplot(s) for x,y,...

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

stripplot(x) plot of the values of x on a line (an alternative
to boxplot() for small sample sizes)

 $coplot(x \sim y \mid z)$  bivariate plot of x and y for each value or interval of values of z

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 fun=mean)

 $\mathbf{matplot}(\mathbf{x}, \mathbf{y})$  bivariate plot of the first column of  $\mathbf{x}$  vs. the first one of  $\mathbf{y}$ , the second one of  $\mathbf{x}$  vs. the second one of  $\mathbf{y}$ , 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

 $\mathbf{pairs}(X)$  draw all bivariate scatterplots between columns of matrix or data frame 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 qqline(x) adds expected line to qq plot.

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 the lengths of the segments

symbols(x, y, ...) draws, at the coordinates given by x and y, symbols (circles, squares, rectangles, stars, thermometres or boxplots) which sizes, colours ... are specified by supplementary arguments

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

The following parameters are common to many plotting functions:

add=FALSE if TRUE superposes the plot on the previous one
 (if it exists)

axes=TRUE if FALSE does not draw the axes and the box
type="p" specifies the type of plot, "p": points, "I": 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, "S": 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 xlim=range(x)

xlab=, ylab= 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)

plotmath(TeX-code) adds math formulas to axis and title commands, using a TeX-like syntax

## 19 Low-level plotting commands

points(x, y) adds points (the option type= can be used)
lines(x, y) id. but with lines

 $\mathbf{text}(\mathbf{x}, \mathbf{y}, \mathbf{labels}, ...)$  adds text given by labels at coordinates (x,y); a typical use is:  $\mathsf{plot}(x, y, \mathsf{type}="n")$ ;  $\mathsf{text}(x, y, \mathsf{names})$ 

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) draw lines from points (x0,y0) to points (x1,y1)

arrows(x0, y0, x1, y1) draw arrows from points (x0,y0) to points (x1,y1)

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

abline(h=y) draw a horizontal line at ordinate y

abline(v=x) draw a vertical line at abcissa x

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

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

polygon(x, y) draw a polygon linking the points with coordinates given by x and y

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

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

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

**box()** draw a box around the current plot

rug(x) draw the data x on the x-axis as small vertical lines
locator(n, type="n", ...) return 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")

## 20 Graphics devices

Open new graphics window: windows() in Windows x11(), or X11() in Unix or Linux, quartz() in Mac OSX

postscript(file) starts the graphics device driver for producing PostScript graphics; use horizontal = FALSE, onefile = FALSE, paper = "special" for EPS files; family= specifies the font (AvantGarde, Bookman, Courier, Helvetica, Helvetica-Narrow, NewCenturySchoolbook, Palatino, Times, or ComputerModern); width= and height= specifies the size of the region in inches (for paper="special", these specify the paper size).

ps.options() set and view (if called without arguments) default values for the arguments to postscript

Other graphics devices (see ?Devices): bitmap(), bmp(), jpeg(),

pdf(), pictex(), png(), tiff(), xfig(); see ?Devices
dev.off() shut down the current/specified device;
 see also dev.cur, dev.set

## 21 Graphical parameters

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

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

bg specifies the colour of the background (ex. : bg="red",
 bg="blue", ... the list of the 657 available colours is
 displayed with colors())

bty controls the type of box drawn around the plot, allowed values are: "o", "I", "7", "c", "u" ou "]" (the box looks like the corresponding character); if bty="n" the box is not drawn

cex a value controlling the size of texts and symbols with respect to the default; the following parameters have the same control for numbers on the axes, cex.axis, the axis labels, cex.lab, the title, cex.main, and the sub-title, cex.sub

col controls the color of symbols and lines; use color names:
 "red", "blue" see colors() or as "#RRGGBB"; see rgb(),
 hsv(), gray(), and rainbow(); as for cex there are: col.axis,
 col.lab, col.main, col.sub

font an integer which controls the style of text (1: normal,2: italics, 3: bold, 4: bold italics); as for cex there are: font.axis, font.lab, font.main, font.sub

las an integer which controls the orientation of the axis labels (0: parallel to the axes, 1: horizontal, 2: perpendicular to the axes, 3: vertical)

lty controls the type of lines, can be an integer or string (1:
 "solid", 2: "dashed", 3: "dotted", 4: "dotdash", 5: "long dash", 6: "twodash", or a string of up to eight characters
 (between "0" and "9") which specifies alternatively the
 length, in points or pixels, of the drawn elements and the
 blanks, for example lty="44" will have the same effect
 than lty=2

lwd a numeric which controls the width of lines, default 1

mar a vector of 4 numeric values which control the space between the axes and the border of the graph of the form c(bottom, left, top, right), the default values are c(5.1, 4.1, 4.1, 2.1)

mfcol a vector of the form c(nr,nc) which partitions the graphic window as a matrix of nr lines and nc columns, the plots are then drawn in columns

mfrow id. but the plots are drawn by row

**pch** controls the type of symbol, either an integer between 1 and 25, or any single character within ""

**ps** an integer which controls the size in points of texts and symbols

pty a character which specifies the type of the plotting region,
 "s": square, "m": maximal

tck a value which specifies the length of tick-marks on the axes
 as a fraction of the smallest of the width or height of the
 plot; if tck=1 a grid is drawn

 ${f tcl}$  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  ${f tcl}{=}{-}0.5)$ 

xaxs, yaxs style of axis interval calculation; default "r" for an extra space; "i" for no extra space

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 conjonction with axis(side=2, ...))

# 22 Lattice (Trellis) graphics

 $\mathbf{xyplot}(\mathbf{y} \sim \mathbf{x})$  bivariate plots (with many functionalities)

 $\mathbf{barchart}(\mathbf{y} \sim \mathbf{x})$  histogram of the values of y with respect to those of x

 $dotplot(y \sim x)$  Cleveland dot plot (stacked plots line-by-line and column-by-column)

 $densityplot(\sim x)$  density functions plot

 $histogram(\sim x)$  histogram of the frequencies of x

 $bwplot(y\sim x)$  "box-and-whiskers" plot

 $qqmath(\sim x)$  quantiles of x with respect to the values expected under a theoretical distribution

 $stripplot(y\sim x)$  single dimension plot, x must be numeric, y may be a factor

 $qq(y\sim x)$  quantiles to compare two distributions, x must be numeric, y may be numeric, character, or factor but must have two 'levels'

 $splom(\sim x)$  matrix of bivariate plots

 $parallel(\sim x)$  parallel coordinates plot

levelplot( $z\sim x^*y|g1^*g2$ ) 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)

 $\mathbf{wireframe}(\mathbf{z}{\sim}\mathbf{x}^*\mathbf{y}|\mathbf{g}\mathbf{1}^*\mathbf{g}\mathbf{2}) \text{ 3d surface plot}$ 

 $cloud(z\sim x*y|g1*g2)$  3d scatter plot

In the normal Lattice formula, y x|g1\*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 ?llines). 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.

Graphics packages

grid The Grid Graphics Package

lattice Lattice Graphics

 $\mathbf{ggplot} \ \mathrm{package}$ 

tcltk Tcl/Tk Interface

## 23 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)

## 24 Variable information

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

length(x) number of elements in x

dim(x) access the dimension of an object, e.g. dim(x) = c(3,2)dimnames(x) access the dimension names of an object

 $\mathbf{nrow}(\mathbf{x})$  number of rows;  $\mathsf{NROW}(\mathbf{x})$  is the same but treats a vector as a one-row matrix

 $\mathbf{ncol}(\mathbf{x})$  and  $\mathsf{NCOL}(\mathbf{x})$  id. for columns

class(x) get or set the class of x; class(x) = "myclass"

unclass(x) remove the class attribute of x

attr(x,which) get or set the attribute which of x

attributes(obj) get or set the list of attributes of obj

## 25 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
rev(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 (factors); 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 i for which x[i] == a (the argument of this function must be a variable of mode logical)

na.omit(x) suppresses the observations with missing data
 (NA) (suppresses the corresponding line if x is a matrix
 or a data frame)

 $\mathbf{na.fail}(\mathbf{x})$  returns an error message if  $\mathbf{x}$  contains at least one  $\mathsf{NA}$ 

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 differents values of x (typically for integers or factors)

 ${f subset(x, ...)}$  returns a selection of x with respect to criteria (..., typically comparisons: x\$V1 < 10); if x is a data frame, the option select gives the variables to be kept or dropped using a minus sign

sample(x, n) sample n elements randomly from x;
replace = TRUE means sample with replacement

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

## 26 Advanced data processing

replicate(n, func(x=x))

apply(x,index,fun) apply function fun to all rows (index=1)
 or all columns (index=2) of the matrix x

 $\label{eq:lapply} \begin{aligned} \textbf{lapply}(\textbf{x}, & \textbf{fun}) \text{ or } \textbf{sapply}(\textbf{x}, & \textbf{fun}) \text{ apply function } \textbf{fun} \text{ to each element of the list } \textbf{x} \end{aligned}$ 

tapply(X,INDEX,FUN=) apply FUN to each cell of a ragged array given by X with indexes INDEX

 $\mathbf{by}(\mathbf{data}, \mathbf{INDEX}, \mathbf{FUN})$  apply  $\mathsf{FUN}$  to data frame  $\mathsf{data}$  subsetted by  $\mathsf{INDEX}$ 

ave(x,...,FUN=mean) subsets of x are averaged (or other function specified by FUN), where each subset consist of those observations with the same factor levels

merge(a,b) merge two data frames by common columns or row names

xtabs(a b,data=x) a contingency table from cross-classifying
factors

aggregate(x,by,FUN) splits the data frame x into subsets,
 computes summary statistics for each, and returns the
 result in a convenient form; by is a list of grouping elements, each as long as the variables in x

unstack(x, ...) inverse of stack()

reshape(x, ...) reshapes a data frame between 'wide' format with repeated measurements in separate columns of the same record and 'long' format with the repeated measurements in separate records; use (direction="wide") or (direction="long")

convolve(y,z) compute the several kinds of convolutions of
 two sequences

fft(z) Fast Fourier Transform of z

mvfft(Z) FFT of each column of a matrix

filter(z,filter) linear filtering of time series

#### 27 Dates and times

The class **Date** has dates without times. POSIXct has dates and times, including time zones. Comparisons (e.g. >), seq(), and difftime() are useful. Date also allows + and -. ?Date-TimeClasses 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).

**%H** Hours (00–23).

**%I** Hours (01–12).

%j Day of year (001–366).

**%m** Month (01−12).

**%M** Minute (00–59).

 $\mathbf{p}$  AM/PM indicator.

%S Second as decimal number (00–61).

%U Week (00–53); the first Sunday as day 1 of week 1.

%w Weekday (0–6, Sunday is 0).

%**W** Week (00–53); the first Monday as day 1 of week 1.

%y Year without century (00–99). Don't use.

**%Y** Year with century.

 $\%\mathbf{z}$  (output only.) Offset from Greenwich; -0800 is 8 hours west of.

%**Z** (output only.) Time zone as a character string (empty if not available).

#### 28Programming

function( arglist ) expr function definition
return(value)

fix(xfunc) opens R editor window for function xfunc

if(cond) expr

if(cond) cons.expr else alt.expr

for(var in seq) expr

while(cond) expr

repeat expr

break

#### next

stopifnot()

browser() and debug() functions

traceback() trace error

Use braces {} around statements

ifelse(test, yes, no) a value with the same shape as test filled with elements from either yes or no

do.call(funname, args) executes a function call from the name of the function and a list of arguments to be passed to it

Programming packages

tools Tools for Package Development

codetools Code Analysis Tools for R

Rprof() identify bottleneck

system.time() measure time elapsed

help.search("integration")

## 29 Optimization and linear programming

optimize() for one-dimensional optimization

nlm(f,p) minimize function f using a Newton-type algorithm with starting values p

#### 30 References

http://cran.r-project.org/ The CRAN site contains online R manuals and other documentation.

Becker, R.A., Chambers, J.M. and Wilks, A.R. (1988). The New S Language. Wadsworth & Brooks/Cole.

Braun, W.J., Murdoch, D.J. (2007). A First Course in Statistical Programming with R. Cambridge University Press, Cambridge.

Crawley, M.J. (2007). The R Book. Wiley, Chichester.

Dalgaard, P. (2002). Introductory Statistics with R. Springer, New York.

Maindonald, J. and Braun, W.J. (2003). Data Analysis and Graphics Using R. Cambridge University Press, Cambridge.

Murrell, P. (2005). R Graphics. Chapman & Hall/CRC, New York.

Venables, W.N. and Ripley, B.D. (2002). Modern Applied Statistics with S. Fourth Edition. Springer, New York.