

## R – principales fonctions

### Need help?

```
# open the help document for function ls
?ls
help(ls)
# search among the help pages for a word
help.search("trigo")
```

Variable information	
is.na(x), is.null(x), is.data.frame(x), is.numeric(x), is.character(x), ...	test for type; for a complete list, use methods(is)
nchar(x)	Number of a character string
print(a)	print to screen with different methods for different objects.
format(x,...)	format an R object for pretty printing

Vector 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; ":" has operator priority; 1:4 + 1 is "2,3,4,5"
seq(from,to,by), seq(from,to,length)	generates a sequence with by to specify increment or length for desired length
rep(x,times)	replicate x times; rep(c(1,2,3),2) is 1 2 3 1 2 3;
a:b	alternative way to create a sequence of integer from a to b. 1:5 is c(1,2,3,4,5)

Indexing vectors	
X[n]	nth element
X[-n]	all but the nth element
X[1:n]	first n elements
X[-(1:n)]	elements from n+1 to the end
X[c(1,4,2)]	specific elements
x["name"]	element named "name"
x[x > 3]	all elements greater than 3
x[x > 3 & x < 5]	all elements between 3 and 5
x[x %in% c("a","and","the")]	elements in the given set

Operations on vector	
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
round(x, n)	rounds the elements of x to n decimals
scale(x)	if x is a matrix, centers and reduces the data; to center only use the option center=FALSE, to reduce only scale=FALSE (by default center=TRUE, scale=TRUE)
pmin(x,y,...) , pmax(x,y,...)	a vector which ith element is the minimum (maximum ) of x[i], y[i], . . .
cumsum(x)	a vector which ith element is the sum from x[1] to x[i]
median(x)	median of the elements of x
quantile(x,probs=)	sample quantiles corresponding to the given probabilities withdefaults probs=c(0,.25,.5,.75,1)
weighted.mean(x, w)	mean of x with weights w
rank(x)	ranks of the elements of x

var(x), cov(x)	variance of the elements of x (calculated on n-1); if 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), 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

<b>Matrices</b>	
matrix(x,nrow=,ncol=)	matrix; elements of x recycle
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"
t(x)	transpose
diag(x)	diagonal
%*%	matrix multiplication
solve(a,b)	solves a %*% x = b for x
solve(a)	matrix inverse of a
rowsum(x)	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
colMeans(x)	id. for columns

<b>Data frames</b>	(matrix indexing plus the following)
data.frame(...)	create a data frame of the named or unnamed arguments.
expand.grid()	a data frame from all combinations of the supplied vectors or factors
x[["name"]]	column named "name"
x\$name	id.

<b>Lists</b>	
list(...)	create a list of the named or unnamed arguments; list(a=c(1,2),b="hi",c=3)
x[n]	list with elements n
x[[n]]	nth element of the list
x[["name"]]	element of the list named "name"
x\$name	id.

<b>Data manipulation</b>	
na.omit(x)	Suppresses the observations with missing data (NA) (suppresses the corresponding line if x is a matrix or a data frame)
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\$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, size)	resample randomly and without replacement size elements in the vector x, the option replace = TRUE allows to resample with replacement
factor(x,levels=)	encodes a vector x as a factor
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)
rbind(...)	combine arguments by rows for matrices, data frames, and others
cbind(...)	id. by columns
length(x)	number of elements in x
dim(x)	Retrieve or set the dimension of an object; dim(x) <- c(3,2)
dimnames(x)	Retrieve or set the dimension names of an object
nrow(x)	number of rows; NROW(x) is the same but treats a vector as a onerow matrix
ncol(x)	and NCOL(x) id. for columns

Input and output	
getwd(), setwd("dir")	filepath representing the current working directory of the R process; setwd(dir) is used to set the working directory to dir
dir()	show files in the current directory
load()	load the datasets (*.rda in the XDR platform independent binary format)
save(file,...)	save the specified objects (...) in the XDR format
save.image(file)	save all objects of the session
data(x)	load specified datasets
read.table("myfile", ...)	read a file in table format and creates a data frame. Use sep="" to define the separator; use header=TRUE to read the first line as a header of column names; use skip=n to skip n lines before reading data. See help(read.table) for other options.
read.csv("filename",header=TRUE)	idem but with defaults set for reading comma-delimited files
write.table(x,file=" myfile ", ...)	save x after converting to a data frame. See help(write.table) for options
read.delim("clipboard")	read / write a table from / to the clipboard for Excel
write.table(x,"clipboard",sep="\t",col.names=NA)	
	for database interaction, see packages RODBC, DBI, RMySQL, RPgSQL.

Statistics	
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
binom.test(), pairwise.t.test(), power.t.test(), prop.test(), t.test(), ...	statistical test. See help.search("test")
filter(x,filter)	applies linear filtering to a univariate time series or to each series separately of a multivariate time series. Many math functions have a logical parameter na.rm=FALSE to specify missing data (NA) removal.

Distributions	
	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 value of quantile (qfunc(p, ...), with $0 < p < 1$ ).
rnorm(n, mean=0, sd=1)	Gaussian (normal)
rexp(n, rate=1)	Exponential
rgamma(n, shape, scale=1)	Gamma
rpois(n, lambda)	Poisson
rweibull(n, shape, scale=1)	Weibull
rcauchy(n, location=0, scale=1)	Cauchy
rbeta(n, shape1, shape2)	Beta
rt(n, df)	'Student' (t)
rf(n, df1, df2)	Fisher-Snedecor (F) (c2)
rchisq(n, df)	Pearson
rbinom(n, size, prob)	Binomial
rgeom(n, prob)	Geometric

rhyper(nn, m, n, k)	Hypergeometric
rlogis(n, location=0, scale=1)	Logistic
rlnorm(n, meanlog=0, sdlog=1)	Lognormal
runif(n, min=0, max=1)	Uniform

Programming structure	Description
for(var in seq) expr	Loop for ... do ...
while(cond) expr	Loop while ... do ...
if(cond) expr	Condition if ... then ...
if(cond) cons.expr else alt.expr	Condition if ... then ... else ...
ifelse(test, yes, no)	a value with the same shape as test filled with elements from either yes or no
function( arglist ) expr	function definition. Use braces { } around statements
return(value)	at the end of a function, return an object

Advanced data processing	
apply(X,INDEX,FUN=)	a vector or array or list of values obtained by applying a function FUN to margins (INDEX) of X
lapply(X,FUN)	apply FUN to each element of the list X
tapply(X,INDEX,FUN=)	apply FUN to each cell of a ragged array given by X with indexes INDEX
by(data,INDEX,FUN)	apply FUN to data frame data subsetted by INDEX
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

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
boxplot(x)	box-and-whiskers plot
pairs(x)	if x is a matrix or a data frame, draws all possible bivariate plots between the columns of x
contour( z )	contour plot (data are interpolated to draw the curves, z must be a matrix)
	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, "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, "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)
points(x, y)	adds points (the option type= can be used)
lines(x, y)	id. but with lines
abline(a,b) , abline(h=y), abline(v=x), abline(lm.obj)	draws a line of slope b and intercept a, a horizontal line at ordinate y, a vertical line at abscissa x, the regression line given by lm.obj
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
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
par(...)	To set graphical parameters globally with many can be passed as parameters to plotting commands. See help(par) for a complete specification.

<b>Optimization and model fitting</b>	
optim(par, fn, method = "Nelder-Mead")	general-purpose optimization; par is initial values, fn is function to optimize (normally minimize) Other algorithm : method = c("Nelder-Mead", "BFGS", "CG", "L-BFGS-B", "SANN"). See Parameter estimation chapter
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 help(family)
nls(formula)	nonlinear least-squares estimates of the nonlinear model parameters. See Parameter estimation chapter
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(fit,...)	predictions from fit based on input data
coef(fit), residuals(fit), deviance(fit), fitted(fit), df.residual(fit), df.residual(fit)	the estimated coefficients (with their standard-errors), the residuals, the deviance, the fitted values, number of residual degrees of freedom
logLik(fit), AIC(fit)	computes the logarithm of the likelihood and the number of parameters, Akaike information criterion or AIC

install.packages("x",repos=getOption("repos"))	install package "x"
library(x)	load package (library) x