Funciones básicas de R

Obteniendo ayuda

help(topic) documentación sobre topic
?topic id.
help.search("topic") busca ayuda sobre objetos o funciones que contengan la cadena topic
help.start() arranca la versión de help en formato HTML
str(a) muestra la estructura interna del objeto
summary(a) obtiene el "resumen" de a, normalmente un resumen estadístico
ls() muestra los objetos existentes en memoria
dir() muestra los ficheros en el directorio actual

Entrada y salida

load() carga un conjunto de datos guardados con save
data(x) carga conjuntos de datos especificos
library(x) carga librerias adicionales
read.table(file) lee un fichero en formato de tabla y genera un data frame; el separador por defecto sep="" es cualquier espacio en blanco; usa header=TRUE para leer la primera línea como nombres de columnas; usa as.is=TRUE para impedir la conversión de los vectores de caracteres en factores; usa comment.char="" para impedir que # sea interpretado como comentario; usa skip=n para omitir n líneas antes de empezar a leer los datos; para más opciones, ver la ayuda
read.csv("filename",header=TRUE) id. pero con parametros por defecto para lectura de ficheros cuyos datos están separados por comas
save(file,...) guarda los objetos especificados (...)
save.image(file) guarda todos los objetos
cat(..., file="", sep=" ") imprime los argumentos despues de convertirlos a caracteres; sep es el carácter separador entre los argumentos
print(a, ...) imprime sus argumentos
write.table(x,file="",row.names=TRUE,col.names=TRUE, sep=" ") imprime x despues de convertirlo en data frame; si quote es TRUE, las variables de tipo caracter y factores se escriben entre comillas ("); sep es el separador de campo utilizado; eol es el carácter que indica final de línea; na es la cadena usada para valores missing

Creación de objetos

c(...) función genérica para concatenación de argumentos, formando por defecto un vector
from:to genera secuencias regulares; el operador ":" tiene prioridad sobre otros operadores; 1:4 + 1 es "2,3,4,5"
seq(from,to) genera secuencias; by= especifica el incremento; length= especifica la longitud deseada
rep(x,times) crea un vector con elementos idénticos; usa each= para repetir cada elemento de x each veces; rep(c(1,2,3),2) is 1 2 3 1 2 3; rep(c(1,2,3),each=2) es 1 1 2 2 3 3
data.frame(...) crea un marco de dstos (data frame) con argumentos nombrados o no; data.frame(v=1:4,ch=c("a","B","c","d"), n=10); los vectores más cortos son reciclados hasta llegar a la longitud del vector más largo
list(...) crea una lista con elementos nombrados o no; list(a=c(1,2),b="hi",c=3i)
matrix(x,nrow=,ncol=) genera una matriz; los elementos de x se pueden reciclar
factor(x,levels=) convierte el vector x en factor
rbind(...) combina argumentos por filas para crear matrices, data frames y otros
cbind(...) id. por columnas

Indexación de objetos

Indexación de vectores

x[n]	elemento en posición n
x[-n]	todo menos el elemento en la posicion n
x[1:n]	los primeros n elementos
x[-(1:n)]	elementos desde n+1 hasta el final
x[c(1,4,2)]	elementos específicos
x["name"]	elemento llamado "name"
x[x > 3]	elementos cuyo valor es mayor que 3
x[x > 3 & x < 5]	elementos cuyo valor está entre 3 y 5
x[x %in% c("a","and","the")]	elementos del conjunto especificado

Indexación de listas

x[n]	lista con elementos n
x[[n]]	elemento en posicion n de la lista
x[["name"]]	elemento de la lista llamado "name"
x$name	id.

Indexación de matrices

x[i,j]	elemento en fila i, columna j
x[i,]	fila i
x[,j]	columna j
x[,c(1,3)]	columnas 1 y 3
x["name",]	fila llamada "name"

Indexación de data frames (indexación de matrices más lo siguiente)

x[["name"]]	columna llamada "name"
x$name	id.

Conversión de variables

as.data.frame(x), as.numeric(x), as.logical(x), as.character(x), ... conversión entre diferentes tipos o clases de objetos; para ver la lista completa usa methods(as)

Información de variables

is.na(x), is.null(x), is.data.frame(x), is.numeric(x) is.character(x), ... comprueba si el objeto es del tipo definido por la función; para ver la lista completa usa methods(is)
length(x) número de elementos en x
dim(x) Obtiene o asigna la dimensión de los objetos; dim(x) <- c(3,2)
dimnames(x) Obtiene o asigna los nombres de las dimensiones de los objetos
nrow(x) número de filas
ncol(x) número de columnas
class(x) obtiene o asigna la clase de x; class(x) <- "myclass"
attr(x,which) obtiene o asigna el atributo which de x
attributes(obj) obtiene o asigna la lista de atributos de obj

Selección y manipulación de datos

sort(x) ordena los elementos de x en orden creciente; para ordenar en orden decreciente: rev(sort(x))
unique(x) si x es un vector o un data frame, devuelve un objeto similar pero suprimiendo los elementos duplicados
table(x) devuelve una tabla con el número de valores diferentes de x (típico para enteros o factores)
subset(x, ...) devuelve la parte de x obtenida según el criterio (..., comparaciones típicas: x$V1 < 10); si x es un un data frame, la opción select proporciona las variables que se obtienen (o se ignoran con -)

Funciones Matemáticas

sin,cos,tan,asin,acos,atan,atan2,log,log10,exp
max(x) el máximo de los valores de x
min(x) el mínimo de los valores de x
range(x) rango de x o c(min(x), max(x))
sum(x) suma de los elementos de x
diff(x) diferencia de los elementos de x
prod(x) producto de los elementos de x
mean(x) media de los elementos de x
median(x) mediana de los elementos de x
quantile(x,probs=) cuantiles; se corresponde con las probabilidades obtenidas (por defecto 0,.25,.5,.75,1)
var(x) o cov(x) varianza de los elementos de x (calculada en $n-1$); si x es matriz o data frame, calcula la matriz de varianza/covarianza
sd(x) desviación estándar de x
cor(x) matriz de correlación de x en el caso de ser matriz o data frame (1 si x es vector)
var(x, y) or cov(x, y) covarianza entre x e y, ó entre las columnas de x y las columnas de y si son matrices o data frames
cor(x, y) correlación lineal entre x e y, ó matriz de correlacion si son matrices o data frames
round(x) redondea los elementos de x a n cifras decimales
log(x, base) logaritmo de x en base base
cumsum(x) vector cuyo iavo elemento es la suma de los elementos de x[1] a x[i]

Muchas funciones matemáticas tienen el parámetro lógico na.rm=FALSE para eliminar los datos faltantes (NA).

Matrices

t(x) matriz traspuesta
diag(x) matriz diagonal
%*% producto de matrices
solve(a) inversa de la matriz a
rowsum(x) suma de filas
colsum(x) suma de columnas

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 a cada elemento de la lista X
merge(a,b) merge two data frames by common columns or row names

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

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

strsplit(x,split) split x according to the substring split

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

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

tolower(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

x %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 difftime() are useful.

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).
%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.
%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).

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

pie(x) circular pie-chart

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

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

The following parameters are common to many plotting functions:

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)

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)

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

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

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

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

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

Graphical parameters

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

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

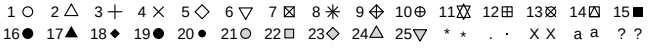
col controls the color of symbols and lines; use color names: "red", "blue" see colors() or as "#RRGGBB"; see rgb(), hsv(), gray(), and rainbow(); as for cex there are: col.axis, col.lab, col.main, col.sub

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

lty controls the type of lines, can be an integer or string (1: "solid", 2: "dashed", 3: "dotted", 4: "dotdash", 5: "longdash", 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

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



Lattice (Trellis) graphics

densityplot(~x) density functions plot

histogram(~x) histogram of the frequencies of x

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

Optimization 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

coef(fit) returns the estimated coefficients (sometimes with their standard-errors)

residuals(fit) returns the residuals

deviance(fit) returns the deviance

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()**, ... use help.search("test")

Distributions

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*) (χ^2)

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

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 (d*func*(x, ...)), the cumulative probability density (p*func*(x, ...)), and the value of quantile (q*func*(p, ...), with $0 < p < 1$).

Programming

function(arglist) expr function definition

return(value)

`if(cond) expr`
`if(cond) cons.expr else alt.expr`
`for(var in seq) expr`
`while(cond) expr`
`break`
`next`
Use braces {} around statements
`ifelse(test, yes, no)` a value with the same shape as `test` filled with elements from either `yes` or `no`