



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 específicos

library(x) carga librerías 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 parámetros 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 después 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` después de convertirlo en data frame; si `quote` es `TRUE`, las variables de tipo carácter 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)` es `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 datos (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]`

`x[-n]`

`x[1:n]`

`x[-(1:n)]`

`x[c(1,4,2)]`

`x["name"]`

`x[x > 3]`

`x[x > 3 & x < 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 posición `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.

elemento en posición `n`

todo menos el elemento en la posición `n`

los primeros `n` elementos

elementos desde `n+1` hasta el final

elementos específicos

elemento llamado "name"

elementos cuyo valor es mayor que 3

elementos cuyo valor está entre 3 y 5

elementos del conjunto especificado

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 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 correlación 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 `i`avo elemento es la suma de los elementos de `x[1: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

Procesamiento de datos avanzado

apply(X, INDEX, FUN=) aplica la función `FUN` a las filas o columnas (`INDEX`) de `X`

lapply(X, FUN) aplica `FUN` a cada elemento de la lista `X`

by(data, INDEX, FUN) aplica `FUN` a cada uno de los elementos clasificados por `INDEX` del data frame `data`

merge(a,b) fusiona dos data frames por columnas o nombres de filas comunes

Cadenas

paste(...) concatena vectores convirtiéndolos en caracteres; sep= es la cadena que separa los términos (el separador por defecto es el espacio en blanco); collapse= cadena opcional, que separa los componentes concatenados

substr(x,start,stop) extrae subcadenas de un vector de caracteres; también puede utilizarse en asignación, modificando la subcadena original,substr(x, start, stop) <- value

strsplit(x,split) divide x en campos según la cadena split

grep(pattern,x) localiza el patron pattern en x y devuelve las posiciones del vector donde aparece; see ?regex

gsub(pattern,replacement,x) sustituye todas las coincidencias en x que verifican un patron; sub() id. pero sustituye solo la primera coincidencia

tolower(x) convierte mayúsculas en minúsculas

toupper(x) convierte minúsculas en mayúsculas

match(x,table) devuelve un vector con las posiciones de las primeras coincidencias de x en table

x %in% table id. pero devuelve un vector lógico

nchar(x) número de caracteres

Gráficos

plot(x) representación de los valores de x (en el eje y) ordenados en el eje x

plot(x, y) gráfico bivariado de x (en el eje x) e y (en el eje y)

hist(x) histograma de frecuencias de x

barplot(x) histograma de los valores de x; usa horiz=FALSE para barras horizontales

pie(x) gráfico circular (o de sectores)

boxplot(x) diagrama de cajas (box-and-whiskers)

pairs(x) si x es una matriz o un data frame, dibuja todas las posibles gráficas bivariadas entre las columnas de x

qqnorm(x) cuantiles de x con respecto a los valores esperados bajo una distribución normal

qqplot(x, y) cuantiles de y frente a los cuantiles de x

Los parametros por defectos son comunes a muchas funciones gráficas:

type="p" especifica el tipo de gráfico, "p": puntos, "l": líneas, "b": puntos conectados por líneas, "o": id. pero las líneas están sobre los puntos, "h": líneas verticales, "s": escaleras, los datos se representan como la parte superior de las líneas verticales, "S": id. pero los datos se representan como la parte inferior de las líneas verticales

xlim=, ylim= especifica los límites inferiores y superiores de los ejes, por ejemplo con xlim=c(1, 10) o xlim=range(x)

xlab=, ylab= títulos de los ejes, deben ser variables de tipo caracter

main= título principal, debe ser una variable de tipo caracter

sub= sub-título (escrito en letra más pequeña)

Comandos de graficación de bajo nivel

points(x, y) añade puntos (se puede usar la opción type=)

lines(x, y) id. pero con líneas

text(x, y, labels, ...) añade el texto labels en las coordenadas (x,y); un uso típico es: plot(x, y, type="n"); text(x, y, names)

abline(a,b) dibuja una línea con pendiente b e intercepto a

abline(h=y) dibuja una línea horizontal en la ordenada y

abline(v=x) dibuja una línea vertical en la abcisa x

abline(lm.obj) dibuja una línea de regresión dada por lm.obj

rect(x1, y1, x2, y2) dibuja un rectángulo cuyas esquinas izquierda, derecha, superior e inferior son x1, x2, y1, y2, respectivamente

polygon(x, y) dibuja un polígono uniendo los puntos con coordenadas dadas por x e y

legend(x, y, legend) añade la legenda en el punto (x,y) con los símbolos dados por legend

title() añade un título y opcionalmente un sub-título

Parámetros gráficos

Se pueden modificar los valores de los parámetros globales mediante la función **par(...)**; muchos de ellos pueden pasarse como parámetros a los comandos de graficación.

adj justificación del texto (0 justificado a la izquierda, 0.5 centrado, 1 justificado a la derecha)

bg especifica el color del fondo (ej. : bg="red", bg="blue",... la lista con los 657 colores disponibles se puede ver con colors())

cex valor que controla el tamaño del texto y los símbolos con respecto al valor por defecto; los siguientes parámetros tienen el mismo control para números en los ejes, cex.axis, títulos de los ejes, cex.lab, título principal, cex.main y sub-título, cex.sub

col color de símbolos y líneas; usa los nombres de colores: "red", "blue" ver colors() o "#RRGGBB"; ver rgb(), hsv(), gray(), y rainbow(); como en cex estos son: col.axis, col.lab, col.main, col.sub

font entero que controla el estilo del texto (1: normal, 2: cursiva, 3: negrita, 4: negrita cursiva); como en cex estos son: font.axis, font.lab, font.main, font.sub

lty controla el tipo de las líneas, puede ser un entero o un caracter (1: "sólida", 2: "discontinua", 3: "punteada", 4: "punto-línea", 5: "línea larga-corta", 6: "línea corta-corta", o una cadena de hasta 8 caracteres (entre "0" y "9") que especifican de forma alternativa la longitud en puntos o píxeles, de los elementos dibujados y los blancos, por ejemplo lty="44" tendrá el mismo efecto que lty=2

lwd valor numérico que controla la anchura de la línea, por defecto 1

mar vector numéricos de 4 elementos, de la forma c(bottom, left, top, right), que controla el espacio entre los ejes y el borde del gráfico; los valores por defecto son c(5.1, 4.1, 4.1, 2.1)

mfcol Vector de la forma c(nr,nc) que divide la ventana gráfica como una matriz con nr filas y nc columnas; las gráficas se dibujan en las columnas

mfrow id. pero las gráficas se dibujan por filas

pch tipo de símbolo, puede ser un entero entre 1 y 25, o cualquier caracter único entre "

1○

2△

3+

4×

5◇

6▽

7⊠

8✱

9⬢

10⊕

11⊗

12⊞

13⊠

14⊞

15■

16●

17▲

18◆

19●

20●

21○

22□

23◇

24△

25▽

*

.

X

X

a

a

?

?

Optimización y ajuste del modelo

lm(formula) ajusta un modelo de regresión lineal; formula es típicamente de la forma response termA + termB + ...

glm(formula,family=) ajusta un modelo lineal generalizado; family es la descripción de la distribución del error y función de enlace para ser usada en el modelo; ver ?family

coef(fit) devuelve los coeficientes estimados (a veces con sus errores estándar)

residuals(fit) devuelve los residuales

deviance(fit) devuelve la suma de cuadrados residual

fitted(fit) devuelve los valores ajustados

AIC(fit) calcula el criterio de información de Akaike o AIC

Estadísticos

aov(formula) análisis de varianza

anova(fit,...) tabla de análisis de varianza (o devianza) de uno o más modelos lineales ajustados

binom.test(), pairwise.t.test(), power.t.test(), prop.test(), t.test(), ... usa help.search("test")

Distribuciones

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

rexp(n, rate=1) exponencial

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, shapel, 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) geométrica

rhyper(nn, m, n, k) hypergeométrica

rlogis(n, location=0, scale=1) logística

rlnorm(n, meanlog=0, sdlog=1) lognormal

rnbinom(n, size, prob) binomial negativa

runif(n, min=0, max=1) uniforme

rwilcox(nn, m, n), rsignrank(nn, n) estadístico de Wilcoxon

Todas estas funciones pueden ser usadas, sustituyendo el caracter r por d, p o q para obtener, la densidad de probabilidad (dfunc(x, ...)), la densida de probabilidad acumulada (pfunc(x, ...)), y el valor del cuartil (qfunc(p, ...), con 0 < p < 1).

Programación

function(arglist) expr definición de funciones

return(value)

if(cond) expr

if(cond) cons.expr else alt.expr

for(var in seq) expr

while(cond) expr

break

next

Utiliza llaves {} alrededor de las declaraciones

ifelse(test, yes, no) versión vectorizada de la estructura condicional if/else