

Inicio y salida de R

Pulsar en el icono de la aplicación o escribir R en la terminal

se puede utilizar de diferentes formas:

- Interactiva: órdenes en la línea de comandos, después de >
- Fichero de instrucciones
- Interfaces como Rcomander R-Studio, Awward, etc.

para salir del programa > q()

TIPOS DE OBJETOS EN R

Hay tres tipos básicos:

- **numérico**
 - Inf (ej: 3/0),
 - NaN (Not a Number) (ej: log(-1))
 - NA(Not Available) (para representar los datos que faltan)
- **lógico** (T o TRUE, F o FALSE)
- **cadenas de caracteres**

Ejemplo (“pepe”, ’casa’) (si se quiere utilizar el carácter “ dentro del texto deberá escribirse \”).

Asignar valores a variables

```
> a<- 3 # crea el elemento numérico a con valor 3  
> A<- 1e4 # crea el objeto numérico A con valor 10000.  
> a;A # R distingue las mayúsculas de las minúsculas  
> name <- "peso"  
> w<- a>A
```

Vectores

Construcción de vectores

Se puede hacer de varias maneras

- listando los elementos
 - > dado<- c("impar", "par","impar")
- Secuencias
 - > x<- 1:10 # lista de enteros del 1 hasta el 10
 - > y<- seq(1,10, by=2) # 1 3 5 7 9
 - > z<- seq(1,10, length=3) # 1.0 5.5 10.0
 - > w<- rep(3,5) # repite el número 3 cinco veces
- > union<- c(x,y,w) # une los vectores x, y, w

Referencia a los elementos de un vector:

se realiza indicando su posición entre []

```
> x[3] ; x[c(1,3)]
```

Eliminar elementos de un vector

Se indica su posición con signo menos

- > x[-1] # elimina el primer elemento del vector x
- > x[-c(1,3)] # ¿Qué resultado dará?

Operaciones sobre vectores:

+,-,*,/,[^]

Las operaciones se efectúan elemento a elemento

Define un vector con los impares del 1 al 10 y otro con los pares

Aplica las operaciones anteriores a los dos vectores

Funciones de vectores

- > `mean(x)` # calcula la media de los datos de x
- > `sd(x)` # calcula la desviación típica de los datos de x
- > `sqrt(x)` # calcula la raíz cuadrada de los elementos de x
- > `quantile(x, c(.25, .50, .75))` # calcula los percentiles 25, 50, 75

Matrices

Crear una matriz: se emplea el comando `matrix`

`matrix(datos, nrow=nfilas, ncol=ncolumnas, byrow=F)`
datos= datos de la matriz; nfilas=número de filas, ncolumnas=número de columnas,
byrow=F (opción por defecto) indica que los datos se van colocando por columnas.

```
> A<- matrix(c(1:10),nrow=2, ncol=5, ); A  
> B<- matrix(c(11:20),5,2); B
```

Referencia a elementos de la matriz

```
> A[1,3]; A[, 2]; A[1,]; A[1,c(1,3,5)]
```

Eliminar filas o columnas

se pone el subíndice correspondiente en negativo

```
> A<-matrix(1:6,2); A[-1]
```

Operadores

Los operadores +, -, *, / entre matrices actúan elemento a elemento.

%*% producto de usual de matrices

```
> A<-matrix(1:6,2); B<-matrix(2,2,3); B*A  
> B%*%t(A) # t(A) es la traspuesta de la matriz A
```

Unir matrices

```
> C1<- cbind(A,B); C1 # Une las matrices A y B por columnas
```

```
> C2<- rbind(A,B); C2 # Une las matrices A y B por filas
```

Gráficos

El comando `plot` sirve para hacer gráficas de una función evaluada en un conjunto de puntos

```
> x<- seq(0, 5, length=100); y<- x^2  
> plot(x,y)
```

Distribuciones de probabilidades

Con R se pueden realizar las siguientes tareas con las distribuciones de probabilidad:

- Cálculo de función de densidad (continuas) o de probabilidad (discretas)
 - > `dbinom(5, 7, .5)` # P(Binomial(7,.5) = 5)
- Cálculo de función de función de distribución,
 - > `ppois(4, 3)` # P[Poisson(landa=3 <= 4)]
- Cálculo de los cuantiles (Inversa de la función de distribución)
 - > `qnorm(0.5, 3, 2)` # El cuantil 0.5, o percentil 50, de la N(3,2)
- Generación de números aleatorios
 - > `rnorm(5, mean=0, sd=1)` # Generación de 5 datos de la normal estandar

Guardar datos

Las formas más sencillas de salvar un fichero de datos son:

- `write.table(x, file="data.txt")` # guarda los datos de x en el fichero “data” con formato texto
- `save(x, file= "datosx.Rdata")` # guarda los datos de x en “datosx” con formato de R

Listado y características de los objetos

Todo objeto tiene dos atributos intrínsecos: longitud **length()** y tipo **mode()**.

> **length(dado); mode(dado)**

ls() se usa para ver los objetos que están almacenados en memoria,

> **ls(pat='b')** # lista los objetos que tienen una b en el nombre

ls.str() muestra la estructura de todos los objetos en memoria

rm() sirve para borrar objetos,

> **rm(B)** # borra el objeto B

> **rm(list=ls())** # borra todos los objetos en memoria

Ayuda

Las órdenes **help()** o **? o ??** sirven para pedir ayuda sobre una función o procedimiento de R

> **?sqrt**

> **help(sqrt)**

Trabajo con ficheros de instrucciones (scripts)

Linux

Etapas para trabajar con ficheros

- Abrir un editor, por ejemplo gedit
- Escribir las instrucciones y grabar el fichero (con extensión .R)
- Ejecutar en R la instrucción **source("dirección y nombre del fichero")**
Por ejemplo para trabajar con el fichero "p1.R" de la carpeta "IE_16-17"
> **source("IE_16-17/p1.R")**

Windows

Etapas para trabajar con ficheros

- Pulsar en el escritorio el icono de la aplicación R, si no está ya abierta.
- Seleccionar el menú **Archivo** y abrir un fichero (Abrir script) o crear uno nuevo (nuevo script)
- Escribir las instrucciones de R, al grabar se hará con extensión .R
- Marcar los comandos que se quieran ejecutar y pulsar simultáneamente las teclas "Control" y "R", o seleccionar el menú editar de esta ventana y ejecutar línea o selección.