

PRÁCTICA 1 DE PROGRAMACIÓN

INTRODUCCIÓN

¿Qué es R?

El proyecto R es un conjunto de programas que se puede utilizar también como lenguaje de programación, además de otras utilidades como representaciones gráficas de funciones, lectura de datos y tratamiento de estos.

Cómo guardar variables

La consola se puede utilizar a modo de calculadora. Si hacemos 3+2 nos saldrá el resultado.

En el caso de que queramos que el resultado quede guardado en una variable. Para ello:

n<- 3+2

sí lo pusiésemos al revés nos daría error:

3+2->n Nos daría error ya que lo que escribimos a la derecha del igual o flecha se guarda en la izquierda.

Las variables donde almacenamos datos pueden ser numéricas o alfanuméricas:

A7<-3+2

Obtener datos al azar

- Para obtener datos al azar usamos **rnorm(n)** poniendo en n el número de datos que queremos que nos dé.

Ejemplo 1

rnorm(1) Solo nos aparecerá un valor

rnorm(5) Nos aparecen 5 valores

- Para obtener valores al azar entre [0,1] usamos la función **runif(n)** siendo n el número de datos que queremos obtener

Funciones de memoria

Para que el programa nos recuerde el nombre de las variables creadas usaremos la función **ls()**

En el interior del paréntesis si no ponemos nada nos dará todas las variables creadas.

En el caso de introducir dentro del paréntesis la función **pat='k'** siendo k una letra cualquiera nos recordará todas las variables que **contienen** esa letra. Si ponemos **pat='^k'** nos dará las variables que **comienzan** por esa letra.

En el caso de querer saber además del nombre el valor (ya sea numérico o alfanumérico) usamos **ls.str()**

Si queremos borrar un elemento usaremos **rm()** introduciendo en el interior del paréntesis el nombre de la variable que queremos borrar.

Operaciones

- **Suma**: Usaremos el símbolo **+**
- **Resta**: Usaremos el símbolo **-**
- **Multiplicación**: Usaremos el símbolo *****
- **División**: Usaremos el símbolo **/**
- **Elevar una potencia**: Usaremos el símbolo ******

¡RECUERDA EL ORDEN DE LAS OPERACIONES!

Potencias

Productos
y divisiones

Sumas y
restas

Símbolos

- **<** Menor
- **>** Mayor
- **<=** Menor o igual
- **>=** Mayor o igual
- **!=** Distinto
- **==** Igualdad lógica
- **Pi** representa al número pi
- **exp(1)** representa el número e
- **7e4** representa a 7×10^4

Funciones

- **log(x)** logaritmo neperiano.
- **log10(x)** logaritmo en base 10.
- **log2(x)** logaritmo en base 2.
- **Log (x, base)** logaritmo en cualquier base.
- **exp(x)** función exponencial.
- **sin(x)** seno.
- **cos(x)** coseno.
- **tan(x)** tangente.
- **asin(x)** arco seno.
- **acos(x)** arco coseno.
- **atan(x)** arco tangente.
- **abs(x)** valor absoluto.
- **sqrt(x)** raíz cuadrada.
- **factorial(x)** factorial.
- **choose(n,x)** binomio de Newton n sobre x

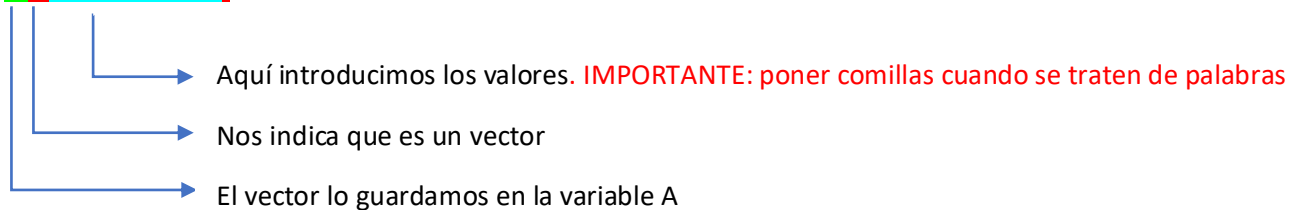
Vectores y matrices

La estructura para definir un vector es `c()`

Los vectores pueden almacenar valores numéricos o palabras.

Ej:

```
A<-c(8, 20, 'amigos')
```



Para acceder a una de las componentes del vector escribimos el nombre de este y entre corchetes la posición de la componente `A[n]` siendo n la posición

Se pueden realizar operaciones con los vectores.

Para hacer el **producto escalar**: `vector1%*%vector2`

La estructura para definir una matriz es `matrix()`

Además, para definir el número de filas y columnas se usan `nrow` y `ncol` respectivamente

También se puede generar una matriz empleando los comandos `cbind` y `rbind` para generar una matriz combinando vectores en forma de columnas o vectores en forma de fila respectivamente. (vectores previamente creados).

Se pueden realizar operaciones con las matrices.

ATENCIÓN: Para hacer el **producto** no se utiliza `*` sino: `%*%`

Otras operaciones con matrices:

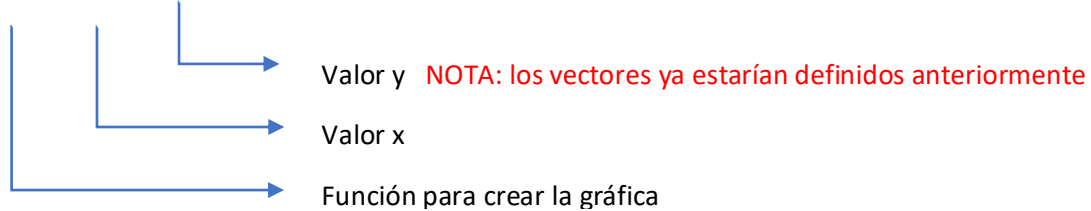
- `t(A)`: transpuesta de la matriz A.
- `solve(A,b)`: solución del sistema de ecuaciones $Ax=b$.
- `solve(A)`: inversa de la matriz A.
- `svd(A)`: descomposición en valores singulares.
- `qr(A)`: descomposición QR.
- `eigen(A)`: valores y vectores propios.
- `diag(b)`: matriz diagonal (b es un vector).
- `diag(A)`: matriz diagonal (A es una matriz).

Tablas y gráficas

Usamos **data.frame()** para almacenar datos en forma de tabla

Usamos **plot()** para crear gráficas

Plot(meses, casas)



Podemos unir los puntos mediante una recta, colocar títulos a los ejes y al gráfico, color del gráfico mediante:

`plot(meses, casas, type="", xlab="", ylab="", col="", main="")`

Type="" Es el tipo de gráfico que puede ser:

- **Type='l'** une los puntos mediante una línea
- **Type='b'** Aparecen tanto líneas como puntos
- ...

Xlab="" Colocar título al eje x

Ylab="" Colocar título al eje y

Col="" Para darle color al gráfico (los nombres de los colores en inglés)

Main="" Para poner título al gráfico

Representación de dos curvas en un mismo gráfico.

Se utiliza la estructura:

plot()

par(new=TRUE)

plot()

Para generar una secuencia de n valores de x entre a y b: `x=seq(a,b,length=n)`

Siendo a, b y n números

ATENCIÓN CON LA PALABRA LENGTH

Podemos acudir a la ayuda de R para obtener más información sobre cualquier comando, para ello ponemos delante del comando una interrogación. Por ejemplo: `?plot`

