

# Comandos Básicos

## Software Estadístico

Dra. Eva Romero Ramos

Dpto. Estadística e Investigación Operativa

- En R, los datos se organizan y manipulan a través de objetos, que son estructuras que contienen datos y funciones asociadas a ellos.
- Los objetos pueden tener diferentes clases y tipos de datos, y R proporciona métodos para trabajar con estos objetos y realizar operaciones específicas para cada tipo.
- Un objeto R pueden contener datos numéricos, caracteres, factores, vectores, matrices, listas, data frames, entre otros tipos de datos.
- Durante una sesión de r se crean diferentes objetos y que se almacenan identificados por su nombre.

- Para nombrar objetos en R se pueden usar todos los símbolos alfanuméricos, además del punto y el guión.
- Hay que tener en cuenta que R es sensible a mayúsculas y minúsculas (A y a son símbolos diferentes).
- La única restricción para nombrar objetos es que un nombre debe comenzar por punto o por letra, y si comienza por punto el siguiente carácter debe ser una letra.

Los comandos elementales pueden ser expresiones o asignaciones:

- 1 En un comando de expresión se evalúa el resultado y se muestra en consola, pero no se guarda el valor obtenido.

**Ejemplo.-**  $10/4$

- 2 En un comando de asignación se evalúa el resultado y se almacena en una variable, pero se muestra en consola.

**Ejemplo.-**  $Div < -10/4$

La asignación se puede hacer mediante los simbolos  $<$   $-$  o  $=$ .

- Los comandos se pueden separar por punto y coma o por salto de línea. Y se pueden agrupar entre llaves.
- Se pueden añadir comentarios en cualquier parte del código empezando por el símbolo `#`.
- Si ejecutamos un comando incompleto aparecerá el símbolo `+` en consola, solicitándonos completarlo. En estos casos se puede abortar la ejecución pulsando la tecla `ESC`.

# Comandos básicos relacionados con el directorio de trabajo

- **getwd()**: Muestra el directorio de trabajo.
- **setwd("ruta del directorio de trabajo" )**: Define o fija la ruta del directorio de trabajo.
- **dir()**: Muestra el contenido del directorio de trabajo.
- **ls()** o **objects()**: Muestra el listado de objetos que tenemos en el directorio de trabajo.
- **rm(objeto)**: Borrar el objeto indicado del directorio de trabajo.
- **q()**: Cierra el programa.

- **help(comando o funcion) o ?comando o funcion** : Nos muestra ayuda sobre el comando o función que queramos conocer.

**Ejemplo.- help(mean)**

- **help.start()**: Proporciona ayuda en general. Nos da acceso a los diferentes manuales y la documentación sobre paquetes r.
- **example(comando o funcion)**: Nos da un ejemplo de uso del comando o función que nos interese.

**Ejemplo.- example(mean)**

# Gestión de paquetes

- Los paquetes juegan un papel fundamental en R, permitiendo añadir numerosas funcionalidades al software.
- La gestión de paquetes se puede realizar desde la pestaña **Packages** en la ventana inferior izquierda.
- Algunos comandos importantes relacionados con la gestión de paquetes son:

- **install.packages(nombre\_del\_paquete)**: Permite instalar un paquete.

**Ejemplo.- install.packages(e1071)**

- **library(nombre\_del\_paquete)**: Carga la librería para que podamos usar sus funcionalidades.
- **search()**: Lista los paquetes cargados.
- **library(help = nombre\_del\_paquete)**: Muestra la ayuda asociada al paquete.
- **data()**: Muestra los conjuntos de datos que contiene el paquete.



# Tipos de datos

Los tipos de datos básicos de R son:

- **integer**: Números enteros (Ejemplo.- 3).
- **numeric**: Números reales (Ejemplo.- 3.42).
- **character**: Secuencias de caracteres. Se deben expresar entre comillas simples o dobles (Ejemplo.- "Hola").
- **logical**: Toma dos posibles valores TRUE o FALSE. Se usarán entre otras cosas para evaluar condiciones. (Ejemplo.- TRUE).

Cuando una variable no tiene asignado valor tomará valor **NA**.

Si al intentar recuperar un dato no se encuentra nada se obtendrá **NULL**.

Para ver el tipo de datos de una variable podemos usar la función (**class**).

# Ejemplo de definición de variables con de los tipos básicos

```
x <- - 3 ; y <- - 7  
class(x); class(y)
```

```
z <- - x < y  
z; class(z)
```

```
z2 <- - "x < y"  
z2; class(z2)
```

```
t <- - x/y  
t; class(t)
```

```
w <- - 7/0
```

# Operadores básicos

Aritméticos		Relacionales	
Suma / resta	+ y -	Mayor / Mayor o igual	> y >=
Multiplicación / División	* y /	Menor / Menor o igual	< y <=
Potencia	^	Igual que	==
Resto	%%	Distinto de	!=
Lógicos		De asignación	
o		Asignar valor a una variable	< - ó =
y	&		
Negación	!		
Es verdadero	isTRUE		
Es falso	isFALSE		

- Coercionar datos implica transformarlos de un tipo a otro.
- La coerción resulta ser una operación imprescindible cuando tratamos utilizar funciones sobre datos de un tipo que no admiten.
- Siempre que ejecutamos una operación en R, el software tratará de coercionar los datos para ofrecer un resultado.
- Las coerciones se dan en el siguiente orden y no pueden darse en orden inverso:  
logical → integer → numeric → character
- Para coercionar se puede utilizar las siguientes funciones **as()**:  
as.integer(), as.numeric(), as.character(), as.factor(),  
as.logical(), as.null()
- Estas funciones admiten datos y vectores como argumento.

# Ejemplos de coerción

```
x_char <- as.character(x)  
x_char; class(x_char)
```

```
x_numeric <- as.numeric(x)  
x_numeric; class(x_numeric)
```

```
z_integer <- as.integer(z)  
z_integer; class(z_integer)
```

# Verificación del tipo de datos

- Ya hemos visto que para verificar el tipo de datos podemos usar la función `Class()`.
- También podemos usar funciones del tipo `is()` como son:  
`is.integer()`, `is.numeric()`, `is.character()`, `is.factor()`,  
`is.logical()`, `is.na()`, `is.null()`
- Estas funciones devuelven el valor `TRUE` o `FALSE` dependiendo de si se verifica que el dato es del tipo indicado o no.