

COMANDOS BÁSICOS USADOS EN R

***Dataframe**: una estructura de datos en R que se utiliza para almacenar y organizar datos en forma de tabla. Está compuesto por filas y columnas, donde cada columna puede contener datos de un tipo diferente (números, caracteres, fechas, etc.), pero todas las filas deben tener la misma longitud.

Manipulación de Datos

- `head(datos)`: Muestra las primeras filas de un dataframe.
- `tail(datos)`: Muestra las últimas filas de un dataframe.
- `nrow(datos)` y `ncol(datos)`: Número de filas y columnas. Lo igualas al número de filas o columnas que quieres que tenga tu matriz por ejemplo.

Manipulación de Vectores y Matrices

- `c(1, 2, 3)`: Crea un vector. La c es para concatenar.
- `seq(1, 10, by=2)`: Secuencia de números. Pones en la primera posición el valor en el que empiezas, luego en el que acabas y por último especificas el incremento o decremento que va realizar la secuencia con "by 2" por ejemplo. Con "length.out" puedes especificar el número total de elementos en la secuencia y con "along.with" creas una secuencia con la misma longitud que otro objeto.
- `seq_along()`: genera una secuencia de enteros desde 1 hasta la longitud que se indique.
- `rep(1:3, times=2)`: Repite elementos.
- `matrix(1:9, nrow=3, ncol=3)`: Crea una matriz. Al poner en este caso 1:9 genera una secuencia de números del 1 al 9. También puedes poner un 0 para inicializar la matriz a cero y que así se vaya rellenando luego con las fórmulas que indiques.
- `apply(matriz, 1, sum)`: Aplica una función por filas (1) o columnas (2). En este caso la función que se va a aplicar es la suma (sum). Se usa con matrices y arrays (estructuras de datos que permiten almacenar información en más de dos dimensiones).
- `sapply()`: Se usa con vectores o listas (más versátil, para diferentes estructuras de datos).

Control de Flujo

- `if (condición) { ... } else { ... }`: Estructura condicional.
- `for (i in 1:10) { ... }`: Bucle for.
- `while (condición) { ... }`: Bucle while.
- `break` y `next`: Interrumpen o continúan bucles.

*Para más información sobre bucles consultar el [recurso de Bucles](#).

Salida de datos

- `print()`: Es la función más básica para mostrar información en la consola. Se usa para imprimir variables, objetos y mensajes. Se puede usar con "paste" imprimir palabras y números (esto no hace falta con "cat").

```
Ej: nombre <- "Ana"
     edad <- 30
     print(paste("Mi nombre es", nombre, "y tengo", edad, "años"))
     # Salida: [1] "Mi nombre es Ana y tengo 30 años"
```

- **cat()**: Permite concatenar e imprimir múltiples elementos, ofreciendo más control sobre el formato de salida.
 - o **cat("\014")**: se utiliza para limpiar la consola, muy útil para ver las cosas más claras.
- **stop()**: Muestra el mensaje de error y deja de evaluar la función.

```
Ej: divide <- function(x, y) {
  if (y == 0) {
    stop("Error: División por cero no permitida")
  }
  return(x / y)
}
```

Funciones

- **function(x) { ... }**: Crea una función personalizada.

IMPORTANTE: no olvidar los corchetes al principio y al final.

```
Ej: suma <- function(a, b) { return(a + b) } suma(2, 3) # Devuelve 5
```

Dibujar en R

1. Gráficos Básicos con la Función plot()

La función `plot()` es la forma más sencilla de crear gráficos en R.

```
# Crear un gráfico de dispersión básico
x <- c(1, 2, 3, 4, 5)
y <- c(2, 4, 6, 8, 10)
plot(x, y)
```

Puedes personalizar el gráfico usando parámetros adicionales como `main` (título del gráfico), `xlab` y `ylab` (etiquetas de los ejes).

```
plot(x, y, main = "Gráfico de Dispersión", xlab = "Eje X", ylab = "Eje Y", col = "blue", pch = 16)
```

- `col`: Define el color de los puntos.
- `pch`: Define el tipo de marcador (por ejemplo, 16 es un círculo sólido).

2. Gráfico de Líneas

Puedes usar `plot()` para crear gráficos de líneas. Esto se hace añadiendo el parámetro `type = "l"`.

```
# Crear un gráfico de líneas
plot(x, y, type = "l", col = "red", main = "Gráfico de Líneas", xlab = "Eje X",
ylab = "Eje Y")
```

Ejemplo de programa:

```
# Pinta los resultados
plot(x, y, type = "l", col = "blue", lwd = 2, xlab = "x", ylab = "Derivada
primera", main = "Derivada primera: Analítica vs Numérica")

lines(x_vals, estimated_first_derivative, col = "red", lwd = 2)
```

```
legend("topright", legend = c("Analítica", "Numérica"), col = c("blue", "red"),  
lty = 1, lwd = 2)
```

RESUMEN DE COMANDOS USADOS PARA DIBUJAR EN R

- `plot()`: crear gráficos
- `main`: indicar el título del gráfico
- `lines()`: se utiliza para añadir líneas a un gráfico existente.
- `legend`: crea una leyenda
 - o `"topright"`: especifica la posición de la leyenda en la esquina superior derecha del gráfico
- `type="p"`: va a dibujar puntos
 - o `pch`: especifica el estilo de los puntos
 - o `cex`: sirve para modificar el tamaño de los puntos
- `type="l"`: es para que dibuje una línea
 - o `col`: para indicar el color del que queremos la línea
 - o `lwd=3`: indica el grosor de la línea
 - o `lty=2`: nos indica que va a ser una línea discontinua

*Los números a lo que se iguala `lwd` o `lty`, por ejemplo, se corresponden con una serie de números, que van a indicar línea continua, discontinua, más gruesa o más fina.