

## UNIDAD N° 1

# ESTADÍSTICA DESCRIPTIVA

Variables, tipos de datos, estructuras de  
datos básicas en R

## INTRODUCCIÓN

R es un lenguaje de programación estadístico que permite manipular, visualizar y analizar datos de manera eficiente. En esta guía, aprenderás a trabajar con los primeros conceptos esenciales como definición de variables, reconocimiento de tipos de datos y uso de las estructuras básicas de almacenamiento que ofrece R.

Esta guía está pensada para estudiantes como vos, que ya cuentan con experiencia previa en programación, por lo que se hace foco en las particularidades del lenguaje R más que en la lógica de programar en sí.

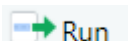
### ¿Cómo ejecutar código en RStudio?

Para probar los ejemplos de esta guía necesitás ingresar el código en la Consola de RStudio o escribirlo en un Script (archivo .R).

#### Opción 1: Usar la Consola

1. Abrí RStudio.
2. Buscá el panel que dice “Console” (abajo a la izquierda).
3. Escribí directamente el código y presioná “Enter” para ejecutarlo.

#### Opción 2: Trabaja con un Script

1. Hacé clic en File > New File > R Script.
2. Escribí ahí tu código. Para ejecutarlo:
  - Podés seleccionar una línea y presionar “Ctrl + Enter” (o “Cmd + Enter” en Mac).
  - O hacer clic en el botón  de la parte superior del script.

Esto permite guardar el código, reutilizarlo y trabajar de forma más ordenada, es la opción que utilizaremos.

## VARIABLES EN R

En R, una variable es un objeto que almacena un valor. Las variables pueden contener números, texto, vectores, tablas u otras estructuras de datos.

La asignación de valores a variables se realiza generalmente con el operador `<-` aunque también puede usarse `=`. Sin embargo, la convención en R es utilizar `<-`

Ejemplo:

```
x <- 10
nombre <- "Juan"
pi_valor <- 3.1416
```

Consejo: Los nombres de variables no deben comenzar con números ni contener espacios. Se recomienda usar letras minúsculas y, en todo caso, guion bajo `_` para separar palabras.

## TIPOS DE DATOS BÁSICOS EN R

R distingue entre diferentes tipos de datos atómicos, que son los bloques fundamentales con los que se construyen estructuras más complejas:

**Numérico:** números reales (decimales o enteros). Ejemplo: `edad <- 25`

**Caracter** (character): texto o cadenas de caracteres.

Ejemplo: `ciudad <- "Rosario"`

**Lógico** (logical): valores TRUE o FALSE. Ejemplo: `aprobado <- TRUE`

**Entero** (integer): números enteros explícitos. Se definen agregando una L. Ejemplo:

```
n <- 100L
```

**Complejo** (complex): números complejos. Ejemplo: `z <- 1 + 2i`

Para conocer el tipo de dato de una variable, usa:

```
class(x)
typeof(x)
```

## ESTRUCTURAS DE DATOS BÁSICAS EN R

### Vectores

Son la estructura más simple y fundamental. Almacenan varios elementos del mismo tipo (números, caracteres, lógicos, etc.). Se crean utilizando la función `c()`, que significa *combine* (“combinar”), y que agrupa los elementos entre paréntesis.

Ejemplo:

```
numeros <- c(2, 4, 6, 8)
nombres <- c("Ana", "Luis", "Marta")
logicos <- c(TRUE, FALSE, TRUE)
```

Para acceder a un elemento del vector:

```
numeros[2] # Devuelve 4 (el segundo elemento)
```

## Listas

Permiten almacenar diferentes tipos de datos juntos (números, textos, otros vectores, etc.).

Ejemplo:

```
mi_lista <- list(nombre = "Carlos", edad = 22, aprobado = TRUE)
```

Para acceder a un elemento por nombre:

```
mi_lista$nombre # Devuelve "Carlos"
```

## Matrices

Estructuras bidimensionales de datos del mismo tipo (como una tabla de filas y columnas).

Ejemplo:

```
matriz <- matrix(1:6, nrow = 2, ncol = 3)
```

Resultado

```
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

Para acceder a un elemento de la matriz:

```
matriz[1, 2] # Fila 1, columna 2: valor 3
```

## Arreglos (Arrays)

Son estructuras multidimensionales (más de dos dimensiones).

Ejemplo:

```
arreglo <- array(1:8, dim = c(2, 2, 2))
```

Aquí se crea un arreglo de 2 x 2 x 2.

## Dataframes

Son la estructura más utilizada en análisis de datos.

Permiten almacenar datos en forma de tabla, con columnas que pueden ser de distintos tipos.

Ejemplo:

```
datos <- data.frame(  
  nombre = c("Ana", "Luis", "Marta"),  
  edad = c(23, 31, 28),  
  aprobado = c(TRUE, FALSE, TRUE)  
)
```

Acceso a una columna:

```
datos$edad
```

Acceso a una celda específica:

```
datos[2, "nombre"] # Segundo registro, columna "nombre"
```

## Factores

Se utilizan para representar variables categóricas (como niveles, categorías o clases).

Son muy útiles para análisis estadístico y creación de gráficos.

Ejemplo:

```
genero <- factor(c("F", "M", "F", "M", "F"))  
levels(genero) # Devuelve los niveles: "F" y "M"
```

Dominar estas estructuras es fundamental para cualquier análisis en R. En unidades siguientes trabajarás con funciones de resumen y visualización de datos, por lo que

entender cómo se organizan y se accede a los datos es el primer paso para aprovechar todo el potencial del lenguaje.

Te sugerimos experimentar en la consola de RStudio, modificar los ejemplos y observar los resultados. La práctica activa te permitirá incorporar con mayor facilidad los conceptos.