

Elementos Básicos de R

Jhonier Rangel

10 de junio de 2025

¿Qué es R?

R es un lenguaje de programación y entorno estadístico. Es ampliamente utilizado para:

- ▶ Análisis estadístico
- ▶ Ciencia de datos
- ▶ Visualización

Vectores

Un vector es una colección de elementos del mismo tipo.

```
1 # Vector num rico
2 numeros <- c(1, 2, 3, 4.5)
3
4 # Vector de caracteres
5 nombres <- c("Ana", "Luis", "Pedro")
6
7 # Acceder a elementos
8 numeros[2]
```

Tipos de Variables

R tiene varios tipos de datos básicos:

```
1 # Num rico
2 x <- 42
3
4 # Entero
5 y <- as.integer(42)
6
7 # L gico
8 z <- TRUE
9
10 # Car cter
11 nombre <- "Juan"
12
13 # Verificar tipo
14 class(x)
```

Matrices

Una matriz es una colección bidimensional de datos del mismo tipo.

```
1 # Crear una matriz 3x3
2 m <- matrix(1:9, nrow=3, ncol=3)
3
4 # Acceder a elementos
5 m[1,2]
```

Arreglos (Arrays)

Los arreglos pueden tener más de dos dimensiones.

```
1 # Crear un arreglo 2x2x2
2 a <- array(1:8, dim = c(2, 2, 2))
3
4 # Acceder a elementos
5 a[1,1,2]
```

Data Frames

Estructura similar a una tabla. Puede contener diferentes tipos de datos por columna.

```
1 # Crear un data frame
2 df <- data.frame(
3   nombre = c("Ana", "Luis"),
4   edad = c(23, 31),
5   aprobado = c(TRUE, FALSE)
6 )
7
8 # Acceder a columnas
9 df$nombre
```

Listas

Las listas pueden contener diferentes tipos de objetos.

```
1 # Crear una lista
2 mi_lista <- list(nombre="Ana", edad=23, notas=c(10, 9,
3           8))
4 # Acceder a elementos
5 mi_lista$edad
```


Funciones en R

Definir y utilizar funciones propias:

```
1 # Definir una funci n
2 suma <- function(a, b) {
3   return(a + b)
4 }
5
6 # Llamar la funci n
7 suma(4, 5)
```

Condicionales y Bucles

Ejemplo de condicional y bucle for:

```
1 # Condicional
2 if (x > 10) {
3     print("Mayor que 10")
4 } else {
5     print("Menor o igual que 10")
6 }
7
8 # Bucle for
9 for (i in 1:5) {
10     print(i)
11 }
```

¿Qué es ggplot2?

ggplot2 es un paquete del tidyverse para crear gráficos de alta calidad.

- ▶ Basado en la "Gramática de los gráficos"
- ▶ Separación entre datos, estética y geometría
- ▶ Altamente personalizable

Instalación y carga

Antes de usar ggplot2, debe instalarse y cargarse:

```
1 # Instalar (solo una vez)
2 install.packages("ggplot2")
3
4 # Cargar la librería
5 library(ggplot2)
```

Sintaxis básica

Estructura general de un gráfico en ggplot2:

```
1 ggplot(data = df, aes(x = var1, y = var2)) +  
2   geom_point()
```

- ▶ data: el data frame
- ▶ aes: mapea variables estéticas (ejes, color, etc.)
- ▶ geom_point(): geometría (tipo de gráfico)

Gráfico de dispersión: geom_point

```
1 ggplot(data = mtcars, aes(x = wt, y = mpg)) +  
2   geom_point()
```

Resultado: gráfico de mpg vs peso (wt) del automóvil.

Gráfico de líneas: geom_line

```
1 ggplot(data = economics, aes(x = date, y = unemploy))  
  +  
2  geom_line()
```

Resultado: evolución del desempleo en el tiempo.

Gráfico de barras: geom_bar

```
1 ggplot(data = diamonds, aes(x = cut)) +  
2   geom_bar()
```

Resultado: cantidad de diamantes por tipo de corte.

Personalización de gráficos

Puedes añadir títulos, etiquetas y temas:

```
1 ggplot(mtcars, aes(x=wt, y=mpg)) +  
2   geom_point() +  
3   labs(title = "MPG vs Peso",  
4         x = "Peso (wt)", y = "Consumo (mpg)") +  
5   theme_minimal()
```

Facetas y colores

Dividir gráficos o agrupar por categoría:

```
1 ggplot(mpg, aes(x=displ, y=hwy, color=class)) +  
2   geom_point() +  
3   facet_wrap(~ class)
```

Facetas: permite comparar entre grupos de forma clara.

Conceptos Básicos de Estadística

La estadística es la ciencia que se encarga de recolectar, analizar e interpretar datos para tomar decisiones.

Población y Muestra

Población: Conjunto completo de elementos o individuos que comparten una característica común.

Muestra: Subconjunto representativo extraído de la población para su estudio.

Ejemplo: Población = todos los estudiantes de una universidad.
Muestra = 200 estudiantes seleccionados aleatoriamente.

Hipótesis

Hipótesis es una afirmación o suposición que se quiere comprobar mediante un estudio estadístico.

- ▶ **Hipótesis nula** (H_0): Afirmación de "no efecto." "no diferencia".
- ▶ **Hipótesis alternativa** (H_a): Lo contrario de H_0 , indica efecto o diferencia.

Ejemplo: H_0 : La media de notas es igual a 70. H_a : La media de notas no es igual a 70.

Tipos de Variables

- ▶ **Cualitativas (Categoricas):** describen cualidades o categorías.
 - ▶ Nominales: sin orden (ej. género, color).
 - ▶ Ordinales: con orden (ej. nivel educativo, satisfacción).
- ▶ **Cuantitativas (Numéricas):** expresan cantidades.
 - ▶ Discretas: valores enteros (ej. número de hijos).
 - ▶ Continuas: cualquier valor en un rango (ej. altura, peso).

Escalas de Medición

- ▶ **Nominal:** categorías sin orden ni jerarquía. Ejemplo: colores, género.
- ▶ **Ordinal:** categorías con orden pero sin distancia fija. Ejemplo: nivel socioeconómico, clasificación de satisfacción.
- ▶ **De intervalo:** valores numéricos con distancias iguales, sin cero absoluto. Ejemplo: temperatura en °C.
- ▶ **De razón:** como intervalo, pero con cero absoluto y proporciones significativas. Ejemplo: peso, altura, ingresos.

Medidas de Tendencia Central y Dispersión

Estas medidas resumen y describen un conjunto de datos numéricos:

- ▶ **Tendencia Central:** valor que representa el centro.^o punto típico de los datos.
- ▶ **Dispersión:** indica qué tanto varían o se alejan los datos respecto al centro.

Medidas de Tendencia Central

- ▶ **Media** (\bar{x}): promedio de todos los valores.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Intuitivamente, suma todos los datos y los divide por la cantidad total.

- ▶ **Mediana**: valor que está en el centro cuando ordenamos los datos.

Intuitivamente, divide los datos en dos partes iguales.

- ▶ **Moda**: valor que aparece con mayor frecuencia.
Es el dato "más común." repetido en la muestra.

Media en R

Usamos la función `mean()` para calcular la media.

```
1 # Media del consumo de combustible (mpg) en mtcars
2 mean(mtcars$mpg)
```

Mediana en R

La función `median()` calcula la mediana.

```
1 # Mediana del consumo de combustible
2 median(mtcars$mpg)
```

Moda en R

R no tiene función directa para moda, pero podemos definirla:

```
1 get_mode <- function(v) {  
2   uniqv <- unique(v)  
3   uniqv[which.max(tabulate(match(v, uniqv)))]  
4 }  
5  
6 # Moda de n mero de cilindros  
7 get_mode(mtcars$cyl)
```

Medidas de Dispersión

- **Rango:** diferencia entre el valor máximo y mínimo.

$$R = x_{\text{máx}} - x_{\text{mín}}$$

Mide la extensión total de los datos.

- **Varianza** (s^2): promedio de los cuadrados de las diferencias entre cada dato y la media.

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Mide cuánto se alejan los datos de la media, penalizando valores lejanos.

- **Desviación estándar** (s): raíz cuadrada de la varianza.

$$s = \sqrt{s^2}$$

Expresa la dispersión en las mismas unidades que los datos originales.

- **Coeficiente de variación** (CV):

Rango en R

Calculamos rango con `max()` y `min()`.

```
1 rango <- max(mtcars$mpg) - min(mtcars$mpg)
2 rango
```

Varianza en R

Función `var()` calcula la varianza muestral.

```
1 var(mtcars$mpg)
```

Desviación Estándar en R

Función `sd()` calcula la desviación estándar.

```
1 sd(mtcars$mpg)
```


Coefficiente de Variación

Cálculo del coeficiente de variación:

$$CV = \frac{\text{desviación estándar}}{\text{media}} \times 100\%$$

```
1 cv <- sd(mtcars$mpg) / mean(mtcars$mpg) * 100
2 cv
```

Interpretación de Medidas

- ▶ **Media y mediana** representan el centro, pero la mediana es más resistente a valores extremos o atípicos.
- ▶ **Varianza y desviación estándar** nos dicen qué tan dispersos están los datos: valores altos indican gran variabilidad.
- ▶ **Coeficiente de variación** permite comparar la dispersión de variables con diferentes unidades o escalas.

Resumen rápido con `summary()`

La función `summary()` da un resumen básico para variables numéricas:

```
1 summary(mtcars$mpg)
```

Incluye mínimo, 1er cuartil, mediana, media, 3er cuartil y máximo.

Medidas para todas las variables

Calcular medias y desviaciones estándar para todas las variables numéricas:

```
1 apply(mtcars, 2, mean)
2 apply(mtcars, 2, sd)
```