Tutoría 1

Sesión de reforzamiento y Estructuras de Datos en R

Sofía Madariaga

Pontificia Universidad Católica de Chile

Aproximación a las Políticas Públicas desde los datos

Taller de Análisis de Datos I

10 de julio de 2023

Materiales

En el portal del diplomado Materiales del Curso > Tutorías.

- `datos.xlsx`
- `tutoria-1.r` ← aquí estaremos trabajando
- `tutoria-respuestas.r`

Objetivos de la Tutoría

1. Repasar contenidos básicos sobre el uso y sintaxis de R.
2. Repasar las principales estructuras de datos en R.
3. Presentar ejemplos prácticos que pueden ser útiles para su tarea.

Outline

Son elementos de la **sintaxis básica** en R:

```
# Comentarios

getwd() # Comprobar directorio
setwd("/home/sofia/Escritorio") # Fijar directorio

# Guardar objeto
objeto <- 1

# Operaciones b sicas
1+1
1-1
1/2
```

- R es un lenguaje de programación orientado a objetos.
- En palabras simples, significa que podemos guardar nuestros resultados, estructuras de datos y valores en un espacio de memoria y este espacio de memoria debe tener un nombre.

```
13 radio <- 2
14 area_circulo <- radio*pi**2
```

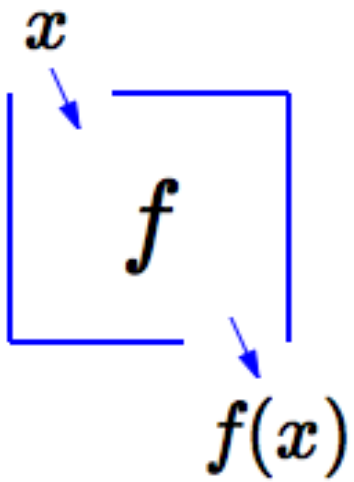
Nombre de los objetos

- No puede empezar con un número.
- No puede tener espacios entre medio.
- Evitar los carácteres extraños como tildes y ñ.

Estilos

- `sanke_case`
- `camelCase`
- `PascalCase`

Funciones



En R, podemos aplicar **funciones** a los objetos:

```
length(objeto)
mean(objeto)
sd(objeto)
table(c(c(0, 0, 0), c(1,
    1)))
```

Documentación. Para obtener más detalles, se recomienda revisar la documentación.

```
?mean
```

Tabmién, es posible consultar **arguentos** y **ejemplos** de la función.

```
args(mean)
example(mean)
```

Algunas funciones pertenecen a **librerías** de R, que requieren ser **instaladas** y **cargadas**.

- **Instalar el paquete.** Se hace sólo una vez. Aconsejo no dejarlo en el código cuando compartan sus proyectos.

```
1 install.packages("dplyr")
```

- **Cargar el paquete.** Se realiza cada vez qeu inicio o reinicio sesión en R, y voy a usar las funciones de ese paquete.

```
1 library(dplyr)
```

Tipos de Datos

Principalmente, existen seis tipos de datos en R:

- **Character:** dato de texto, y se introduce e imprime en la consola acompañado de comillas .
- **Entero:** dato entero (discreto).
- **Numéricos:** Incluye enteros y decimales (también denomniado *double*).
- **Booleano:** clase particular para los valores `TRUE` y `FALSE`.
- **Perdidos:** un caso perdido es lo que se denomina **NA** en R, es cuando falta el dato de una observación en particular de una variable (está en blanco).
- **No existe:** denominado `NULL`, es cuando estás buscando un elemento que no existe en ese objeto.

Para verificar la* **clase *usar la función* `class` *o familia de funciones con salida lógica* `is` *(e.g.* `is.character`*).*

Extructuras de Datos

En esta oportunidad, nos concentraremos en las principales estructuras de datos en R:

- **Vectores.** Arreglo unidimensional con un solo tipo de dato.

```r
vector <- c(1, 2, 3, 4, 5, 6)
```

- **Matrices.** Conjunto de datos ordenados en dos dimensiones. Solo un tipo de dato.

```r
matrix <- cbind(vector_1, vector_2)
```

- **Data Frames.** Base de Datos importada como objeto a R.

```r
data <- data.table::fread("data.csv")
```

Vectores

$$\vec{v_i} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Definición y propiedades

- Arreglo unidimensional de valores.
- Solo puede contener un tipo de dato.
- En R, principalmente se generan con las siguientes funciones: `c()`, `seq()`, `rep()`.

```
vector <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
secuencia <- seq(from = 1, to = 5)
repeticion <- rep(5, times = 10)
```

- $X \sim Normal(\mu_X = 4, \sigma_X = 1{,}2)$:

```
x <- rnorm(100, mean = 4, sd = 1.2)
```

Estructuras de Datos

Coerción Implícita

Existen dos tipos de coherción:

1. Explícita: es cuando se aplica una función (generalmente, del tipo: `as.numeric()`) para cambiar la naturaleza de la variable.

```
vector_bool <- c(TRUE, FALSE)
as.numeric(vector_bool)
```

2. Implícita: cuando R modifica de maenra automática y forzosa la naturaleza de los datos.
   - Ejemplo: unaoperación o estructura que exige dimensionalidad múltiplo, pero se involucran unidades . R recicla uno de lso términos para çalzarlos".

Veamos algunos ejemplos en R →

Matrices

$$A_{ij} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} \\ a_{21} & a_{22} & \dots & a_{2j} \\ \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} \end{bmatrix}$$

Definición y propiedades

- Arreglo bidimensional de valores.
- Solo puede contener un tipo de dato *(si no, se genera coerción)*.
- En R, principalmente se generan con las siguientes funciones: `matrix()`, `cbind()`, `rbind()`.

```
matrix <- matrix(c(1, 2, 3, 4, 5, 6))
vec_columna <- cbind(vector_1, vector_2, vector_3)
vec_fila <- rbind(vector_1, vector_2, vector_3)
```

Estructuras de Datos

Coerción

- Explícita: coerción **deliberada**.

```
1 matrix(c(1, 2, 3, TRUE), ncol = 2)
```

```
     [,1] [,2]
[1,]    1    3
[2,]    2    1
```

- Implícita: coerción **no deliberada**.

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

§ Operaciones

- **Diagonal:** `diag()`: calcular diagonal de la matriz.
- **Determinante:** `det()`: cacular el determinante de una matriz.

§ **Transpuesta** (`t()`)

Sea A una matriz con dimensiones 2×2: $A = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$

$$A^T = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

```
1 t(A)
```

§ Suma de Matrices (+)

Sea A y B dos matricescon dimensiones 2×2, tal que:

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \qquad B = \begin{bmatrix} 3 & 4 \\ 7 & 10 \end{bmatrix}$$

$$A + B = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} + \begin{bmatrix} 3 & 4 \\ 7 & 10 \end{bmatrix} = \begin{bmatrix} 7 & 4 \\ 9 & 14 \end{bmatrix}$$

```
1 A + B
```

§ Multiplicación de Matrices (`solve()`)

Sea A y B una matriz con dimensiones 2×2 y x una matriz de un vector 2×1, tal que:

$$A = \begin{bmatrix} 1 \text{ y } 3 \\ 2 \quad 4 \end{bmatrix} \qquad x = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$Ax = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 + 3 \cdot 2 \\ 2 + 4 \cdot 2 \end{bmatrix} = \begin{bmatrix} 7 \\ 10 \end{bmatrix}$$

```
A %*% x
```

§ **Inversa**

Sea A una matriz con dimensiones 2×2, la inversa A^{-1} es una matriz que cumple simultáneamente:

$$AA^{-1} = I \ \& \ A^{-1}A = I$$

```
1 solve(A)
```

Ejercicios

1. Muestre que $AA^{-1} = I$
2. Calcule Ax
3. Pruebe que: $X^T tX = \sum x_i^2$
4. Calcule correlación. Para ello, considere: $S = Z^T ZXn^{-1}$

Comentario sobre el Ejercicio 1.c.

La solución matricial **estimadordes mínimos cuadrados** se obtiene de esta manera:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

Para ello, deben considerar la **notación matricial** de un modelo de regresión lineal:

$$y_i = \beta_0 + \beta_1 x_i + e_i$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & x1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_i \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_i \end{bmatrix}$$

Comentario sobre el Ejercicio 1.c.

Importante:

- **Compruebe** sus resultados, calculando la regresión para esas variables con la función `lm()`:

`lm(Y` $\sim X, data = sus\_datos$)

Comentario sobre el Ejercicio 1.c.

Ejemplo:

```
> (operación censurda)
             [,1]
[1,] -17.579095
[2,]   3.932409

> lm(dist ~ speed, data = cars)
Coefficients:
(Intercept)        speed
    -17.579        3.932
```

Data Frames

	edad	**sexo**	**nivel educacional**
1	24	1	Básica
2	60	0	Universitaria
3	15	0	Media
4	31	1	Posgrado
5	56	0	Posgrado
6	70	0	Básica

Cuadro: Ejemplo de Data Frame (elaboración propia)

Estructuras de Datos

Propiedades y características

- Es lo que entiende R por Base de Datos.
- Es un tipo de estructura de datos **heterogéneo**, ya que contiene variables con diferentes tipos de valor: texto, continuas, categórico-discretas, booleanos, entre otros.

Vamos a R $\rightarrow$

Extracción de datos, vía APIs (Ejercicio 2)

- En el **Ejercicio 2**, deben usar un paquete `WDI` que facilita la consulta a la **API del Banco Mundial**.
- Primero, deben identificar el ID del indicador de interés.
- Luego, extraen los datos con la función `WDI`.

Extracción de datos, vía APIs (Ejercicio 2)

§ **Paso 1:** Buscar el identificador de mi indicador de interés.

- Usar la función `WDIsearch()`.
- Portal de indicadores del Banco Mundial: https://data.worldbank.org/indicator.

Extracción de datos, vía APIs (Ejercicio 2)

§ **Paso 2:** Extracción de los datos.

```
data_life_expectancy <- WDI(indicator = 'SP.DYN.LE00.
    IN',
                            country = "all",
                            start = 2015,
                            end = 2020,
                            extra = TRUE)
```