

Datos Solidarios con R: Primeros pasos

Análisis y visualización de datos para el sector solidario en Colombia

En este primer tutorial enseñaré a l@s participantes del curso a usar la interfase de R Studio para realizar tareas básicas.

Usando R como calculadora

En su funcionalidad más básica R puede ser utilizada como una calculadora tradicional.

```
520*40
```

```
## [1] 20800
```

```
(120-75)/2
```

```
## [1] 22.5
```

Objetos, Funciones y Paquetes

Una parte importante de programar usando lenguaje R son: *los objetos*, *las funciones* y *los paquetes*. A continuación, se explica que son cada uno de estos elementos.

Objetos

Un objeto es cualquier cosa, que queremos que R almacene para poder acceder a él. Puede ser desde un único valor hasta una colección de valores, un conjunto de datos, una función o incluso una lista de otros objetos. *La única limitación para los objetos* es que **cada uno debe tener un nombre único**, que se utiliza para llamarlo.

Para crear un objeto, utilizamos el operador de asignación (= o <-), donde colocamos el nombre del objeto que queremos crear a la izquierda de la flecha y el objeto en sí a la derecha.

```
valor = 10
value <- 50
suma = valor - value
suma
```

```
## [1] -40
```

```
valor = 80
suma = valor - value
suma
```

```
## [1] 30
```

Dentro de la categoría de objetos, uno de ellos tiene un destaque especial: ***data frames***. En R, un *data frame* es una estructura de datos fundamental y ampliamente utilizada para almacenar datos

tabulares. Puede conceptualizarse como una hoja de cálculo o una tabla en una base de datos, con filas que representan observaciones y columnas que representan variables.

```
base <- data.frame( nombre = c("Cooperativa A", "Cooperativa B",  
                              "Cooperativa C", "Cooperativa D",  
                              "Cooperativa E", "Cooperativa F"),  
                  ingresos = c(120000, 450000, 180000, 250000,  
                              400000, 150000))
```

base

```
##           nombre ingresos  
## 1 Cooperativa A   120000  
## 2 Cooperativa B   450000  
## 3 Cooperativa C   180000  
## 4 Cooperativa D   250000  
## 5 Cooperativa E   400000  
## 6 Cooperativa F   150000
```

Funciones

Mientras que los objetos son todo lo que almacenamos en R, las **funciones** son todo lo que hacemos en R. Las funciones son bloques de código autónomos diseñados para tomar uno o varios objetos y realizar tareas específicas; ellas reciben entradas (argumentos), las procesan y devuelven una salida.

Como ejemplo de funciones que vienen incorporadas en *lenguaje de R Base*, podemos mencionar las siguientes:

- Suma

```
sum(base$ingresos)
```

```
## [1] 1550000
```

- Média

```
mean(base$ingresos)
```

```
## [1] 258333.3
```

Si no estamos familiarizados con una función siempre podemos pedir ayuda y consultar la documentación del paquete, usando el siguiente comando:

```
#help("mean")
```

Paquetes

Los **paquetes en R** son colecciones de funciones, datos y documentación agrupadas en un formato estandarizado. Amplían las capacidades de R al proporcionar funcionalidades especializadas más allá de la instalación base. Los paquetes suelen almacenarse en un directorio llamado “biblioteca” dentro del entorno de R.

Tidyverse Tidyverse es una colección de paquetes de R diseñados para que los flujos de trabajo de ciencia de datos sean más consistentes, eficientes y amenos. Abarca un conjunto de paquetes que comparten una filosofía de diseño, gramática y estructuras de datos comunes, centrados principalmente en los principios de “datos ordenados”.



Figure 1: Fuente: Rpubs

Paquetes principales:

Incluye paquetes esenciales para diversas tareas de ciencia de datos:

dplyr: Para la manipulación de datos (p. ej., `filter()`, `mutate()`, `group_by()`, `summary()`).

ggplot2: Para la visualización de datos, creando gráficos estadísticos de alta calidad.

tidyr: Para ordenar datos, transformándolos entre formatos anchos y largos (p. ej., `pivot_wider()`, `pivot_longer()`).

readr: Para importar datos desde varios formatos de archivo (p. ej., `read_csv()`, `read_tsv()`).

purrr: Para la programación funcional, facilitando el trabajo con listas y la aplicación iterativa de funciones.

stringr: Para la manipulación de secuencias de caracteres.

forcats: Para trabajar con factores (variables categóricas).

Operador de canalización (%>%): Tidyverse utiliza ampliamente el operador de canalización del paquete *magrittr*, que permite encadenar múltiples operaciones de forma legible e intuitiva.

A continuación, realizaremos la instalación del paquete **tidyverse** y su activación para comenzar a usarlo en el próximo tutorial. Para lo cual, debemos usar el siguiente comando:

```

#install.packages("tidyverse")
library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v ggplot2    3.5.2      v tibble    3.3.0
## v lubridate  1.9.4      v tidyr     1.3.1
## v purrr      1.1.0
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco

```

Próximos pasos

En el próximo tutorial aprenderemos:

- Cómo limpiar y procesar los datos de una base
- Calcular estadísticas básicas
- Crear gráficos