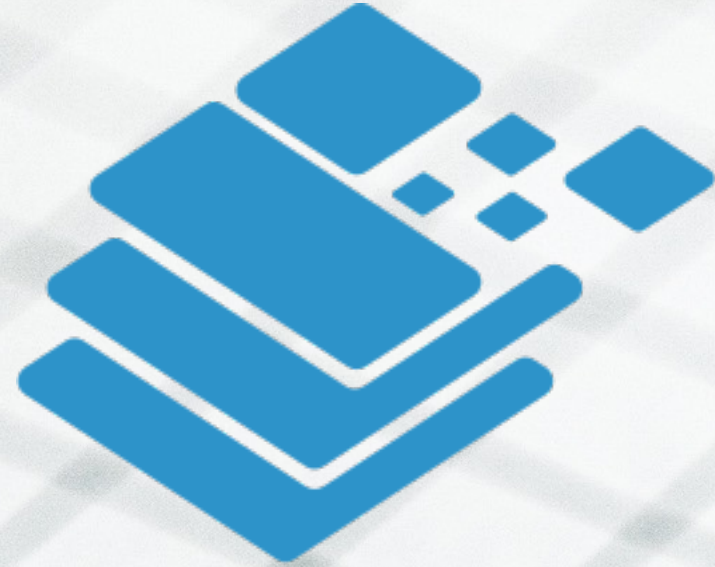




**Spatial
Lab
Analytics**

partimos pronto



**Spatial
Lab
Analytics**

ANÁLISIS DE DATOS CON R

Introducción al lenguaje R

Bastían Olea Herrera - baolea@uc.cl

Primera aproximación al lenguaje de programación R orientado al análisis de datos. Conceptos principales, elementos fundamentales del lenguaje, flujo de trabajo, y buenas prácticas.

01 **Funciones**

Uso y creación de herramientas

02 **Condicionales**

Creación de nuevas variables en base a comparaciones

03 **Iteraciones**

Operaciones consecutivas

04 **Data frames**

Datos ordenados de forma tabular

funciones() { ... }

01

Uso y creación de herramientas
programáticas para trabajar con
datos y acelerar tu flujo de
trabajo

Funciones

- Las funciones son pequeños programas que nos permiten realizar distintas operaciones.
- Las funciones tienen **argumentos**, que son los inputs que recibe la función
- Si no sabemos cómo usar una función, podemos buscar su nombre en el panel de Ayuda de RStudio, o escribir el nombre de la función con un signo de interrogación antes: `?funcion`

`mean()` # calcular promedio

`median()` # calcular mediana

`sum()` # sumar elementos

`min()` # valor mínimo

`max()` # valor máximo

`typeof()` # tipo del objeto

`is.na()` # evaluar si es missing

`as.numeric()` # convertir a numérico

Funciones

paste() # pegar textos

paste0() # pegar sin espacios

typeof() # obtener tipo de un objeto

class() # obtener clase de un objeto

toupper() # texto a mayúsculas

tolower() # texto a minúsculas

as.numeric()

as.integer()

as.character()

sample() # muestra aleatoria

seq() # secuencia regular

rep() # repetir datos

round() # redondear decimales

signif() # redondear cifras

Funciones

- Para crear una función, usamos la función `function()`
- Dentro de esta función se especifican los inputs, y luego en el cuerpo se define lo que hará la función.
- Dentro del **cuerpo** de la función podremos usar los inputs para realizar lo que necesitemos.
- Lo último que hagamos en la función será lo que la función **retorna**; es decir, su output

```
calculo1 <- function(input1) {  
  # cuerpo  
  input1 * 1000  
}
```

`ifelse()` # condicionales

02

Creación de nuevas variables en
base a comparaciones

ifelse()

- Una función condicional permite hacer algo si se cumple o no una condición.
- “if else” significa en castellano “si pasa esto, entonces haz esto”.
- El primer argumento es una **comparación**, y luego hay que especificar dos argumentos más: el primero es lo que queremos si la comparación es **verdadera**, y lo segundo es lo que queremos si la comparación es **falsa**.

if () { } else { }

- También podemos definir condicionalidad sin usar una función, sino, en su lugar, controlando el flujo de la ejecución del código.

```
explicación <- TRUE
```

```
if (explicación == TRUE) {  
  "el código dentro de los  
  paréntesis de llave sólo se  
  ejecutará si la condición  
  establecida arriba retorna TRUE"  
} else {  
  "si la condición es FALSE,  
  entonces podemos definir que se  
  ejecute un código distinto, como  
  éste, o bien, no hacer nada"  
}
```



```
for (i in iteraciones)
```

03

Operaciones consecutivas

Iteraciones

- Una iteración permite que, en base a una secuencia de elementos o un vector, R repita la operación que definamos para cada uno de los elementos.
- Al crear la iteración, lo primero que se define es cómo se llamará el objeto que contiene cada paso, y luego el conjunto de elementos sobre los que se iterará

```
pasos <- 1:10
```

nombre del
objeto que
representa a
los pasos

objeto o vector
sobre el cual se
repetirá la
operación



```
for (num in pasos) {  
  paste("paso:", num)  
}
```



```
a = c(0.796671831281856, 0.749656155938283, 0.656508937245235,  
0.375385080929846, 0.480514499824494, 0.266876166220754, 0.70684823  
0.099679464707151, 0.982302132295445, 0.0498262464534491), b = c(0  
0.77340058144182, 0.816512330435216, 0.851083924024218, 0.948183559  
0.61821505217813, 0.775022355739805, 0.7942303505726159, 0.840941876  
0.204416815424338), c = c(0.0505761282984167, 0.644072845578194,  
0.846621100557968, 0.891164530534297, 0.977461001370102, 0.91304313  
0.852099474752322, 0.503274511080235, 0.525078371865675, 0.09693358
```

data.frame()

04

Datos ordenados de forma
tabular

Tablas o dataframes

- Los data frames son lo que comúnmente conocemos como tablas o planillas de datos
- Son una forma de estructurar datos en filas y columnas
- Comúnmente, las filas son *observaciones*, y las columnas son *variables*
- Se caracterizan por ser **rectangulares**; es decir, todas las columnas tienen la misma cantidad de filas
- En R, los dataframes están constituidos por vectores, dado que cada **columna** de un dataframe es realmente un **vector**.
- En otras palabras, los dataframes están hechos de vectores
- Lo que hemos aprendido a hacer sobre los vectores, podremos hacerlo sobre las columnas de nuestras tablas de datos



**Spatial
Lab
Analytics**

Soluciones en análisis de datos

www.spatiallab.cl