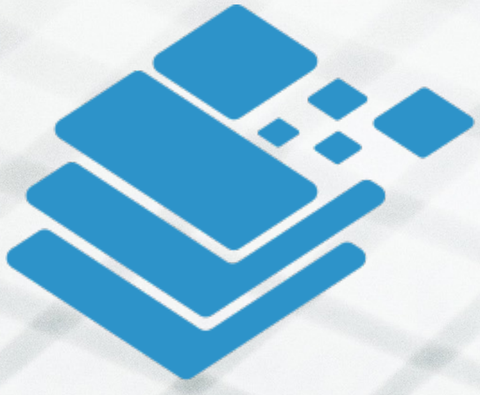
Spatial
Lab
Analytics

partimos pronto



Spatial
Lab
Analytics

ANÁLISIS DE DATOS CON R
Introducción al lenguaje R

Bastián 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)
}
```

`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

data.frame()

- Una forma de crear una tabla de datos es con `data.frame()`
- Dentro van los vectores que constituirán las columnas (variables) de la tabla

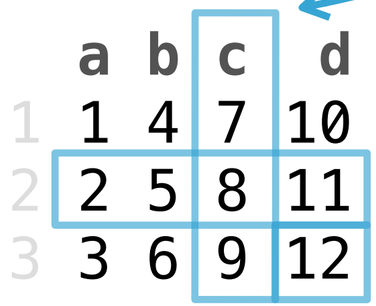
```
data.frame("a" = 1:3,
           "b" = 4:6,
           "c" = 7:9,
           "d" = 10:12)
```

	a	b	c	d
1	1	4	7	10
2	2	5	8	11
3	3	6	9	12

data.frame()

- Ahora que tenemos una tabla, podemos volver a descomponerla en sus partes iniciales

  `x[filas, columnas]`



	a	b	c	d
1	1	4	7	10
2	2	5	8	11
3	3	6	9	12

```
tabla$c
tabla[["c"]]
tabla[, 3]
# tercera columna
```

```
tabla[2, ]
# segunda fila
```

```
tabla[3, 4]
tabla$d[3]
# tercera fila,
  cuarta columna
```

Spatial Lab Analytics

Soluciones en análisis de datos

www.spatiallab.cl