

Programación en R para el análisis de datos

Introducción y precalentamiento

Nicolás Schmidt

`mail::nschmidt@cienciassociales.edu.uy`

`GitHub::@Nicolas-Schmidt`

Departamento de Ciencia Política

Facultad de Ciencias Sociales

1. Punto de partida
2. Objetivos del curso
3. Contenido
4. Dinámica
5. Precalentamiento
6. Muestra

Punto de partida

IMPORTANTE:

Este curso es la continuación del curso introductorio '**IntRo**'.
Esto supone que el estudiante usa R con cierta autonomía relativa.

El curso '**IntRo**' está disponible y actualizado en la [web](#). Pueden recurrir y consultar los contenidos del mismo siempre que sea necesario!

Objetivos del curso

- Dotar a los estudiantes con los conocimientos fundamentales de la creación de funciones y logren utilizar la programación funcional para sus análisis de datos.
- Lograr que los estudiantes se enfrenten a sus problemas relativos al análisis de datos de manera general de tal manera que puedan generar rutinas automatizadas para sus tareas y que esto impacte positivamente en el tiempo invertido y en la replicabilidad de las tareas.

Contenido

- Programación orientada a objetos
- Programación funcional
- Entornos
- Estructura de los Paquetes
- Estructuras de control e iteraciones
- Funciones
- Programación eficiente

Dinámica

Todas las clases van a estar estructuradas y van a tener una presentación específica que van a poder consultar.

Pero la dinámica de clase va a ser un uso intensivo de R aplicando los distintos tópicos del curso a distintos conjuntos de datos.

Precalentamiento

Hagamos un test para verificar el nivel de memoria...

`!sum(!1 + !0) + 0`

`mean(1,2,3,4,5)`

```
x <- c(NA, 5, NA, 10)
```

```
x == NA
```

```
z <- matrix(, 10, 10)
```

```
z[] <- 0
```

```
z <- matrix(, 10, 10)
```

```
z <- 0
```



```
data.frame(  
  x = 1:10,  
  y = letters[1:10]  
)
```

```
x <- 10
```

```
y <- function() print(x)
```

```
y()
```

```
x <- 10
```

```
y <-function(x) print(x)
```

```
y(10 + x)
```

```
c <- function(...){  
  base::c("holaaaa", ...)  
}
```

```
c <- 10
```

```
c(c = c)
```

```
matrix(1:5, byrow = TRUE)
```

```
dim(c(3, 1, 3, 6)) <- c(2, 2)
```

```
foo <- function(f){  
  f <- f  
  foo2 <- function(){  
    f2 <- f + 1  
    print(ls())  
    f2  
  }  
  foo2()  
}
```


Muestra

Limpiar datos con expresiones regulares para luego buscar patrones con expresiones regulares:

```
texto <- "SEÑOR presidente, pido la palabra. Diputado SEÑOR Gonzalez..."  
  
separate_sir(texto)  
  
## [1] "SEÑOR presidente, pido la palabra. Diputado SEÑOR Gonzalez..."
```

Trabajando con texto

```
archivo = "ascasdaaaaasdadsdadadshgywyefgqoyefqeuyr2t3642876294872"

names(which.max(table(unlist(strsplit(
  stringr::str_remove_all(archivo, pattern = "[^A-z]]"), "")))))

## [1] "a"

library(magrittr)
archivo %>%
  stringr::str_remove_all(pattern = "[^A-z]]") %>%
  strsplit(., "") %>%
  unlist() %>%
  table() %>%
  which.max() %>%
  names()

## [1] "a"
```