



# Taller R – Rifa Valentina

## Funciones y procesos iterativos

Nicolás Ratto

2022/02/02

# Contenidos curso

- Manipulación avanzada de data frames (funciones `pivot` y combinación de data frames)
- Manipulación de fechas: paquete `lubridate`
- Visualización de datos con `ggplot2`
- Estimar desde diseños muestrales complejos (paquete `survey` y `srvyr`)
- **Procesos iterativos y funciones**

# Interacción con pc y environment

```
ls(),list.files(),getwd()
```

# Introducción

**Siempre** R está trabajando desde una ruta del computador

Desde esta ruta comienza la lectura y carga de archivos

Para conocer esta ruta:

```
getwd()
```

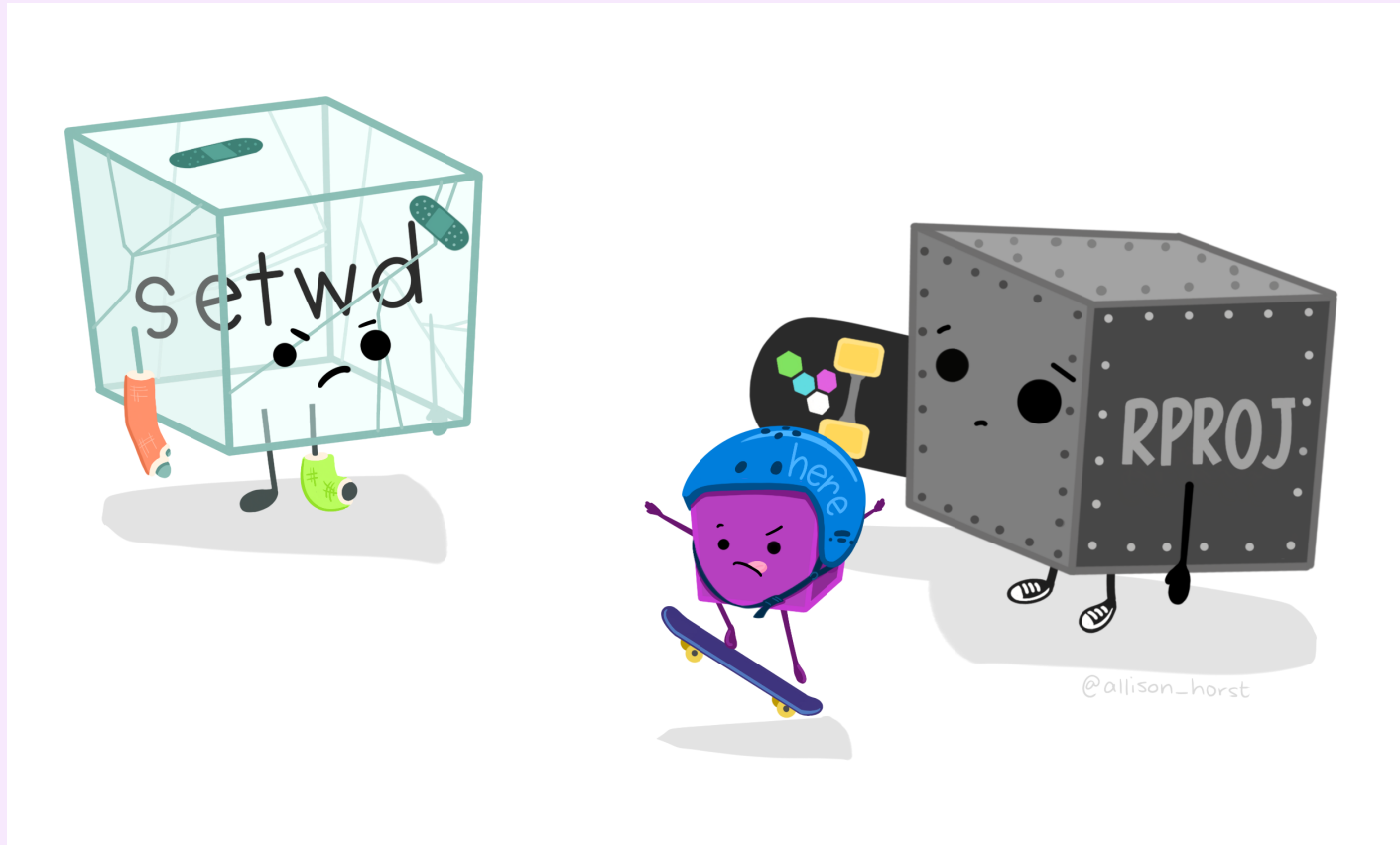
```
## [1] "C:/Users/Nratto/Documents/Github/tallerR_rifa"
```

Para modificarla solo dos formas recomendadas.

- Crear un nuevo RProject
- Abrir un nuevo documento RMarkdown

Es decir, **nunca utilizar setwd()**

# ¡Muerte a direct absolutos!



# Introducción

Ya sabemos cargar desde el pc archivos en diferentes formatos.

Pero, ¿como saber todo lo que existe en una ruta? (una vez que definimos la que nos interesa)

```
list.files()
```

```
## [1] "1.-trasponer.html"      "1.-trasponer_files"    "1. trasponer.Rmd"
## [4] "2.-tiempo.html"         "2.-tiempo_files"       "2. tiempo.Rmd"
## [7] "3.-ggplot2.html"        "3.-ggplot2_files"      "3. ggplot2.Rmd"
## [10] "4.-survey.html"         "4. survey.Rmd"         "5.-iteraciones.html"
## [13] "5.-iteraciones.Rmd"     "5. iteraciones.Rmd"    "6.-openxlsx.html"
## [16] "6. openxlsx.Rmd"        "data"                   "datos-encuesta"
## [19] "datos_ene"              "datos_pt"               "imagenes"
## [22] "libs"                   "output"                  "pdf"
## [25] "README.md"              "tallerR_rifa.Rproj"     "xaringan-themer.css"
```

Para retroceder e ir a otra carpeta

```
list.files(path = "../vignettes")
```

# Introducción

En nuestra ruta de interés, podemos preguntarnos si existe una carpeta o archivo

```
file.exists("1.-trasponer.html")
```

```
## [1] TRUE
```

```
file.exists("6.-hacer-paquetes.html")
```

```
## [1] FALSE
```

```
file.exists("datos_ene")
```

```
## [1] TRUE
```

# Introducción

Si la carpeta no existe, podemos crearla

```
ifelse(  
  dir.exists("datos_ene"),  
  "Directorio existe",  
  dir.create("datos_ene")  
)
```

```
## [1] TRUE
```

¿Funcionó?

```
ifelse(  
  dir.exists("datos_ene"),  
  "Directorio existe",  
  dir.create("datos_ene")  
)
```

```
## [1] "Directorio existe"
```



# Introducción

Si es un archivo también se puede

Crear dato basura

```
basura <- data.frame(x=c(1:10),y=c(100:109))
```

Exportar dato si no existe

```
ifelse(  
  file.exists("datos_ene/basura.xlsx"),  
  "dato existe",  
  writexl::write_xlsx(basura,"datos_ene/basura.xlsx")  
)
```

```
## [1] "C:\\Users\\Nratto\\Documents\\Github\\tallerR_rifa\\datos_ene\\basura."
```

# Introducción

Creemos más basura

```
basura2 <- data.frame(x=c(1:10),y=c(100:109))  
data3 <- data.frame(x=c(1:10),y=c(100:109))  
data4 <- data.frame(x=c(1:10),y=c(100:109))
```

¿Para saber que hay en nuestro ambiente?

```
ls()
```

```
## [1] "basura"      "basura2"     "blue"        "dark_yellow" "data3"  
## [6] "data4"      "gray"        "light_yellow"
```

Según patrón

```
ls(pattern = "basura")
```

```
## [1] "basura" "basura2"
```

```
rm(list=ls(pattern = "basura")) # borrar
```

# Descargar datos desde R

```
download.file()
```

# Descargar datos desde R

Para automatizar todo o hacerlo lo más reproducible posible, nos evitaremos incluso descargar los datos mediante *clicks*.

Salvo que queramos descargar .rar, `download.file` sirve:

```
download.file(url="link a pagina",  
              destfile = "ruta del pc en donde guardar archivo",  
              method = "curl",  
              mode="wb")
```

```
url <- "https://www.ine.cl/docs/default-source/ocupacion-y-desocupacion/  
destfile <- "datos_ene/ene_2021_10.csv"  
download.file(url, destfile, mode = "wb")
```

```
if(!file.exists(destfile)){  
  download.file(url, destfile, mode = "wb")  
}
```

# Proceso iterativo y condiciones

`for()` e `if()`

# if()

Ya lo vimos un poco, la lógica es:

```
if(condicion){  
    acción a implementar si la condicion es verdadera  
}
```

```
if((1+1)==2){  
    "suma correcta"  
}
```

```
## [1] "suma correcta"
```

```
if((1+2)==2){  
    "suma correcta"  
}
```

Condición false, no pasa nada...

# if()

Para tener respuesta con condición incorrecta, agregar `else`

```
n <- 1
if((n+2)==2){
  "suma correcta"
} else{
  "suma incorrecta"
}
```

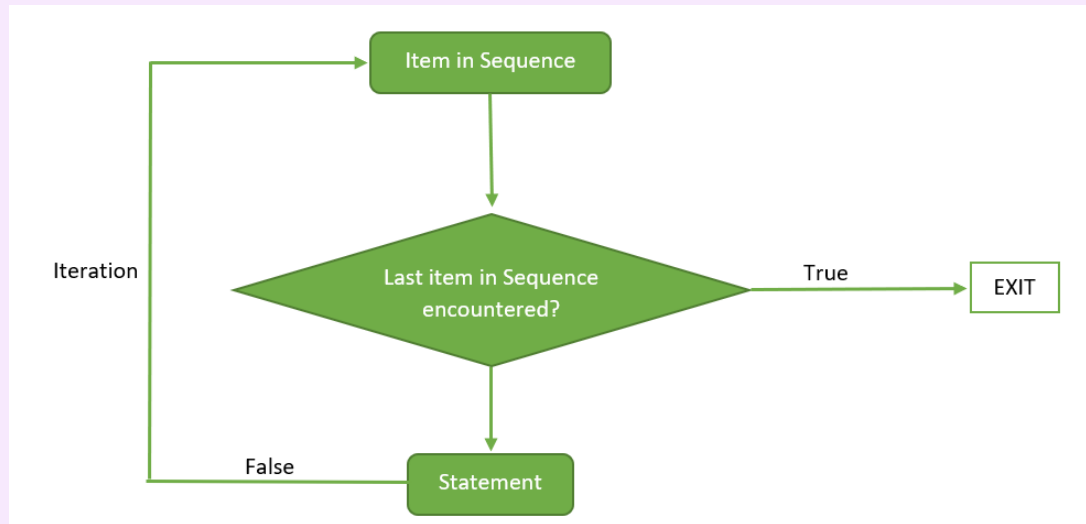
```
## [1] "suma incorrecta"
```

```
n <- 0
if(n+n > 0){
  "n es positivo"
} else if(n+n < 0) {
  "n es negativo"
} else{
  "n es cero"
}
```

```
## [1] "n es cero"
```

# for()

Para cada valor X en una secuencia que va de Y a Z, ejecuta una acción.



`while()` y `repeat()` pueden ser útiles. En los personal no he tenido que usarlos.

Entendiendo la lógica de `for()` e `if()` estas funciones no tienen mayor complejidad.



# for()

Ejemplo simple:

```
for(i in 1:5){  
  print(i + 1)  
}
```

```
## [1] 2  
## [1] 3  
## [1] 4  
## [1] 5  
## [1] 6
```

Imaginen la potencia de esto para exportar o importar cientos de archivos...

Veamos un ejemplo con las bases de puestos de trabajo de la EMR

# for()

Crear secuencia de fechas

```
library(dplyr)
library(lubridate)

inicio <- "2016-01-01"
fin <- "2017-02-01"

meses <- ((ymd(inicio) %--% ymd(fin)) / months(1) )

ano_mes_dia <- ymd(inicio) %m+% months(0:meses) %>% as.character()
ano_mes_dia
```

```
## [1] "2016-01-01" "2016-02-01" "2016-03-01" "2016-04-01" "2016-05-01"
## [6] "2016-06-01" "2016-07-01" "2016-08-01" "2016-09-01" "2016-10-01"
## [11] "2016-11-01" "2016-12-01" "2017-01-01" "2017-02-01"
```

```
meses <- length(ano_mes_dia)
```

# for()

Crear secuencia de meses en español

```
meses_esp <- rep(c("enero", "febrero", "marzo", "abril", "mayo", "junio",  
                  "julio", "agosto", "septiembre", "octubre", "noviembre", "diciembre"),  
                trunc(meses/12))
```

```
meses_esp <- c(meses_esp, meses_esp[0:(meses%%12)])
```

```
meses_esp
```

```
## [1] "enero"      "febrero"    "marzo"      "abril"      "mayo"  
## [6] "junio"      "julio"      "agosto"    "septiembre" "octubre"  
## [11] "noviembre"  "diciembre"  "enero"      "febrero"
```

```
length(meses_esp)==length(ano_mes_dia)
```

```
## [1] TRUE
```

# for()

Descargar bases

```
for(i in 1:meses){  
  download.file(paste0("https://www.ine.cl/docs/default-source/sueldos-y-s  
    year(ano_mes_dia[i]),"/",  
    meses_esp[i],"-",  
    year(ano_mes_dia[i]),".csv?sfvrsn=54c360d8_6&download=true"),  
    destfile = paste0("datos_pt/",  
                      year(ano_mes_dia[i]),"_",  
                      sprintf("%02d", month(ano_mes_dia[i])), ".csv"),  
    method = "libcurl")  
}
```

```
list.files("datos_pt/")
```

```
## [1] "2016_01.csv" "2016_02.csv" "2016_03.csv" "2016_04.csv" "2016_05.csv"  
## [6] "2016_06.csv" "2016_07.csv" "2016_08.csv" "2016_09.csv" "2016_10.csv"  
## [11] "2016_11.csv" "2016_12.csv" "2017_01.csv" "2017_02.csv"
```

# for()

Ahora cargar bases en R

# Funciones

`function()`

# Funciones

# Recursos web utilizados

Xaringan: [Presentation Ninja](#), de [Yihui Xie](#). Para generar esta presentación.

Ilustraciones de [Allison Horst](#)

[Loops in R](#)