

OPSO79-1-UCSH2021

Diseños de investigación, Rproject y cargar data. Bloque práctico (4b)

03/09/2021

Revisión tarea 2

Módulo I

Cargar base guaguas desde paquete o creando objeto.

```
# instalar dplyr y guaguas
library(dplyr)
guaguas <- guaguas::guaguas
```

1. Explique que representa cada fila en la data frame guaguas

Cada fila de la data frame es un nombre inscrito en un año en el Registro Civil de Chile (por ejemplo, Andrea en 2014, Silvia en 1920, etc).

La data entrega información sobre el número de veces que fue inscrito el nombre en cada año, el sexo al que corresponde el nombre y la proporción de inscripciones totales que representa el nombre en el año.

Módulo I

2. ¿Que valor tiene la fila 3 y la columna 5?

```
guaguas[3,5]
```

```
## # A tibble: 1 x 1
##   proporcion
##       <dbl>
## 1     0.0312
```

3. ¿Cuáles son los valores de la observación 789.111?

```
guaguas[789111, ]
```

```
## # A tibble: 1 x 5
##   anio nombre sexo      n proporcion
##   <dbl> <chr>  <chr> <dbl>      <dbl>
## 1 2017 Nashly F        16  0.0000730
```

Módulo I

4. Indique medidas de distribución de la variable n e interprete brevemente

```
summary(guaguas$n)
```

```
##      Min.    1st Qu.     Median      Mean    3rd Qu.      Max.
##    1.00    1.00    1.00    25.96    3.00  21448.00
```

La idea era entender que no correspondía hacer un `table()`.

Módulo I

5. Indique la clase de las cinco columnas de la data frame y especifique su nivel de medida

```
#str(guaguas), o variable por variable: class(guagua$n)  
apply(guaguas, 2, class)
```

```
##           anio        nombre         sexo          n    proporcion  
## "character" "character" "character" "character" "character"
```

Respecto del nivel de medida hay que indicar que:

- anio es una variable cuantitativa discreta
- nombre es una variable categórica - nominal
- sexo es una variable categórica - nominal
- n es una variable cuantitativa discreta
- proporcion es una variable cuantitativa continua

Módulo II

1. ¿En que años se ha inscrito el nombre "khaleesi"?

```
filter(guaguas, nombre == "Khaleesi")  
  
## # A tibble: 8 x 5  
##   anio  nombre  sexo      n  proporcion  
##   <dbl> <chr>    <chr> <dbl>        <dbl>  
## 1 2012 Khaleesi F     1 0.00000410  
## 2 2013 Khaleesi F     1 0.00000412  
## 3 2015 Khaleesi F     4 0.0000163  
## 4 2016 Khaleesi F     4 0.0000172  
## 5 2017 Khaleesi F     4 0.0000182  
## 6 2018 Khaleesi F     8 0.0000359  
## 7 2019 Khaleesi F    10 0.0000473  
## 8 2020 Khaleesi F     3 0.0000154
```

Módulo II

2. ¿Cuántas guaguas se inscribieron con el nombre "Arya", "Daenerys" y "Khaleesi" en el año 2011?

- Arya

```
arya <- filter(guaguas, nombre == "Arya" & anio == 2011)
select(arya,n)
```

```
## # A tibble: 1 x 1
##       n
##   <dbl>
## 1     3
```

- Daenerys

```
daenerys <- filter(guaguas, nombre == "Daenerys" & anio == 2011)
select(daenerys,n)
```

```
## # A tibble: 1 x 1
##       n
```

Módulo II

- Khaleesi

```
khaleesi <- filter(guaguas, nombre == "Khaleesi" & anio == 2011)
select(khaleesi,n)
```

```
## # A tibble: 0 x 1
## # ... with 1 variable: n <dbl>
```

Otra posibilidad

```
nombres <- filter(guaguas, nombre %in% c("Arya", "Khaleesi", "Daenerys") &
                     anio == 2011)
select(nombres,nombre,n)

## # A tibble: 2 x 2
##   nombre      n
##   <chr>     <dbl>
## 1 Arya       3
## 2 Daenerys   1
```

Módulo II

3. ¿Cuántas guaguas se inscribieron con esos nombres en el año 2018?

```
select(filter(guaguas, nombre == "Arya" & anio == 2018),n)  
  
## # A tibble: 1 x 1  
##       n  
##   <dbl>  
## 1     10  
  
select(filter(guaguas, nombre == "Daenerys" & anio == 2018),n)  
  
## # A tibble: 1 x 1  
##       n  
##   <dbl>  
## 1     23
```

Módulo II

```
select(filter(guaguas, nombre == "Khaleesi" & anio == 2018),n)

## # A tibble: 1 x 1
##       n
##   <dbl>
## 1     8
```

4. ¿Se inscribió algún "Tyrion"?

```
filter(guaguas, nombre == "Tyrion")

## # A tibble: 1 x 5
##       anio nombre sexo      n proporcion
##   <dbl> <chr>  <chr> <dbl>      <dbl>
## 1  2018 Tyrion M         1  0.00000449
```

Módulo III

1. (Optativa) Llegue al resultado 1 desde la base guaguas

```
sum(select(filter(guaguas, anio==2010),proporcion))  
## [1] 1
```

2. Crea una nueva data frame desde guaguas que se llame "companers"

Esta data solo puede tener los nombres 3 compañeros o compañeras del curso. La data puede tener hasta 20 nombres.

```
companers <- filter(guaguas, nombre=="Nicolás" | nombre=="Kevin")
```

Módulo III

3. ¿Cuál es la distribución de la variable "nombre" de la data "companers"?

```
table(companers$nombre)
```

```
##  
##      Kevin Nicolás  
##          62      102
```

4. En no más de dos líneas, ¿qué puede concluir de la distribución de la variable "nombre" de la data "companers"?

Que el nombre Nicolás es la categoría modal. Este nombre ha sido inscrito más años (102 veces) entre 1920 y 2020 que el nombre Kevin (62 veces).

Reaso dplyr::mutate()

mutate()

"Muta" nuestra data, agregando una nueva columna.

```
mutate(data, nuevavariable = valorotorgado)
```

Multiplicar por 100:

```
mutate(guaguas, porcentaje=proporcion*100)
```

```
## # A tibble: 3 x 6
##   anio nombre sexo     n  proporcion  porcentaje
##   <dbl> <chr>  <chr> <dbl>       <dbl>        <dbl>
## 1 1920 María   F      2130      0.104       10.4
## 2 1920 José    M      984       0.0483      4.83
## 3 1920 Juan   M      636       0.0312      3.12
```

mutate()



mutate() para condiciones

La gracia de `mutate()` está en asignar valores en base a **condiciones**.

Vimos una forma simple para asignar dos valores, uno para la condición verdadera y otra para la condición falsa.

```
mutate(data, nuevavariable = if_else(condicion,  
                                      verdadero,  
                                      falso))
```

```
guaguas <- mutate(guaguas, populares = if_else(proporcion >= 0.03,  
                                              "populares",  
                                              "no populares"))
```

```
table(guaguas$populares)
```

```
##  
## no populares    populares  
##      845162        238
```

mutate() para condiciones

La condición pueden ser muchas cosas.

Por ejemplo, una lista de nombres:

```
## Creamos un vector de nombres
curso<-c("Isidora","Alejandra","Noemí","Leslie","Javier","Valentina",
       "Andrés","Matías","Kevin")  
  
guaguas <- mutate(guaguas,
                  curso = if_else(nombre %in% curso, 1, 0))
```

mutate() para condiciones

```
head(guaguas, 3)
```

```
## # A tibble: 3 x 7
##   anio nombre sexo     n proporcion populares curso
##   <dbl> <chr>  <chr> <dbl>      <dbl> <chr>       <dbl>
## 1 1920 María   F     2130      0.104 populares    0
## 2 1920 José    M     984       0.0483 populares    0
## 3 1920 Juan    M     636       0.0312 populares    0
```

```
table(guaguas$curso)
```

```
##
##          0         1
## 844469    931
```

mutate() para condiciones

Esto tiene mucha potencia para filtra data frames según condiciones.

A partir de la nueva variable podemos solo dejar en guaguas2 los nombres del curso:

```
guaguas2 <- filter(guaguas, curso==1)
```

```
table(guaguas2$nombre)
```

```
##  
## Alejandra      Andrés      Isidora      Javier      Kevin       Leslie      Matías  
##          99          109          106          104          62         134          104  
## Valentina  
##          105
```

R project e importación de datos

R project

Si tuviésemos recursos ilimitados, y mucho tiempo, produciríamos nuestras propias bases de datos.

Con esto, podríamos llegar a tener operacionalizaciones precisas de nuestros conceptos.

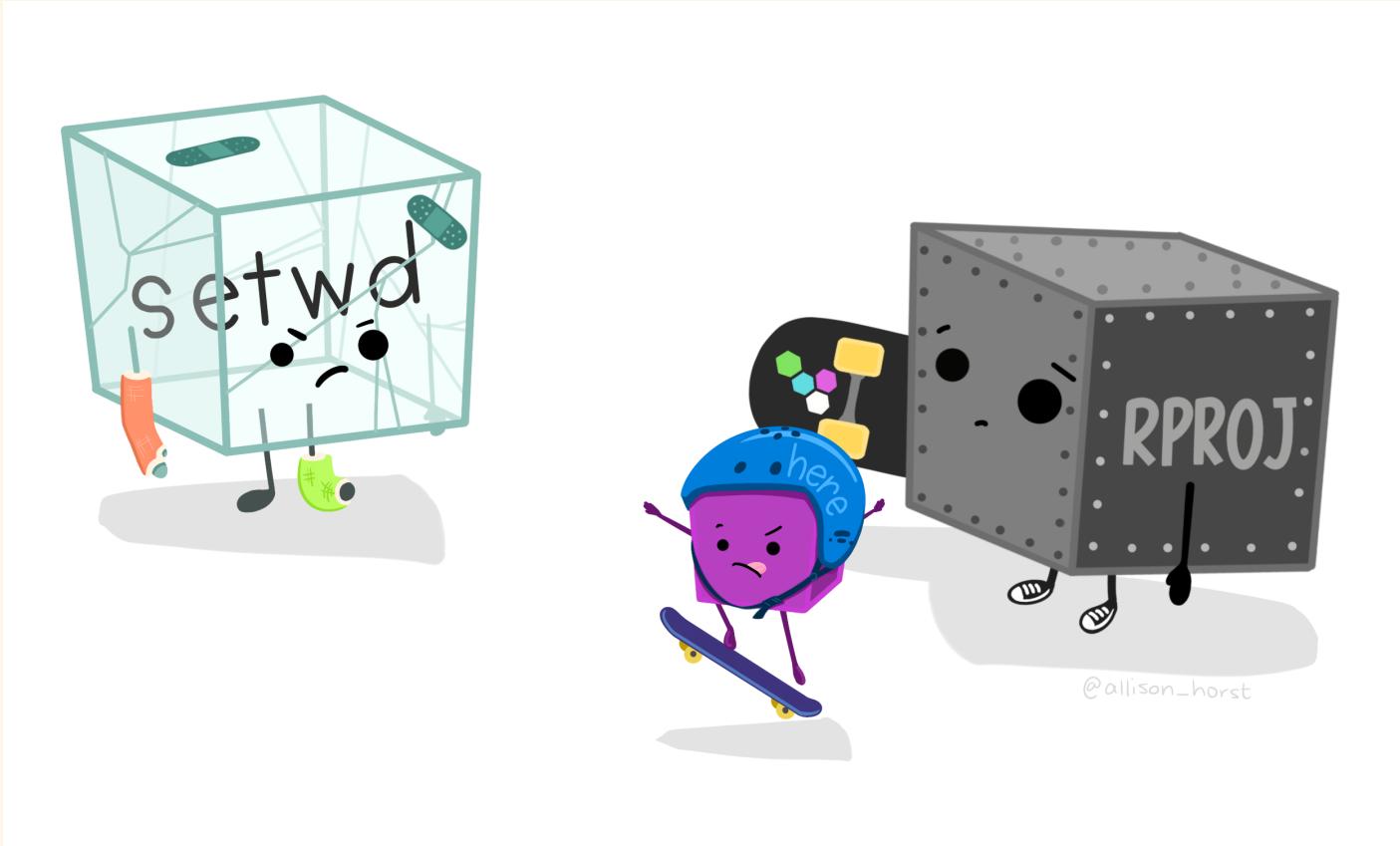
Por lo general, tendremos que contentarnos y saber utilizar bases producidas por otros equipos de investigación.

Análisis secundario de datos. El dato es recogido y procesado por un investigador, y luego este es reanalizado por otro para un propósito diferente.

Desde hoy comenzaremos a revisar las principales data frame que existen en ciencias sociales y que son de acceso libre.

Aprenderemos a abrir las y a conocer sus principales características antes de hacer análisis más complejos.

Antes: directorios de trabajo



@allison_horst

Antes: directorios de trabajo

Hasta ahora no nos hemos preocupado de este tema: cargábamos guaguas desde el mismo paquete.

¿Como cargamos conjuntos de datos que descargamos y almacenamos en nuestro computador?

Si descargamos y guardamos los datos en el **directorio o carpeta correcta** podremos abrir nuestros datos desde R.

¿Cuál es la carpeta o directorio correcto? Cualquiera...

El directorio es la ruta de nuestro computador desde la cuál R busca archivos para cargar.

Antes: directorios de trabajo

¿Cuál es mi ruta?

```
getwd()
```

```
## [1] "C:/Users/Nratto/Documents/Github/0PS079_1_UCSH2021/clases/clase4"
```

Desde esta ruta mi R está trabajando. Desde acá debo indicar si quiero avanzar o retroceder para buscar archivos.

Así puedo conocer los archivos que están en la ruta definida por defecto:

```
list.files()
```

```
## [1] "clase4a_diseno.html"  "clase4a_diseno.rmd"    "clase4b_rproj.html"
## [4] "clase4b_rproj.pdf"     "clase4b_rproj.Rmd"   "clase4b_rproj_files"
## [7] "data"                  "Imagenes"        "libs"
## [10] "xaringan-themer.css"
```

Es lo mismo que especificar la ruta:

Antes: directorios de trabajo

Si quiero conocer los archivos que están en otra ruta, solo hay que especificarla:

```
list.files("C:/Users/Nratto/Documents/Github/OPS079_1_UCSH2021/clases/c1
```

```
## [1] "casen.png"           "create_project.PNG"
## [3] "diseno.png"          "dplyr_mutate.png"
## [5] "est_transversal.PNG"   "est_transversal2.PNG"
## [7] "Ficha_tecnica_senadis.JPG" "h1_sindicatos.JPG"
## [9] "lieberman.PNG"         "mec_causal.PNG"
## [11] "mill.PNG"              "mill_eeuu_france.PNG"
## [13] "owidR.PNG"             "panel.PNG"
## [15] "pipes.jpg"             "Rproj.JPG"
## [17] "RprojvsSetwd.url"     "tidyvsmessy.url"
```

Otra opción mas sencilla es llegar a imágenes desde la ruta ya definida por R:

```
list.files("imagenes")
```

```
## [1] "casen.png"           "create_project.PNG"
## [3] "diseno.png"          "dplyr_mutate.png"
## [5] "est_transversal.PNG"   "est_transversal2.PNG"
```

Antes: directorios de trabajo

¿Y si quiero retroceder en la estructura de carpetas?

```
list.files("../")  
  
## [1] "clase1"  "clase10" "clase11" "clase2"  "clase3"  "clase4"  "clase5"  
## [8] "clase6"  "clase7"  "clase8"  "clase9"
```

Que sería lo mismo que decir:

```
list.files("C:/Users/Nratto/Documents/Github/OPS079_1_UCSH2021/clases/".  
  
## [1] "clase1"  "clase10" "clase11" "clase2"  "clase3"  "clase4"  "clase5"  
## [8] "clase6"  "clase7"  "clase8"  "clase9"
```

Antes: directorios de trabajo

¿Qué pasa si no me es cómodo el directorio desde el cuál R está trabajando?

Se puede cambiar. La lógica es ocupar directorios diferentes para cada uno de los proyectos en los que trabajamos.

Mi directorio de trabajo para esta clase es:

```
## [1] "C:/Users/Nratto/Documents/Github/OPS079_1_UCSH2021/clases/clase4"
```

Pero para la clase anterior ocupé

```
## [1] "C:/Users/Nratto/Documents/Github/OPS079_1_UCSH2021/clases/clase3"
```

Antes: directorios de trabajo

Los directorios se pueden cambiar, pero además se puede hacer que estos actúen de manera **relativa**, no **absoluta**.

Por ejemplo, cuando a R le damos instrucciones desde acá:

```
## [1] "C:/Users/Nratto/Documents/Github/0PS079_1_UCSH2021/clases/clase4"
```

Estas instrucciones solo funcionarán en mi computador (Nratto).

Pero la gracia de R es la contraria. Que en código pueda ser utilizado por terceros en cualquier otro computador.

Por la tanto, las rutas absolutas son un problema. Solo funciona desde nuestras computadoras personales.

R Project

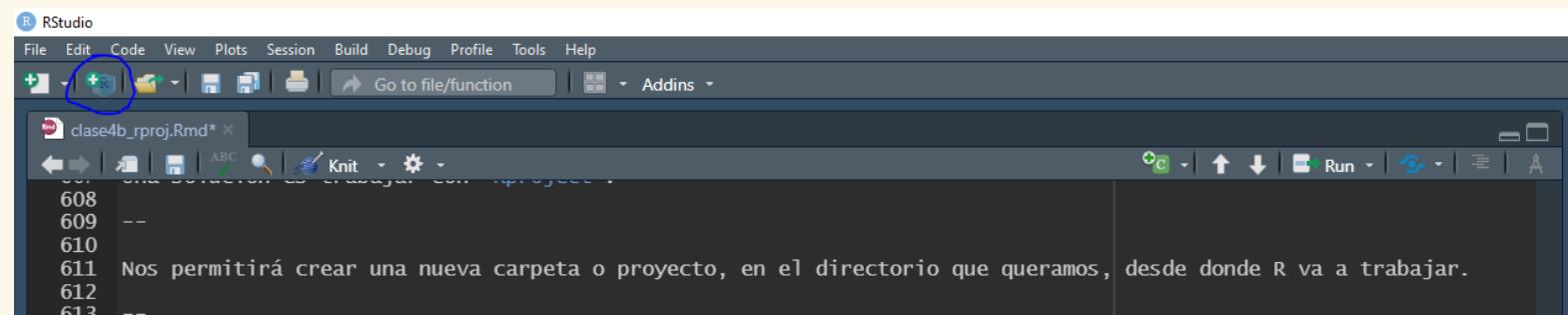
Una solución es trabajar con Rproject.

Nos permitirá crear una nueva carpeta o proyecto, en el directorio que queramos, desde donde R va a trabajar.

Esta carpeta se puede adjuntar y compartir con otras personas, lo que permite el trabajo colaborativo.

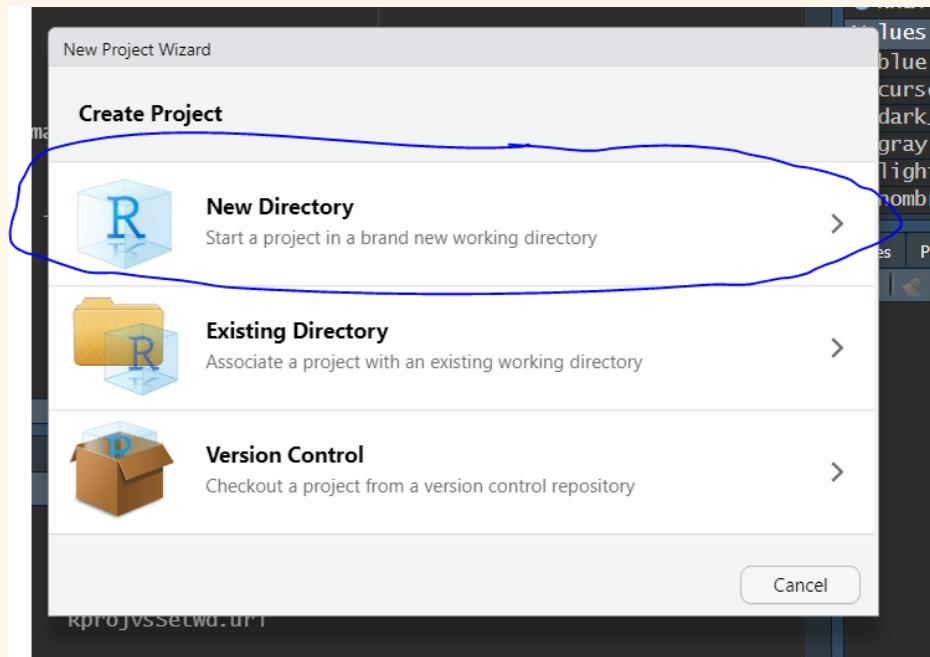
R se abrirá desde cualquier otro computador.

Para crear un nuevo proyecto:



R Project

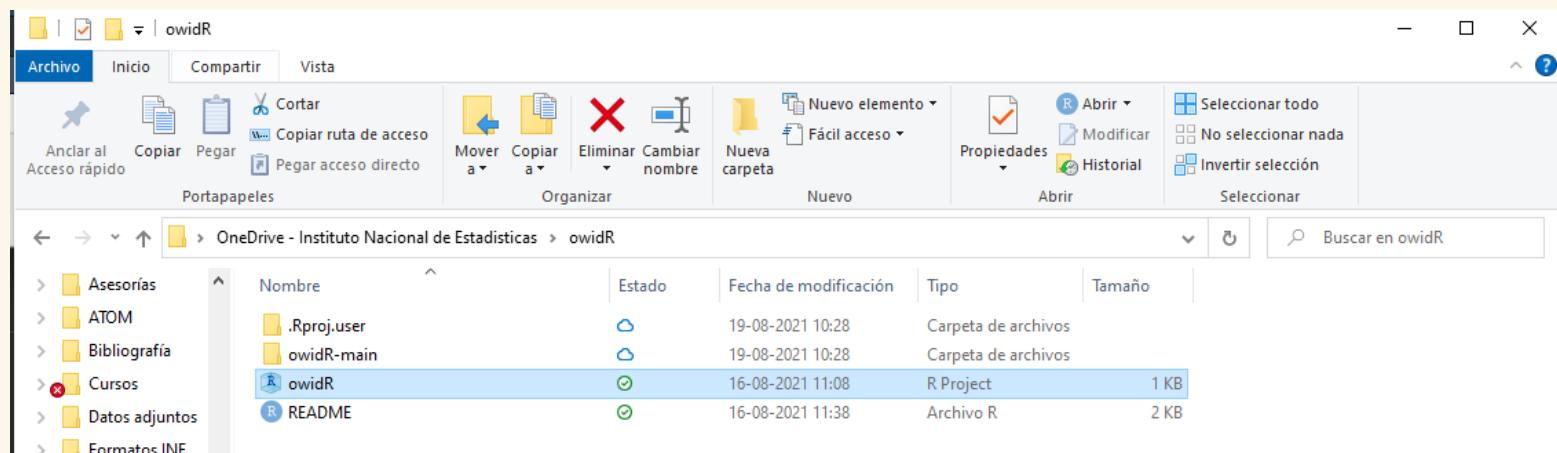
Señalar que queremos crear un nuevo directorio:



Definir donde crear la carpeta del nuevo proyecto y darle "aceptar".

R Project

Si todo sale bien, se nos habrá creado una carpeta con el nombre del proyecto, la cuál tendrá un archivo "nombrecarpeta.Rproj"



Ese archivo debe abrirse cada vez que queramos trabajar en el proyecto, para que quede definida la ruta de trabajo.

Una vez abierta la sesión, crear R script, RMD, subcarpetas y colocar archivos. Lo que quieran.

R Project

Veamos como funciona (ejercicio práctico)

Creemos un R Project donde quieran, pero que se llame "afghanistan"



R Project

Una vez creado el proyecto, descargar el siguiente set de datos:

[afghanistan.rds](#)

Guardarlos dentro de la carpeta "afghanistan" que acaban de crear.

Luego carguemos el dato:

```
data <- readRDS(file = "data/afghanistan.rds")  
dim(data)  
  
## [1] 61 1443
```

R Project

Tenemos 1443 variables sobre Afganistán

Como la lista es interminable, vemos los nombres de las 10 primeras:

```
names(data)[1:10]
```

```
## [1] "anio"
## [2] "internally_displaced_persons_new_displacement_associated_with_conflic"
## [3] "mercaderias_exportadas_hacia_economias_en_desarrollo_en_europa_y_asia"
## [4] "indice_de_terminos_netos_de_intercambio_2000_100"
## [5] "mercaderias_importadas_desde_economias_de_ingreso_alto_percent_del_to"
## [6] "participacion_de_lineas_arancelarias_con_maximos_internacionales_prod"
## [7] "participacion_de_lineas_arancelarias_con_maximos_internacionales_prod"
## [8] "poblacion_urbana_percent_del_total"
## [9] "poblacion_total"
## [10] "poblacion_de_65_anos_de_edad_y_mas_hombres_percent_del_total"
```

R Project

```
## Resumen de la variable
summary(data$poblacion_total)

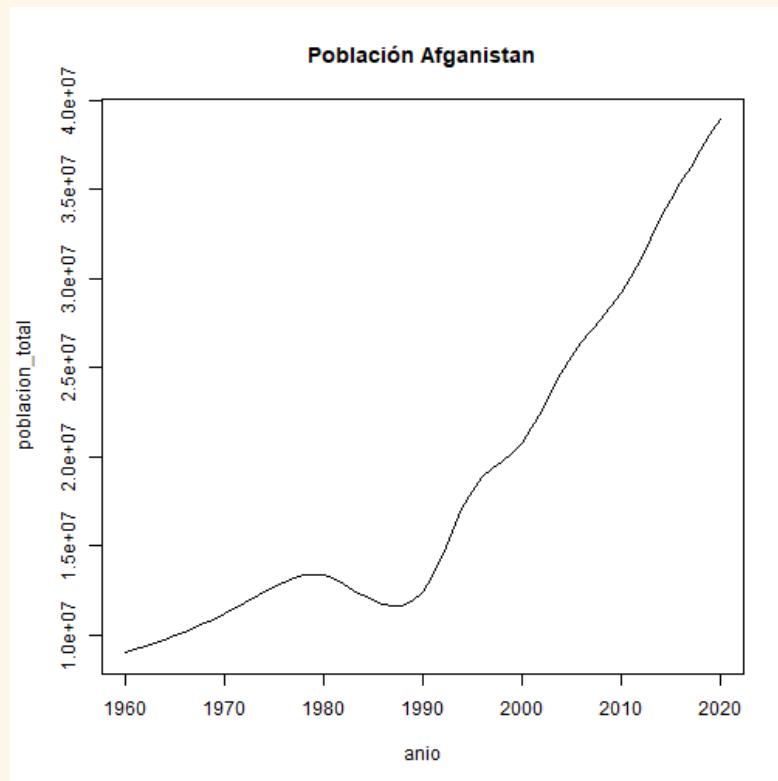
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## 8996967 11791222 13356500 18699394 25654274 38928341

## dato del último año
select(filter(data,anio==2020),poblacion_total)

## # A tibble: 1 x 1
##   poblacion_total
##   <dbl>
## 1 38928341
```

R Project

```
plot(poblacion_total ~ anio,data=data, type = "l", main="Población Afganistán")
```



R Project

Buscar variable de interés.

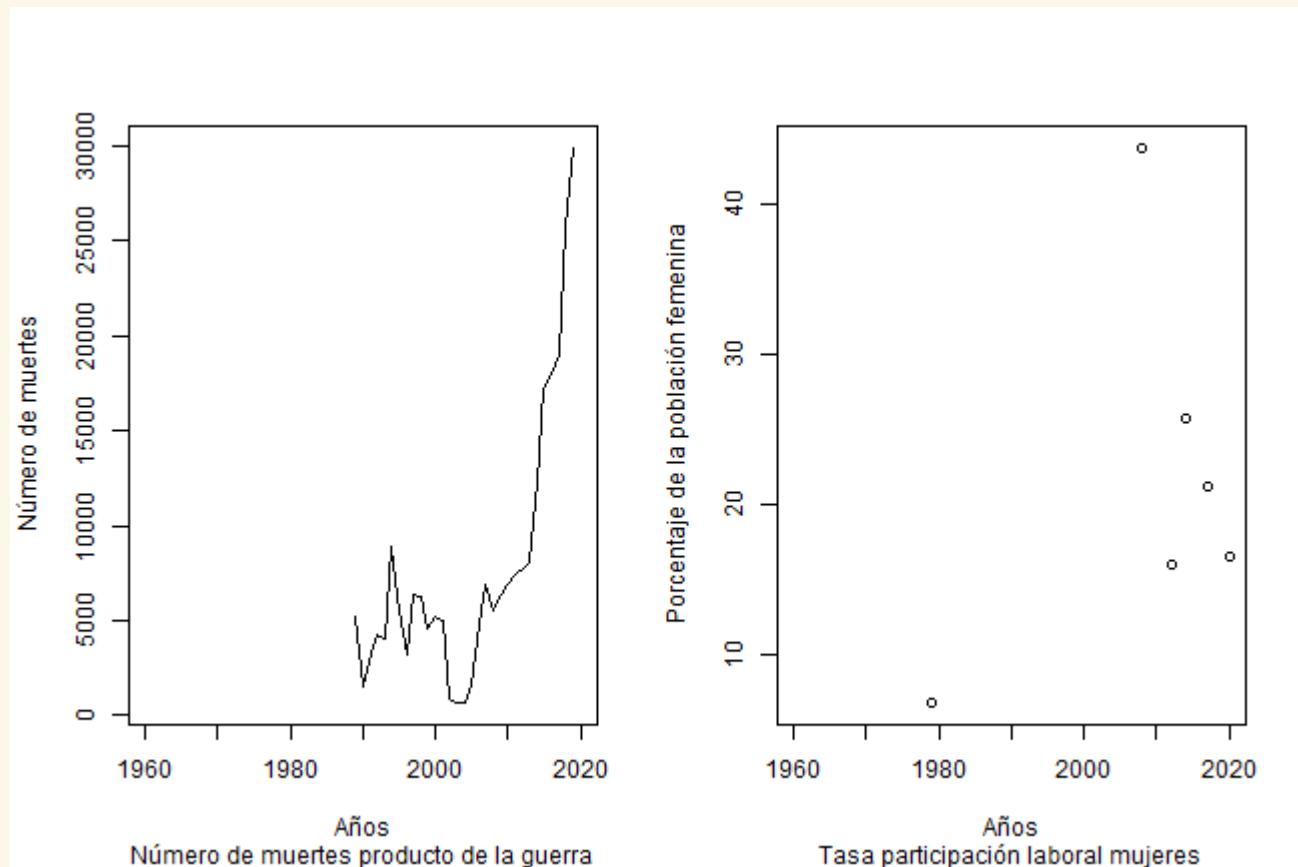
Para hacerlo de forma sencilla descarguemos y carguemos este paquete:

```
#install.packages("sjmisc")
library(sjmisc)

find_var(data, "muerte")

## [1] "numero_de_muertes_maternas"
## [2] "numero_de_muertes_infantiles"
## [3] "numero_de_muertes_de_menores_de_5_anos"
## [4] "muertes_producto_de_la_guerra_cantidad_de_personas"
## [5] "numero_de_muertes_de_recien_nacidos"
## [6] "exhaustividad_del_registro_de_muertes_con_informacion_sobre_causa_de_m
```

R Project



Ejercicio práctico 2

Como ejercicio práctico identifiquen y seleccionen dos variables de interés sobre Afganistán.

Guarden esas **dos variables** más la variable **anio** en una nueva data llamada "data2"

Describan la distribución de las dos variables y reporten el dato más reciente de ambas (no siempre es 2020)

Intentemos hacer un gráfico de una de las dos variables.

Al terminar ejecuten estas dos funciones:

```
saveRDS(data2, file = "data2.rds") # para exportar la data2  
file.remove("afganistan.rds")      # para eliminar data afganistan
```

Comprimir carpeta del proyecto y enviar a Nicolás y Kevin

(solo tres archivos: .R, .Rproj y .rds)

Recursos web utilizados

Xaringan: Presentation Ninja, de Yihui Xie. Para generar esta presentación.

Ilustraciones de Allison Horst

Datos de Afganistán descargados de [World Data Bank](#)

Para reforzar y seguir aprendiendo

Otra explicación de los R Project

Capítulo 8 "Flujo de trabajo: proyectos" de Wickham **muy** recomendado.

Bibliografía utilizada

Wickham, H. (2021). *R Para Ciencia de Datos*.