

# Problem Set 3

Juan Camilo Parada Sandoval

06/12/2023

## Punto 1

```
## initial setup
rm(list=ls())
require(pacman)

## Loading required package: pacman

## Warning: package 'pacman' was built under R version 4.3.2

p_load(tidyverse,rio,data.table)
```

### Punto 1.1

```
rutas <- list.files("input" , recursive=T , full.names=T)
```

Cargamos todas las bases de datos como una cadena de listas.

### Punto 1.2

```
rutas_resto <- str_subset(string = rutas , pattern = "Resto - Ca")
```

Filtramos y extraemos en una lista únicamente las bases que sean “Resto - Características generales”.

```
lista_resto <- import_list(file = rutas_resto)

## Warning in (function (input = "", file = NULL, text = NULL, cmd = NULL, :
## Detected 33 column names but the data has 32 columns. Filling rows
## automatically. Set fill=TRUE explicitly to avoid this warning.

## Warning in (function (input = "", file = NULL, text = NULL, cmd = NULL, :
## Stopped early on line 10. Expected 33 fields but found 34. Consider fill=TRUE
## and comment.char=. First discarded non-empty line: <<5550967 1 1 1 10 1 1 6
## 1989 32 1 6 2 2 2 1 2 1 1 1 48000 2 1 2 5 11 2 2 11 09 68 2105.2640588>>
```

Cargamos la lista ya filtrada y extraida.

```
rutas_resto[1]

## [1] "input/2019/Abril.csv/Resto - Características generales (Personas).csv"

str_sub(rutas_resto[35],start = 14 , 17)

## [1] "tubr"
```

Identificamos los caracteres en donde aparecen los meses y años.

```
View(lista_resto[[1]])
lista_resto[[1]]$path <- rutas_resto[1]
```

Creamos la ruta de identificacion de las bases de datos.

```
for (i in 1:length(lista_resto)){
  lista_resto[[i]]$path <- rutas_resto[i]
  lista_resto[[i]]$year <- str_sub(lista_resto[[i]]$path,start = 14 , 17)
}
View(lista_resto[[20]])
```

Organizamos los dataframes por mes y año mediante un bucle.

### Punto 1.3

```
lista_resto[[36]] <- NULL
cg <- rbindlist(l=lista_resto , use.names=T , fill=T)
```

Eliminamos la 36va base de datos porque estaba mal especificada entre los archivos del enunciado.

Cargamos la base de datos completa con los datos organizados por mes y año.

```
export(cg,"output/db_full.rds")
```

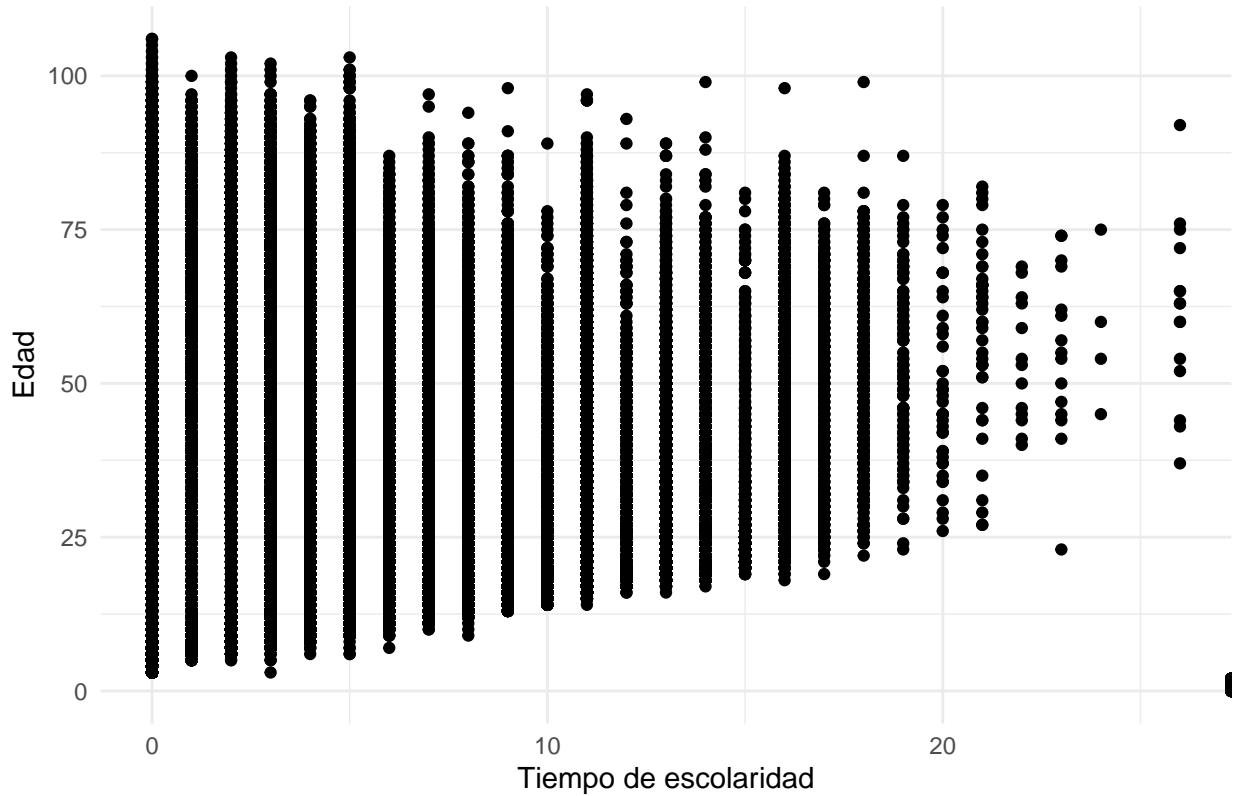
Exportamos la base de datos ya creada.

### Punto 2

```
ggplot(data = cg,aes(x=ESC, y= P6040)) + geom_point() +
  labs(title = "Relacion entre edad y tiempo de escolaridad",
       x = "Tiempo de escolaridad",
       y = "Edad") +
  theme_minimal()
```

```
## Warning: Removed 9464 rows containing missing values ('geom_point()').
```

## Relacion entre edad y tiempo de escolaridad



```
df <- cg %>%
  filter(P3246 == 1) %>%
  select(P6040, ESC)
```

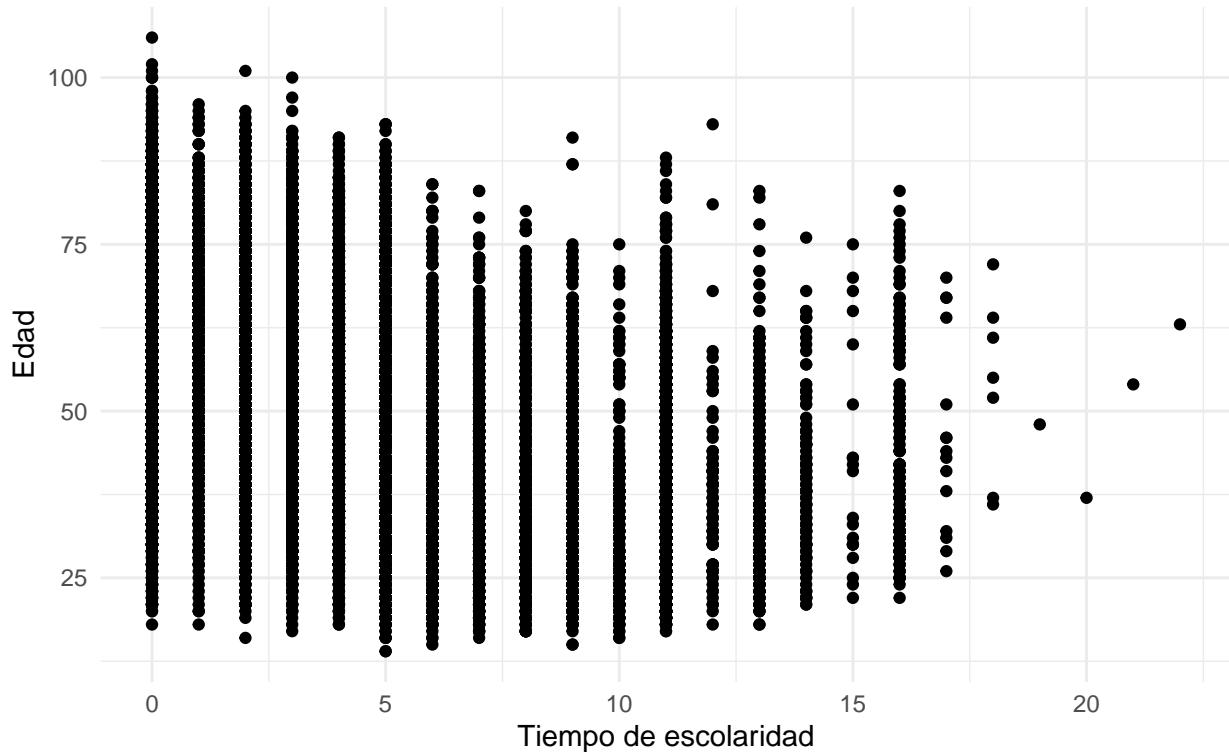
Hacemos un ajuste para ver los datos de las personas que se consideran pobres. Posteriormente se hará lo mismo para aquellos que no se consideran pobres.

```
ggplot(data = df, aes(x=ESC, y= P6040)) + geom_point() +
  labs(title = "Relacion entre edad y tiempo de escolaridad",
       subtitle = "Datos para personas que se consideran pobres segun la GEIH",
       x = "Tiempo de escolaridad",
       y = "Edad") +
  theme_minimal()
```

```
## Warning: Removed 2 rows containing missing values ('geom_point()').
```

## Relacion entre edad y tiempo de escolaridad

Datos para personas que se consideran pobres segun la GEIH



```
df2 <- cg %>%
  filter(P3246 == 2) %>%
  select(P6040, ESC)

ggplot(data = df2, aes(x=ESC, y= P6040)) + geom_point() +
  labs(title = "Relacion entre edad y tiempo de escolaridad",
       subtitle = "Datos para personas que no se consideran pobres segun la GEIH",
       x = "Tiempo de escolaridad",
       y = "Edad") +
  theme_minimal()
```

## Warning: Removed 2 rows containing missing values ('geom\_point()').

## Relacion entre edad y tiempo de escolaridad

Datos para personas que no se consideran pobres segun la GEIH

