R para el análisis estadístico de datos

Ana Escoto

6/10/24

Table of contents

Sobre el curso

1. Introducción a R y Rstudio (4 horas)

Objetivo: que el estudiantado sea familiarice con la interfase de trabajo y la programación por objetos, del mismo modo que sean capaces de realizar tareas básicas tales como crear un script, un proyecto, objetos, ambientes e instalar paqueterías.

2. Importación de información y primera revisión de fuentes demográficas (4 horas)

a. Importación de información a R en diferentes formatos
b. Revisión de encuestas y ejemplos de importación de datos en formatos diferentes
c. Revisión preliminar y limpieza de información

Objetivo: que el estudiantado sea capaz de importar información desde diferentes formatos (.txt, .csv, .xlsx, .dta, .dbf) a R, así como de exportar sus resultados en estos formatos. Del mismo modo que sean capaces de revisar de manera preliminar los objetos de tipo `data.frame`*: funciones* `dplyr::glimpse()`*,* `skimr::skim()`*, manejo de etiquetas y hacer subconjuntos de información*

3. Revisión de elementos estadísticos básicos desde `{tidyverse}` (8 horas)

a. Tabulados con `janitor::tabyl()` y uso de factores de expansión con `pollster::topline()`, `pollster::crosstab`. Lectura e interpretación de tablas de doble entrada
b. Estadística descriptiva básica (medidas de tendencia central, dispersión y de posición) con el paquete `{dplyr}`
c. Gráficos univariados y bivariados usando `{ggplot2}` y extensiones de `{ggplot2}`
d. Fusionado de información

Objetivo: que el estudiantado sea capaz de realizar análisis estadísticos básicos utilizando encuestas mexicanas

4. Estimaciones por intervalo y diseño complejo muestral (4 horas)

a. Estimaciones para medias

b. Estimaciones para proporciones

c. Estimaciones para totales

d. Estimaciones lineales de coeficientes

Objetivo: que el estudiantado sea capaz de realizar intervalos de confianza, calculo de errores estándar con diseño muestral complejo y sepa evaluar la confiabilidad de sus estimaciones

Facilitadora

Ana Ruth Escoto Castillo

Profesora de tiempo completo en la Facultad de Ciencias Políticas y Sociales, UNAM. Doctora en Estudios de Población por El Colegio de México, cuenta con el nivel I en el Sistema Nacional de Investigadores. Coorganizadora del capítulo de la CDMX de la iniciativa global Rladies. Le interesa el bienestar de la población, en el presente, analizando los procesos de desigualdad y exclusión en los mercados laborales latinoamericanos; y, en el futuro, a través del estudio de la sustentabilidad. Su experiencia en el ámbito académico se ha concentrado en el estudio de este bienestar, específicamente en el análisis sociodemográfico de las condiciones laborales y la vinculación del comercio exterior con el mercado de trabajo, en la relación del cambio climático y la distribución de ingresos, el consumo energético de los hogares y sus implicaciones ambientales. Posee experiencia en recolección de información estadística, diseño y control de procesos de recolección y su procesamiento. Ha aplicado diversos métodos y herramientas multivariadas, homologación de información y comparabilidad de fuentes en sus investigaciones, así como usa de diversos softwares estadísticos, y ha impartido clases de estadítica aplicada a nivel de licenciatura y posgrado.

Instalación de R y Rstudio

Introducción a R

https://youtu.be/YkN5urybh2A

Instalación en OS

1. Necesito que instalen la versión más nueva de R: Download R-4.4.0 of MAC. *The R-project for statistical computing*. https://cran.r-project.org/bin/macosx/

Elije la versión de acuerdo a tu procesador, intel o ARM.

2. Instalar también las herramientas Quartz, xcode y fortran

- https://www.xquartz.org/
- https://developer.apple.com/xcode/resources/
- https://mac.r-project.org/tools/gfortran-12.2-universal.pkg

3. Después de eso instalar el Rstudio, que hoy se encuentra alojado en el sitio posit, que vaya acorde con MAC

https://posit.co/download/rstudio-desktop/

Algunas indicaciones en video, pero son algo viejitas y pueden cambiar las versiones de R.

https://youtu.be/icWV8jzYOtA

Algunas indicaciones en video, pero son algo viejitas y pueden cambiar las versiones de R.

Instalación en PC

1. Necesito que instalen la versión más nueva de R: Download R-4.4.0 for Windows. *The R-project for statistical computing.* https://cran.r-project.org/bin/windows/base/
2. Instalar también la herramienta RTools https://cran.r-project.org/bin/windows/Rtools/rtools44/rtools.html
3. Después de eso instalar el Rstudio, que hoy se encuentra alojado en el sitio posit, que vaya acorde con Windows https://posit.co/download/rstudio-desktop/

Algunas indicaciones en video, pero son algo viejitas y pueden cambiar las versiones de R.

https://youtu.be/TNSQikMfgJI

Ojo

Desde octubre de 2022, RStudio se volvió **"Posit"**

1 Primer acercamiento al uso del programa

1.1 Introducción

En RStudio podemos tener varias ventanas que nos permiten tener más control de nuestro "ambiente", el historial, los "scripts" o códigos que escribimos y por supuesto, tenemos nuestra consola, que también tiene el símbolo ">" con R. Podemos pedir operaciones básicas

```
  2+5
```

```
[1] 7
```

```
  5*3
```

```
[1] 15
```

```
  #Para escribir comentarios y que no los lea como operaciones ponemos el símbolo de gato
  # Lo podemos hacer para un comentario en una línea o la par de una instrucción

  1:5               # Secuencia 1-5
```

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

```
  seq(1, 10, 0.5)   # Secuencia con incrementos diferentes a 1
```

```
 [1]  1.0  1.5  2.0  2.5  3.0  3.5  4.0  4.5  5.0  5.5  6.0  6.5  7.0  7.5  8.0
[16]  8.5  9.0  9.5 10.0
```

```
  c('a','b','c')  # Vector con caracteres
```

```
[1] "a" "b" "c"
```

```r
1:7             # Entero
```

```
[1] 1 2 3 4 5 6 7
```

```r
40 < 80         # Valor logico
```

```
[1] TRUE
```

```r
2 + 2 == 5      # Valor logico
```

```
[1] FALSE
```

```r
T == TRUE       # T expresion corta de verdadero
```

```
[1] TRUE
```

R es un lenguaje de programación por objetos. Por lo cual vamos a tener objetos a los que se les asigna su contenido. Si usamos una flechita "<-" o "->" le estamos asignando algo al objeto que apunta la felcha.

```r
x <- 24         # Asignacion de valor 24 a la variable x para su uso posterior (OBJETO)
x/2             # Uso posterior de variable u objeto x
```

```
[1] 12
```

```r
x               # Imprime en pantalla el valor de la variable u objeto
```

```
[1] 24
```

```r
x <- TRUE       # Asigna el valor logico TRUE a la variable x OJO: x toma el ultimo valor
x
```

```
[1] TRUE
```

1.2 Vectores

Los vectores son uno de los objetos más usados en R.

```r
y <- c(2, 4, 6)     # Vector numerico
y <- c('Primaria', 'Secundaria') # Vector caracteres
```

Dado que poseen elementos, podemos también observar y hacer operaciones con sus elementos, usando "[ ]" para acceder a ellos

```r
y[2]              # Acceder al segundo valor del vector y
```

```
[1] "Secundaria"
```

```r
y[3] <- 'Preparatoria y más' # Asigna valor a la tercera componente del vector
sex <-1:2         # Asigna a la variable sex los valores 1 y 2
names(sex) <- c("Femenino", "Masculino") # Asigna nombres al vector de elementos sexo
sex[2]            # Segundo elemento del vector sex
```

```
Masculino
        2
```

1.3 Matrices

Las matrices son muy importantes, porque nos permiten hacer operaciones y casi todas nuestras bases de datos tendran un aspecto de matriz.

```r
m <- matrix (nrow=2, ncol=3, 1:6, byrow = TRUE) # Matrices Ejemplo 1
m
```

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

```r
m <- matrix (nrow=2, ncol=3, 1:6, byrow = FALSE) # Matrices Ejemplo 1
m
```

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

```
dim(m)
```

```
[1] 2 3
```

```
attributes(m)
```

```
$dim
[1] 2 3
```

¿Qué hace “byrow”?

```
n <- 1:6     # Matrices Ejemplo 2
dim(n) <- c(2,3)
n
```

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

```
xx <-10:12   # Matrices Ejemplo 3
yy<-14:16
cbind(xx,yy) # Une vectores por Columnas
```

```
     xx yy
[1,] 10 14
[2,] 11 15
[3,] 12 16
```

```
rbind(xx,yy) # Une vectores por Renglones
```

```
   [,1] [,2] [,3]
xx   10   11   12
yy   14   15   16
```

```
mi_matrix<-cbind(xx,yy) # este resultado lo puedo asignar a un objeto
```

1.4 data.frames o conjuntos de datos

```
mi_dataframe<-as.data.frame(m)
```

El formato matricial sigue sirviendo:

```
mi_dataframe[2,]
```

```
  V1 V2 V3
2  2  4  6
```

```
mi_dataframe[,2]
```

```
[1] 3 4
```

Pero también podemos utilizar el símbolo de peso para cada variable:

```
mi_dataframe$V2
```

```
[1] 3 4
```

Puedo agregar variables columnas:

```
cbind(mi_dataframe, c("a", "b"), c(T, F))
```

```
  V1 V2 V3 c("a", "b") c(T, F)
1  1  3  5           a    TRUE
2  2  4  6           b   FALSE
```

Qué pasa con las matrices

```
cbind(m, c("a", "b"),  c(T, F))
```

```
     [,1] [,2] [,3] [,4] [,5]
[1,] "1"  "3"  "5"  "a"  "TRUE"
[2,] "2"  "4"  "6"  "b"  "FALSE"
```

Checa cómo cambian los elementos. En una matriz todos los elementos deben ser del mismo tipo.

Podemos crear "a mano" dataframes:

```
data<-data.frame(
  "entero" = 1:4,
  "factor" = as.factor(c("a", "b", "c", "d")),
  "numero" = c(1/1, 1/2, 1/3, 1/4),
  "cadena" = as.character(c("a", "b", "c", "d"))
)
```

Los data.frames tienen una estructura

```
str(data)
```

```
'data.frame':   4 obs. of  4 variables:
 $ entero: int  1 2 3 4
 $ factor: Factor w/ 4 levels "a","b","c","d": 1 2 3 4
 $ numero: num  1 0.5 0.333 0.25
 $ cadena: chr  "a" "b" "c" "d"
```

1.5 Valores y perdidos

Además de caracteres, numéricos y lógicos hay también valores perdidos. Y de varios tipos

```
vector<-c(1:5, # numérico
          T, # lógico
          NA, # perdido
          "a", # caracter
          5/0, # no es un número
          sqrt(-1))
```

```
Warning in sqrt(-1): Se han producido NaNs
```

Si lo imprimimos vamos a ir viendo cómo se convierten ciertos valores a otros al quererlos incluir en un mismo conjunto:

```
vector
```

```
 [1] "1"    "2"    "3"    "4"    "5"    "TRUE" NA     "a"    "Inf"  "NaN"
```

Quitaremos el caracter

```
vector<-c(1:5, # numérico
          T, # lógico
          NA, # perdido
          5/0, # Infinito
          sqrt(-1))
```

```
Warning in sqrt(-1): Se han producido NaNs
```

```
vector
```

```
[1]   1   2   3   4   5   1  NA Inf NaN
```

¿Qué le pasó al valor lógico?

Hay unos operadores que nos señalan si los valores son perdidos o infinitos o "Not a number"

```
is.na(vector)
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
```

```
is.nan(vector)
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
```

```
is.infinite(vector)
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
```

1.6 Funciones

Algunas funciones básicas son las siguientes. Vamos a ir viendo más funciones, pero para entender cómo funcionan, haremos unos ejemplos y cómo pedir ayuda sobre ellas.

```r
sum(10, 20, 30)    # Función suma
```

```
[1] 60
```

```r
rep('R', times = 3) # Repite la letra R el numero de veces que se indica
```

```
[1] "R" "R" "R"
```

```r
sqrt(9)           # Raiz cuadrada de 9
```

```
[1] 3
```

1.7 Listas

Las listas son conjuntos de objetos y pueden ser de varios tipos

```r
milista<- list(data, m, xx, "a")
```

```r
milista
```

```
[[1]]
  entero factor    numero cadena
1      1      a 1.0000000      a
2      2      b 0.5000000      b
3      3      c 0.3333333      c
4      4      d 0.2500000      d

[[2]]
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

[[3]]
[1] 10 11 12

[[4]]
[1] "a"
```

Ojo con los corchetes

```
milista[[1]]
```

```
  entero factor    numero cadena
1      1      a 1.0000000      a
2      2      b 0.5000000      b
3      3      c 0.3333333      c
4      4      d 0.2500000      d
```

Si queremos ponerle nombres a los elementos

```
milista<- list(datos = data,
               matriz = m,
               vector = xx,
               valor = "a")

milista
```

```
$datos
  entero factor    numero cadena
1      1      a 1.0000000      a
2      2      b 0.5000000      b
3      3      c 0.3333333      c
4      4      d 0.2500000      d

$matriz
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6

$vector
[1] 10 11 12

$valor
[1] "a"
```

1.8 Ayuda

Pedir ayuda es indispensable para aprender a escribir nuestros códigos. A prueba y error, es el mejor sistema para aprender. Podemos usar la función help, example y ?

```r
help(sum)         # Ayuda sobre función sum
example(sum)      # Ejemplo de función sum
```

```
sum> ## Pass a vector to sum, and it will add the elements together.
sum> sum(1:5)
[1] 15

sum> ## Pass several numbers to sum, and it also adds the elements.
sum> sum(1, 2, 3, 4, 5)
[1] 15

sum> ## In fact, you can pass vectors into several arguments, and everything gets added.
sum> sum(1:2, 3:5)
[1] 15

sum> ## If there are missing values, the sum is unknown, i.e., also missing, ....
sum> sum(1:5, NA)
[1] NA

sum> ## ... unless  we exclude missing values explicitly:
sum> sum(1:5, NA, na.rm = TRUE)
[1] 15
```

1.9 Mi ambiente

Todos los objetos que hemos declarado hasta ahora son parte de nuestro "ambiente" (environment). Para saber qué está en nuestro ambiente usamos el comando

```r
ls()
```

```
 [1] "data"         "m"            "mi_dataframe" "mi_matrix"    "milista"
 [6] "n"            "sex"          "vector"       "x"            "xx"
[11] "y"            "yy"
```

```r
gc()           # Garbage collection, reporta memoria en uso
```

```
         used (Mb) gc trigger (Mb) limit (Mb) max used (Mb)
Ncells  639616 34.2    1346240 71.9         NA  1346240 71.9
Vcells 1200769  9.2    8388608 64.0      16384  2147389 16.4
```

Para borrar todos nuestros objetos, usamos el siguiente comando, que equivale a usar la escobita de la venta de environment

```
rm(list=ls())  # Borrar objetos actuales
```

1.10 Directorio de trabajo

Es muy útil saber dónde estamos trabajando y donde queremos trabajar. Por eso podemos utilizar los siguientes comandos para saberlo

Ojo, checa, si estás desdes una PC, cómo cambian las " " por"/" o por "\"

```
getwd()           # Directorio actual
```

```
[1] "/Users/anaescoto/Dropbox/2024/Curso_R_inter/r_analisis_datos/r_analisis_datos"
```

```
#setwd("")# Cambio de directorio

list.files()      # Lista de archivos en ese directorio
```

```
 [1] "Mi_Exportación.xlsx"      "P1.html"
 [3] "P1.qmd"                   "P1.rmarkdown"
 [5] "P2.qmd"                   "P3.qmd"
 [7] "README.md"                "_quarto.yml"
 [9] "datos"                    "docs"
[11] "index.html"               "index.qmd"
[13] "instala.html"             "instala.qmd"
[15] "intro1.png"               "mi_exportacion.sav"
[17] "mi_primer_ambiente.RData" "miexportacion.dta"
[19] "nombres_limpios.xlsx"     "r_analisis_datos.Rproj"
[21] "scripts"                  "site_libs"
[23] "videos.qmd"               "~$nombres_limpios.xlsx"
```

Checar que esto también se puede hacer desde el menú:

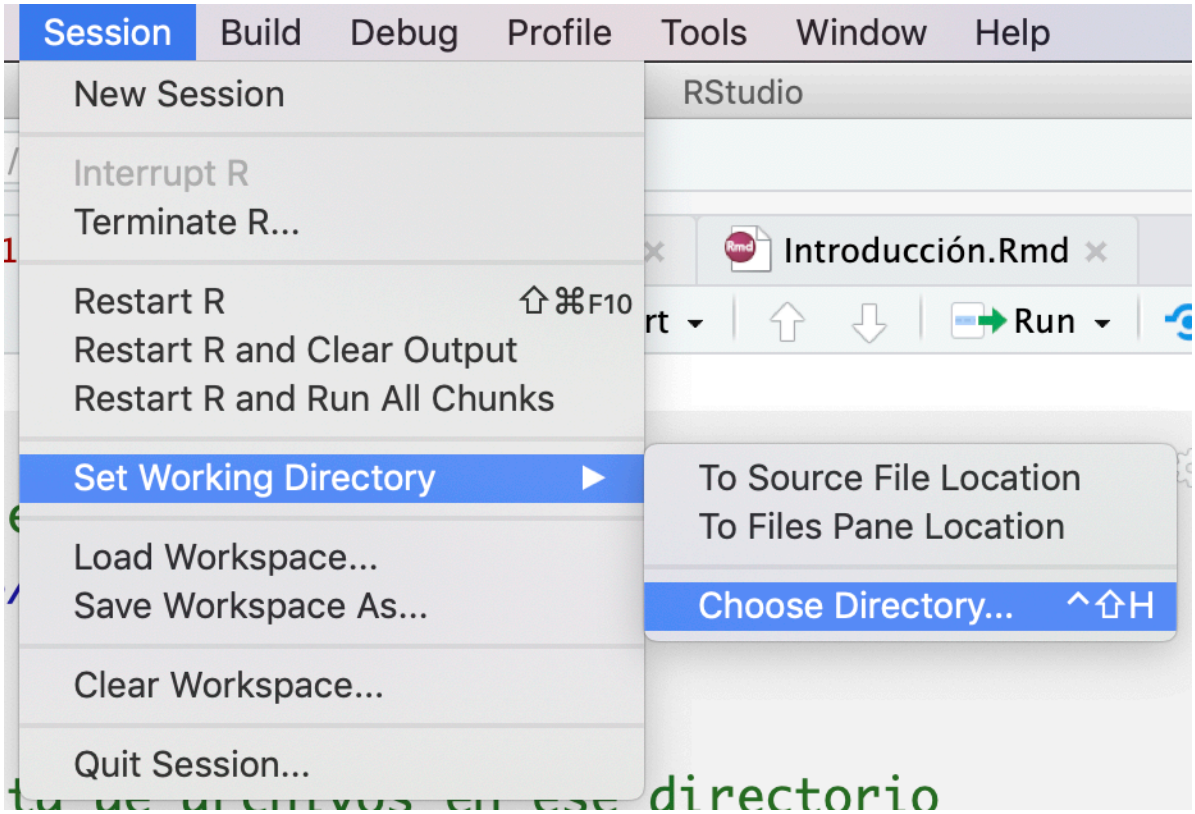

Session
Build
Debug
Profile
Tools
Window
Help
New Session
Interrupt R
Terminate R...
Restart R
Restart R and Clear Output
Restart R and Run All Chunks
Set Working Directory
Load Workspace...
Save Workspace As...
Clear Workspace...
Quit Session...
To Source File Location
To Files Pane Location
Choose Directory...
RStudio
Introducción.Rmd
Run

Figure 1.1: i0

1.11 Proyectos

Pero... a veces preferimos trabajar en proyectos, sobre todo porque nos da más control.

Hay gente que lo dice mejor que yo, como Hadley Wickham: https://es.r4ds.hadley.nz/flujo-de-trabajo-proyectos.html

1.12 Instalación de paquetes

Los paquetes son útiles para realizar funciones especiales. La especialización de paquetes es más rápida en R que en otros programas por ser software libre.

Vamos a instalar el paquete `{foreign}`, como su nombre lo indica, nos permite leer elementos "extranjeros" en R.

Para instalar las paqueterías usamos el siguiente comando `install.packages()` Checa que adentro del paréntesis va el nombre de la librería, con comillas.

Vamos a instalar dos librerías que nos permiten importar formatos.

```
#install.packages("foreign", dependencies = TRUE)
#install.packages("haven", dependencies = TRUE)
```

Este proceso no hay que hacerlo siempre. Si no sólo la primera vez. Una vez instalado un paquete de librería, la llamamos con el comando "library"

```
library(haven)
library(foreign)
```

`{foreing}` nos permite leer archivos en formato de *dBase*, con extensión ".dbf". Si bien no es un formato muy común para los investigadores, sí para los que generan la información, puesto que dBase es uno de los principales programas de administración de bases de datos.

He puesto un ejemplo de una base de datos mexicana en dbf, en este formato.

```
ejemplo_dbf<-foreign::read.dbf("datos/ejemplo_dbf.DBF") #checa cómo nos vamos adentro de n
```

Los `::` sirven para tres cosas:

- cargar un comando de un paquete, sin haberlo cargado
- para identificar de qué paquete viene el comando.
- para especificar en caso que hayan dos comandos iguales en un paquete, usar el que tenemos de los paquetes.

1.13 Paquete {pacman}

En general, cuando hacemos nuestro código querremos verificar que nuestras librerías estén instaladas. Si actualizamos nuestro R y Rstudio es probable (sobre todo en MAC) que hayamos perdido alguno.

Este es un ejemplo de un código. Y vamos a introducir un paquete muy útil llamado {pacman}

```
if (!require("pacman")) install.packages("pacman") # instala pacman si se requiere
```

```
Cargando paquete requerido: pacman
```

```
pacman::p_load(tidyverse, readxl, writexl, haven, sjlabelled, foreign) #carga los paquetes
```

Hay muchos formatos de almacenamiento de bases de datos. Vamos a aprender a importar información desde ellos.

1.14 Estilos

Escribir código tiene su gramática. Por lo general en este curso seguiremos el estilo de Google

https://google.github.io/styleguide/Rguide.html

1.15 Ejercicio 1

Realice en un **nuevo script** lo siguiente:

1. Escriba un vector "x", con los elementos 2,3,7,9. Muestre el resultado
2. Escriba un vector "y", con los elementos 9, 7, 3, 2. Muestre el resultado
3. Escriba un vector "year" con los años que van desde 1990 a 1993. Muestre el resultado
4. Escriba un vector "name" con los nombres de 4 de sus compañeros de curso. Muestre el resultado
5. Cree una matrix "m" 2x4 que incluya los valores 101 a 108, que se ordene según fila
6. ¿Cuáles son las dimensiones de la matriz "m"?

7. Cree una matriz “m2” juntado los vectores “x” y “y”, por sus filas ¿Cuáles son las dimensiones de la matriz “m2”?
8. Convierta esa matriz en un *data.frame*
9. Escriba una lista

Entregue su resultado en este formulario

2 Manejo de datos: importación, selección y revisión

2.1 Datos

Guarda en tu carpeta de datos, la información que está acá (ayer descargamos algunos archivos, agregué dos más)

2.2 Paquetes

Vamos a llamar algunas paqueterías básicas para la práctica de hoy.

```
if (!require("pacman")) install.packages("pacman") # instala pacman si se requiere
```

```
Cargando paquete requerido: pacman
```

```
pacman::p_load(tidyverse,
               readr,
               readxl,
               writexl,
               haven,
               magrittr,
               skimr,
               sjlabelled,
               foreign,
               janitor) #carga los paquetes necesarios para esta práctica
```

2.3 Importación de datos

2.3.1 Desde Excel

El paquete más compatible con RStudio es `{readxl}`. Como su nombre dice "lee" los archivos de excel

```
ejemploxl <- readxl::read_excel("datos/ejemplo_xlsx.xlsx", sheet = "para_importar")
```

```
New names:
* `` -> `...128`
* `` -> `...129`
* `` -> `...132`
* `PIB (Paridad de Poder Adquisitivo)` -> `PIB (Paridad de Poder
  Adquisitivo)...135`
* `PIB (Paridad de Poder Adquisitivo)` -> `PIB (Paridad de Poder
  Adquisitivo)...136`
* `PIB per cápita (Paridad de Poder Adquisitivo)` -> `PIB per cápita (Paridad
  de Poder Adquisitivo)...137`
* `PIB per cápita (Paridad de Poder Adquisitivo)` -> `PIB per cápita (Paridad
  de Poder Adquisitivo)...138`
* `PIB per cápita` -> `PIB per cápita...139`
* `PIB per cápita` -> `PIB per cápita...140`
* `PIB` -> `PIB...141`
* `PIB` -> `PIB...142`
```

Como el nombre de paquete lo indica, sólo lee. Para "escribir" en este formato, recomiendo el paquete `{writexl}`. Lo instalamos anteriormente.

Si quisiéramos exportar un objeto a Excel, se hace de la siguiente forma:

```
writexl::write_xlsx(ejemploxl, path = "Mi_Exportación.xlsx")
```

2.3.2 Desde STATA y SPSS

Si bien también se puede realizar desde el paquete `{foreign}` Pero este no importa algunas características como las etiquetas y tampoco funciona con las versiones más nuevas de STATA. Vamos a instalar otro paquete, compatible con el mundo `{tidyverse}`.

Recuerda que no hay que instalarlo (viene adentro de `{tidyverse}`).

```
concentradohogar <- haven::read_dta("datos/concentradohogar.dta")
```

!Importante, a R no le gustan los objetos con nombres que empiezan en números

El paquete haven sí exporta información.

```
haven::write_dta(concentradohogar,
                 "datos/mi_exportación.dta",
                 version = 12)
```

Con SSPS es muy parecido. Dentro de {haven} hay una función específica para ello.

```
concentradohogarr<- haven::read_sav("datos/concentradohogar.sav")
```

Para escribir

```
haven::write_sav(concentradohogar , "mi_exportacion.sav")
```

Checa que en todas las exportaciones en los nombres hay que incluir la extensión del programa. Si quieres guardar en un lugar diferente al directorio del trabajo, hay que escribir toda la ruta dentro de la computadora.

2.3.3 Desde archivos de texto y de una url

Desde el portal https://datos.gob.mx/ tenemos acceso a directo a varias fuentes de información, al ser datos abiertos, los archivos de texto son muy comunes.

Leeremos parte de esa información, específicamente de las proyecciones de CONAPO

```
mig_inter_quin_proyecciones <- read.csv("https://conapo.segob.gob.mx/work/models/CONAPO/Da
names(mig_inter_quin_proyecciones)
```

```
[1] "RENGLON"     "AÑO"         "ENTIDAD"     "CVE_GEO"     "EDAD"
[6] "SEXO"        "EMIGRANTES"  "INMIGRANTES"
```

```
mig_inter_quin_proyecciones <- readr::read_csv("https://conapo.segob.gob.mx/work/models/CO
```

```
Rows: 23904 Columns: 8
-- Column specification --------------------------------------------------------
Delimiter: ","
```

```
chr (4): AÑO, ENTIDAD, EDAD, SEXO
dbl (4): RENGLON, CVE_GEO, EMIGRANTES, INMIGRANTES

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
names(mig_inter_quin_proyecciones)
```

```
[1] "RENGLON"     "AÑO"         "ENTIDAD"     "CVE_GEO"     "EDAD"
[6] "SEXO"        "EMIGRANTES"  "INMIGRANTES"
```

2.4 Revisión de nuestro conjunto de datos

2.4.1 con base

Vamos a revisar la base, brevemente la base

```
class(concentradohogar) # tipo de objeto
```

```
[1] "tbl_df"     "tbl"        "data.frame"
```

```
names(concentradohogar) # lista las variables
```

```
 [1] "folioviv"   "foliohog"   "ubica_geo"  "tam_loc"    "est_socio"
 [6] "est_dis"    "upm"        "factor"     "clase_hog"  "sexo_jefe"
[11] "edad_jefe"  "educa_jefe" "tot_integ"  "hombres"    "mujeres"
[16] "mayores"    "menores"    "p12_64"     "p65mas"     "ocupados"
[21] "percep_ing" "perc_ocupa" "ing_cor"    "ingtrab"    "trabajo"
[26] "sueldos"    "horas_extr" "comisiones" "aguinaldo"  "indemtrab"
[31] "otra_rem"   "remu_espec" "negocio"    "noagrop"    "industria"
[36] "comercio"   "servicios"  "agrope"     "agricolas"  "pecuarios"
[41] "reproducc"  "pesca"      "otros_trab" "rentas"     "utilidad"
[46] "arrenda"    "transfer"   "jubilacion" "becas"      "donativos"
[51] "remesas"    "bene_gob"   "transf_hog" "trans_inst" "estim_alqu"
[56] "otros_ing"  "gasto_mon"  "alimentos"  "ali_dentro" "cereales"
[61] "carnes"     "pescado"    "leche"      "huevo"      "aceites"
[66] "tuberculo"  "verduras"   "frutas"     "azucar"     "cafe"
```

```
 [71] "especias"   "otros_alim" "bebidas"    "ali_fuera"  "tabaco"
 [76] "vesti_calz" "vestido"    "calzado"    "vivienda"   "alquiler"
 [81] "pred_cons"  "agua"       "energia"    "limpieza"   "cuidados"
 [86] "utensilios" "enseres"    "salud"      "atenc_ambu" "hospital"
 [91] "medicinas"  "transporte" "publico"    "foraneo"    "adqui_vehi"
 [96] "mantenim"   "refaccion"  "combus"     "comunica"   "educa_espa"
[101] "educacion"  "esparci"    "paq_turist" "personales" "cuida_pers"
[106] "acces_pers" "otros_gas"  "transf_gas" "percep_tot" "retiro_inv"
[111] "prestamos"  "otras_perc" "ero_nm_viv" "ero_nm_hog" "erogac_tot"
[116] "cuota_viv"  "mater_serv" "material"   "servicio"   "deposito"
[121] "prest_terc" "pago_tarje" "deudas"     "balance"    "otras_erog"
[126] "smg"
```

```
head(concentradohogar) # muestra las primeras 6 líneas
```

```
# A tibble: 6 x 126
  folioviv   foliohog ubica_geo tam_loc est_socio est_dis upm   factor clase_hog
  <chr>      <chr>    <chr>     <chr>   <chr>     <chr>   <chr>  <dbl> <chr>
1 0100005002 1        01001     1       4         003     0000~    206 3
2 0100005003 1        01001     1       4         003     0000~    206 2
3 0100005004 1        01001     1       4         003     0000~    206 2
4 0100012002 1        01001     1       3         002     0000~    167 3
5 0100012002 2        01001     1       3         002     0000~    167 1
6 0100012004 1        01001     1       3         002     0000~    167 2
# i 117 more variables: sexo_jefe <chr>, edad_jefe <dbl>, educa_jefe <chr>,
#   tot_integ <dbl>, hombres <dbl>, mujeres <dbl>, mayores <dbl>,
#   menores <dbl>, p12_64 <dbl>, p65mas <dbl>, ocupados <dbl>,
#   percep_ing <dbl>, perc_ocupa <dbl>, ing_cor <dbl>, ingtrab <dbl>,
#   trabajo <dbl>, sueldos <dbl>, horas_extr <dbl>, comisiones <dbl>,
#   aguinaldo <dbl>, indemtrab <dbl>, otra_rem <dbl>, remu_espec <dbl>,
#   negocio <dbl>, noagrop <dbl>, industria <dbl>, comercio <dbl>, ...
```

```
table(concentradohogar$sexo_jefe) # un tabulado simple
```

```
    1     2
61905 28197
```

2.4.2 Revisión con `{skimr}`

Esto se puede tardar un poquito

```
skimr::skim(concentradohogar)
```

Table 2.1: Data summary

Name	concentradohogar
Number of rows	90102
Number of columns	126
_____	
Column type frequency:	
character	10
numeric	116
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
folioviv	0	1	10	10	0	88823	0
foliohog	0	1	1	1	0	5	0
ubica_geo	0	1	5	5	0	1132	0
tam_loc	0	1	1	1	0	4	0
est_socio	0	1	1	1	0	4	0
est_dis	0	1	3	3	0	560	0
upm	0	1	7	7	0	10211	0
clase_hog	0	1	1	1	0	5	0
sexo_jefe	0	1	1	1	0	2	0
educa_jefe	0	1	2	2	0	11	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
factor	0	1	416.86	419.41	6.0	156.00	283.00	530.00	6470.00	
edad_jefe	0	1	51.23	15.91	13.0	39.00	50.00	63.00	109.00	
tot_integ	0	1	3.44	1.78	1.0	2.00	3.00	4.00	19.00	
hombres	0	1	1.66	1.12	0.0	1.00	1.00	2.00	12.00	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
mujeres	0	1	1.78	1.17	0.0	1.00	2.00	2.00	11.00	
mayores	0	1	2.78	1.36	1.0	2.00	3.00	4.00	15.00	
menores	0	1	0.65	0.95	0.0	0.00	0.00	1.00	9.00	
p12_64	0	1	2.45	1.49	0.0	1.00	2.00	3.00	15.00	
p65mas	0	1	0.33	0.62	0.0	0.00	0.00	1.00	4.00	
ocupados	0	1	1.66	1.09	0.0	1.00	2.00	2.00	11.00	
percep_ing	0	1	2.26	1.18	0.0	1.00	2.00	3.00	16.00	
perc_ocupa	0	1	1.61	1.06	0.0	1.00	1.00	2.00	12.00	
ing_cor	0	1	61489.96	78324.84	0.0	28385.69	46073.68	74343.65	7153770.46	
ingtrab	0	1	40359.54	52327.25	0.0	12170.67	29899.10	54327.95	5891913.57	
trabajo	0	1	33514.63	44828.95	0.0	49.18	23867.40	47445.65	3771994.00	
sueldos	0	1	28866.79	37377.13	0.0	0.00	21130.43	41086.95	2655737.70	
horas_extr	0	1	292.47	1995.00	0.0	0.00	0.00	0.00	110543.47	
comisiones	0	1	879.37	6484.49	0.0	0.00	0.00	0.00	690491.80	
aguinaldo	0	1	1489.57	3781.67	0.0	0.00	0.00	1475.40	293478.25	
indemtrab	0	1	129.04	3639.71	0.0	0.00	0.00	0.00	449999.99	
otra_rem	0	1	654.77	3389.42	0.0	0.00	0.00	0.00	225245.89	
remu_espec	0	1	1202.61	11869.30	0.0	0.00	0.00	0.00	3015428.52	
negocio	0	1	5525.91	28303.90	0.0	0.00	0.00	1549.18	5712891.84	
noagrop	0	1	4374.96	17327.48	0.0	0.00	0.00	0.00	1114679.34	
industria	0	1	997.87	8092.96	0.0	0.00	0.00	0.00	1114679.34	
comercio	0	1	1466.54	11309.68	0.0	0.00	0.00	0.00	889369.06	
servicios	0	1	1910.54	10479.11	0.0	0.00	0.00	0.00	889524.59	
agrope	0	1	1150.96	22407.72	0.0	0.00	0.00	0.00	5712891.84	
agricolas	0	1	584.18	9493.09	0.0	0.00	0.00	0.00	1473748.04	
pecuarios	0	1	489.26	19957.84	0.0	0.00	0.00	0.00	5712891.84	
reproducc	0	1	29.05	1651.04	0.0	0.00	0.00	0.00	424032.78	
pesca	0	1	48.46	1442.87	0.0	0.00	0.00	0.00	154190.54	
otros_trab	0	1	1319.00	7439.20	0.0	0.00	0.00	0.00	1257295.08	
rentas	0	1	3585.66	51810.69	0.0	0.00	0.00	0.00	6854754.09	
utilidad	0	1	2922.56	50880.21	0.0	0.00	0.00	0.00	6854754.09	
arrenda	0	1	663.09	7270.20	0.0	0.00	0.00	0.00	733695.65	
transfer	0	1	10901.15	21915.52	0.0	0.00	3692.92	12913.03	1001867.40	
jubilacion	0	1	4722.85	17245.75	0.0	0.00	0.00	0.00	688524.59	
becas	0	1	105.17	1548.61	0.0	0.00	0.00	0.00	153831.51	
donativos	0	1	1305.04	5071.83	0.0	0.00	0.00	0.00	413114.75	
remesas	0	1	848.70	5791.98	0.0	0.00	0.00	0.00	238524.59	
bene_gob	0	1	1849.19	3366.51	0.0	0.00	0.00	2611.22	63195.64	
transf_hog	0	1	1512.32	4697.95	0.0	0.00	0.00	771.42	465000.00	
trans_inst	0	1	557.90	7606.45	0.0	0.00	0.00	0.00	978260.86	
estim_alqu	0	1	6592.66	8338.00	0.0	2903.22	4500.00	8709.67	348387.09	

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
otros_ing	0	1	50.95	973.27	0.0	0.00	0.00	0.00	73369.56	
gasto_mon	0	1	37615.18	34649.23	0.0	18561.37	29678.83	45901.43	1703575.17	
alimentos	0	1	14046.16	11010.37	0.0	7482.73	11957.05	17961.27	849839.62	
ali_dentro	0	1	11533.55	7834.39	0.0	6197.01	10144.16	15158.38	302942.31	
cereales	0	1	2052.96	1570.90	0.0	977.12	1761.40	2751.39	32631.39	
carnes	0	1	2493.10	2847.37	0.0	359.99	1774.27	3689.96	157448.49	
pescado	0	1	248.92	857.41	0.0	0.00	0.00	0.00	60685.70	
leche	0	1	1011.34	1277.08	0.0	0.00	642.85	1427.11	32014.25	
huevo	0	1	515.33	595.19	0.0	0.00	385.71	848.56	9218.54	
aceites	0	1	214.21	446.77	0.0	0.00	0.00	347.14	13395.84	
tuberculo	0	1	198.87	337.12	0.0	0.00	0.00	321.42	12857.14	
verduras	0	1	1241.79	1234.53	0.0	321.42	977.12	1810.20	54784.26	
frutas	0	1	469.28	841.42	0.0	0.00	0.00	642.85	38571.42	
azucar	0	1	109.78	293.17	0.0	0.00	0.00	0.00	25714.28	
cafe	0	1	120.40	377.98	0.0	0.00	0.00	0.00	12471.42	
especias	0	1	108.18	284.00	0.0	0.00	0.00	64.28	13139.99	
otros_alim	0	1	1651.95	2858.15	0.0	0.00	321.42	2314.28	55028.54	
bebidas	0	1	1097.43	1570.63	0.0	218.57	719.99	1480.50	128237.11	
ali_fuera	0	1	2445.31	6907.11	0.0	0.00	0.00	2571.42	802542.78	
tabaco	0	1	67.31	449.34	0.0	0.00	0.00	0.00	19234.27	
vesti_calz	0	1	1470.03	2611.47	0.0	0.00	684.77	1858.68	131191.33	
vestido	0	1	906.82	1863.60	0.0	0.00	273.91	1105.42	96050.32	
calzado	0	1	563.21	1092.30	0.0	0.00	107.60	733.69	47739.11	
vivienda	0	1	3361.12	5429.41	0.0	1020.00	2160.00	3862.21	604814.51	
alquiler	0	1	1031.01	4667.52	0.0	0.00	0.00	0.00	581951.61	
pred_cons	0	1	180.63	1100.52	0.0	0.00	0.00	87.50	116400.00	
agua	0	1	407.19	776.34	0.0	0.00	240.00	540.00	78000.00	
energia	0	1	1742.29	1743.06	0.0	553.69	1400.80	2400.00	113080.64	
limpieza	0	1	2345.68	3932.56	0.0	771.00	1379.01	2481.43	302696.82	
cuidados	0	1	1745.97	2833.03	0.0	678.00	1158.36	1896.00	128197.70	
utensilios	0	1	218.77	1378.17	0.0	0.00	0.00	0.00	294740.20	
enseres	0	1	380.94	1649.55	0.0	0.00	0.00	0.00	90098.34	
salud	0	1	1270.55	5662.22	0.0	0.00	146.73	841.29	324547.74	
atenc_ambu	0	1	903.67	4754.37	0.0	0.00	0.00	538.03	283695.64	
hospital	0	1	152.99	2368.65	0.0	0.00	0.00	0.00	241434.73	
medicinas	0	1	213.90	1294.84	0.0	0.00	0.00	97.82	127464.42	
transporte	0	1	7538.59	12471.10	0.0	1975.75	4803.31	9299.92	491168.39	
publico	0	1	1365.08	2596.60	0.0	0.00	0.00	1851.41	137031.32	
foraneo	0	1	205.99	1200.95	0.0	0.00	0.00	0.00	88011.04	
adqui_vehi	0	1	1045.70	9662.96	0.0	0.00	0.00	0.00	489130.43	
mantenim	0	1	3122.62	4882.43	0.0	0.00	929.03	4770.49	113684.95	

skim_variable	missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
refaccion	0	1	275.12	1029.55	0.0	0.00	0.00	0.00	39130.43	
combus	0	1	2847.50	4483.70	0.0	0.00	870.96	4354.83	113225.80	
comunica	0	1	1799.20	2322.07	0.0	580.64	1380.00	2409.98	105366.77	
educa_espa	0	1	3506.12	8866.90	0.0	0.00	753.00	3641.60	475717.08	
educacion	0	1	2466.90	7454.48	0.0	0.00	0.00	2177.41	451161.26	
esparci	0	1	718.84	2310.87	0.0	0.00	0.00	747.00	96406.00	
paq_turist	0	1	320.39	2610.74	0.0	0.00	0.00	0.00	166304.34	
personales	0	1	2953.55	4402.36	0.0	998.69	1884.00	3452.78	296006.96	
cuida_pers	0	1	2239.89	2335.69	0.0	888.34	1628.67	2824.79	69462.00	
acces_pers	0	1	94.06	433.90	0.0	0.00	0.00	0.00	23996.72	
otros_gas	0	1	619.60	3203.71	0.0	0.00	0.00	48.91	295256.96	
transf_gas	0	1	1123.38	5306.60	0.0	0.00	0.00	225.08	737704.91	
percep_tot	0	1	3940.89	40469.54	0.0	0.00	0.00	2842.18	8171344.09	
retiro_inv	0	1	993.91	14080.54	0.0	0.00	0.00	0.00	3480662.98	
prestamos	0	1	407.08	4801.37	0.0	0.00	0.00	0.00	540983.60	
otras_perc	0	1	877.86	24750.31	0.0	0.00	0.00	0.00	4402173.91	
ero_nm_viv	0	1	40.11	2162.50	0.0	0.00	0.00	0.00	440217.38	
ero_nm_hog	0	1	1621.92	27703.08	0.0	0.00	0.00	1349.97	8171344.09	
erogac_tot	0	1	7659.37	43226.39	0.0	0.00	489.13	6208.52	5733149.15	
cuota_viv	0	1	712.97	3500.36	0.0	0.00	0.00	0.00	150000.00	
mater_serv	0	1	441.57	4152.57	0.0	0.00	0.00	0.00	391304.34	
material	0	1	267.11	2604.68	0.0	0.00	0.00	0.00	244565.21	
servicio	0	1	174.46	2301.69	0.0	0.00	0.00	0.00	195652.17	
deposito	0	1	4006.25	31415.73	0.0	0.00	0.00	1320.65	4176795.57	
prest_terc	0	1	125.48	4000.35	0.0	0.00	0.00	0.00	1100543.47	
pago_tarje	0	1	990.75	9069.74	0.0	0.00	0.00	0.00	1640883.97	
deudas	0	1	430.28	2720.39	0.0	0.00	0.00	0.00	122282.60	
balance	0	1	420.11	4608.27	0.0	0.00	0.00	0.00	496467.39	
otras_erog	0	1	531.96	14050.93	0.0	0.00	0.00	0.00	2934782.60	
smg	0	1	15558.30	0.00	15558.3	15558.30	15558.30	15558.30	15558.30	

2.5 Un poquito de `{dplyr}` y limpieza

2.5.1 Primero, los pipes

R utiliza dos pipes el nativo `|>` y el pipe que está en `{dplyr}` `%>%`. Algunas de las diferencias las puedes checar acá https://eliocamp.github.io/codigo-r/2021/05/r-pipa-nativa/

Aquí hay un *tuit*, o *post de x.com* que lo explica bien.

https://x.com/ArthurWelle/status/1535429654760284161

En estas prácticas utilizaremos el segundo, son muy parecidos y así esta instructora pueda reciclar algunos de sus códigos viejos. Pero funcionan igual:

```
concentradohogar|> #pipe nativo, no necesita instalación
  head()
```

```
# A tibble: 6 x 126
  folioviv   foliohog ubica_geo tam_loc est_socio est_dis upm   factor clase_hog
  <chr>      <chr>    <chr>     <chr>   <chr>     <chr>   <chr>  <dbl> <chr>
1 0100005002 1        01001     1       4         003     0000~    206 3
2 0100005003 1        01001     1       4         003     0000~    206 2
3 0100005004 1        01001     1       4         003     0000~    206 2
4 0100012002 1        01001     1       3         002     0000~    167 3
5 0100012002 2        01001     1       3         002     0000~    167 1
6 0100012004 1        01001     1       3         002     0000~    167 2
# i 117 more variables: sexo_jefe <chr>, edad_jefe <dbl>, educa_jefe <chr>,
#   tot_integ <dbl>, hombres <dbl>, mujeres <dbl>, mayores <dbl>,
#   menores <dbl>, p12_64 <dbl>, p65mas <dbl>, ocupados <dbl>,
#   percep_ing <dbl>, perc_ocupa <dbl>, ing_cor <dbl>, ingtrab <dbl>,
#   trabajo <dbl>, sueldos <dbl>, horas_extr <dbl>, comisiones <dbl>,
#   aguinaldo <dbl>, indemtrab <dbl>, otra_rem <dbl>, remu_espec <dbl>,
#   negocio <dbl>, noagrop <dbl>, industria <dbl>, comercio <dbl>, ...
```

```
concentradohogar %>%  #pipe de dplyr, necesita instalación de dplyr en tidyverse
  head()
```

```
# A tibble: 6 x 126
  folioviv   foliohog ubica_geo tam_loc est_socio est_dis upm   factor clase_hog
  <chr>      <chr>    <chr>     <chr>   <chr>     <chr>   <chr>  <dbl> <chr>
1 0100005002 1        01001     1       4         003     0000~    206 3
2 0100005003 1        01001     1       4         003     0000~    206 2
3 0100005004 1        01001     1       4         003     0000~    206 2
4 0100012002 1        01001     1       3         002     0000~    167 3
5 0100012002 2        01001     1       3         002     0000~    167 1
6 0100012004 1        01001     1       3         002     0000~    167 2
# i 117 more variables: sexo_jefe <chr>, edad_jefe <dbl>, educa_jefe <chr>,
#   tot_integ <dbl>, hombres <dbl>, mujeres <dbl>, mayores <dbl>,
#   menores <dbl>, p12_64 <dbl>, p65mas <dbl>, ocupados <dbl>,
#   percep_ing <dbl>, perc_ocupa <dbl>, ing_cor <dbl>, ingtrab <dbl>,
```

```
#   trabajo <dbl>, sueldos <dbl>, horas_extr <dbl>, comisiones <dbl>,
#   aguinaldo <dbl>, indemtrab <dbl>, otra_rem <dbl>, remu_espec <dbl>,
#   negocio <dbl>, noagrop <dbl>, industria <dbl>, comercio <dbl>, ...
```

2.5.2 Limpieza de nombres con {janitor}

Este paso también nos permitirá enseñar otro *pipe* que está en el paquete {magrittr}.

Los nombres de una base de datos son los nombres de las columnas.

```
names(concentradohogar)
```

```
  [1] "folioviv"   "foliohog"   "ubica_geo"  "tam_loc"    "est_socio"
  [6] "est_dis"    "upm"        "factor"     "clase_hog"  "sexo_jefe"
 [11] "edad_jefe"  "educa_jefe" "tot_integ"  "hombres"    "mujeres"
 [16] "mayores"    "menores"    "p12_64"     "p65mas"     "ocupados"
 [21] "percep_ing" "perc_ocupa" "ing_cor"    "ingtrab"    "trabajo"
 [26] "sueldos"    "horas_extr" "comisiones" "aguinaldo"  "indemtrab"
 [31] "otra_rem"   "remu_espec" "negocio"    "noagrop"    "industria"
 [36] "comercio"   "servicios"  "agrope"     "agricolas"  "pecuarios"
 [41] "reproducc"  "pesca"      "otros_trab" "rentas"     "utilidad"
 [46] "arrenda"    "transfer"   "jubilacion" "becas"      "donativos"
 [51] "remesas"    "bene_gob"   "transf_hog" "trans_inst" "estim_alqu"
 [56] "otros_ing"  "gasto_mon"  "alimentos"  "ali_dentro" "cereales"
 [61] "carnes"     "pescado"    "leche"      "huevo"      "aceites"
 [66] "tuberculo"  "verduras"   "frutas"     "azucar"     "cafe"
 [71] "especias"   "otros_alim" "bebidas"    "ali_fuera"  "tabaco"
 [76] "vesti_calz" "vestido"    "calzado"    "vivienda"   "alquiler"
 [81] "pred_cons"  "agua"       "energia"    "limpieza"   "cuidados"
 [86] "utensilios" "enseres"    "salud"      "atenc_ambu" "hospital"
 [91] "medicinas"  "transporte" "publico"    "foraneo"    "adqui_vehi"
 [96] "mantenim"   "refaccion"  "combus"     "comunica"   "educa_espa"
[101] "educacion"  "esparci"    "paq_turist" "personales" "cuida_pers"
[106] "acces_pers" "otros_gas"  "transf_gas" "percep_tot" "retiro_inv"
[111] "prestamos"  "otras_perc" "ero_nm_viv" "ero_nm_hog" "erogac_tot"
[116] "cuota_viv"  "mater_serv" "material"   "servicio"   "deposito"
[121] "prest_terc" "pago_tarje" "deudas"     "balance"    "otras_erog"
[126] "smg"
```

```
names(ejemploxl)
```

```
 [1] "Indicador"
 [2] "Protección de derechos humanos"
 [3] "Homicidios dolosos"
 [4] "Costos de la delincuencia en los negocios"
 [5] "Confianza en la policía"
 [6] "Imparcialidad de las cortes"
 [7] "Independencia del poder judicial"
 [8] "Protección de derechos de propiedad"
 [9] "Piratería Informática"
[10] "Protección a acreedores"
[11] "Tiempo para resolver quiebras"
[12] "Cumplimiento de contratos"
[13] "Índice de Estados Frágiles"
[14] "Índice de Estado de Derecho"
[15] "Contaminación del aire"
[16] "Emisiones de CO2"
[17] "Recursos hídricos renovables"
[18] "Estrés hídrico"
[19] "Áreas naturales protegidas"
[20] "Superficie forestal perdida"
[21] "Uso de fertilizantes en la agricultura"
[22] "Uso de pesticidas"
[23] "Fuentes de energía no contaminantes"
[24] "Empresas certificadas como limpias"
[25] "Índice de vulnerabilidad a efectos del cambio climático"
[26] "Índice de Gini"
[27] "Índice Global de Brecha de Género"
[28] "Mujeres en la PEA"
[29] "Dependientes de la PEA"
[30] "Acceso a agua"
[31] "Acceso a alcantarillado"
[32] "Analfabetismo"
[33] "Cobertura en nivel preescolar"
[34] "Escolaridad promedio"
[35] "Calidad educativa"
[36] "Nivel de inglés"
[37] "Esperanza de vida"
[38] "Mortalidad infantil"
[39] "Cobertura de vacunación"
[40] "Embarazos adolescentes"
[41] "Impactos en salud por sobrepeso y obesidad"
[42] "Prevalencia de diabetes"
[43] "Suicidios"
```

```
[44] "Médicos y médicas"
[45] "Gasto en salud per cápita"
[46] "Gasto en salud por cuenta propia"
[47] "Estabilidad política y ausencia de violencia"
[48] "Interferencia militar en el Estado de derecho o en el proceso político"
[49] "Derechos políticos"
[50] "Libertades civiles"
[51] "Libertad de prensa"
[52] "Índice de Percepción de Corrupción"
[53] "Disponibilidad de información pública"
[54] "Participación electoral"
[55] "Índice de efectividad del gobierno"
[56] "Miembro de la Alianza para el Gobierno Abierto"
[57] "Índice de desarrollo de Gobierno Electrónico"
[58] "Economía informal"
[59] "Facilidad para abrir una empresa"
[60] "Tiempo de altos ejecutivos a temas burocráticos"
[61] "Tiempo para preparar y pagar impuestos"
[62] "Presupuesto balanceado"
[63] "Deuda total del gobierno central"
[64] "Ingresos fiscales"
[65] "Impuesto sobre el ingreso"
[66] "Carga impositiva"
[67] "Edad efectiva de retiro"
[68] "Flexibilidad de las leyes laborales"
[69] "Productividad media del trabajo"
[70] "Valor agregado de la industria"
[71] "Valor agregado de la agricultura"
[72] "Eficiencia energética"
[73] "Cambio en inventarios"
[74] "Índice de transparencia y regulación de la propiedad privada"
[75] "Crecimiento del PIB"
[76] "Crecimiento promedio del PIB"
[77] "Variabilidad del crecimiento del PIB"
[78] "Inflación"
[79] "Inflación promedio"
[80] "Variabilidad de la inflación"
[81] "Desempleo"
[82] "Deuda externa"
[83] "Calificación de deuda"
[84] "Activos del sector financiero"
[85] "Activos de los depositantes"
[86] "Reservas"
```

```
 [87] "Índice de riesgos de seguridad energética"
 [88] "Pérdidas de electricidad"
 [89] "Líneas móviles"
 [90] "Usuarios de internet"
 [91] "Servidores de internet seguros"
 [92] "Transporte intraurbano de alta capacidad"
 [93] "Índice calidad de carreteras"
 [94] "Flujo de pasajeros aéreos"
 [95] "Índice de desempeño logístico (transporte)"
 [96] "Índice de infraestructura portuaria"
 [97] "Tráfico portuario de contenedores"
 [98] "Penetración del sistema financiero privado"
 [99] "Capitalización del mercado de valores"
[100] "Cambio en empresas listadas"
[101] "Rotación de activos bursátiles"
[102] "Índice de competencia de Boone"
[103] "Organizaciones internacionales a las que pertenece"
[104] "Acuerdos comerciales"
[105] "Socios comerciales efectivos"
[106] "Apertura comercial"
[107] "Diversificación de las exportaciones"
[108] "Diversificación de las importaciones"
[109] "Barreras ocultas a la importación"
[110] "Aranceles agrícolas"
[111] "Aranceles manufactureros"
[112] "Inversión extranjera directa (neta)"
[113] "Inversión Extranjera Directa neta promedio"
[114] "Variabilidad de la IED"
[115] "Ingresos por turismo"
[116] "Flujo de pasajeros aéreos internacionales"
[117] "Gasto militar"
[118] "Gasto en investigación y desarrollo"
[119] "Coeficiente de invención"
[120] "Artículos científicos y técnicos"
[121] "Exportaciones de alta tecnología"
[122] "Índice de Complejidad Económica"
[123] "Crecimiento de la productividad total de los factores"
[124] "Empresas en Fortune 500"
[125] "Empresas ISO 9001"
[126] "Población en grandes ciudades"
[127] "PIB en servicios"
[128] "...128"
[129] "...129"
```

```
[130] "Inversión (Formación bruta de capital fijo)"
[131] "Talento"
[132] "...132"
[133] "Población total"
[134] "Densidad de población"
[135] "PIB (Paridad de Poder Adquisitivo)...135"
[136] "PIB (Paridad de Poder Adquisitivo)...136"
[137] "PIB per cápita (Paridad de Poder Adquisitivo)...137"
[138] "PIB per cápita (Paridad de Poder Adquisitivo)...138"
[139] "PIB per cápita...139"
[140] "PIB per cápita...140"
[141] "PIB...141"
[142] "PIB...142"
```

Como vemos en las bases hay mayúsculas, caracteres especiales y demás. Esto lo podemos cambiar

```
ejemploxl<-ejemploxl %>%
  janitor::clean_names()

names(ejemploxl)
```

```
 [1] "indicador"
 [2] "proteccion_de_derechos_humanos"
 [3] "homicidios_dolosos"
 [4] "costos_de_la_delincuencia_en_los_negocios"
 [5] "confianza_en_la_policia"
 [6] "imparcialidad_de_las_cortes"
 [7] "independencia_del_poder_judicial"
 [8] "proteccion_de_derechos_de_propiedad"
 [9] "pirateria_informatica"
[10] "proteccion_a_acreedores"
[11] "tiempo_para_resolver_quiebras"
[12] "cumplimiento_de_contratos"
[13] "indice_de_estados_fragiles"
[14] "indice_de_estado_de_derecho"
[15] "contaminacion_del_aire"
[16] "emisiones_de_co2"
[17] "recursos_hidricos_renovables"
[18] "estres_hidrico"
[19] "areas_naturales_protegidas"
[20] "superficie_forestal_perdida"
```

```
[21] "uso_de_fertilizantes_en_la_agricultura"
[22] "uso_de_pesticidas"
[23] "fuentes_de_energia_no_contaminantes"
[24] "empresas_certificadas_como_limpias"
[25] "indice_de_vulnerabilidad_a_efectos_del_cambio_climatico"
[26] "indice_de_gini"
[27] "indice_global_de_brecha_de_genero"
[28] "mujeres_en_la_pea"
[29] "dependientes_de_la_pea"
[30] "acceso_a_agua"
[31] "acceso_a_alcantarillado"
[32] "analfabetismo"
[33] "cobertura_en_nivel_preescolar"
[34] "escolaridad_promedio"
[35] "calidad_educativa"
[36] "nivel_de_ingles"
[37] "esperanza_de_vida"
[38] "mortalidad_infantil"
[39] "cobertura_de_vacunacion"
[40] "embarazos_adolescentes"
[41] "impactos_en_salud_por_sobrepeso_y_obesidad"
[42] "prevalencia_de_diabetes"
[43] "suicidios"
[44] "medicos_y_medicas"
[45] "gasto_en_salud_per_capita"
[46] "gasto_en_salud_por_cuenta_propia"
[47] "estabilidad_politica_y_ausencia_de_violencia"
[48] "interferencia_militar_en_el_estado_de_derecho_o_en_el_proceso_politico"
[49] "derechos_politicos"
[50] "libertades_civiles"
[51] "libertad_de_prensa"
[52] "indice_de_percepcion_de_corrupcion"
[53] "disponibilidad_de_informacion_publica"
[54] "participacion_electoral"
[55] "indice_de_efectividad_del_gobierno"
[56] "miembro_de_la_alianza_para_el_gobierno_abierto"
[57] "indice_de_desarrollo_de_gobierno_electronico"
[58] "economia_informal"
[59] "facilidad_para_abrir_una_empresa"
[60] "tiempo_de_altos_ejecutivos_a_temas_burocraticos"
[61] "tiempo_para_preparar_y_pagar_impuestos"
[62] "presupuesto_balanceado"
[63] "deuda_total_del_gobierno_central"
```

```
 [64] "ingresos_fiscales"
 [65] "impuesto_sobre_el_ingreso"
 [66] "carga_impositiva"
 [67] "edad_efectiva_de_retiro"
 [68] "flexibilidad_de_las_leyes_laborales"
 [69] "productividad_media_del_trabajo"
 [70] "valor_agregado_de_la_industria"
 [71] "valor_agregado_de_la_agricultura"
 [72] "eficiencia_energetica"
 [73] "cambio_en_inventarios"
 [74] "indice_de_transparencia_y_regulacion_de_la_propiedad_privada"
 [75] "crecimiento_del_pib"
 [76] "crecimiento_promedio_del_pib"
 [77] "variabilidad_del_crecimiento_del_pib"
 [78] "inflacion"
 [79] "inflacion_promedio"
 [80] "variabilidad_de_la_inflacion"
 [81] "desempleo"
 [82] "deuda_externa"
 [83] "calificacion_de_deuda"
 [84] "activos_del_sector_financiero"
 [85] "activos_de_los_depositantes"
 [86] "reservas"
 [87] "indice_de_riesgos_de_seguridad_energetica"
 [88] "perdidas_de_electricidad"
 [89] "lineas_moviles"
 [90] "usuarios_de_internet"
 [91] "servidores_de_internet_seguros"
 [92] "transporte_intraurbano_de_alta_capacidad"
 [93] "indice_calidad_de_carreteras"
 [94] "flujo_de_pasajeros_aereos"
 [95] "indice_de_desempeno_logistico_transporte"
 [96] "indice_de_infraestructura_portuaria"
 [97] "trafico_portuario_de_contenedores"
 [98] "penetracion_del_sistema_financiero_privado"
 [99] "capitalizacion_del_mercado_de_valores"
[100] "cambio_en_empresas_listadas"
[101] "rotacion_de_activos_bursatiles"
[102] "indice_de_competencia_de_boone"
[103] "organizaciones_internacionales_a_las_que_pertenece"
[104] "acuerdos_comerciales"
[105] "socios_comerciales_efectivos"
[106] "apertura_comercial"
```

```
[107] "diversificacion_de_las_exportaciones"
[108] "diversificacion_de_las_importaciones"
[109] "barreras_ocultas_a_la_importacion"
[110] "aranceles_agricolas"
[111] "aranceles_manufactureros"
[112] "inversion_extranjera_directa_neta"
[113] "inversion_extranjera_directa_neta_promedio"
[114] "variabilidad_de_la_ied"
[115] "ingresos_por_turismo"
[116] "flujo_de_pasajeros_aereos_internacionales"
[117] "gasto_militar"
[118] "gasto_en_investigacion_y_desarrollo"
[119] "coeficiente_de_invencion"
[120] "articulos_cientificos_y_tecnicos"
[121] "exportaciones_de_alta_tecnologia"
[122] "indice_de_complejidad_economica"
[123] "crecimiento_de_la_productividad_total_de_los_factores"
[124] "empresas_en_fortune_500"
[125] "empresas_iso_9001"
[126] "poblacion_en_grandes_ciudades"
[127] "pib_en_servicios"
[128] "x128"
[129] "x129"
[130] "inversion_formacion_bruta_de_capital_fijo"
[131] "talento"
[132] "x132"
[133] "poblacion_total"
[134] "densidad_de_poblacion"
[135] "pib_paridad_de_poder_adquisitivo_135"
[136] "pib_paridad_de_poder_adquisitivo_136"
[137] "pib_per_capita_paridad_de_poder_adquisitivo_137"
[138] "pib_per_capita_paridad_de_poder_adquisitivo_138"
[139] "pib_per_capita_139"
[140] "pib_per_capita_140"
[141] "pib_141"
[142] "pib_142"
```

Si quisiéramos que la acción quedará en una sola operación, podemos usar un pipe diferente:

```
concentradohogar %<>%
  clean_names()
```

```
names(concentradohogar)
```

```
  [1] "folioviv"   "foliohog"   "ubica_geo"  "tam_loc"    "est_socio"
  [6] "est_dis"    "upm"        "factor"     "clase_hog"  "sexo_jefe"
 [11] "edad_jefe"  "educa_jefe" "tot_integ"  "hombres"    "mujeres"
 [16] "mayores"    "menores"    "p12_64"     "p65mas"     "ocupados"
 [21] "percep_ing" "perc_ocupa" "ing_cor"    "ingtrab"    "trabajo"
 [26] "sueldos"    "horas_extr" "comisiones" "aguinaldo"  "indemtrab"
 [31] "otra_rem"   "remu_espec" "negocio"    "noagrop"    "industria"
 [36] "comercio"   "servicios"  "agrope"     "agricolas"  "pecuarios"
 [41] "reproducc"  "pesca"      "otros_trab" "rentas"     "utilidad"
 [46] "arrenda"    "transfer"   "jubilacion" "becas"      "donativos"
 [51] "remesas"    "bene_gob"   "transf_hog" "trans_inst" "estim_alqu"
 [56] "otros_ing"  "gasto_mon"  "alimentos"  "ali_dentro" "cereales"
 [61] "carnes"     "pescado"    "leche"      "huevo"      "aceites"
 [66] "tuberculo"  "verduras"   "frutas"     "azucar"     "cafe"
 [71] "especias"   "otros_alim" "bebidas"    "ali_fuera"  "tabaco"
 [76] "vesti_calz" "vestido"    "calzado"    "vivienda"   "alquiler"
 [81] "pred_cons"  "agua"       "energia"    "limpieza"   "cuidados"
 [86] "utensilios" "enseres"    "salud"      "atenc_ambu" "hospital"
 [91] "medicinas"  "transporte" "publico"    "foraneo"    "adqui_vehi"
 [96] "mantenim"   "refaccion"  "combus"     "comunica"   "educa_espa"
[101] "educacion"  "esparci"    "paq_turist" "personales" "cuida_pers"
[106] "acces_pers" "otros_gas"  "transf_gas" "percep_tot" "retiro_inv"
[111] "prestamos"  "otras_perc" "ero_nm_viv" "ero_nm_hog" "erogac_tot"
[116] "cuota_viv"  "mater_serv" "material"   "servicio"   "deposito"
[121] "prest_terc" "pago_tarje" "deudas"     "balance"    "otras_erog"
[126] "smg"
```

2.5.3 Ojeando

```
dplyr::glimpse(concentradohogar)
```

```
Rows: 90,102
Columns: 126
$ folioviv   <chr> "0100005002", "0100005003", "0100005004", "0100012002", "01~
$ foliohog   <chr> "1", "1", "1", "1", "2", "1", "1", "1", "1", "1", "1", "1",~
$ ubica_geo  <chr> "01001", "01001", "01001", "01001", "01001", "01001", "0100~
$ tam_loc    <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1",~
```

```
$ est_socio  <chr> "4", "4", "4", "3", "3", "3", "3", "4", "4", "4", "4", "3",~
$ est_dis    <chr> "003", "003", "003", "002", "002", "002", "002", "003", "00~
$ upm        <chr> "0000001", "0000001", "0000001", "0000002", "0000002", "000~
$ factor     <dbl> 206, 206, 206, 167, 167, 167, 167, 212, 212, 212, 212, 184,~
$ clase_hog  <chr> "3", "2", "2", "3", "1", "2", "2", "1", "2", "2", "2", "1",~
$ sexo_jefe  <chr> "2", "1", "1", "1", "1", "1", "2", "2", "2", "1", "1", "1",~
$ edad_jefe  <dbl> 91, 68, 56, 87, 27, 57, 47, 75, 70, 69, 48, 73, 64, 55, 58,~
$ educa_jefe <chr> "03", "08", "10", "11", "08", "08", "10", "06", "10", "04",~
$ tot_integ  <dbl> 3, 2, 3, 4, 1, 4, 4, 1, 3, 2, 5, 1, 4, 3, 1, 6, 4, 2, 3, 1,~
$ hombres    <dbl> 0, 1, 2, 2, 1, 2, 2, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 3, 0,~
$ mujeres    <dbl> 3, 1, 1, 2, 0, 2, 2, 1, 2, 1, 4, 0, 3, 2, 1, 5, 3, 1, 0, 1,~
$ mayores    <dbl> 3, 2, 3, 4, 1, 3, 4, 1, 3, 2, 5, 1, 4, 2, 1, 4, 4, 2, 1, 1,~
$ menores    <dbl> 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 2, 0, 0, 2, 0,~
$ p12_64     <dbl> 2, 1, 3, 2, 1, 3, 4, 0, 2, 0, 5, 0, 4, 2, 1, 4, 3, 0, 1, 1,~
$ p65mas     <dbl> 1, 1, 0, 2, 0, 0, 0, 1, 1, 2, 0, 1, 0, 0, 0, 0, 1, 2, 0, 0,~
$ ocupados   <dbl> 1, 2, 2, 0, 1, 3, 1, 0, 3, 1, 1, 0, 1, 1, 0, 3, 1, 1, 1, 1,~
$ percep_ing <dbl> 3, 2, 2, 2, 1, 4, 2, 1, 3, 2, 1, 1, 2, 2, 1, 3, 2, 2, 1, 1,~
$ perc_ocupa <dbl> 1, 2, 2, 0, 1, 3, 1, 0, 3, 1, 1, 0, 1, 1, 0, 3, 1, 1, 1, 1,~
$ ing_cor    <dbl> 56123.75, 108048.87, 133852.88, 105054.15, 24211.95, 121649~
$ ingtrab    <dbl> 35706.51, 66766.28, 93081.50, 0.00, 22255.43, 40255.41, 333~
$ trabajo    <dbl> 35706.51, 66766.28, 51603.24, 0.00, 17364.13, 40255.41, 327~
$ sueldos    <dbl> 33749.99, 61630.42, 41086.95, 0.00, 17364.13, 36586.94, 246~
$ horas_extr <dbl> 0.00, 0.00, 978.26, 0.00, 0.00, 0.00, 7092.39, 0.00, 0.00, ~
$ comisiones <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ aguinaldo  <dbl> 1956.52, 4646.73, 5135.86, 0.00, 0.00, 3668.47, 1027.17, 0.~
$ indemtrab  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ otra_rem   <dbl> 0.00, 489.13, 4402.17, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, ~
$ remu_espec <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ negocio    <dbl> 0.00, 0.00, 41478.26, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0~
$ noagrop    <dbl> 0.00, 0.00, 41478.26, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0~
$ industria  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ comercio   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ servicios  <dbl> 0.00, 0.00, 41478.26, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0~
$ agrope     <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ agricolas  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ pecuarios  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ reproducc  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ pesca      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ otros_trab <dbl> 0.00, 0.00, 0.00, 0.00, 4891.30, 0.00, 586.95, 0.00, 0.00, ~
$ rentas     <dbl> 0.00, 32282.60, 11739.13, 0.00, 0.00, 72684.78, 0.00, 0.00,~
$ utilidad   <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 72684.78, 0.00, 0.00, 16007.2~
$ arrenda    <dbl> 0.00, 32282.60, 11739.13, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0~
$ transfer   <dbl> 8804.34, 8999.99, 0.00, 90538.03, 1956.52, 0.00, 26902.17, ~
```

```
$ jubilacion <dbl> 0.00, 0.00, 0.00, 79239.13, 0.00, 0.00, 0.00, 73369.56, 440~
$ becas      <dbl> 391.3, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.~
$ donativos  <dbl> 0.00, 0.00, 0.00, 0.00, 1956.52, 0.00, 26902.17, 0.00, 0.00~
$ remesas    <dbl> 978.26, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0~
$ bene_gob   <dbl> 7434.78, 0.00, 0.00, 11298.90, 0.00, 0.00, 0.00, 5649.45, 0~
$ transf_hog <dbl> 0.00, 8999.99, 0.00, 0.00, 0.00, 0.00, 0.00, 2442.84, 0.00,~
$ trans_inst <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ estim_alqu <dbl> 11612.90, 0.00, 29032.25, 14516.12, 0.00, 8709.67, 0.00, 14~
$ otros_ing  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ gasto_mon  <dbl> 35091.17, 78670.73, 101647.27, 46702.31, 26927.85, 51176.07~
$ alimentos  <dbl> 9514.19, 17524.25, 18321.36, 14759.90, 12458.47, 6351.40, 1~
$ ali_dentro <dbl> 6814.20, 5181.41, 16907.08, 6274.20, 7315.63, 951.42, 11828~
$ cereales   <dbl> 1465.70, 231.42, 1362.84, 1928.53, 308.56, 617.14, 1915.67,~
$ carnes     <dbl> 617.14, 4114.28, 5142.85, 1928.57, 2442.84, 0.00, 6685.69, ~
$ pescado    <dbl> 0.00, 0.00, 0.00, 0.00, 1799.99, 0.00, 1414.28, 0.00, 0.00,~
$ leche      <dbl> 269.99, 578.57, 0.00, 1414.26, 0.00, 334.28, 0.00, 565.70, ~
$ huevo      <dbl> 0.00, 257.14, 1028.57, 0.00, 321.42, 0.00, 0.00, 0.00, 1002~
$ aceites    <dbl> 0.00, 0.00, 565.71, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0~
$ tuberculo  <dbl> 0.00, 0.00, 0.00, 0.00, 1028.56, 0.00, 321.42, 621.38, 0.00~
$ verduras   <dbl> 2288.53, 0.00, 1735.69, 1002.84, 642.85, 0.00, 951.41, 2069~
$ frutas     <dbl> 1954.27, 0.00, 0.00, 0.00, 0.00, 0.00, 539.99, 1234.27, 195~
$ azucar     <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ cafe       <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ especias   <dbl> 218.57, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0~
$ otros_alim <dbl> 0.00, 0.00, 5142.85, 0.00, 0.00, 0.00, 0.00, 3857.14, 1928.~
$ bebidas    <dbl> 0.00, 0.00, 1928.57, 0.00, 771.41, 0.00, 0.00, 462.84, 2378~
$ ali_fuera  <dbl> 2699.99, 12342.84, 1414.28, 8485.70, 5142.84, 5399.98, 5528~
$ tabaco     <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ vesti_calz <dbl> 2445.64, 684.78, 0.00, 1369.56, 0.00, 1751.06, 9782.60, 489~
$ vestido    <dbl> 2445.64, 684.78, 0.00, 1369.56, 0.00, 1751.06, 5380.43, 489~
$ calzado    <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 4402.17, 0.00, 0.00, 0.~
$ vivienda   <dbl> 1736.75, 29649.66, 3232.25, 2850.00, 2700.00, 3660.00, 1822~
$ alquiler   <dbl> 0.00, 24677.41, 0.00, 0.00, 0.00, 0.00, 13935.48, 0.00, 0.0~
$ pred_cons  <dbl> 116.75, 2032.25, 2032.25, 150.00, 0.00, 150.00, 0.00, 750.0~
$ agua       <dbl> 780, 540, 750, 450, 450, 1200, 1410, 420, 840, 420, 900, 87~
$ energia    <dbl> 840.00, 2400.00, 450.00, 2250.00, 2250.00, 2310.00, 2876.61~
$ limpieza   <dbl> 2075.80, 2816.11, 1422.55, 1228.04, 890.36, 3518.67, 2386.3~
$ cuidados   <dbl> 2075.80, 2816.11, 1422.55, 1228.04, 792.54, 3518.67, 2386.3~
$ utensilios <dbl> 0.00, 0.00, 0.00, 0.00, 97.82, 0.00, 0.00, 0.00, 489.13, 23~
$ enseres    <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ salud      <dbl> 2641.29, 0.00, 0.00, 0.00, 0.00, 1007.60, 8902.16, 3277.16,~
$ atenc_ambu <dbl> 2641.29, 0.00, 0.00, 0.00, 0.00, 1007.60, 7923.90, 3277.16,~
$ hospital   <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
```

```
$ medicinas  <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 978.26, 0.00, 978.26, 0~
$ transporte <dbl> 6773.62, 6706.44, 23312.90, 23574.19, 5080.63, 20601.28, 84~
$ publico    <dbl> 2314.28, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 771.42, 1157.1~
$ foraneo    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ adqui_vehi <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ mantenim   <dbl> 2903.22, 4354.83, 11612.90, 20322.58, 4064.51, 17709.67, 53~
$ refaccion  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ combus     <dbl> 2903.22, 4354.83, 11612.90, 20322.58, 4064.51, 17709.67, 53~
$ comunica   <dbl> 1556.12, 2351.61, 11700.00, 3251.61, 1016.12, 2891.61, 3033~
$ educa_espa <dbl> 2903.22, 0.00, 34728.25, 0.00, 4209.66, 6967.74, 9058.05, 0~
$ educacion  <dbl> 2903.22, 0.00, 0.00, 0.00, 0.00, 6967.74, 6735.47, 0.00, 0.~
$ esparci    <dbl> 0.00, 0.00, 5380.43, 0.00, 4209.66, 0.00, 2322.58, 0.00, 0.~
$ paq_turist <dbl> 0.00, 0.00, 29347.82, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0~
$ personales <dbl> 4097.44, 3870.14, 13416.08, 2920.62, 1344.17, 812.90, 4918.~
$ cuida_pers <dbl> 673.53, 3745.14, 1916.09, 2920.62, 1344.17, 812.90, 4708.95~
$ acces_pers <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ otros_gas  <dbl> 3423.91, 125.00, 11499.99, 0.00, 0.00, 0.00, 210.00, 0.00, ~
$ transf_gas <dbl> 2903.22, 17419.35, 7213.88, 0.00, 244.56, 6505.42, 73.36, 4~
$ percep_tot <dbl> 0.00, 0.00, 0.00, 0.00, 3214.27, 0.00, 0.00, 0.00, 0.00, 0.~
$ retiro_inv <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ prestamos  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ otras_perc <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ ero_nm_viv <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ ero_nm_hog <dbl> 0.00, 0.00, 0.00, 0.00, 3214.27, 0.00, 0.00, 0.00, 0.00, 0.~
$ erogac_tot <dbl> 0.00, 19565.21, 0.00, 28124.99, 0.00, 5771.73, 360.97, 2445~
$ cuota_viv  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ mater_serv <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ material   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ servicio   <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ deposito   <dbl> 0.00, 19565.21, 0.00, 28124.99, 0.00, 5771.73, 0.00, 2445.6~
$ prest_terc <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ pago_tarje <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ deudas     <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ balance    <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ otras_erog <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 360.97, 0.00, 0.00, 0.0~
$ smg        <dbl> 15558.3, 15558.3, 15558.3, 15558.3, 15558.3, 15558.3, 15558~
```

```r
dplyr::glimpse(concentradohogar[,1:10]) # en corchete del lado derecho podemos ojear colum
```

```
Rows: 90,102
Columns: 10
```

```
$ folioviv  <chr> "0100005002", "0100005003", "0100005004", "0100012002", "010~
$ foliohog  <chr> "1", "1", "1", "1", "2", "1", "1", "1", "1", "1", "1", "1", ~
$ ubica_geo <chr> "01001", "01001", "01001", "01001", "01001", "01001", "01001~
$ tam_loc   <chr> "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", ~
$ est_socio <chr> "4", "4", "4", "3", "3", "3", "3", "4", "4", "4", "4", "3", ~
$ est_dis   <chr> "003", "003", "003", "002", "002", "002", "002", "003", "003~
$ upm       <chr> "0000001", "0000001", "0000001", "0000002", "0000002", "0000~
$ factor    <dbl> 206, 206, 206, 167, 167, 167, 167, 212, 212, 212, 212, 184, ~
$ clase_hog <chr> "3", "2", "2", "3", "1", "2", "2", "1", "2", "2", "2", "1", ~
$ sexo_jefe <chr> "2", "1", "1", "1", "1", "1", "2", "2", "2", "1", "1", "1", ~
```

2.5.4 Selección de casos y de variables

Poco a poco vamos comprendiendo más la lógica de R. Hay varias "formas" de programar. Por lo que no te asustes si varios códigos llegan al mismo resultado

Para revisar el contenido de un data frame podemos usar, como lo hicimos anteriormente, el formato basededatos$var o usar corchete, checa como estas cuatro formas tan el mismo resultado.

```
x<-concentradohogar$ing_cor
x<-concentradohogar[["ing_cor"]]  # ¡Ojo con las comillas!
x<-concentradohogar[,23]
x<-concentradohogar[,"ing_cor"]
```

Ahora, con el formato de dplyr podemos llegar a lo mismo

```
x<-concentradohogar %>%
  dplyr::select(ing_cor)
```

2.6 "Subsetting"

Selección "inversa" O sea no "botar algo", es con el negativo. No funciona con todos los formatos

```
x<-concentradohogar %>%
  select(-ing_cor)

rm(x) #rm sólo bota objetos
```

Pero con los otros formatos podemos "asignar" valores adentro de un data.frame, y uno de eso valores puede ser "la nada""

```
concentradohogar$ing_cor2<-concentradohogar$ing_cor
concentradohogar$ing_cor2<-NULL

concentradohogar %<>%
  dplyr::mutate(ing_cor2=ing_cor) # crea o cambia variables

concentradohogar %<>%
  dplyr::mutate(ing_cor2=NULL) # crea o cambia variables
```

De aquí viene esa cuesta en el aprendizaje; tenemos que comprender en qué forma programó el que hizo el paquete e incluso a veces cómo aprendió quién te está enseñando o el foro que estás leyendo.

Rara vez utilizamos una base de datos completa, y rara vez queremos hacer operaciones completas con ellas.

Vamos a pedir cosas más específicas y podemos seleccionar observaciones o filas. Como nuestra base de datos es muy grande, guardaremos el filtro o selección en un objeto.

```
subset1<-concentradohogar[concentradohogar$ing_cor>4,]
```

También podemos seleccionar columnas

```
subset2<- concentradohogar[, c("sexo_jefe", "edad_jefe", "ing_cor")]
```

podemos combinar los dos tipos de selección

```
subset3<- concentradohogar[(concentradohogar$ing_cor>2000 & concentradohogar$sexo_jefe==1
```

Con `{dplyr}`, podemos usar `dplyr::filter()` y `dplyr::select`

```
subset4<-concentradohogar %>%
  dplyr::filter(ing_cor>2000 & sexo_jefe==1) %>%
  dplyr::select(sexo_jefe, edad_jefe, ing_cor)
```

2.7 Etiquetas y cómo usarlas

Podemos ver que los objetos *data.frame*

```
class(concentradohogar$sexo_jefe)
```

```
[1] "character"
```

¿Cómo etiquetamos?

1. Creamos un vector de la etiqueta
2. Convertimos la variable a numérica
3. Usamos `dplyr::mutate()` y `sjlabelled:set_labels()`

```
etiqueta_sex<-c("Hombre", "Mujer")

concentradohogar<-concentradohogar %>%
  mutate(sexo_jefe=as_numeric(sexo_jefe)) %>% # para quitar el "string"
  sjlabelled::set_labels(sexo_jefe, labels=etiqueta_sex)
```

Veamos hoy nuestra variable

```
class(concentradohogar$sexo_jefe)
```

```
[1] "numeric"
```

```
class(sjlabelled::as_label(concentradohogar$sexo_jefe))
```

```
[1] "factor"
```

```
concentradohogar %>%
  mutate(sexo_jefe=sjlabelled::as_label(sexo_jefe)) %>%
  tabyl(sexo_jefe)
```

```
 sexo_jefe     n   percent
    Hombre 61905 0.6870547
     Mujer 28197 0.3129453
```

2.8 Ejercicio

- Escoja una base con la que usted esté trabajando. Impórtela
- Replique la exploración de la práctica: incluya limpiar nombre, alguna revisión global y, opcionalmente, alguna selección de variables o casos de acuerdo a su interés
- Utilice al menos un comando con `{dplyr}` para revisar algo
- Adjunte un archivo con información de la base (para que yo verifique su importación.), así como el código en .R

Envíe al siguiente formulario

3 Revisión de elementos estadísticos básicos

3.1 Análisis descriptivo

Vamos a llamar algunas paquetes que nos ayudarán en esta práctica

```
if (!require("pacman")) install.packages("pacman") # instala pacman si se requiere
```

```
Cargando paquete requerido: pacman
```

```
pacman::p_load(tidyverse,
               writexl,
               haven,
               sjlabelled,
               janitor,
               magrittr,
               GGally,
               wesanderson,
               gt,
               pollster,
               dineq
)
```

3.2 Datos

E importamos la base

```
concentradohogar <- haven::read_sav("datos/concentradohogar.sav")
```

3.2.1 Variables nominales

La variable nominal "sexo_jefe", se captura con "1" para hombres y con un "2" para mujeres en la base de datos. Podemos establecer una operación de igual y además sumar los casos que cumplan con esta condición:

```
  concentradohogar %>%
    dplyr::count(sexo_jefe=="2") # cuentan los casos que cumplen con la condición "sexo_jefe
```

```
# A tibble: 2 x 2
  `sexo_jefe == "2"`     n
  <lgl>              <int>
1 FALSE              61905
2 TRUE               28197
```

Esto es a lo que nos referimos con contar frecuencias. Podemos contar casos que cumplan con una operación de igualdad.

```
  concentradohogar %>%
    with(
      table(sexo_jefe)
      )
```

```
sexo_jefe
    1     2
61905 28197
```

3.2.2 Recordemos nuestro etiquetado

```
  etiqueta_sex<-c("Hombre", "Mujer")

  concentradohogar<-concentradohogar %>%
    mutate(sexo_jefe=as_numeric(sexo_jefe)) %>% # para quitar el "string"
    sjlabelled::set_labels(sexo_jefe, labels=etiqueta_sex)
```

```
  concentradohogar<-concentradohogar %>%
    mutate(clase_hog=as_numeric(clase_hog)) %>% # para quitar el "string"
    sjlabelled::set_labels(clase_hog, labels=c("unipersonal",
                                               "nuclear",
```

```
                                                "ampliado",
                                                "compuesto",
                                                "corresidente"))
```

Con “tabyl()”” de “janitor””

```
  concentradohogar %>%
    dplyr::mutate(sexo_jefe=as_label(sexo_jefe))  %>%
    janitor::tabyl(sexo_jefe)
```

```
 sexo_jefe     n   percent
    Hombre 61905 0.6870547
     Mujer 28197 0.3129453
```

Para ver que esto es una distribución de frecuencias sería muy útil ver la proporción total, ello se realiza agregando un elemento más en nuestro código con una “tubería”:

```
  concentradohogar %>%
    dplyr::mutate(sexo_jefe=as_label(sexo_jefe))  %>%
    janitor::tabyl(sexo_jefe) %>%
    janitor::adorn_totals()
```

```
 sexo_jefe     n   percent
    Hombre 61905 0.6870547
     Mujer 28197 0.3129453
     Total 90102 1.0000000
```

Hoy revisamos algunos tipos de variables

```
  class(concentradohogar$sexo_jefe) # variable sin etiqueta
```

```
[1] "numeric"
```

```
  class(as_label(concentradohogar$sexo_jefe)) # variable con etiqueta
```

```
[1] "factor"
```

```
  class(as_label(concentradohogar$educa_jefe)) # variable ordinal
```

```
[1] "factor"
```

```
  class(concentradohogar$ing_cor) # variable de intervalo/razón
```

```
[1] "numeric"
```

En general, tendremos variables de factor que podrían ser consideradas como cualitativas y numéricas. Aunque en realidad, R tiene muchas formas de almacenamiento. Como mostramos con el comando "glimpse()" en la práctica anterior, podemos revisar una variable en específico:

```
  dplyr::glimpse(concentradohogar$sexo_jefe)
```

```
 num [1:90102] 2 1 1 1 1 1 2 2 2 1 ...
 - attr(*, "labels")= Named num [1:2] 1 2
  ..- attr(*, "names")= chr [1:2] "Hombre" "Mujer"
 - attr(*, "label")= chr "Sexo del jefe del hogar"
```

```
  concentradohogar %>% mutate(sexo_jefe=as_label(sexo_jefe)) %>% # cambia los valores de la
                  tabyl(sexo_jefe) %>% # para hacer la tabla
                  adorn_totals() %>% # añade totales
                  adorn_pct_formatting()  # nos da porcentaje en lugar de proporción
```

```
 sexo_jefe     n percent
    Hombre 61905   68.7%
     Mujer 28197   31.3%
     Total 90102  100.0%
```

La tubería o "pipe" %>% nos permite ir agregando elementos de manera sencilla nuestros comandos. En este caso decimos que dentro del objeto haga el cambio, luego la tabla, que le ponga porcentajes y finalmente que nos dé los totales.

3.2.3 Variables ordinales

Son variables que dan cuenta de cualidades o condiciones a través de categorías que guardan un orden entre sí.

Vamos a darle una "ojeada" a esta variable

```
glimpse(concentradohogar$educa_jefe)
```

```
chr+lbl [1:90102] 03, 08, 10, 11, 08, 08, 10, 06, 10, 04, 06, 08, 10, 11, ...
@ label        : chr "Educación formal del jefe del hogar"
@ format.spss  : chr "A2"
@ display_width: int 6
@ labels       : Named chr [1:11] "10" "01" "11" "02" ...
 ..- attr(*, "names")= chr [1:11] "Profesional completa" "Sin instrucción" "Posgrado" "Prees
```

Etiquetemos también nuestra variable ordinal

```
concentradohogar <-concentradohogar %>%
  mutate(educa_jefe=as_numeric(educa_jefe)) %>%
  set_labels(educa_jefe,
             labels=c("Sin instrucción",
                      "Preescolar",
                      "Primaria incompleta",
                      "Primaria completa",
                      "Secundaria incompleta",
                      "Secundaria completa",
                      "Preparatoria incompleta",
                      "Preparatoria completa",
                      "Profesional incompleta",
                      "Profesional completa",
                      "Posgrado"))
```

Hoy hacemos la tabla, con las etiquetas y vemos que se ve más bonita:

```
concentradohogar %>%
  mutate(educa_jefe=as_label(educa_jefe)) %>%
  tabyl(educa_jefe)
```

```
          educa_jefe     n     percent
      Sin instrucción  5495 0.060986438
```

```
            Preescolar    32 0.000355153
    Primaria incompleta 13328 0.147921245
      Primaria completa 14928 0.165678897
  Secundaria incompleta  2728 0.030276797
    Secundaria completa 24581 0.272813034
Preparatoria incompleta  3032 0.033650751
  Preparatoria completa 11782 0.130762913
 Profesional incompleta  2645 0.029355619
   Profesional completa  9788 0.108632439
               Posgrado  1763 0.019566713
```

Para que no nos salgan las categorías sin datos podemos *apagar* la opción `show_missing_levels=F` dentro del comando "tabyl()"

```
concentradohogar %>%
  mutate(educa_jefe=as_label(educa_jefe)) %>%
  tabyl(educa_jefe, show_missing_levels=F ) %>% # esta opción elimina los valores con 0
  adorn_totals()
```

```
            educa_jefe     n     percent
        Sin instrucción  5495 0.060986438
            Preescolar    32 0.000355153
    Primaria incompleta 13328 0.147921245
      Primaria completa 14928 0.165678897
  Secundaria incompleta  2728 0.030276797
    Secundaria completa 24581 0.272813034
Preparatoria incompleta  3032 0.033650751
  Preparatoria completa 11782 0.130762913
 Profesional incompleta  2645 0.029355619
   Profesional completa  9788 0.108632439
               Posgrado  1763 0.019566713
                  Total 90102 1.000000000
```

3.2.4 Bivariado cualitativo

3.2.4.1 Cálculo de frecuencias

Las tablas de doble entrada tiene su nombre porque en las columnas entran los valores de una variable categórica, y en las filas de una segunda. Básicamente es como hacer un conteo de todas las combinaciones posibles entre los valores de una variable con la otra.

Por ejemplo, si quisiéramos combinar las dos variables que ya estudiamos lo podemos hacer, con una tabla de doble entrada:

```
concentradohogar %>%
  mutate(clase_hog=as_label(clase_hog)) %>%
  mutate(sexo_jefe=as_label(sexo_jefe)) %>% # para que las lea como factor
                tabyl(clase_hog, sexo_jefe, show_missing_levels=F ) %>% # incluimos aquí
                adorn_totals()
```

```
    clase_hog Hombre Mujer
  unipersonal   6519  5367
      nuclear  41919 13621
     ampliado  12898  8888
    compuesto    372   211
 corresidente    197   110
        Total  61905 28197
```

Observamos que en cada celda confluyen los casos que comparten las mismas características:

```
concentradohogar %>%
    count(clase_hog==1 & sexo_jefe==1) # nos da la segunda celda de la izquierda
```

```
# A tibble: 2 x 2
  `clase_hog == 1 & sexo_jefe == 1`     n
  <lgl>                             <int>
1 FALSE                             83583
2 TRUE                               6519
```

3.2.4.2 Totales y porcentajes

De esta manera se colocan todos los datos. Si observa al poner la función "adorn_totals()" lo agregó como una nueva fila de totales, pero también podemos pedirle que agregue una columna de totales.

```
concentradohogar %>%
  mutate(clase_hog=as_label(clase_hog)) %>%
  mutate(sexo_jefe=as_label(sexo_jefe)) %>% # para que las lea como factor
  tabyl(clase_hog, sexo_jefe, show_missing_levels=F ) %>% # incluimos aquí dos variables
  adorn_totals("col")
```

```
   clase_hog Hombre Mujer Total
 unipersonal   6519  5367 11886
     nuclear  41919 13621 55540
    ampliado  12898  8888 21786
   compuesto    372   211   583
corresidente    197   110   307
```

O bien agregar los dos, introduciendo en el argumento `c("col", "row")` un vector de caracteres de las dos opciones requeridas:

```
concentradohogar %>%
  mutate(clase_hog = as_label(clase_hog)) %>%
  mutate(sexo_jefe = as_label(sexo_jefe)) %>% # para que las lea como factor
  tabyl(clase_hog, sexo_jefe, show_missing_levels = F ) %>% # incluimos aquí dos variable
  adorn_totals(c("col", "row"))
```

```
   clase_hog Hombre Mujer Total
 unipersonal   6519  5367 11886
     nuclear  41919 13621 55540
    ampliado  12898  8888 21786
   compuesto    372   211   583
corresidente    197   110   307
       Total  61905 28197 90102
```

Del mismo modo, podemos calcular los porcentajes. Pero los podemos calcular de tres formas. Uno es que lo calculemos para los totales calculados para las filas, para las columnas o para el gran total poblacional.

Para columnas tenemos el siguiente código y los siguientes resultados:

```
concentradohogar %>%
  mutate(clase_hog = as_label(clase_hog)) %>%
  mutate(sexo_jefe = as_label(sexo_jefe)) %>% # para que las lea como factor
  tabyl(clase_hog, sexo_jefe, show_missing_levels = F ) %>% # incluimos aquí dos variable
  adorn_totals(c("col", "row")) %>%
  adorn_percentages("col") %>% # Divide los valores entre el total de la columna
  adorn_pct_formatting() # lo vuelve porcentaje
```

```
   clase_hog Hombre  Mujer  Total
 unipersonal  10.5%  19.0%  13.2%
     nuclear  67.7%  48.3%  61.6%
```

```
    ampliado  20.8%  31.5%  24.2%
   compuesto   0.6%   0.7%   0.6%
corresidente   0.3%   0.4%   0.3%
       Total 100.0% 100.0% 100.0%
```

Cuando se hagan cuadros de distribuciones (que todas sus partes suman 100), los porcentajes pueden ser una gran ayuda para la interpretación, sobre todos cuando se comparar poblaciones de categorías de diferente tamaño. Por lo general, queremos que los cuadros nos den información de donde están los totales y su 100%, de esta manera el lector se puede guiar de porcentaje con respecto a qué está leyendo. En este caso, vemos que el 100% es común en la última fila.

Veamos la diferencia de cómo podemos leer la misma celda, pero hoy, hemos calculado los porcentajes a nivel de fila:

```
concentradohogar %>%
  mutate(clase_hog = as_label(clase_hog)) %>%
  mutate(sexo_jefe = as_label(sexo_jefe)) %>% # para que las lea como factor
  tabyl(clase_hog, sexo_jefe, show_missing_levels = F ) %>% # incluimos aquí dos variable
  adorn_totals(c("col", "row")) %>%
  adorn_percentages("row") %>% # Divide los valores entre el total de la fila
  adorn_pct_formatting() # lo vuelve porcentaje
```

```
  clase_hog Hombre Mujer  Total
 unipersonal  54.8% 45.2% 100.0%
    nuclear  75.5% 24.5% 100.0%
   ampliado  59.2% 40.8% 100.0%
  compuesto  63.8% 36.2% 100.0%
corresidente  64.2% 35.8% 100.0%
       Total  68.7% 31.3% 100.0%
```

Finalmente, podemos calcular los porcentajes con referencia a la población total en análisis. Es decir la celda en la esquina inferior derecha de nuestra tabla original.

```
concentradohogar %>%
  mutate(clase_hog = as_label(clase_hog)) %>%
  mutate(sexo_jefe = as_label(sexo_jefe)) %>% # para que las lea como factor
  tabyl(clase_hog, sexo_jefe, show_missing_levels = F ) %>% # incluimos aquí dos variable
  adorn_totals(c("col", "row")) %>%
  adorn_percentages("all") %>% # Divide los valores entre el total de la población
  adorn_pct_formatting() # lo vuelve porcentaje
```

```
   clase_hog Hombre Mujer  Total
 unipersonal   7.2%  6.0%  13.2%
     nuclear  46.5% 15.1%  61.6%
    ampliado  14.3%  9.9%  24.2%
   compuesto   0.4%  0.2%   0.6%
corresidente   0.2%  0.1%   0.3%
       Total  68.7% 31.3% 100.0%
```

3.3 Factores de expansión y algunas otras medidas

3.3.1 La función `tally()`

El comando `tabyl()` del paquete `{janitor}` es muy útil pero no es compatible con los factores del expansión. En realidad, `tabyl()` nos ahorra un poco el hecho de tener que agrupar nuestra base en categorías y luego hacer un conteo para cada una de ellas. `tally()` es un comando que nos hace ese conteo y `group_by()` nos agrupa las observaciones de nuestra base de datos para hacer cualquier operación.

```
concentradohogar %>%
 group_by(as_label(sexo_jefe)) %>%
  tally(factor) %>% #nombre del factor
  adorn_totals()  # Agrega total
```

```
as_label(sexo_jefe)        n
             Hombre 25397559
              Mujer 12162564
              Total 37560123
```

Podemos usar funciones de `adorns...` de ‘{janitor}

```
concentradohogar %>%
 group_by(as_label(sexo_jefe)) %>%
  tally(factor) %>% #nombre del factor
  adorn_totals() %>% # Agrega total
  adorn_percentages("all")  %>%
  adorn_pct_formatting()
```

```
as_label(sexo_jefe)      n
             Hombre  67.6%
```

```
Mujer  32.4%
Total 100.0%
```

3.3.2 Con `dplyr::count()`

La función `count()` también permite dar pesos a la operaciones de frecuencias, con el argumento `wt =`

```
concentradohogar %>%
  count(sexo_jefe, clase_hog,  wt = factor)
```

```
# A tibble: 10 x 3
   sexo_jefe clase_hog        n
       <dbl>     <dbl>    <dbl>
 1         1         1  2560548
 2         1         2 17303329
 3         1         3  5324985
 4         1         4   132517
 5         1         5    76180
 6         2         1  2316541
 7         2         2  5769256
 8         2         3  3932043
 9         2         4    96921
10         2         5    47803
```

Es compatible con etiquetas

```
concentradohogar %>%
  count(as_label(sexo_jefe), as_label(clase_hog),  wt = factor)
```

```
# A tibble: 10 x 3
   `as_label(sexo_jefe)` `as_label(clase_hog)`        n
   <fct>                 <fct>                    <dbl>
 1 Hombre                unipersonal            2560548
 2 Hombre                nuclear               17303329
 3 Hombre                ampliado               5324985
 4 Hombre                compuesto               132517
 5 Hombre                corresidente             76180
 6 Mujer                 unipersonal            2316541
 7 Mujer                 nuclear                5769256
```

```
 8 Mujer                ampliado              3932043
 9 Mujer                compuesto               96921
10 Mujer                corresidente            47803
```

3.3.3 con {pollster}

Para una variable

```
# tabulado simple con factor de expansión

concentradohogar %>%
  dplyr::mutate(sexo_jefe = sjlabelled::as_label(sexo_jefe)) %>%
  pollster::topline(sexo_jefe , weight = factor)
```

```
# A tibble: 2 x 5
  Response Frequency Percent `Valid Percent` `Cumulative Percent`
  <fct>        <dbl>   <dbl>           <dbl>                <dbl>
1 Hombre    25397559    67.6            67.6                 67.6
2 Mujer     12162564    32.4            32.4                100
```

Para dos variables

```
# tabulado simple con factor de expansión

concentradohogar %>%
  dplyr::mutate(sexo_jefe = sjlabelled::as_label(sexo_jefe)) %>%
  dplyr::mutate(clase_hog = sjlabelled::as_label(clase_hog)) %>%
  pollster::crosstab(sexo_jefe, clase_hog, weight = factor)
```

```
# A tibble: 2 x 7
  sexo_jefe unipersonal nuclear ampliado compuesto corresidente        n
  <fct>           <dbl>   <dbl>    <dbl>     <dbl>        <dbl>    <dbl>
1 Hombre           10.1    68.1     21.0     0.522        0.300 25397559
2 Mujer            19.0    47.4     32.3     0.797        0.393 12162564
```

3.4 Descriptivos para variables cuantitativas

3.4.1 Medidas numéricas básicas

5 números

```r
summary(concentradohogar$ing_cor) ## educación
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      0   28386   46074   61490   74344 7153770 
```

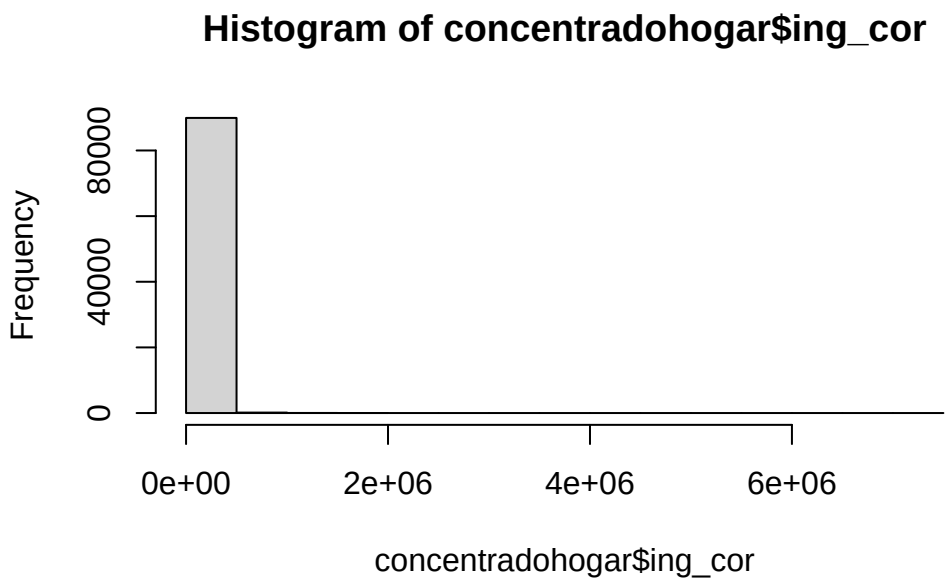
Con pipes se pueden crear "indicadores" de nuestras variables es un tibble

```r
concentradohogar %>%
  summarise(nombre_indicador=mean(ing_cor, na.rm=T))
```

```
# A tibble: 1 x 1
  nombre_indicador
             <dbl>
1           61490.
```

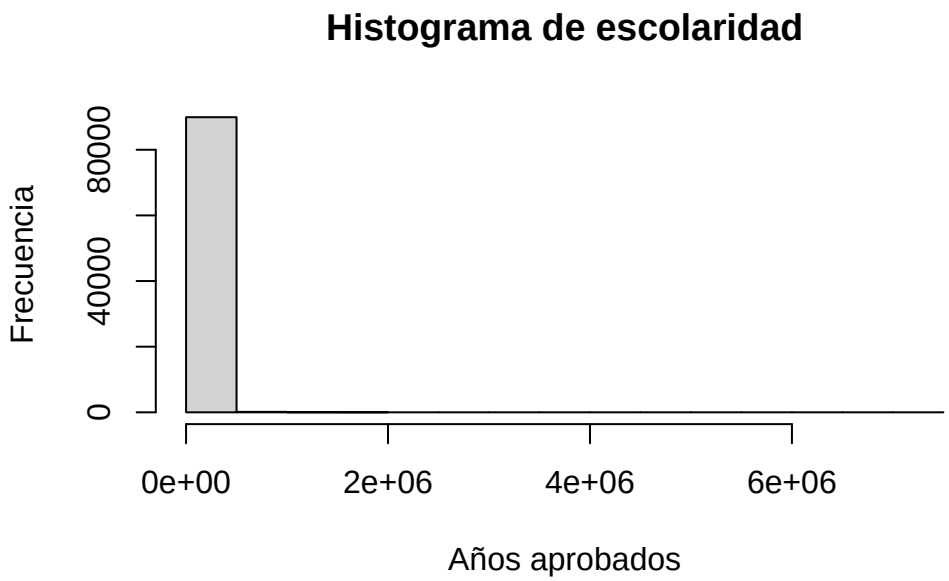
3.4.2 Histograma básico

```r
hist(concentradohogar$ing_cor)
```



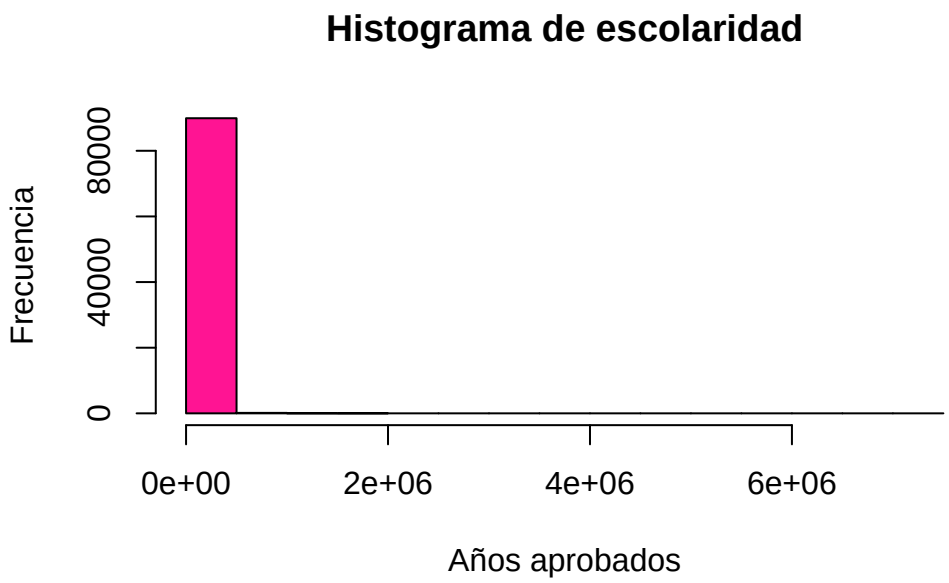
Le podemos modificar el título del eje de las x y de las y

```
hist(concentradohogar$ing_cor,
     main="Histograma de escolaridad",
     xlab="Años aprobados", ylab="Frecuencia")
```



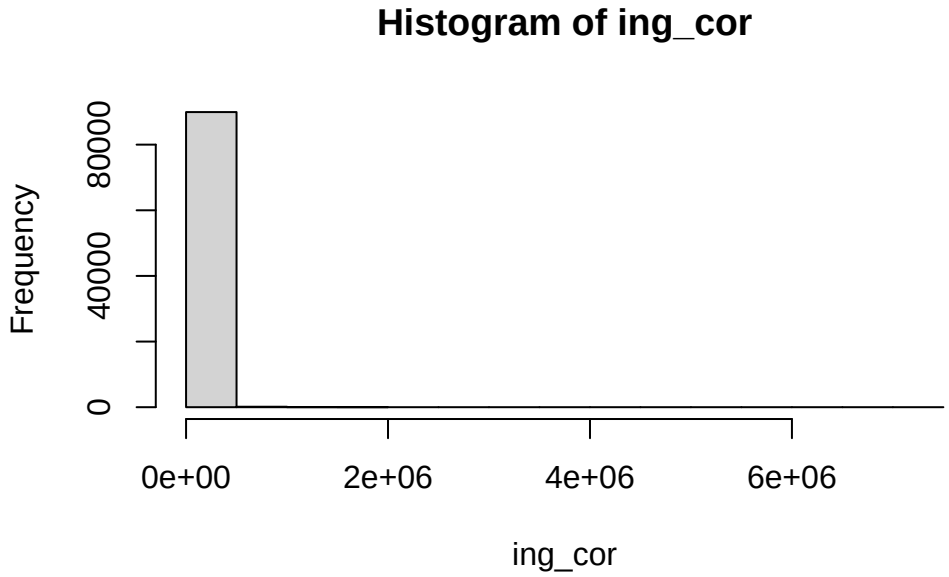
¡A ponerle colorcitos! Aquí hay una lista http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf

```
hist(concentradohogar$ing_cor,
     main="Histograma de escolaridad",
     xlab="Años aprobados",
     ylab="Frecuencia", col="deeppink1")
```



Con pipes:

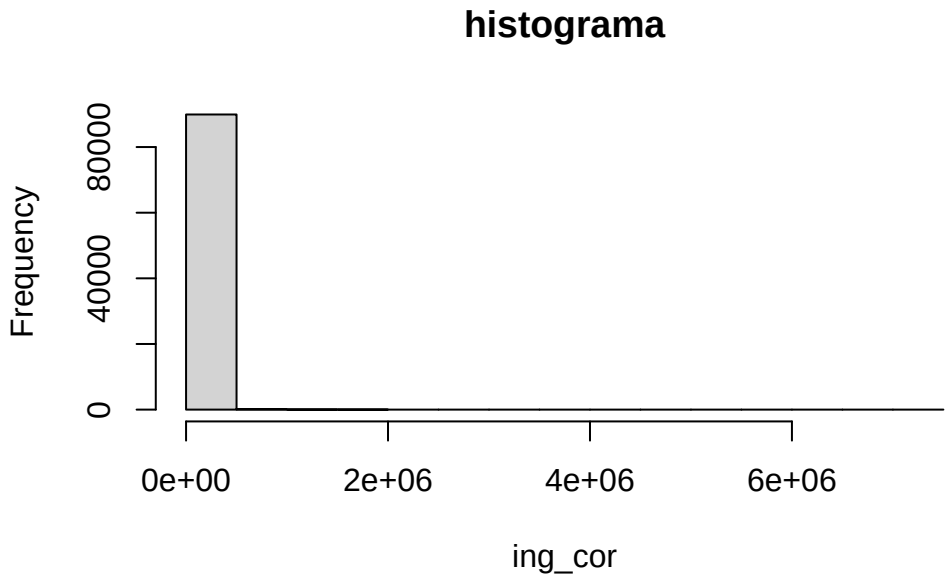
```
concentradohogar %>%
    with(hist(ing_cor)) # con with, para que entienda
```



Cuando usamos pipes, se debe de recordar que no es necesario escribir el nombre del data.frame en el filtro porque es lo primero que colocamos en nuestro "pipe".

Checa que cualquier aditamiento debe ir en el pipe donde está el comando de hist(). Ten cuidado con los paréntesis.

```
concentradohogar %>%
  filter(!is.na(ing_cor)) %>% # la ventaja de esta forma es que podemos hacer más operacio
    with(hist(ing_cor, main= "histograma"))
```



3.5 Recodificación de variables

Por ejemplo, si quisiéramos hacer una variable que separara a los hogares de acuerdo al grupo etario del jefe

3.5.1 `dplyr::if_else()`

```
concentradohogar %<>%
  mutate(joven=dplyr::if_else(edad_jefe<30, 1, 0))

concentradohogar %>% tabyl(edad_jefe,joven)
```

```
edad_jefe    0    1
       13    0    2
       14    0    3
```

15	0	10
16	0	10
17	0	29
18	0	125
19	0	152
20	0	220
21	0	306
22	0	451
23	0	561
24	0	634
25	0	794
26	0	828
27	0	1015
28	0	1183
29	0	1180
30	1458	0
31	1280	0
32	1554	0
33	1488	0
34	1481	0
35	1665	0
36	1675	0
37	1652	0
38	1932	0
39	1751	0
40	2067	0
41	1571	0
42	2265	0
43	1903	0
44	1762	0
45	2080	0
46	1830	0
47	2066	0
48	2076	0
49	2022	0
50	2436	0
51	1746	0
52	2286	0
53	1936	0
54	1966	0
55	1810	0
56	1917	0
57	1839	0

58	1775	0
59	1617	0
60	1880	0
61	1373	0
62	1851	0
63	1633	0
64	1457	0
65	1697	0
66	1289	0
67	1396	0
68	1333	0
69	1036	0
70	1157	0
71	808	0
72	1136	0
73	965	0
74	858	0
75	903	0
76	704	0
77	712	0
78	785	0
79	520	0
80	628	0
81	372	0
82	537	0
83	370	0
84	398	0
85	364	0
86	298	0
87	257	0
88	197	0
89	150	0
90	152	0
91	87	0
92	105	0
93	89	0
94	47	0
95	36	0
96	33	0
97	21	0
98	21	0
99	20	0
100	7	0

```
      101     1     0
      102     6     0
      103     1     0
      104     1     0
      106     1     0
      109     1     0
```

3.5.2 `dplyr::case_when()`

Esto nos ayuda para recodificación múltiple

```
concentradohogar %<>%
  mutate(grupo_edad2=dplyr::case_when(edad_jefe<30 ~ 1,
                                      edad_jefe>29 & edad_jefe<45 ~ 2,
                                      edad_jefe>44 & edad_jefe<65 ~ 3,
                                      edad_jefe>64 ~ 4))


#TRUE~ 4

concentradohogar %>% tabyl(edad_jefe,grupo_edad2)
```

```
edad_jefe    1    2    3    4
       13    2    0    0    0
       14    3    0    0    0
       15   10    0    0    0
       16   10    0    0    0
       17   29    0    0    0
       18  125    0    0    0
       19  152    0    0    0
       20  220    0    0    0
       21  306    0    0    0
       22  451    0    0    0
       23  561    0    0    0
       24  634    0    0    0
       25  794    0    0    0
       26  828    0    0    0
       27 1015    0    0    0
       28 1183    0    0    0
       29 1180    0    0    0
       30    0 1458    0    0
       31    0 1280    0    0
```

32	0	1554	0	0
33	0	1488	0	0
34	0	1481	0	0
35	0	1665	0	0
36	0	1675	0	0
37	0	1652	0	0
38	0	1932	0	0
39	0	1751	0	0
40	0	2067	0	0
41	0	1571	0	0
42	0	2265	0	0
43	0	1903	0	0
44	0	1762	0	0
45	0	0	2080	0
46	0	0	1830	0
47	0	0	2066	0
48	0	0	2076	0
49	0	0	2022	0
50	0	0	2436	0
51	0	0	1746	0
52	0	0	2286	0
53	0	0	1936	0
54	0	0	1966	0
55	0	0	1810	0
56	0	0	1917	0
57	0	0	1839	0
58	0	0	1775	0
59	0	0	1617	0
60	0	0	1880	0
61	0	0	1373	0
62	0	0	1851	0
63	0	0	1633	0
64	0	0	1457	0
65	0	0	0	1697
66	0	0	0	1289
67	0	0	0	1396
68	0	0	0	1333
69	0	0	0	1036
70	0	0	0	1157
71	0	0	0	808
72	0	0	0	1136
73	0	0	0	965
74	0	0	0	858

75	0	0	0	903
76	0	0	0	704
77	0	0	0	712
78	0	0	0	785
79	0	0	0	520
80	0	0	0	628
81	0	0	0	372
82	0	0	0	537
83	0	0	0	370
84	0	0	0	398
85	0	0	0	364
86	0	0	0	298
87	0	0	0	257
88	0	0	0	197
89	0	0	0	150
90	0	0	0	152
91	0	0	0	87
92	0	0	0	105
93	0	0	0	89
94	0	0	0	47
95	0	0	0	36
96	0	0	0	33
97	0	0	0	21
98	0	0	0	21
99	0	0	0	20
100	0	0	0	7
101	0	0	0	1
102	0	0	0	6
103	0	0	0	1
104	0	0	0	1
106	0	0	0	1
109	0	0	0	1

3.5.3 `dplyr::rename()`

Para cambiar los nombres de las variables podemos cambiarlos nombres

```
concentradohogar %<>%
  dplyr::rename(nuevo_nombre=grupo_edad2)
```

Esto en base sería similar a

```r
names(concentradohogar)[128]<-"grupo_edad2"
names(concentradohogar)
```

```
  [1] "folioviv"    "foliohog"    "ubica_geo"   "tam_loc"     "est_socio"
  [6] "est_dis"     "upm"         "factor"      "clase_hog"   "sexo_jefe"
 [11] "edad_jefe"   "educa_jefe"  "tot_integ"   "hombres"     "mujeres"
 [16] "mayores"     "menores"     "p12_64"      "p65mas"      "ocupados"
 [21] "percep_ing"  "perc_ocupa"  "ing_cor"     "ingtrab"     "trabajo"
 [26] "sueldos"     "horas_extr"  "comisiones"  "aguinaldo"   "indemtrab"
 [31] "otra_rem"    "remu_espec"  "negocio"     "noagrop"     "industria"
 [36] "comercio"    "servicios"   "agrope"      "agricolas"   "pecuarios"
 [41] "reproducc"   "pesca"       "otros_trab"  "rentas"      "utilidad"
 [46] "arrenda"     "transfer"    "jubilacion"  "becas"       "donativos"
 [51] "remesas"     "bene_gob"    "transf_hog"  "trans_inst"  "estim_alqu"
 [56] "otros_ing"   "gasto_mon"   "alimentos"   "ali_dentro"  "cereales"
 [61] "carnes"      "pescado"     "leche"       "huevo"       "aceites"
 [66] "tuberculo"   "verduras"    "frutas"      "azucar"      "cafe"
 [71] "especias"    "otros_alim"  "bebidas"     "ali_fuera"   "tabaco"
 [76] "vesti_calz"  "vestido"     "calzado"     "vivienda"    "alquiler"
 [81] "pred_cons"   "agua"        "energia"     "limpieza"    "cuidados"
 [86] "utensilios"  "enseres"     "salud"       "atenc_ambu"  "hospital"
 [91] "medicinas"   "transporte"  "publico"     "foraneo"     "adqui_vehi"
 [96] "mantenim"    "refaccion"   "combus"      "comunica"    "educa_espa"
[101] "educacion"   "esparci"     "paq_turist"  "personales"  "cuida_pers"
[106] "acces_pers"  "otros_gas"   "transf_gas"  "percep_tot"  "retiro_inv"
[111] "prestamos"   "otras_perc"  "ero_nm_viv"  "ero_nm_hog"  "erogac_tot"
[116] "cuota_viv"   "mater_serv"  "material"    "servicio"    "deposito"
[121] "prest_terc"  "pago_tarje"  "deudas"      "balance"     "otras_erog"
[126] "smg"         "joven"       "grupo_edad2"
```

3.6 Ejercicio 3

- Genere una tabla de frecuencias o una tabla de estadísticas con su conjunto de datos con al menos dos variables
- Recuerde respetar las características de sus variables.

Envíelo al siguiente formulario

Videos y extras

Sesión 1

https://youtu.be/N78ZLRTZeLg

Código

Sesión 2

https://youtu.be/w2c45eNz5_0

Código

Sitio web demos

Sesión 3

hhttps://youtu.be/XBXzgVZ4Rs4

Código

Jamboard repaso de paquetes aquí

Cheatsheets

`{dplyr}`

La puedes descargar de aquí