R para el monitoreo de la política de desarrollo social

Ana Escoto Mónica Lara

09/30/2022

Table of contents

Introducción al curso **4**
Objetivo general . . . 4
Temas . . . 4
Metodología . . . 5
Facilitadoras . . . 5
Ana Ruth Escoto Castillo . . . 5
Mónica Lara Escalante . . . 5

Instalación de R y Rstudio **7**
Introducción a R . . . 7
Instalación en OS . . . 7
Instalación en PC . . . 7
Ojo . . . 7

1 Primer acercamiento al uso del programa **8**
1.1 Introducción . . . 8
1.2 Vectores . . . 9
1.3 Matrices . . . 10
1.4 Funciones . . . 11
1.5 Ayuda . . . 12
1.6 Mi ambiente . . . 12
1.7 Directorio de trabajo . . . 13
1.8 Proyectos . . . 13
1.9 Instalación de paquetes . . . 14
1.10 Paquete pacman . . . 15

2 Manejo de datos: importación, selección y revisión **16**
2.1 Previo . . . 16
2.2 Importación de datos . . . 16
2.2.1 Desde Excel . . . 16
2.2.2 Desde STATA y SPSS . . . 17
2.3 Revisión de nuestra base . . . 18
2.4 Revisión con dplyr . . . 19

2.5 Etiquetas y cómo usarlas 20
2.5.1 Ejemplo de etiquetado 20
2.5.2 Ojeando 21
2.5.3 Selección de casos y de variables 25
2.6 "Subsetting" 26

3 Análisis descriptivo básico **28**
3.1 Leer desde archivos de texto y desde una url 28
3.2 Análisis descriptivo básico 28
3.3 Variables nominales 29
3.3.1 Recordemos nuestro etiquetado 29
3.4 Variables ordinales 31
3.5 Bivariado cualitativo 33
3.5.1 Cálculo de frecuencias 33
3.5.2 Totales y porcentajes 34
3.6 Descriptivos para variables cuantitativas 36
3.6.1 Medidas numéricas básicas 36
3.6.2 Histograma básico 36

4 Factores de expansión y algunas otras medidas **41**
4.1 Paquetes 41
4.2 Cargando los datos 41
4.3 La función tally 42
4.4 Otras formas 43
4.5 Diseño complejo 44
4.6 Creación de quintiles y otros grupos 46
4.7 Recodificación de variables 56
4.7.1 if_else() 56
4.7.2 case_when() 59
4.7.3 rename() 61
4.8 Práctica 62

5 Fusionado de conjuntos de datos **63**
5.1 Importación bases ENIGH 2020 63
5.2 Juntando bases 63
5.3 Merge con id compuesto 66
5.4 Bases de distinto tamaño 74
5.5 Cuatro formas de hacer un fusionado 77
5.5.1 Casos en ambas bases 77
5.5.2 Todos los casos 77
5.5.3 Casos en la base 1 77
5.5.4 Casos de la base 2 78
5.6 Las cuatro formas en `dplyr` 78
5.7 Práctica 79

6 Funciones, condicionales, bucles y mapeos **80**

6.1 Paquetes 80
6.2 Datos 80
6.3 Mi primera función 81
6.4 Una función para hacer edades 82
6.5 Bucles 83
6.5.1 `for` 83
6.5.2 `while()` 83
6.6 Condicionales 84
6.7 `purrr::map()` 85
6.8 Combinando funciones con purr::map 92
6.9 Una aplicación para exportar los resultados de una base 122

7 Visualización de datos (I) 127
7.1 Paquetes y datos 127
7.2 Visualización de datos: introducción 127
7.3 Variables cuantitativas 128
7.3.1 Sobre los colores en R: 131
7.4 Práctica en clase: 141

8 Visualización de datos (II) 142
8.1 Paquetes y datos 142
8.2 Variables cualitativas 142
8.3 Práctica 152

Introducción al curso

Objetivo general

El objetivo del curso es que las personas adscritas a la CGMEFFI desarrollen habilidades en el uso del software especializado "R" para fortalecer el análisis y potenciar el alcance de la información derivada del monitoreo de la política de desarrollo social.

Temas

1. Manejo y procesamiento de datos

1.1 Tipos y estructuras de datos

1.2 Operaciones básicas

1.3 Manejo de datos

1.4 Ciclos, secuencias y condicionales

1.5 Funciones

2. Visualización de datos

2.1 Generación de gráficas con ggplot

2.2 Edición de gráficas con ggplot

2.3 Visualización espacial

2.4 Creación de tableros

3. Análisis de texto

3.1 Estructura y carga de datos

3.2 Análisis de palabras

3.3 Relación de texto

Metodología

La metodología del curso consistirá en lo siguiente:

1. La exposición de la facilitadora. Durante la primera parte de la sesión, se expondrán los comandos necesarios para llevar a cabo cada tema. Se dará una introducción sobre la temática y se buscará dar ejemplos concretos para facilitar el aprendizaje. Se espera que el personal exponga sus dudas o comentarios a lo largo de la explicación.

2. Realización de ejercicios prácticos. Al final de cada sesión, corresponderá a las personas asistentes del curso realizar individualmente o en parejas un ejercicio relacionado con lo visto en la primera parte de la clase.

3. Consulta autónoma de material. Tanto la exposición como los ejercicios serán acompañado de material de consulta realizado ad hoc para el curso y el contenido, de tal manera que el estudiantado pueda volver a los códigos y las explicaciones posteriormente.

Facilitadoras

Ana Ruth Escoto Castillo

Doctora en Estudios de Población. Centro de Estudios Demográficos y Urbanos, El Colegio de México.

Semblanza Profesora de tiempo completo en la Facultad de Ciencias Políticas y Sociales. Investigadora nivel I en elSistema Nacional de Investigadores. Maestra en Población y Desarrollo por la Facultad Latinoamericana de Ciencias Sociales (FLACSO) – Sede México. 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. Es co-coordinadora del Capítulo de CDMX de la iniciativa RLadies.

Mónica Lara Escalante

Doctora en Ciencia Política. Centro de Investigación y Docencia Económicas (CIDE) México.

Semblanza Gerente de Información y Políticas Públicas en Sertech MX, asistente de docencia en FLACSO México y profesora de asignatura de Estadística en la UNAM. Maestra en Gobierno y Asuntos Públicos por la Facultad Latinoamericana de Ciencias Sociales (FLACSO) – Sede México. También se ha desempeñado como Analista de Datos Senior en ThinkData MX; como profesora de asignatura y adjunta de diversos cursos de métodos cuantitativos para

el análisis de políticas públicas en la Universidad Autónoma de San Luis Potosí, FLACSO México, Centro de Investigación y Docencia Económicas (CIDE) y Universidad Nacional Autónoma de México (UNAM). Sus líneas de investigación son los estudios legislativos, análisis de políticas públicas a nivel local, instituciones y partidos políticos en América Latina.

Instalación de R y Rstudio

Introducción a R

https://youtu.be/YkN5urybh2A Video en YouTube

Instalación en OS

https://youtu.be/icWV8jzYOtA Viedo en YouTube

Instalación en PC

https://youtu.be/TNSQikMfgJI Video en YouTube

Ojo

Pronto RStudio se volverá **"posit"**

Chapter 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"
```

```
  1:7             # Entero
```

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

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

```
[1] TRUE
```

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

```
[1] FALSE
```

```
  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.

```
  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
```

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

```
[1] 24
```

```
  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
```

```r
dim(m)
```

```
[1] 2 3
```

```r
attributes(m)
```

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

```
  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 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.

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

```
[1] 60
```

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

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

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

```
[1] 3
```

1.5 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 ?

```
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
```

1.6 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

```
ls()
```

```
[1] "m"         "mi_matrix" "n"         "sex"       "x"         "xx"
[7] "y"         "yy"
```

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

```
         used (Mb) gc trigger (Mb) limit (Mb) max used (Mb)
Ncells  598921 32.0    1303332 69.7         NA  1303332 69.7
Vcells 1112623  8.5    8388608 64.0      16384  1839370 14.1
```

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
```

1.7 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/2022/Curso_r_cnvl/coneval"
```

```
  #setwd("C:/Users/anaes/Dropbox/2021/CursoR-posgrado")# Cambio de directorio

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

```
 [1] "01_ppt20221003.pptx" "Icon\r"              "LICENSE"
 [4] "Mi_Exportación.xlsx" "P1.html"             "P1.qmd"
 [7] "P1.rmarkdown"        "P2.qmd"              "P3.qmd"
[10] "P4.R"                "P4.qmd"              "P5.R"
[13] "P5.qmd"              "P6.qmd"              "P7.qmd"
[16] "P8.qmd"              "Pendiente.qmd"       "README.md"
[19] "_quarto.yml"         "coneval.Rproj"       "datos"
[22] "docs"                "index.html"          "index.qmd"
[25] "instala.html"        "instala.qmd"         "intro1.png"
[28] "mds.xlsx"            "modelos.xlsx"        "rrefine.R"
[31] "site_libs"           "tabs.xlsx"
```

Checar que esto también se puede hacer desde el menú:

1.8 Proyectos

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

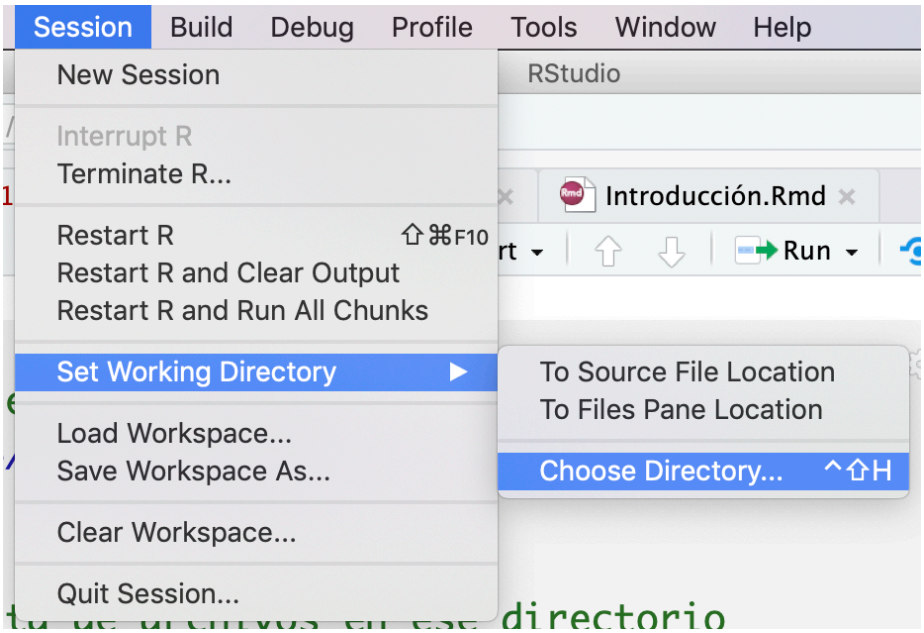


Figure 1.1: i0

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

1.9 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 un software libre.

Vamos a instalar el paquete "foreign", como su nombre lo indica, nos permite leer elementos "extranjeros" en R. Es sumamente útil porque nos permite leer casi todos los formatos, sin necesidad de usar paquetes especializados como StatTransfer.

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

Con la opción "dependencies = TRUE" R nos instalará no sólo la librería o paquete que estamos pidiendo, sino todo aquellos paquetes que necesite la librería en cuestión. Muchas veces los diseños de los paquetes implican el uso de algún otro anterior. Por lo que poner esta sentencia nos puede ahorrar errores cuando estemos usando el paquete. Piensa que esto es similar a cuando enciendes tu computadora y tu sistema operativo te pide que mantengas las actualizaciones.

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(foreign)
library(haven)
```

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

1.10 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
```

```
Loading required package: pacman
```

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

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

Chapter 2

Manejo de datos: importación, selección y revisión

2.1 Previo

Vamos a llamar algunas librerías básicas, el tidyverse (que son muchas librerías) y sjlabelled que nos sirve para el manejo de etiquetas

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

```
Loading required package: pacman
```

```
pacman::p_load(tidyverse, haven, sjlabelled, foreign, janitor) #carga los paquetes necesa
```

2.2 Importación de datos

2.2.1 Desde Excel

El paquete más compatible con RStudio es readxl. A veces, otros paquetes tienen más problemas de configuración entre R y el Java.

```
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

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

2.2.2 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). Se instalasólo la primera vez. Una vez instalado un paquete, lo llamamos con el comando "library"

```
concentrado2020 <- haven::read_dta("datos/concentrado2020.dta")
```

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

El paquete haven sí exporta información.

```
haven::write_dta(concentrado2020, "datos/mi_exportación.dta", version = 12)
```

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

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

Para escribir

```
#haven::write_sav(concentrado2020 , "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 Revisión de nuestra base

Vamos a revisar la base, brevemente la base

```
class(concentrado2020) # tipo de objeto
```

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

```
names(concentrado2020) # 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(concentrado2020) # muestra las primeras 6 líneas
```

```
# A tibble: 6 x 126
  folioviv  folio~1 ubica~2 tam_loc est_s~3 est_dis upm   factor clase~4 sexo_~5
  <chr>     <chr>   <chr>   <chr>   <chr>   <chr>   <chr>  <dbl> <chr>   <chr>
1 01000136~ 1       01001   1       3       002     0000~    190 2       2
2 01000136~ 1       01001   1       3       002     0000~    190 2       1
3 01000178~ 1       01001   1       3       002     0000~    189 2       1
4 01000178~ 1       01001   1       3       002     0000~    189 2       1
5 01000178~ 1       01001   1       3       002     0000~    189 2       1
6 01000178~ 1       01001   1       3       002     0000~    189 2       1
# ... with 116 more variables: 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(concentrado2020$clase_hog) # un tabulado simple
```

```
    1     2     3     4     5
10842 55339 21819   717   289
```

2.4 Revisión con dplyr

Operador de "pipe" o "tubería" %>% (Ctrl+Shift+M) Antes de continuar, presentemos el operador "pipe" %>%. dplyr importa este operador de otro paquete (magrittr). Este operador le permite canalizar la salida de una función a la entrada de otra función. En lugar de funciones de anidamiento (lectura desde adentro hacia afuera), la idea de la tubería es leer las funciones de izquierda a derecha.

```
concentrado2020 %>%
    dplyr::select(sexo_jefe, edad_jefe) %>%
    head
```

```
# A tibble: 6 x 2
  sexo_jefe edad_jefe
  <chr>         <dbl>
1 2                48
2 1                46
3 1                26
```

```
4 1                29
5 1                63
6 1                33
```

```
concentrado2020 %>%
    dplyr::select(sexo_jefe, edad_jefe) %>%
    glimpse
```

```
Rows: 89,006
Columns: 2
$ sexo_jefe <chr> "2", "1", "1", "1", "1", "1", "1", "1", "1", "1", "1", "2", ~
$ edad_jefe <dbl> 48, 46, 26, 29, 63, 33, 60, 76, 74, 37, 76, 79, 37, 80, 46, ~
```

2.5 Etiquetas y cómo usarlas

Podemos ver que los objetos "data.frame"(*spoiler*, ya hablaremos de ellos)

```
class(concentrado2020$sexo_jefe)
```

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

2.5.1 Ejemplo de etiquetado

Para que se vea mejor nuestro tabulado, sería bueno que nuestras variables tuvieran etiqueta. Para ello utilizaremos el paquete "sjlabelled"

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

concentrado2020<-concentrado2020 %>%
  mutate(sexo_jefe=as_numeric(sexo_jefe)) %>% # para quitar el "string"
  sjlabelled::set_labels(sexo_jefe, labels=etiqueta_sex)
```

Etiquetemos también la variable "clase_hog". Podemos checar cómo está estructurada esta base acá https://www.inegi.org.mx/rnm/index.php/catalog/685/data-dictionary

```
concentrado2020<-concentrado2020 %>%
  mutate(clase_hog=as_numeric(clase_hog)) %>% # para quitar el "string"
  sjlabelled::set_labels(clase_hog, labels=c("unipersonal",
                                             "nuclear",
                                             "ampliado",
                                             "compuesto",
                                             "corresidente"))
```

```
table(concentrado2020$sexo_jefe)
```

```
    1     2
63230 25776
```

```
table(sjlabelled::as_label(concentrado2020$sexo_jefe))
```

```
Hombre  Mujer
 63230  25776
```

2.5.2 Ojeando

```
dplyr::glimpse(concentrado2020)
```

```
Rows: 89,006
Columns: 126
$ folioviv   <chr> "0100013605", "0100013606", "0100017801", "0100017802", "01~
$ foliohog   <chr> "1", "1", "1", "1", "1", "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> "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "3",~
$ est_dis    <chr> "002", "002", "002", "002", "002", "002", "002", "002", "00~
$ upm        <chr> "0000001", "0000001", "0000002", "0000002", "0000002", "000~
$ factor     <dbl> 190, 190, 189, 189, 189, 189, 189, 168, 168, 168, 168, 168,~
$ clase_hog  <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 1, 1, 1, 3, 3, 2, 3, 5, 2,~
$ sexo_jefe  <dbl> 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 1, 1, 2, 1, 1,~
$ edad_jefe  <dbl> 48, 46, 26, 29, 63, 33, 60, 76, 74, 37, 76, 79, 37, 80, 46,~
$ educa_jefe <chr> "09", "08", "10", "08", "10", "06", "03", "08", "03", "06",~
$ tot_integ  <dbl> 3, 4, 2, 2, 2, 4, 3, 2, 2, 6, 6, 1, 1, 1, 2, 3, 3, 2, 2, 5,~
$ hombres    <dbl> 1, 3, 1, 2, 1, 2, 2, 1, 1, 3, 4, 0, 1, 0, 0, 2, 1, 1, 2, 3,~
$ mujeres    <dbl> 2, 1, 1, 0, 1, 2, 1, 1, 1, 3, 2, 1, 0, 1, 2, 1, 2, 1, 0, 2,~
$ mayores    <dbl> 3, 3, 2, 1, 2, 2, 3, 2, 2, 3, 6, 1, 1, 1, 2, 3, 2, 2, 2, 5,~
$ menores    <dbl> 0, 1, 0, 1, 0, 2, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,~
$ p12_64     <dbl> 3, 3, 2, 1, 2, 2, 3, 0, 0, 3, 4, 0, 1, 0, 2, 1, 2, 1, 2, 5,~
$ p65mas     <dbl> 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 2, 1, 0, 1, 0, 2, 0, 1, 0, 0,~
$ ocupados   <dbl> 1, 1, 2, 1, 1, 1, 1, 0, 1, 3, 1, 1, 1, 0, 2, 0, 1, 1, 2, 2,~
$ percep_ing <dbl> 2, 2, 2, 1, 1, 1, 2, 1, 2, 2, 5, 1, 1, 1, 1, 2, 1, 2, 2, 3,~
$ perc_ocupa <dbl> 1, 1, 2, 1, 1, 1, 1, 0, 1, 2, 1, 1, 1, 0, 1, 0, 1, 1, 2, 2,~
$ ing_cor    <dbl> 16229.49, 31425.68, 33979.16, 71557.37, 90703.26, 30368.84,~
$ ingtrab    <dbl> 13278.68, 22254.09, 33979.16, 71557.37, 48113.11, 30368.84,~
$ trabajo    <dbl> 0.00, 22254.09, 24098.35, 71557.37, 48113.11, 30368.84, 148~
$ sueldos    <dbl> 0.00, 21639.34, 23606.55, 67868.85, 47213.11, 29508.19, 140~
```

```
$ horas_extr <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ comisiones <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ aguinaldo  <dbl> 0.00, 614.75, 491.80, 3688.52, 0.00, 860.65, 737.70, 0.00, ~
$ 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, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ remu_espec <dbl> 0.00, 0.00, 0.00, 0.00, 900.00, 0.00, 0.00, 0.00, 0.00, 0.0~
$ negocio    <dbl> 1573.77, 0.00, 9880.81, 0.00, 0.00, 0.00, 0.00, 0.00, 8659.~
$ noagrop    <dbl> 1573.77, 0.00, 9880.81, 0.00, 0.00, 0.00, 0.00, 0.00, 8659.~
$ industria  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ comercio   <dbl> 1573.77, 0.00, 9880.81, 0.00, 0.00, 0.00, 0.00, 0.00, 8659.~
$ servicios  <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ 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> 11704.91, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0~
$ rentas     <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 15497~
$ utilidad   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 154979, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ arrenda    <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ transfer   <dbl> 2459.01, 1671.59, 0.00, 0.00, 22131.14, 0.00, 25967.21, 130~
$ jubilacion <dbl> 0.00, 0.00, 0.00, 0.00, 22131.14, 0.00, 25967.21, 7336.95, ~
$ becas      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ donativos  <dbl> 885.24, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 29.34, 0.~
$ remesas    <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ bene_gob   <dbl> 1573.77, 1573.77, 0.00, 0.00, 0.00, 0.00, 0.00, 5086.95, 10~
$ transf_hog <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 606.51, 0.00, 0.0~
$ trans_inst <dbl> 0.00, 97.82, 0.00, 0.00, 0.00, 0.00, 0.00, 58.69, 0.00, 0.0~
$ estim_alqu <dbl> 0.00, 7500.00, 0.00, 0.00, 18000.00, 0.00, 12000.00, 11612.~
$ otros_ing  <dbl> 491.80, 0.00, 0.00, 0.00, 2459.01, 0.00, 0.00, 0.00, 0.00, ~
$ gasto_mon  <dbl> 24626.04, 20397.10, 44955.73, 82950.42, 30140.68, 39991.94,~
$ alimentos  <dbl> 14732.80, 9321.32, 15081.32, 26921.53, 11969.93, 7547.03, 1~
$ ali_dentro <dbl> 13549.96, 9321.32, 9295.63, 22164.39, 3355.69, 7547.03, 112~
$ cereales   <dbl> 3990.78, 1324.26, 1594.26, 2441.54, 0.00, 1529.96, 1259.98,~
$ carnes     <dbl> 989.99, 3882.84, 0.00, 4513.33, 3034.27, 4204.25, 2031.41, ~
$ pescado    <dbl> 0.00, 0.00, 0.00, 1025.87, 0.00, 0.00, 771.42, 0.00, 0.00, ~
$ leche      <dbl> 1613.54, 925.71, 0.00, 449.99, 321.42, 321.42, 2494.25, 707~
$ huevo      <dbl> 822.85, 745.70, 925.71, 0.00, 0.00, 244.28, 642.85, 0.00, 0~
$ aceites    <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 1067.13, 0.00, 0.00, 0.00, 0.~
$ tuberculo  <dbl> 347.14, 0.00, 0.00, 197.48, 0.00, 0.00, 0.00, 411.42, 0.00,~
$ verduras   <dbl> 655.70, 1157.10, 385.71, 2413.26, 0.00, 0.00, 1896.37, 3439~
$ frutas     <dbl> 0.00, 0.00, 0.00, 1367.85, 0.00, 0.00, 642.85, 1504.25, 0.0~
$ azucar     <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ cafe       <dbl> 925.71, 0.00, 0.00, 86.52, 0.00, 0.00, 0.00, 77.14, 0.00, 0~
$ especias   <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ otros_alim <dbl> 3304.26, 1285.71, 3278.56, 9668.55, 0.00, 179.99, 1542.85, ~
```

```
$ bebidas    <dbl> 899.99, 0.00, 3111.39, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, ~
$ ali_fuera  <dbl> 1182.84, 0.00, 5785.69, 4757.14, 8614.24, 0.00, 0.00, 0.00,~
$ 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> 0.00, 0.00, 1006.60, 4509.73, 0.00, 371.73, 0.00, 215.21, 0~
$ vestido    <dbl> 0.00, 0.00, 1006.60, 4294.52, 0.00, 0.00, 0.00, 215.21, 0.0~
$ calzado    <dbl> 0.00, 0.00, 0.00, 215.21, 0.00, 371.73, 0.00, 0.00, 0.00, 0~
$ vivienda   <dbl> 2850.00, 2308.50, 11097.00, 13984.50, 3179.50, 12450.00, 34~
$ alquiler   <dbl> 0.00, 0.00, 9000.00, 12000.00, 0.00, 10500.00, 0.00, 0.00, ~
$ pred_cons  <dbl> 0.0, 0.0, 0.0, 0.0, 212.5, 0.0, 300.0, 100.0, 100.0, 150.0,~
$ agua       <dbl> 750.00, 990.00, 420.00, 756.00, 408.00, 1500.00, 600.00, 39~
$ energia    <dbl> 2100.00, 1318.50, 1677.00, 1228.50, 2559.00, 450.00, 2550.0~
$ limpieza   <dbl> 375.00, 924.00, 2530.16, 708.00, 920.80, 408.00, 845.73, 72~
$ cuidados   <dbl> 375.00, 924.00, 2403.00, 708.00, 429.00, 408.00, 699.00, 72~
$ utensilios <dbl> 0.00, 0.00, 39.13, 0.00, 0.00, 0.00, 146.73, 0.00, 0.00, 0.~
$ enseres    <dbl> 0.00, 0.00, 88.03, 0.00, 491.80, 0.00, 0.00, 0.00, 0.00, 0.~
$ salud      <dbl> 0.00, 782.60, 4509.77, 39.13, 2412.39, 229.87, 213.25, 309.~
$ atenc_ambu <dbl> 0.00, 782.60, 3913.04, 0.00, 0.00, 229.87, 0.00, 309.12, 0.~
$ 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, 596.73, 39.13, 2412.39, 0.00, 213.25, 0.00, 426~
$ transporte <dbl> 5447.24, 4915.68, 7029.68, 7022.39, 7154.75, 16171.31, 4200~
$ publico    <dbl> 1812.82, 1465.68, 514.28, 899.99, 0.00, 1594.27, 0.00, 1092~
$ foraneo    <dbl> 634.42, 0.00, 1475.40, 1475.40, 0.00, 0.00, 0.00, 0.00, 0.0~
$ adqui_vehi <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 7377.04, 0.00, 0.00, 0.00, 0.~
$ mantenim   <dbl> 0.00, 1200.00, 3000.00, 0.00, 6014.75, 1950.00, 3000.00, 11~
$ refaccion  <dbl> 0.00, 0.00, 0.00, 0.00, 2114.75, 0.00, 0.00, 538.04, 0.00, ~
$ combus     <dbl> 0.00, 1200.00, 3000.00, 0.00, 3900.00, 1950.00, 3000.00, 58~
$ comunica   <dbl> 3000.00, 2250.00, 2040.00, 4647.00, 1140.00, 5250.00, 1200.~
$ educa_espa <dbl> 120.00, 0.00, 693.44, 26408.75, 1440.00, 1035.00, 0.00, 0.0~
$ educacion  <dbl> 120.00, 0.00, 0.00, 7650.00, 0.00, 1035.00, 0.00, 0.00, 0.0~
$ esparci    <dbl> 0.00, 0.00, 693.44, 13840.72, 1440.00, 0.00, 0.00, 0.00, 0.~
$ paq_turist <dbl> 0.00, 0.00, 0.00, 4918.03, 0.00, 0.00, 0.00, 0.00, 0.00, 0.~
$ personales <dbl> 1101.00, 2145.00, 2766.78, 2767.30, 112.50, 1779.00, 521.50~
$ cuida_pers <dbl> 1101.00, 2145.00, 2082.00, 2601.00, 0.00, 1029.00, 384.00, ~
$ acces_pers <dbl> 0.00, 0.00, 684.78, 166.30, 0.00, 0.00, 0.00, 0.00, 0.00, 1~
$ otros_gas  <dbl> 0.00, 0.00, 0.00, 0.00, 112.50, 750.00, 137.50, 125.00, 0.0~
$ transf_gas <dbl> 0.00, 0.00, 240.98, 589.09, 2950.81, 0.00, 0.00, 386.40, 0.~
$ percep_tot <dbl> 0.00, 2571.42, 6014.03, 1799.99, 4885.71, 5528.56, 0.00, 22~
$ retiro_inv <dbl> 0.00, 0.00, 3442.61, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.~
$ prestamos  <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ otras_perc <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 48.91, 0.00, 2445~
$ ero_nm_viv <dbl> 0, 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, 2571.42, 2571.42, 1799.99, 4885.71, 5528.56, 0.00, 22~
$ erogac_tot <dbl> 0.00, 2360.65, 1062.28, 885.24, 5901.63, 0.00, 0.00, 0.00, ~
$ cuota_viv  <dbl> 0, 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, 78.68, 0.00, 0.00, 0.00, 0.00, 0.00, 29.34, 0.0~
$ material   <dbl> 0.00, 0.00, 78.68, 0.00, 0.00, 0.00, 0.00, 0.00, 29.34, 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, 0.00, 983.60, 0.00, 5901.63, 0.00, 0.00, 0.00, 0.00, ~
$ prest_terc <dbl> 0.00, 0.00, 0.00, 885.24, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0~
$ pago_tarje <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,~
$ deudas     <dbl> 0.00, 2360.65, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.~
$ 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, 0.00, 0.00, 0.00, 0.00,~
$ smg        <dbl> 11089.8, 11089.8, 11089.8, 11089.8, 11089.8, 11089.8, 11089~
```

```
dplyr::glimpse(concentrado2020[,20:30]) # en corchete del lado derecho podemos ojear colu
```

```
Rows: 89,006
Columns: 11
$ ocupados   <dbl> 1, 1, 2, 1, 1, 1, 1, 0, 1, 3, 1, 1, 1, 0, 2, 0, 1, 1, 2, 2,~
$ percep_ing <dbl> 2, 2, 2, 1, 1, 1, 2, 1, 2, 2, 5, 1, 1, 1, 1, 2, 1, 2, 2, 3,~
$ perc_ocupa <dbl> 1, 1, 2, 1, 1, 1, 1, 0, 1, 2, 1, 1, 1, 0, 1, 0, 1, 1, 2, 2,~
$ ing_cor    <dbl> 16229.49, 31425.68, 33979.16, 71557.37, 90703.26, 30368.84,~
$ ingtrab    <dbl> 13278.68, 22254.09, 33979.16, 71557.37, 48113.11, 30368.84,~
$ trabajo    <dbl> 0.00, 22254.09, 24098.35, 71557.37, 48113.11, 30368.84, 148~
$ sueldos    <dbl> 0.00, 21639.34, 23606.55, 67868.85, 47213.11, 29508.19, 140~
$ horas_extr <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
$ comisiones <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00,~
$ aguinaldo  <dbl> 0.00, 614.75, 491.80, 3688.52, 0.00, 860.65, 737.70, 0.00, ~
$ indemtrab  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

Podemos hacer un tipo "labelbook", usando una función que viene de la librería "sjlabelled", "get_labels". Funciona para toda la base o para columnas, o para variables.

```
#print(get_labels(concentrado2020)) #todas
print(get_labels(concentrado2020[, 20:30])) #de las segundas 10 variables
```

```
$ocupados
NULL

$percep_ing
NULL

$perc_ocupa
NULL

$ing_cor
NULL

$ingtrab
NULL
```

```
$trabajo
NULL

$sueldos
NULL

$horas_extr
NULL

$comisiones
NULL

$aguinaldo
NULL

$indemtrab
NULL
```

No tienen :(

En singular nos da las etiquetas de las variables, no de los valores:

```
  #print(get_label(concentrado2020)) #todas
  print(get_label(concentrado2020[, 1:10])) #de las primeras 10 variables
```

```
                     folioviv                      foliohog
"Identificador de la vivienda"     "Identificador del hogar"
                    ubica_geo                      tam_loc
       "Ubicación geográfica"         "Tamaño de localidad"
                    est_socio                      est_dis
     "Estrato socioeconómico"  "Estrato de diseño muestral"
                          upm                       factor
 "Unidad primaria de muestreo"         "Factor de expansión"
                    clase_hog                    sexo_jefe
            "Clase de hogar"     "Sexo del jefe del hogar"
```

```
  print(get_label(concentrado2020$clase_hog)) #
```

```
[1] "Clase de hogar"
```

2.5.3 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<-concentrado2020$ing_cor
x<-concentrado2020[["ing_cor"]]  # ¡Ojo con las comillas!
x<-concentrado2020[,23]
x<-concentrado2020[,"ing_cor"]
```

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

```
x<-concentrado2020 %>%
  select(ing_cor)
```

2.6 "Subsetting"

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

```
x<-concentrado2020 %>%
  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""

```
concentrado2020$aproba2<-concentrado2020$ing_cor
concentrado2020$aproba2<-NULL
```

De aquí viene esa cuesta en el aprendizaje; tenemos que comprender en qué forma programó el que hizo la librería 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<-concentrado2020[concentrado2020$ing_cor>5000,]
```

También podemos seleccionar columnas

```
subset2<- concentrado2020[, c("sexo_jefe", "edad_jefe", "ing_cor")]
```

podemos combinar los dos tipos de selección

```
subset3<- concentrado2020[(concentrado2020$ing_cor>5000 & concentrado2020$sexo_jefe==1 ),
```

Con dplyr, podemos usar “filter” y “select”

```
subset4<-concentrado2020 %>%
  dplyr::filter(ing_cor>5000 & sexo_jefe==1) %>%
    dplyr::select(sexo_jefe, edad_jefe, ing_cor)
```

Chapter 3

Análisis descriptivo básico

3.1 Leer desde archivos de texto y desde 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 la de CONAPO https://datos.gob.mx/busca/dataset/proyecciones-de-la-poblacion-de-mexico-y-de-las-entidades-federativas-2016-2050

En estas bases hay acentos y otros carecteres especiales del español, por lo que agregaremos una opción de "encoding", de lo contrario da error.

```
mig_inter_quin_proyecciones <- read.csv("http://www.conapo.gob.mx/work/models/CONAPO/Datc
#View(mig_inter_quin_proyecciones)
names(mig_inter_quin_proyecciones)
```

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

3.2 Análisis descriptivo básico

Vamos a llamar algunas librerías básicas, el tidyverse (que son muchas librerías) y sjlabelled que nos sirve para el manejo de etiquetas

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

```
Loading required package: pacman
```

```
pacman::p_load(tidyverse, haven, sjlabelled, foreign, janitor) #carga los paquetes necesa
```

E importamos la base

```
concentrado2020 <- haven::read_dta("datos/concentrado2020.dta")
```

3.3 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:

```
concentrado2020 %>%
  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            63230
2 TRUE             25776
```

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

```
concentrado2020 %>%
  with(
    table(sexo_jefe)
    )
```

```
sexo_jefe
    1     2
63230 25776
```

3.3.1 Recordemos nuestro etiquetado

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

concentrado2020<-concentrado2020 %>%
  mutate(sexo_jefe=as_numeric(sexo_jefe)) %>% # para quitar el "string"
  sjlabelled::set_labels(sexo_jefe, labels=etiqueta_sex)
```

```
concentrado2020<-concentrado2020 %>%
  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""

```
concentrado2020 %>%
  dplyr::mutate(sexo_jefe=as_label(sexo_jefe))  %>%
  janitor::tabyl(sexo_jefe)
```

```
 sexo_jefe     n   percent
    Hombre 63230 0.7104015
     Mujer 25776 0.2895985
```

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":

```
concentrado2020 %>%
  dplyr::mutate(sexo_jefe=as_label(sexo_jefe))  %>%
  janitor::tabyl(sexo_jefe) %>%
  janitor::adorn_totals()
```

```
 sexo_jefe     n   percent
    Hombre 63230 0.7104015
     Mujer 25776 0.2895985
     Total 89006 1.0000000
```

Hoy revisamos algunos tipos de variables

```
class(concentrado2020$sexo_jefe) # variable sin etiqueta
```

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

```
class(as_label(concentrado2020$sexo_jefe)) # variable con etiqueta
```

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

```
class(as_label(concentrado2020$educa_jefe)) # variable ordinal
```

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

```
class(concentrado2020$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(concentrado2020$sexo_jefe)
```

```
num [1:89006] 2 1 1 1 1 1 1 1 1 1 ...
- attr(*, "labels")= Named num [1:2] 1 2
 ..- attr(*, "names")= chr [1:2] "Hombre" "Mujer"
- attr(*, "label")= chr "Sexo del jefe del hogar"
```

```
concentrado2020 %>% 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 63230   71.0%
    Mujer 25776   29.0%
    Total 89006  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. El total del 100% no aparece, por un elemento propio del programa.

3.4 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(concentrado2020$educa_jefe)
```

```
chr [1:89006] "09" "08" "10" "08" "10" "06" "03" "08" "03" "06" "03" "03" ...
- attr(*, "label")= chr "Educación formal del jefe del hogar"
- attr(*, "format.stata")= chr "%2s"
```

Etiquetemos también nuestra variable ordinal

```
concentrado2020 <-concentrado2020 %>%
  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:

```
concentrado2020 %>%
  mutate(educa_jefe=as_label(educa_jefe)) %>%
  tabyl(educa_jefe)
```

```
           educa_jefe     n     percent
       Sin instrucción  6160 0.069208817
            Preescolar    20 0.000224704
    Primaria incompleta 14577 0.163775476
      Primaria completa 15136 0.170055951
  Secundaria incompleta  2974 0.033413478
    Secundaria completa 23865 0.268127991
Preparatoria incompleta  3029 0.034031414
  Preparatoria completa 10550 0.118531335
 Profesional incompleta  2535 0.028481226
   Profesional completa  8474 0.095207065
             Posgrado  1686 0.018942543
```

Para que no nos salgan las categorías sin datos podemos poner una opción dentro del comando "tabyl()"

```
concentrado2020 %>%
  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  6160 0.069208817
            Preescolar    20 0.000224704
```

```
     Primaria incompleta 14577 0.163775476
       Primaria completa 15136 0.170055951
   Secundaria incompleta  2974 0.033413478
     Secundaria completa 23865 0.268127991
 Preparatoria incompleta  3029 0.034031414
   Preparatoria completa 10550 0.118531335
  Profesional incompleta  2535 0.028481226
    Profesional completa  8474 0.095207065
                Posgrado  1686 0.018942543
                   Total 89006 1.000000000
```

3.5 Bivariado cualitativo

3.5.1 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. Basicamente 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:

```
concentrado2020 %>%
  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   6010  4832
      nuclear  43151 12188
     ampliado  13410  8409
    compuesto    477   240
 corresidente    182   107
        Total  63230 25776
```

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

```
concentrado2020 %>%
    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                                82996
2 TRUE                                  6010
```

3.5.2 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.

```
concentrado2020 %>%
  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   6010  4832 10842
     nuclear  43151 12188 55339
    ampliado  13410  8409 21819
   compuesto    477   240   717
corresidente    182   107   289
```

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

```
concentrado2020 %>%
  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   6010  4832 10842
     nuclear  43151 12188 55339
    ampliado  13410  8409 21819
   compuesto    477   240   717
corresidente    182   107   289
       Total  63230 25776 89006
```

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:

```
concentrado2020 %>%
  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   9.5%  18.7%  12.2%
     nuclear  68.2%  47.3%  62.2%
    ampliado  21.2%  32.6%  24.5%
   compuesto   0.8%   0.9%   0.8%
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:

```
concentrado2020 %>%
  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  55.4% 44.6% 100.0%
     nuclear  78.0% 22.0% 100.0%
    ampliado  61.5% 38.5% 100.0%
   compuesto  66.5% 33.5% 100.0%
corresidente  63.0% 37.0% 100.0%
       Total  71.0% 29.0% 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.

```r
concentrado2020 %>%
  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   6.8%  5.4%  12.2%
     nuclear  48.5% 13.7%  62.2%
    ampliado  15.1%  9.4%  24.5%
   compuesto   0.5%  0.3%   0.8%
 corresidente   0.2%  0.1%   0.3%
       Total  71.0% 29.0% 100.0%
```

3.6 Descriptivos para variables cuantitativas

Vamos a empezar a revisar los gráficos para variables cuantitativas.

3.6.1 Medidas numéricas básicas

5 números

```r
summary(concentrado2020$ing_cor) ## ingresos
```

```
   Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
      0    21392    35172    47838    57640 10702107
```

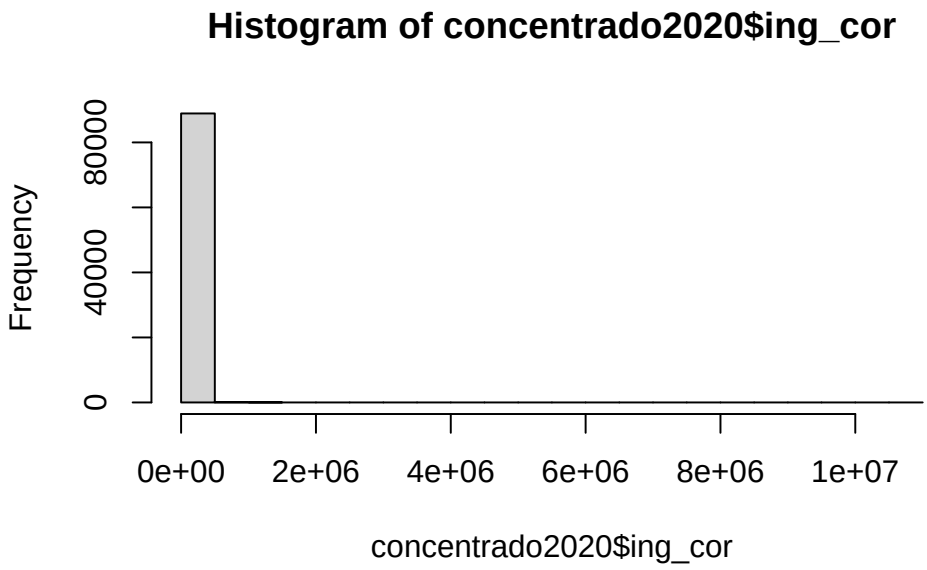
Con pipes se pueden crear "indicadores" de nuestras variables es un tibble

```r
concentrado2020 %>%
  summarise(nombre_indicador=mean(ing_cor, na.rm=T))
```

```
# A tibble: 1 x 1
  nombre_indicador
             <dbl>
1           47838.
```

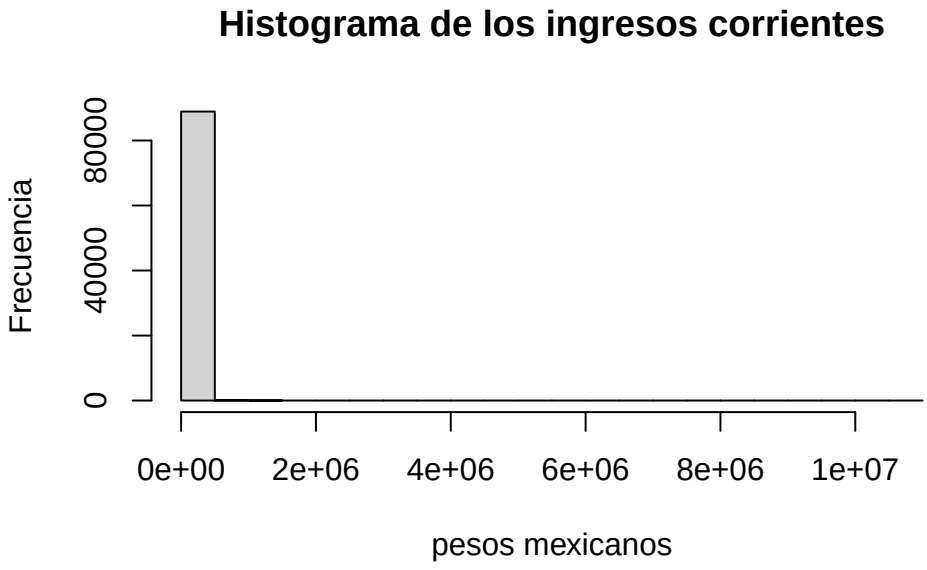
3.6.2 Histograma básico

```r
hist(concentrado2020$ing_cor)
```



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

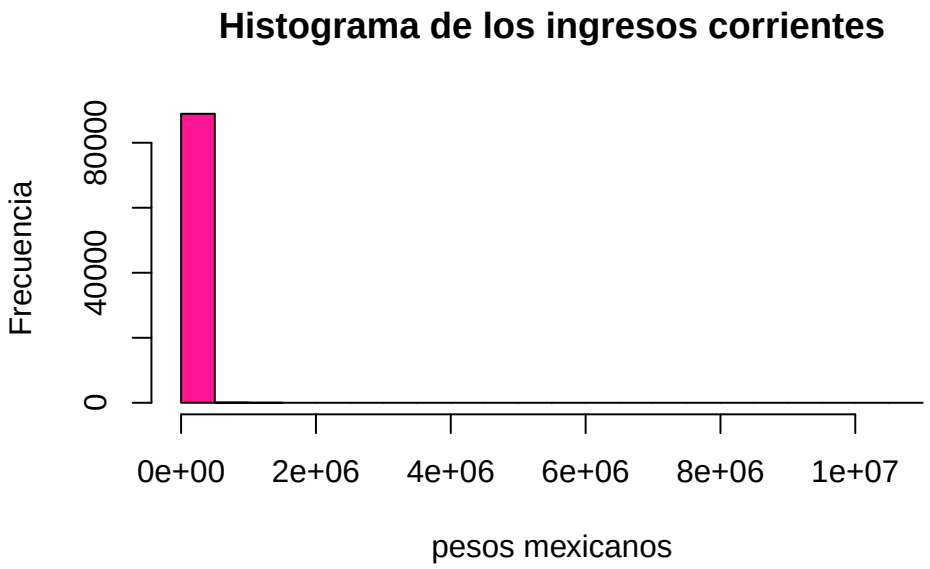
```
hist(concentrado2020$ing_cor,
     main="Histograma de los ingresos corrientes",
     xlab="pesos mexicanos", ylab="Frecuencia")
```



¡A ponerle colorcitos! Aquí hay una lista http://www.stat.columbia.edu/~tzh

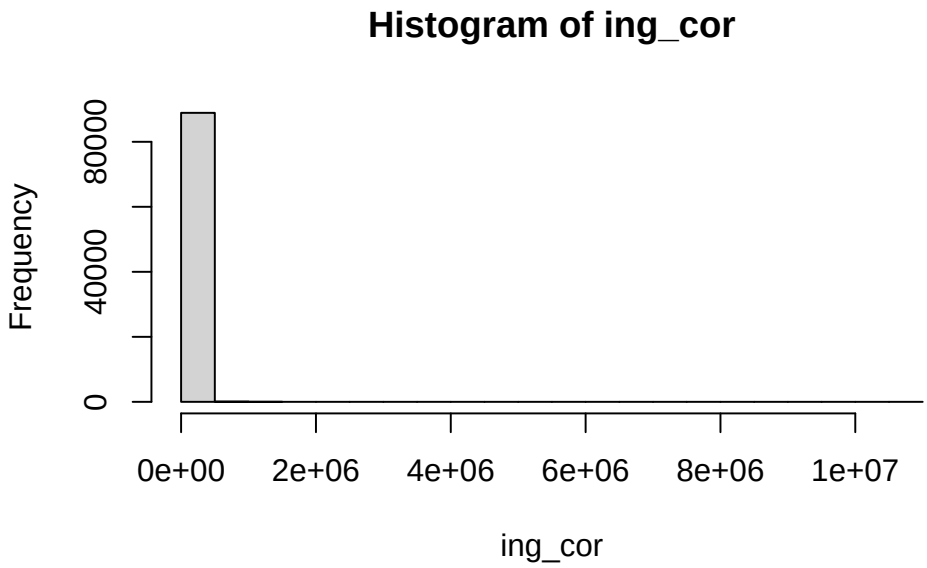
eng/files/Rcolor.pdf

```
hist(concentrado2020$ing_cor,
     main="Histograma de los ingresos corrientes",
     xlab="pesos mexicanos", ylab="Frecuencia",
     col="deeppink1")
```



Con pipes:

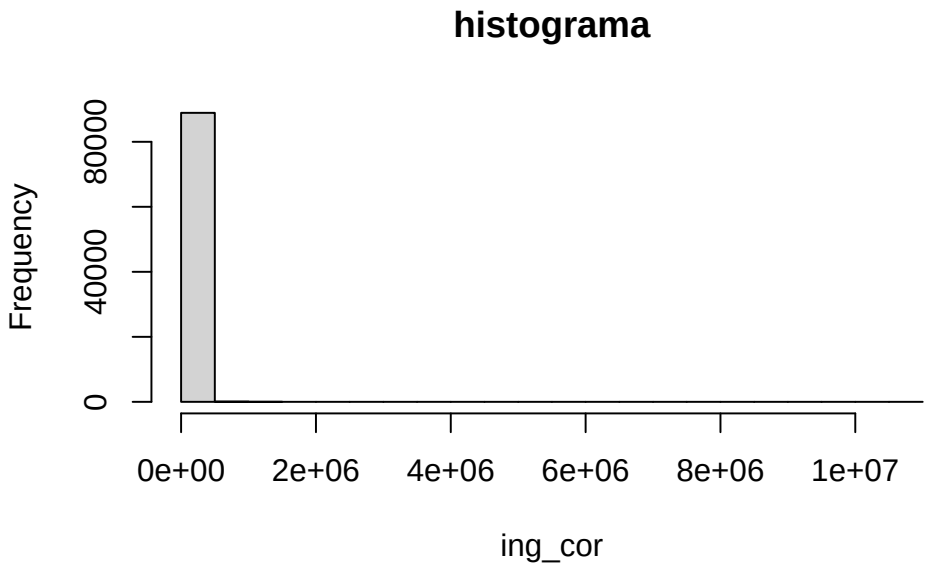
```
concentrado2020 %>%
    with(hist(ing_cor)) # con with, para que entienda
```



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.

```
concentrado2020 %>%
  filter(!is.na(ing_cor)) %>% # la ventaja de esta forma es que podemos hacer más operaci
    with(hist(ing_cor, main= "histograma"))
```


histograma
Frequency
0
40000
80000
0e+00
2e+06
4e+06
6e+06
8e+06
1e+07
ing_cor

Chapter 4

Factores de expansión y algunas otras medidas

4.1 Paquetes

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

```
Loading required package: pacman
```

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

4.2 Cargando los datos

Desde STATA y haremos unos cambios...

```
concentrado2020 <- read_dta("datos/concentrado2020.dta") %>%
  mutate(across(c(sexo_jefe, clase_hog, educa_jefe), as.numeric)) %>% # ojo aquí
  set_labels(sexo_jefe, labels=c("Hombre", "Mujer")) %>%
  set_labels(clase_hog, labels=c("unipersonal","nuclear", "ampliado",
                                 "compuesto","corresidente"))   %>%
   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"))
```

4.3 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.

```
concentrado2020 %>%
 group_by(as_label(sexo_jefe)) %>%
  tally(factor) %>% #nombre del factor
  adorn_totals()  # Agrega total
```

```
as_label(sexo_jefe)        n
             Hombre 25072652
              Mujer 10677007
              Total 35749659
```

Podemos usar funciones de tabyl

```
concentrado2020 %>%
 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  70.1%
               Mujer  29.9%
               Total 100.0%
```

4.4 Otras formas

La función "count()" también permite dar pesos

```
  concentrado2020 %>%
    count(sexo_jefe, clase_hog,  wt = factor)
```

```
# A tibble: 10 x 3
   sexo_jefe clase_hog        n
       <dbl>     <dbl>    <dbl>
 1         1         1  2288234
 2         1         2 17103678
 3         1         3  5408464
 4         1         4   179580
 5         1         5    92696
 6         2         1  1944813
 7         2         2  4989763
 8         2         3  3591323
 9         2         4    98773
10         2         5    52335
```

Es compatible con etiquetas

```
  concentrado2020 %>%
    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            2288234
 2 Hombre                nuclear               17103678
 3 Hombre                ampliado               5408464
 4 Hombre                compuesto               179580
 5 Hombre                corresidente             92696
 6 Mujer                 unipersonal            1944813
 7 Mujer                 nuclear                4989763
 8 Mujer                 ampliado               3591323
```

```
 9 Mujer                 compuesto               98773
10 Mujer                 corresidente            52335
```

Podemos mover un poquito con pivot_wider para que se vea más a lo que acostumbramos a una tabla de frecuencias

```
concentrado2020 %>%
  mutate_at(vars(sexo_jefe, clase_hog), as_label) %>%
  count(sexo_jefe, clase_hog,  wt = factor) %>%
  tidyr::pivot_wider(names_from = sexo_jefe,
              values_from = n)
```

```
# A tibble: 5 x 3
  clase_hog      Hombre   Mujer
  <fct>           <dbl>   <dbl>
1 unipersonal   2288234 1944813
2 nuclear      17103678 4989763
3 ampliado      5408464 3591323
4 compuesto      179580   98773
5 corresidente    92696   52335
```

```
concentrado2020 %>%
  mutate_at(vars(sexo_jefe, clase_hog), as_label) %>% # otra forma de mutate y as_label
  count(sexo_jefe, clase_hog,  wt = factor) %>%
  pivot_wider(names_from = sexo_jefe,
              values_from = n) %>%
  adorn_totals() %>% # Agrega total
  adorn_percentages("col")  %>%
  adorn_pct_formatting()
```

```
   clase_hog Hombre  Mujer
 unipersonal   9.1%  18.2%
     nuclear  68.2%  46.7%
    ampliado  21.6%  33.6%
   compuesto   0.7%   0.9%
corresidente   0.4%   0.5%
       Total 100.0% 100.0%
```

4.5 Diseño complejo

Hay muchos diseños muestrales, asumiremos el diseño simple, pero hay que revisar la documentación de la base

```
# Muestreo aleatorio
ags_srvy <- concentrado2020 %>%
  as_survey_design(weights = factor)
```

Si revisamos las encuestas tiene un diseño complejo, hay estratos y unidades primarias de muestreo

```
# Muestreo estratificado
ags_srvy <- concentrado2020 %>%
  as_survey_design(
    upm = upm,
    strata = est_dis,
    weights = factor,
    nest = TRUE)
```

Como vemos esto es un archivo bien grande, por lo que mejor vamos a seleccionar un par de variables:

```
# simple random sample
ags_srvy <- concentrado2020 %>%
  select(upm, est_dis, factor, clase_hog,
         sexo_jefe, edad_jefe, educa_jefe,ing_cor, factor) %>%
  as_survey_design(
    upm=upm,
    strata = est_dis,
    weights = factor,
    nest = TRUE)
```

Para una media ponderada

```
ags_srvy %>%
  filter(ing_cor>0) %>% # sólo con ingresos
  summarise(
    media_ponderada = survey_mean(ing_cor, na.rm=T))
```

```
# A tibble: 1 x 2
  media_ponderada media_ponderada_se
            <dbl>               <dbl>
1          50315.                341.
```

Si queremos los intervalos de confianza (*spoiler*):

```
ags_srvy %>%
  summarize(
```

```
      media_ponderada = survey_mean(ing_cor,
                                    vartype = "ci") )
```

```
# A tibble: 1 x 3
  media_ponderada media_ponderada_low media_ponderada_upp
            <dbl>               <dbl>               <dbl>
1          50309.              49640.              50979.
```

```
  ags_srvy %>%
    summarize(
      mediana_ponderada = survey_median(ing_cor,
                                    vartype = "ci") )
```

```
# A tibble: 1 x 3
  mediana_ponderada mediana_ponderada_low mediana_ponderada_upp
              <dbl>                 <dbl>                 <dbl>
1            36624.                36365.                36882.
```

```
  ags_srvy %>%
    mutate(sexo_jefe=as_label(sexo_jefe)) %>%
    group_by(sexo_jefe) %>% #variables cuali
    summarize(proportion = survey_mean(), # proporción
              total = survey_total() ) # totales
```

```
# A tibble: 2 x 5
  sexo_jefe proportion proportion_se    total total_se
  <fct>          <dbl>         <dbl>    <dbl>    <dbl>
1 Hombre         0.701       0.00217 25072652   80320.
2 Mujer          0.299       0.00217 10677007   77840.
```

4.6 Creación de quintiles y otros grupos

Uno de los elementos más comunes es crear grupos. Por ejemplo, la función cut, nos ayuda a crear variables con ciertos cortes. Por ejemplo, para recodificar por grupos etarios

```
  concentrado2020 %<>%
    mutate(grupo=cut(edad_jefe,
                     breaks=c(0, 25, 50, 75, 100)))


  concentrado2020 %>%
    tabyl(grupo)
```

```
   grupo     n      percent valid_percent
  (0,25]  3327 0.0373795025    0.03738328
 (25,50] 42558 0.4781475406    0.47819589
 (50,75] 36085 0.4054221064    0.40546311
(75,100]  7027 0.0789497337    0.07895772
    <NA>     9 0.0001011168            NA
```

Algunas opciones se pueden modificar dentro de la función cut

```
concentrado2020 %<>%
  mutate(grupo=cut(edad_jefe,
                   breaks=c(0, 25, 50, 75, 100),
                   include.lowest=T,
                   right= F))


concentrado2020 %>%
  tabyl(grupo)
```

```
   grupo     n      percent valid_percent
  [0,25)  2502 0.0281104645    0.02811331
 [25,50) 41068 0.4614070962    0.46145376
 [50,75) 37488 0.4211850886    0.42122768
[75,100]  7939 0.0891962340    0.08920525
    <NA>     9 0.0001011168            NA
```

Esto nos puede ayudar para hacer variables de rangos de cualquier tipo.

Otro tipo de variables muy importante son los quintiles y demás.

```
concentrado2020 %<>%
  mutate(quintil0=ntile(ing_cor, n=5))

concentrado2020 %>%
  tabyl(quintil0)
```

```
quintil0     n   percent
       1 17802 0.2000090
       2 17801 0.1999978
       3 17801 0.1999978
       4 17801 0.1999978
       5 17801 0.1999978
```

Pero quizás nos interesa más los quintiles que toman en cuenta el factor de expansión

```
concentrado2020 %<>%
  mutate(quintil1=dineq::ntiles.wtd(ing_cor, n=5, weights=factor))

concentrado2020 %>%
  tabyl(quintil1)
```

```
quintil1     n   percent
       1 19133 0.2149630
       2 18253 0.2050761
       3 17803 0.2000202
       4 17609 0.1978406
       5 16208 0.1821001
```

```
concentrado2020 %>%
  count(quintil1, wt=factor) %>%
  mutate(p=n/sum(n)*100) %>%
  adorn_totals()
```

```
quintil1        n         p
       1  7150004  20.00020
       2  7150151  20.00061
       3  7149344  19.99836
       4  7150470  20.00151
       5  7149690  19.99932
   Total 35749659 100.00000
```

Podemos también ver la diferencia en los máximos y minimos de ambas variables

```
concentrado2020 %>%
  group_by(quintil0) %>%
  summarise(min=min(ing_cor),
            max=max(ing_cor))
```

```
# A tibble: 5 x 3
  quintil0    min       max
     <int>  <dbl>     <dbl>
1        1     0     18934.
2        2 18935.    29188.
3        3 29188.    42257.
4        4 42257.    65267.
5        5 65268. 10702107.
```

Veamos con la ponderación:

```
  concentrado2020 %>%
    group_by(quintil1) %>%
    summarise(min=min(ing_cor),
              max=max(ing_cor))
```

```
# A tibble: 5 x 3
  quintil1    min       max
     <dbl>  <dbl>     <dbl>
1        1     0     19666.
2        2 19668.    30326.
3        3 30326.    44017.
4        4 44017.    68533.
5        5 68534. 10702107.
```

La flexibilidad de dplyr nos permite además hacer quintiles fácilmente adentro de grupos. Por ejemplo si quisiéramos hacer quintiles estatales… Claro para eso debemos tener la variable.

La variable "ubica_geo", nos da esa información pero junta

```
  concentrado2020 %>%
    select(ubica_geo) %>%
    head
```

```
# A tibble: 6 x 1
  ubica_geo
  <chr>
1 01001
2 01001
3 01001
4 01001
5 01001
6 01001
```

Vamos a crear dos variables, uan que nos diga la entidad y la otra el municipio

```
  concentrado2020 %<>%
    mutate(ent=stringr::str_sub(ubica_geo, start = 1, end = 2)) %>%
    mutate(mun=stringr::str_sub(ubica_geo, start = 3, end = 5))

  concentrado2020 %>% tabyl(ent)
```

```
 ent    n    percent
  01 2669 0.02998674
  02 4142 0.04653619
  03 2717 0.03052603
```

```
 04 2174 0.02442532
 05 3922 0.04406445
 06 3282 0.03687392
 07 2123 0.02385232
 08 4572 0.05136732
 09 2570 0.02887446
 10 2746 0.03085185
 11 3083 0.03463811
 12 2490 0.02797564
 13 2213 0.02486349
 14 2779 0.03122261
 15 3568 0.04008719
 16 2047 0.02299845
 17 2564 0.02880705
 18 2103 0.02362762
 19 3502 0.03934566
 20 2596 0.02916657
 21 2141 0.02405456
 22 3769 0.04234546
 23 2196 0.02467249
 24 2521 0.02832393
 25 3429 0.03852549
 26 2420 0.02718918
 27 2088 0.02345909
 28 2311 0.02596454
 29 2159 0.02425679
 30 2717 0.03052603
 31 2889 0.03245849
 32 2504 0.02813293
```

```
concentrado2020 %>% tabyl(mun)
```

```
mun    n      percent
001 4929 5.537829e-02
002 4164 4.678336e-02
003 3196 3.590769e-02
004 3636 4.085118e-02
005 3578 4.019954e-02
006 3230 3.628969e-02
007 3069 3.448082e-02
008 2271 2.551513e-02
009 1779 1.998742e-02
010 2050 2.303216e-02
011 1819 2.043682e-02
012 1738 1.952677e-02
```

```
013 1317 1.479676e-02
014 2189 2.459385e-02
015  855 9.606094e-03
016 1022 1.148237e-02
017 2582 2.900928e-02
018 1436 1.613374e-02
019 1277 1.434735e-02
020 1733 1.947060e-02
021  963 1.081950e-02
022  601 6.752354e-03
023  355 3.988495e-03
024  569 6.392827e-03
025  716 8.044402e-03
026  606 6.808530e-03
027  925 1.039256e-02
028 1017 1.142620e-02
029  882 9.909444e-03
030 1851 2.079635e-02
031  853 9.583624e-03
032  889 9.988091e-03
033 1339 1.504393e-02
034  407 4.572725e-03
035 1637 1.839202e-02
036  467 5.246837e-03
037 1695 1.904366e-02
038  789 8.864571e-03
039 1434 1.611127e-02
040  309 3.471676e-03
041  756 8.493809e-03
042  402 4.516549e-03
043  445 4.999663e-03
044  337 3.786262e-03
045  269 3.022268e-03
046  524 5.887244e-03
047  259 2.909916e-03
048  634 7.123115e-03
049  234 2.629036e-03
050 1125 1.263960e-02
051  427 4.797429e-03
052  340 3.819967e-03
053  593 6.662472e-03
054  266 2.988563e-03
055  557 6.258005e-03
056  437 4.909781e-03
057  320 3.595263e-03
058  289 3.246972e-03
```

```
059  240 2.696447e-03
060  143 1.606633e-03
061  206 2.314451e-03
062  252 2.831270e-03
063  232 2.606566e-03
064  167 1.876278e-03
065  203 2.280745e-03
066  146 1.640339e-03
067  381 4.280610e-03
068   91 1.022403e-03
069  246 2.763859e-03
070   83 9.325214e-04
071  157 1.763926e-03
072   60 6.741119e-04
073  251 2.820035e-03
074  147 1.651574e-03
075   38 4.269375e-04
076  306 3.437970e-03
077  159 1.786396e-03
078  127 1.426870e-03
079  277 3.112150e-03
080   21 2.359392e-04
081   70 7.864638e-04
082  157 1.763926e-03
083  157 1.763926e-03
084   89 9.999326e-04
085  181 2.033571e-03
086   92 1.033638e-03
087  257 2.887446e-03
088   77 8.651102e-04
089  237 2.662742e-03
090   41 4.606431e-04
091  119 1.336989e-03
092   78 8.763454e-04
093   81 9.100510e-04
094   63 7.078175e-04
095   23 2.584095e-04
096  176 1.977395e-03
097  356 3.999730e-03
098  263 2.954857e-03
099  140 1.572928e-03
100   94 1.056109e-03
101  537 6.033301e-03
102  359 4.033436e-03
104  222 2.494214e-03
105  222 2.494214e-03
```

106	308	3.460441e-03
107	125	1.404400e-03
108	364	4.089612e-03
109	114	1.280813e-03
110	64	7.190526e-04
111	95	1.067344e-03
112	92	1.033638e-03
113	83	9.325214e-04
114	639	7.179291e-03
115	123	1.381929e-03
116	16	1.797632e-04
117	23	2.584095e-04
118	65	7.302878e-04
119	34	3.819967e-04
120	381	4.280610e-03
121	120	1.348224e-03
122	49	5.505247e-04
123	76	8.538750e-04
124	110	1.235872e-03
125	29	3.258207e-04
127	21	2.359392e-04
128	59	6.628767e-04
129	24	2.696447e-04
130	34	3.819967e-04
131	70	7.864638e-04
132	80	8.988158e-04
133	32	3.595263e-04
134	20	2.247040e-04
135	8	8.988158e-05
136	24	2.696447e-04
138	22	2.471743e-04
140	36	4.044671e-04
141	109	1.224637e-03
142	20	2.247040e-04
143	25	2.808799e-04
144	64	7.190526e-04
145	42	4.718783e-04
149	18	2.022336e-04
153	24	2.696447e-04
154	42	4.718783e-04
156	96	1.078579e-03
157	7	7.864638e-05
160	66	7.415230e-04
163	44	4.943487e-04
164	35	3.932319e-04
167	23	2.584095e-04

169	24	2.696447e-04
170	63	7.078175e-04
171	33	3.707615e-04
173	58	6.516415e-04
174	43	4.831135e-04
175	72	8.089342e-04
176	24	2.696447e-04
177	22	2.471743e-04
179	23	2.584095e-04
181	17	1.909984e-04
183	38	4.269375e-04
184	159	1.786396e-03
186	20	2.247040e-04
187	20	2.247040e-04
188	19	2.134688e-04
189	73	8.201694e-04
191	16	1.797632e-04
193	161	1.808867e-03
194	22	2.471743e-04
197	21	2.359392e-04
199	21	2.359392e-04
200	24	2.696447e-04
201	83	9.325214e-04
202	43	4.831135e-04
204	21	2.359392e-04
205	23	2.584095e-04
206	42	4.718783e-04
208	43	4.831135e-04
212	14	1.572928e-04
227	5	5.617599e-05
234	20	2.247040e-04
261	23	2.584095e-04
266	20	2.247040e-04
271	16	1.797632e-04
277	39	4.381727e-04
278	47	5.280543e-04
293	16	1.797632e-04
295	22	2.471743e-04
302	22	2.471743e-04
309	20	2.247040e-04
315	19	2.134688e-04
318	42	4.718783e-04
324	143	1.606633e-03
334	66	7.415230e-04
348	24	2.696447e-04
349	21	2.359392e-04

```
350   11 1.235872e-04
364   23 2.584095e-04
365   15 1.685280e-04
372   18 2.022336e-04
385   70 7.864638e-04
390   20 2.247040e-04
394   20 2.247040e-04
397   16 1.797632e-04
399   11 1.235872e-04
401   21 2.359392e-04
403    5 5.617599e-05
406   33 3.707615e-04
413   42 4.718783e-04
418   48 5.392895e-04
439   41 4.606431e-04
441   24 2.696447e-04
447   22 2.471743e-04
460   21 2.359392e-04
466   20 2.247040e-04
467   25 2.808799e-04
469   37 4.157023e-04
482   22 2.471743e-04
483   22 2.471743e-04
491   21 2.359392e-04
504   22 2.471743e-04
515   21 2.359392e-04
539   19 2.134688e-04
546   21 2.359392e-04
551   43 4.831135e-04
553    9 1.011168e-04
559   43 4.831135e-04
570   46 5.168191e-04
```

Hoy sí podemos hacer nuestras variables dentro de cada entidad federativa

```
concentrado2020 %<>%
  group_by(ent) %>%
  mutate(quintil2=dineq::ntiles.wtd(ing_cor, n=5, weights=factor)) %>%
  ungroup()
```

¿Discreparán muchos los hogares en sus distribuciones a nivel nacional y por entidad?

```
concentrado2020 %>%
  tabyl(quintil1,quintil2) %>%
```

```
  adorn_totals(c("row", "col"))
```

```
quintil1     1     2     3     4     5 Total
       1 15878  3088   167     0     0 19133
       2  4413 10071  3503   266     0 18253
       3     0  5583  8917  3221    82 17803
       4     0     0  5089 10301  2219 17609
       5     0     0     0  2969 13239 16208
   Total 20291 18742 17676 16757 15540 89006
```

Y si queremos este tabulado más bonito

```
concentrado2020 %>%
  tabyl(quintil1,quintil2) %>%
  adorn_totals(c("row", "col")) %>%
  gt()
```

quintil1	1	2	3	4	5	Total
1	15878	3088	167	0	0	19133
2	4413	10071	3503	266	0	18253
3	0	5583	8917	3221	82	17803
4	0	0	5089	10301	2219	17609
5	0	0	0	2969	13239	16208
Total	20291	18742	17676	16757	15540	89006

concentrado2020 %>% tabyl(quintil1,quintil2) %>% adorn_totals(c("row", "col")) %>% gt() %>% tab_header(title = md("Distribución de los hogares en **México**"), subtitle = md("Según quintiles y *quintiles*")) %>% tab_footnote(footnote = paste(get_label(concentrado2020$ing_cor)))

4.7 Recodificación de variables

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

4.7.1 if_else()

```
concentrado2020 %<>%
  mutate(joven=dplyr::if_else(edad_jefe<30, 1, 0))

concentrado2020 %>% tabyl(edad_jefe,joven)
```

edad_jefe	0	1
14	0	1
15	0	2
16	0	19
17	0	28
18	0	93
19	0	153
20	0	243
21	0	312
22	0	423
23	0	588
24	0	640
25	0	825
26	0	914
27	0	1028
28	0	1163
29	0	1193
30	1461	0
31	1221	0
32	1495	0
33	1426	0
34	1493	0
35	1627	0
36	1654	0
37	1619	0
38	1899	0
39	1769	0
40	2146	0
41	1590	0
42	2232	0
43	2029	0
44	1827	0
45	2163	0
46	1975	0
47	2138	0
48	2153	0
49	2028	0
50	2315	0
51	1672	0
52	2044	0
53	1859	0
54	1942	0
55	1900	0
56	1900	0
57	1644	0
58	1704	0

59	1589	0
60	1930	0
61	1283	0
62	1563	0
63	1545	0
64	1344	0
65	1514	0
66	1187	0
67	1216	0
68	1331	0
69	1003	0
70	1255	0
71	837	0
72	1027	0
73	965	0
74	919	0
75	912	0
76	754	0
77	655	0
78	764	0
79	532	0
80	695	0
81	404	0
82	463	0
83	406	0
84	430	0
85	402	0
86	317	0
87	250	0
88	215	0
89	144	0
90	171	0
91	72	0
92	85	0
93	76	0
94	54	0
95	45	0
96	33	0
97	29	0
98	19	0
99	7	0
100	5	0
101	2	0
102	2	0
103	1	0
104	2	0

```
       105     1     0
       107     1     0
```

4.7.2 case_when()

Esto nos ayuda para recodificación múltiple

```
concentrado2020 %<>%
  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

concentrado2020 %>% tabyl(edad_jefe,grupo_edad2)
```

```
edad_jefe    1    2    3    4
       14    1    0    0    0
       15    2    0    0    0
       16   19    0    0    0
       17   28    0    0    0
       18   93    0    0    0
       19  153    0    0    0
       20  243    0    0    0
       21  312    0    0    0
       22  423    0    0    0
       23  588    0    0    0
       24  640    0    0    0
       25  825    0    0    0
       26  914    0    0    0
       27 1028    0    0    0
       28 1163    0    0    0
       29 1193    0    0    0
       30    0 1461    0    0
       31    0 1221    0    0
       32    0 1495    0    0
       33    0 1426    0    0
       34    0 1493    0    0
       35    0 1627    0    0
       36    0 1654    0    0
       37    0 1619    0    0
       38    0 1899    0    0
       39    0 1769    0    0
       40    0 2146    0    0
       41    0 1590    0    0
```

42	0	2232	0	0
43	0	2029	0	0
44	0	1827	0	0
45	0	0	2163	0
46	0	0	1975	0
47	0	0	2138	0
48	0	0	2153	0
49	0	0	2028	0
50	0	0	2315	0
51	0	0	1672	0
52	0	0	2044	0
53	0	0	1859	0
54	0	0	1942	0
55	0	0	1900	0
56	0	0	1900	0
57	0	0	1644	0
58	0	0	1704	0
59	0	0	1589	0
60	0	0	1930	0
61	0	0	1283	0
62	0	0	1563	0
63	0	0	1545	0
64	0	0	1344	0
65	0	0	0	1514
66	0	0	0	1187
67	0	0	0	1216
68	0	0	0	1331
69	0	0	0	1003
70	0	0	0	1255
71	0	0	0	837
72	0	0	0	1027
73	0	0	0	965
74	0	0	0	919
75	0	0	0	912
76	0	0	0	754
77	0	0	0	655
78	0	0	0	764
79	0	0	0	532
80	0	0	0	695
81	0	0	0	404
82	0	0	0	463
83	0	0	0	406
84	0	0	0	430
85	0	0	0	402
86	0	0	0	317
87	0	0	0	250

88	0	0	0	215
89	0	0	0	144
90	0	0	0	171
91	0	0	0	72
92	0	0	0	85
93	0	0	0	76
94	0	0	0	54
95	0	0	0	45
96	0	0	0	33
97	0	0	0	29
98	0	0	0	19
99	0	0	0	7
100	0	0	0	5
101	0	0	0	2
102	0	0	0	2
103	0	0	0	1
104	0	0	0	2
105	0	0	0	1
107	0	0	0	1

4.7.3 rename()

Para cambiar los nombres de las variables podemos cambiarlos nombres

```
concentrado2020 %<>%
  dplyr::rename(nuevo_nombre=grupo_edad2)
```

Esto en base sería similar a

```
names(concentrado2020)[134]<-"grupo_edad2"
names(concentrado2020)
```

```
 [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"         "grupo"       "quintil0"    "quintil1"    "ent"
[131] "mun"         "quintil2"    "joven"       "grupo_edad2"
```

4.8 Práctica

- Genere una variable de deciles de ingresos dentro de cada tamaño de localidad tam_loc
- Etiquete los valores de los deciles con números romanos
- Encuentre el coeficiente de variación para las estimaciones dentro de esa variable, sexo del jefe y tamaño de localidad

Chapter 5

Fusionado de conjuntos de datos

5.1 Importación bases ENIGH 2020

Vamos a trabajar con esta base que tiene elementos separados.

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

```
Loading required package: pacman
```

```
pacman::p_load(skimr,tidyverse, magrittr, # sobretodo para dplyr
               haven, readxl, #importación
               janitor,
               sjlabelled)
```

Hoy cargamos la versión seccionada de la base

```
viviendas <- haven::read_dta("datos/viviendas2020.dta")
hogares <- haven::read_dta("datos/hogares2020.dta")
poblacion<- haven::read_dta("datos/poblacion2020.dta")
```

5.2 Juntando bases

Muchas bases de datos están organizadas en varias tablas. La ventaja de la programación por objetos de R, nos permite tener las bases cargadas en nuestro ambiente y llamarlas y juntarlas cuando sea necesario.

```
  dim(viviendas)
```

```
[1] 87754    64
```

```
  names(viviendas[,1:15])
```

```
 [1] "folioviv"   "tipo_viv"   "mat_pared"  "mat_techos" "mat_pisos"
 [6] "antiguedad" "antigua_ne" "cocina"     "cocina_dor" "cuart_dorm"
[11] "num_cuarto" "disp_agua"  "dotac_agua" "excusado"   "uso_compar"
```

```
  dim(hogares)
```

```
[1] 89006   137
```

```
  names(hogares[,1:15])
```

```
 [1] "folioviv"   "foliohog"   "huespedes"  "huesp_come" "num_trab_d"
 [6] "trab_come"  "acc_alim1"  "acc_alim2"  "acc_alim3"  "acc_alim4"
[11] "acc_alim5"  "acc_alim6"  "acc_alim7"  "acc_alim8"  "acc_alim9"
```

```
  dim(poblacion)
```

```
[1] 315743    184
```

```
  names(poblacion[,1:15])
```

```
 [1] "folioviv"   "foliohog"   "numren"     "parentesco" "sexo"
 [6] "edad"       "madre_hog"  "madre_id"   "padre_hog"  "padre_id"
[11] "disc_camin" "disc_ver"   "disc_brazo" "disc_apren" "disc_oir"
```

Para juntar bases usamos el comando “merge”

En “by” ponemos el id, correspondiente a la variable o variables que forman el id, entrecomillado. Cuando estamos mezclando bases del mismo nivel de análisis el id es igual en ambas bases. Cuando estamos incoporando información de bases de distinto nivel debemos escoger

En general ponemos el id de la base de mayor nivel. En este caso, sabemos que a una vivienda corresponde más de un hogar. Tal como revisamos nuestra documentación, sabemos que el id de la tabla “viviendas” es “folioviv”

```
  merge_data<- merge(viviendas, hogares, by="folioviv")
```

Revisemos la base creada

```r
names(merge_data)
```

```
  [1] "folioviv"    "tipo_viv"    "mat_pared"   "mat_techos"  "mat_pisos"
  [6] "antiguedad"  "antigua_ne"  "cocina"      "cocina_dor"  "cuart_dorm"
 [11] "num_cuarto"  "disp_agua"   "dotac_agua"  "excusado"    "uso_compar"
 [16] "sanit_agua"  "biodigest"   "bano_comp"   "bano_excus"  "bano_regad"
 [21] "drenaje"     "disp_elect"  "focos_inca"  "focos_ahor"  "combustible"
 [26] "estufa_chi"  "eli_basura"  "tenencia"    "renta"       "estim_pago"
 [31] "pago_viv"    "pago_mesp"   "tipo_adqui"  "viv_usada"   "tipo_finan"
 [36] "num_dueno1"  "hog_dueno1"  "num_dueno2"  "hog_dueno2"  "escrituras"
 [41] "lavadero"    "fregadero"   "regadera"    "tinaco_azo"  "cisterna"
 [46] "pileta"      "calent_sol"  "calent_gas"  "medidor_luz" "bomba_agua"
 [51] "tanque_gas"  "aire_acond"  "calefacc"    "tot_resid"   "tot_hom"
 [56] "tot_muj"     "tot_hog"     "ubica_geo"   "tam_loc"     "est_socio"
 [61] "est_dis"     "upm"         "factor"      "procaptar"   "foliohog"
 [66] "huespedes"   "huesp_come"  "num_trab_d"  "trab_come"   "acc_alim1"
 [71] "acc_alim2"   "acc_alim3"   "acc_alim4"   "acc_alim5"   "acc_alim6"
 [76] "acc_alim7"   "acc_alim8"   "acc_alim9"   "acc_alim10"  "acc_alim11"
 [81] "acc_alim12"  "acc_alim13"  "acc_alim14"  "acc_alim15"  "acc_alim16"
 [86] "alim17_1"    "alim17_2"    "alim17_3"    "alim17_4"    "alim17_5"
 [91] "alim17_6"    "alim17_7"    "alim17_8"    "alim17_9"    "alim17_10"
 [96] "alim17_11"   "alim17_12"   "acc_alim18"  "telefono"    "celular"
[101] "tv_paga"     "conex_inte"  "num_auto"    "anio_auto"   "num_van"
[106] "anio_van"    "num_pickup"  "anio_pickup" "num_moto"    "anio_moto"
[111] "num_bici"    "anio_bici"   "num_trici"   "anio_trici"  "num_carret"
[116] "anio_carret" "num_canoa"   "anio_canoa"  "num_otro"    "anio_otro"
[121] "num_ester"   "anio_ester"  "num_grab"    "anio_grab"   "num_radio"
[126] "anio_radio"  "num_tva"     "anio_tva"    "num_tvd"     "anio_tvd"
[131] "num_dvd"     "anio_dvd"    "num_video"   "anio_video"  "num_licua"
[136] "anio_licua"  "num_tosta"   "anio_tosta"  "num_micro"   "anio_micro"
[141] "num_refri"   "anio_refri"  "num_estuf"   "anio_estuf"  "num_lavad"
[146] "anio_lavad"  "num_planc"   "anio_planc"  "num_maqui"   "anio_maqui"
[151] "num_venti"   "anio_venti"  "num_aspir"   "anio_aspir"  "num_compu"
[156] "anio_compu"  "num_impre"   "anio_impre"  "num_juego"   "anio_juego"
[161] "esc_radio"   "er_aparato"  "er_celular"  "er_compu"    "er_aplicac"
[166] "er_tv"       "er_otro"     "recib_tvd"   "tsalud1_h"   "tsalud1_m"
[171] "habito_1"    "habito_2"    "habito_3"    "habito_4"    "habito_5"
[176] "habito_6"    "consumo"     "nr_viv"      "tarjeta"     "pagotarjet"
[181] "regalotar"   "regalodado"  "autocons"    "regalos"     "remunera"
[186] "transferen"  "parto_g"     "embarazo_g"  "negcua"      "est_alim"
[191] "est_trans"   "bene_licon"  "cond_licon"  "lts_licon"   "otros_lts"
[196] "diconsa"     "frec_dicon"  "cond_dicon"  "pago_dicon"  "otro_pago"
```

```r
dim(merge_data)
```

```
[1] 89006   200
```

Algunos elementos

(1) El orden de las variables corresponde al orden que pusimos las bases en las opciones.

(2) También vemos que las variables que se repetían en ambas bases se repiten en la nueva base, seguida de un un punto y una "x", para lo que proviene de la primera base y con una "y", lo que proviene de la segunda. R dejará las variables intactas y son coincidentes, en nuestro caso, porque las variables son iguales. R hace esto para precaver que por error tengamos alguna variable con un nombre igual y no sea la misma

5.3 Merge con id compuesto

Los identificadores pueden estar compuestos de más de una variable:

- Viviendas {viviendas} es "folioviv"

```
viviendas %>%
  janitor::get_dupes(folioviv)
```

```
No duplicate combinations found of: folioviv
```

```
# A tibble: 0 x 65
# ... with 65 variables: folioviv <chr>, dupe_count <int>, tipo_viv <chr>,
#   mat_pared <chr>, mat_techos <chr>, mat_pisos <chr>, antiguedad <dbl>,
#   antigua_ne <chr>, cocina <chr>, cocina_dor <chr>, cuart_dorm <dbl>,
#   num_cuarto <dbl>, disp_agua <chr>, dotac_agua <chr>, excusado <chr>,
#   uso_compar <chr>, sanit_agua <chr>, biodigest <chr>, bano_comp <dbl>,
#   bano_excus <dbl>, bano_regad <dbl>, drenaje <chr>, disp_elect <chr>,
#   focos_inca <dbl>, focos_ahor <dbl>, combustible <chr>, ...
```

- Hogares {hogares} es "folioviv", "foliohog"

```
hogares %>%
  janitor::get_dupes(c(folioviv, foliohog))
```

```
No duplicate combinations found of: folioviv, foliohog
```

```
# A tibble: 0 x 138
# ... with 138 variables: folioviv <chr>, foliohog <chr>, dupe_count <int>,
#   huespedes <dbl>, huesp_come <dbl>, num_trab_d <dbl>, trab_come <dbl>,
#   acc_alim1 <chr>, acc_alim2 <chr>, acc_alim3 <chr>, acc_alim4 <chr>,
#   acc_alim5 <chr>, acc_alim6 <chr>, acc_alim7 <chr>, acc_alim8 <chr>,
#   acc_alim9 <chr>, acc_alim10 <chr>, acc_alim11 <chr>, acc_alim12 <chr>,
#   acc_alim13 <chr>, acc_alim14 <chr>, acc_alim15 <chr>, acc_alim16 <chr>,
```

```
#   alim17_1 <dbl>, alim17_2 <dbl>, alim17_3 <dbl>, alim17_4 <dbl>, ...
```

- Poblacion {individuos} es "folioviv", "foliohog", "numren"

```
poblacion %>%
  janitor::get_dupes(c(folioviv, foliohog, numren))
```

```
No duplicate combinations found of: folioviv, foliohog, numren
```

```
# A tibble: 0 x 185
# ... with 185 variables: folioviv <chr>, foliohog <chr>, numren <chr>,
#   dupe_count <int>, parentesco <chr>, sexo <chr>, edad <dbl>,
#   madre_hog <chr>, madre_id <chr>, padre_hog <chr>, padre_id <chr>,
#   disc_camin <chr>, disc_ver <chr>, disc_brazo <chr>, disc_apren <chr>,
#   disc_oir <chr>, disc_vest <chr>, disc_habla <chr>, disc_acti <chr>,
#   cau_camin <chr>, cau_ver <chr>, cau_brazo <chr>, cau_apren <chr>,
#   cau_oir <chr>, cau_vest <chr>, cau_habla <chr>, cau_acti <chr>, ...
```

Esto significa que tenemos un id compuesto. No es una sola variable. Para esto modificamos ligeramente cómo ponemos el "by", pero siempre eligiendo el id de la base de mayor nivel. (Tené cuidado con los paréntesis)

```
merge_data2<- merge(hogares, poblacion, by=c("folioviv", "foliohog"))
dim(merge_data2)
```

```
[1] 315743    319
```

Revisemos la base

```
merge_data2 %>%
  tail()
```

```
        folioviv foliohog huespedes huesp_come num_trab_d trab_come acc_alim1
315738 3260770717        1         0         NA          0        NA         2
315739 3260770717        1         0         NA          0        NA         2
315740 3260770717        1         0         NA          0        NA         2
315741 3260770718        1         0         NA          0        NA         2
315742 3260770718        1         0         NA          0        NA         2
315743 3260770718        1         0         NA          0        NA         2
       acc_alim2 acc_alim3 acc_alim4 acc_alim5 acc_alim6 acc_alim7 acc_alim8
315738         2         2         2         2         2
315739         2         2         2         2         2
315740         2         2         2         2         2
315741         2         2         2         2         2
315742         2         2         2         2         2
315743         2         2         2         2         2
       acc_alim9 acc_alim10 acc_alim11 acc_alim12 acc_alim13 acc_alim14
```

```
315738
315739
315740
315741
315742
315743
       acc_alim15 acc_alim16 alim17_1 alim17_2 alim17_3 alim17_4 alim17_5
315738                              7        2        2        3        2
315739                              7        2        2        3        2
315740                              7        2        2        3        2
315741                              7        2        3        4        2
315742                              7        2        3        4        2
315743                              7        2        3        4        2
       alim17_6 alim17_7 alim17_8 alim17_9 alim17_10 alim17_11 alim17_12
315738        7        0        7        2         7         2         2
315739        7        0        7        2         7         2         2
315740        7        0        7        2         7         2         2
315741        4        0        7        3         7         3         7
315742        4        0        7        3         7         3         7
315743        4        0        7        3         7         3         7
       acc_alim18 telefono celular tv_paga conex_inte num_auto anio_auto
315738          1        2       1       2          1        0
315739          1        2       1       2          1        0
315740          1        2       1       2          1        0
315741          1        2       1       2          1        1        19
315742          1        2       1       2          1        1        19
315743          1        2       1       2          1        1        19
       num_van anio_van num_pickup anio_pickup num_moto anio_moto num_bici
315738       0                   1          97        0                   0
315739       0                   1          97        0                   0
315740       0                   1          97        0                   0
315741       0                   0                    0                   0
315742       0                   0                    0                   0
315743       0                   0                    0                   0
       anio_bici num_trici anio_trici num_carret anio_carret num_canoa
315738                   0                     0                     0
315739                   0                     0                     0
315740                   0                     0                     0
315741                   0                     0                     0
315742                   0                     0                     0
315743                   0                     0                     0
       anio_canoa num_otro anio_otro num_ester anio_ester num_grab anio_grab
315738                   0                   1         00        0
315739                   0                   1         00        0
315740                   0                   1         00        0
315741                   0                   0                   0
```

```
315742                  0                  0                  0
315743                  0                  0                  0
       num_radio anio_radio num_tva anio_tva num_tvd anio_tvd num_dvd anio_dvd
315738         0                  0                1       10       1       16
315739         0                  0                1       10       1       16
315740         0                  0                1       10       1       16
315741         0                  1       05       1       16       0
315742         0                  1       05       1       16       0
315743         0                  1       05       1       16       0
       num_video anio_video num_licua anio_licua num_tosta anio_tosta num_micro
315738         0                    1         17         0                    1
315739         0                    1         17         0                    1
315740         0                    1         17         0                    1
315741         0                    1         17         0                    0
315742         0                    1         17         0                    0
315743         0                    1         17         0                    0
       anio_micro num_refri anio_refri num_estuf anio_estuf num_lavad
315738         00         1         19         1         05         1
315739         00         1         19         1         05         1
315740         00         1         19         1         05         1
315741                    1         06         1         10         1
315742                    1         06         1         10         1
315743                    1         06         1         10         1
       anio_lavad num_planc anio_planc num_maqui anio_maqui num_venti
315738         00         1         15         0                    0
315739         00         1         15         0                    0
315740         00         1         15         0                    0
315741         10         1         05         0                    0
315742         10         1         05         0                    0
315743         10         1         05         0                    0
       anio_venti num_aspir anio_aspir num_compu anio_compu num_impre
315738                    0                    0                    0
315739                    0                    0                    0
315740                    0                    0                    0
315741                    0                    0                    0
315742                    0                    0                    0
315743                    0                    0                    0
       anio_impre num_juego anio_juego esc_radio er_aparato er_celular er_compu
315738                    0                    1          1
315739                    0                    1          1
315740                    0                    1          1
315741                    0                    2
315742                    0                    2
315743                    0                    2
       er_aplicac er_tv er_otro recib_tvd tsalud1_h tsalud1_m habito_1 habito_2
315738                                          2         0
```

```
315739                                          2         0
315740                                          2         0
315741                                          1         0
315742                                          1         0
315743                                          1         0
       habito_3 habito_4 habito_5 habito_6 consumo nr_viv tarjeta pagotarjet
315738        3                                 1              2          2
315739        3                                 1              2          2
315740        3                                 1              2          2
315741        3                 5               2              2          2
315742        3                 5               2              2          2
315743        3                 5               2              2          2
       regalotar regalodado autocons regalos remunera transferen parto_g
315738         2          2        2       1        2          2       2
315739         2          2        2       1        2          2       2
315740         2          2        2       1        2          2       2
315741         2          2        2       1        2          1       2
315742         2          2        2       1        2          1       2
315743         2          2        2       1        2          1       2
       embarazo_g negcua est_alim est_trans bene_licon cond_licon lts_licon
315738          2      2     4000         0          2
315739          2      2     4000         0          2
315740          2      2     4000         0          2
315741          2      2     3000         0          1          2         2
315742          2      2     3000         0          1          2         2
315743          2      2     3000         0          1          2         2
       otros_lts diconsa frec_dicon cond_dicon pago_dicon otro_pago numren
315738        NA       1          2          2          3        NA     05
315739        NA       1          2          2          3        NA     06
315740        NA       1          2          2          3        NA     07
315741        NA       1          2          2          3        NA     01
315742        NA       1          2          2          3        NA     02
315743        NA       1          2          2          3        NA     03
       parentesco sexo edad madre_hog madre_id padre_hog padre_id disc_camin
315738        301    2   14         1       02         1       01          4
315739        301    1   12         1       02         1       01          4
315740        301    2    5         1       02         1       01          4
315741        101    2   64         2                  2                   4
315742        301    1   33         1       01         2                   4
315743        301    1   31         1       01         2                   4
       disc_ver disc_brazo disc_apren disc_oir disc_vest disc_habla disc_acti
315738        4          4          4        4         4          4         4
315739        4          4          4        4         4          4         4
315740        4          4          4        4         4          4         4
315741        4          3          4        4         4          4         3
315742        4          4          4        4         4          4         4
```

```
315743       4          4          4        4         4          4         4
       cau_camin cau_ver cau_brazo cau_apren cau_oir cau_vest cau_habla
315738
315739
315740
315741                          2
315742
315743
       cau_acti hablaind lenguaind hablaesp comprenind etnia alfabetism
315738                2                               2     2          1
315739                2                               2     2          1
315740                2                               2     2          2
315741        1       2                               2     2          1
315742                2                               2     2          1
315743                2                               2     2          1
       asis_esc nivel grado tipoesc tiene_b otorg_b forma_b tiene_c otorg_c
315738        1    07     3       1       2
315739        1    07     1       1       2
315740        1    01     3       1       2
315741        2
315742        2
315743        2
       forma_c nivelaprob gradoaprob antec_esc residencia edo_conyug pareja_hog
315738                3           2                  32          6
315739                2           6                  32          6
315740                1           2                  32
315741                3           3                  32          5
315742                3           3                  32          3
315743                7           5         3        32          6
       conyuge_id segsoc ss_aa ss_mm redsoc_1 redsoc_2 redsoc_3 redsoc_4
315738                2    NA    NA        3        3        3        3
315739                2    NA    NA        3        3        2        3
315740                     NA    NA
315741                2    NA    NA        2        3        2        2
315742                1     2     0        2        2        2        2
315743                1     3     0        2        2        2        2
       redsoc_5 redsoc_6 hor_1 min_1 usotiempo1 hor_2 min_2 usotiempo2 hor_3
315738        3        3    NA    NA          9    30     0                NA
315739        2        3    NA    NA          9    15     0                 1
315740                 3    NA    NA               NA    NA               NA
315741        3        3    NA    NA          9    NA    NA           9     3
315742        2        2    81     0               NA    NA           9    NA
315743        2        2    35     0               NA    NA           9    NA
       min_3 usotiempo3 hor_4 min_4 usotiempo4 hor_5 min_5 usotiempo5 hor_6
315738    NA          9     3     0                NA    NA           9     3
315739     0          9     2     0                NA    NA           9     2
```

```
315740    NA               NA    NA               NA    NA               NA
315741     0               NA    NA          9    NA    NA          9    56
315742    NA          9    NA    NA          9    NA    NA          9    NA
315743    NA          9    NA    NA          9    NA    NA          9    12
       min_6 usotiempo6 hor_7 min_7 usotiempo7 hor_8 min_8 usotiempo8
315738     0               NA    NA          9     7    00
315739     0                1     0              14    00
315740    NA               NA    NA              NA
315741     0               NA    NA          9    14    00
315742    NA          9    NA    NA          9    17    00
315743     0               NA    NA          9    21    00
       pop_insabi atemed inst_1 inst_2 inst_3 inst_4 inst_5 inst_6 inscr_1
315738          1      2
315739          1      2
315740          1      2
315741          2      1             2
315742          2      2
315743          2      2
       inscr_2 inscr_3 inscr_4 inscr_5 inscr_6 inscr_7 inscr_8 prob_anio
315738                                                              2019
315739                                                              2019
315740                                                              2020
315741                                                 7            2020
315742                                                              2018
315743                                                              2016
       prob_mes prob_sal aten_sal servmed_1 servmed_2 servmed_3 servmed_4
315738       08        1        1        01
315739       12        1        1        01
315740       04        1        1        01
315741       08        1        1        01
315742       08        2
315743       01        1        1        01
       servmed_5 servmed_6 servmed_7 servmed_8 servmed_9 servmed_10 servmed_11
315738
315739
315740
315741
315742
315743
       servmed_12 hh_lug mm_lug hh_esp mm_esp pagoaten_1 pagoaten_2 pagoaten_3
315738                 0     15      0     30
315739                 0     15      0     30
315740                 0     15      0     30
315741                 0     20      0     10
315742                NA     NA     NA     NA
315743                 0     50      0     30          1
```

```
       pagoaten_4 pagoaten_5 pagoaten_6 pagoaten_7 noatenc_1 noatenc_2
315738                                             7
315739                                             7
315740                                             7
315741                                             7
315742
315743
       noatenc_3 noatenc_4 noatenc_5 noatenc_6 noatenc_7 noatenc_8 noatenc_9
315738
315739
315740
315741
315742
315743
       noatenc_10 noatenc_11 noatenc_12 noatenc_13 noatenc_14 noatenc_15
315738
315739
315740
315741
315742
315743
       noatenc_16 norecib_1 norecib_2 norecib_3 norecib_4 norecib_5 norecib_6
315738
315739
315740
315741
315742         16
315743
       norecib_7 norecib_8 norecib_9 norecib_10 norecib_11 norecib_12 razon_1
315738
315739
315740
315741
315742
315743
       razon_2 razon_3 razon_4 razon_5 razon_6 razon_7 razon_8 razon_9 razon_10
315738
315739
315740
315741
315742
315743
       razon_11 diabetes pres_alta peso segvol_1 segvol_2 segvol_3 segvol_4
315738                1          1    1
315739                2          1    1
315740                                 1
```

```
315741                1         1    1
315742                2         2    2
315743                2         2    2
       segvol_5 segvol_6 segvol_7 hijos_viv hijos_mue hijos_sob trabajo_mp
315738                6                 0        NA        NA          2
315739                6                NA        NA        NA          2
315740                                 NA        NA        NA
315741                6                10         2         8          2
315742                6                NA        NA        NA          1
315743                6                NA        NA        NA          1
       motivo_aus act_pnea1 act_pnea2 num_trabaj c_futuro ct_futuro
315738                  4
315739                  4
315740
315741                  3
315742                                        1
315743                                        1
```

5.4 Bases de distinto tamaño

Hasta ahorita hemos hecho merge que son de unidades de distinto nivel y son incluyentes. A veces tenemos bases de datos que son de distinto tamaño y del mismo nivel. A veces las dos aportan casos y a veces aportan variables, y a veces, las dos aportan las dos cosas.

Vamos a revisar qué pasaría si quisiéramos incorporar la información los ingresos

```
rm(merge_data, merge_data2) # botamos otros ejemplos
ingresos<- haven::read_dta("datos/ingresos2020.dta")
```

Esta base tiene otro ID

- Ingresos {clave de ingreso} es "folioviv", "foliohog", "numren", clave

```
ingresos %>%
  janitor::get_dupes(c(folioviv, foliohog, numren, clave))
```

```
No duplicate combinations found of: folioviv, foliohog, numren, clave
```

```
# A tibble: 0 x 18
# ... with 18 variables: folioviv <chr>, foliohog <chr>, numren <chr>,
#   clave <chr>, dupe_count <int>, mes_1 <chr>, mes_2 <chr>, mes_3 <chr>,
#   mes_4 <chr>, mes_5 <chr>, mes_6 <chr>, ing_1 <dbl>, ing_2 <dbl>,
#   ing_3 <dbl>, ing_4 <dbl>, ing_5 <dbl>, ing_6 <dbl>, ing_tri <dbl>
```

¿Cuántas claves de ingreso hay?

```r
ingresos %>%
  tabyl(clave)
```

clave	n	percent
P001	99992	2.532007e-01
P002	2619	6.631857e-03
P003	6260	1.585163e-02
P004	5990	1.516794e-02
P005	3098	7.844786e-03
P006	7113	1.801161e-02
P007	6754	1.710254e-02
P008	16600	4.203468e-02
P009	48715	1.233566e-01
P011	258	6.533101e-04
P012	3282	8.310712e-03
P013	3	7.596629e-06
P014	5056	1.280285e-02
P015	67	1.696581e-04
P016	643	1.628211e-03
P018	11	2.785431e-05
P019	239	6.051981e-04
P021	1567	3.967973e-03
P022	19138	4.846143e-02
P023	872	2.208087e-03
P024	2541	6.434345e-03
P025	188	4.760554e-04
P026	233	5.900049e-04
P027	292	7.394052e-04
P028	116	2.937363e-04
P029	12	3.038652e-05
P030	5	1.266105e-05
P031	133	3.367839e-04
P032	16768	4.246009e-02
P033	679	1.719370e-03
P034	44	1.114172e-04
P035	152	3.848959e-04
P036	1538	3.894539e-03
P037	522	1.321813e-03
P038	3334	8.442387e-03
P039	603	1.526922e-03
P040	24129	6.109969e-02
P041	6323	1.601116e-02
P043	2275	5.760777e-03
P045	894	2.263795e-03
P048	1671	4.231322e-03

```
P049  2097 5.310044e-03
P050    36 9.115955e-05
P051 14683 3.718044e-02
P052  1326 3.357710e-03
P053  6920 1.752289e-02
P054   492 1.245847e-03
P055     8 2.025768e-05
P056     6 1.519326e-05
P057    78 1.975124e-04
P058   426 1.078721e-03
P059    48 1.215461e-04
P060   182 4.608622e-04
P061   338 8.558869e-04
P062  1129 2.858865e-03
P063  4868 1.232680e-02
P064    48 1.215461e-04
P065    62 1.569970e-04
P066   230 5.824082e-04
P067  1917 4.854246e-03
P068  5383 1.363088e-02
P069  7154 1.811543e-02
P070  7854 1.988798e-02
P071  5130 1.299024e-02
P072  4537 1.148864e-02
P073   193 4.887165e-04
P074   229 5.798760e-04
P075   892 2.258731e-03
P076  1176 2.977879e-03
P077  1460 3.697026e-03
P078  1290 3.266551e-03
P079  1280 3.241228e-03
P080   131 3.317195e-04
P081    61 1.544648e-04
P101  6411 1.623400e-02
P102  6044 1.530468e-02
P103   719 1.820659e-03
P104 17014 4.308302e-02
P105  1413 3.578012e-03
P106   200 5.064419e-04
P107    67 1.696581e-04
P108   651 1.648469e-03
```

```
ingresos_sueldos<-ingresos %>%
    filter(clave=="P001")
dim(ingresos_sueldos)
```

```
[1] 99992    17
```

Vamos a hacer el primer tipo de merge

```
merge_data3<-merge(poblacion, ingresos_sueldos, by=c("folioviv", "foliohog", "numren"))
dim(merge_data3)
```

```
[1] 99992   198
```

¡La base nueva no tiene a todas las observaciones, solo la que tiene en la base más pequeña! Tenemos sólo 99,9992 individuos.

5.5 Cuatro formas de hacer un fusionado

En realidad hay cuatro formas de hacer un "merge"

5.5.1 Casos en ambas bases

Por *default*, el comando tiene activado la opción "all = FALSE", que nos deja los datos de ambas bases comunes. (tipo una intersección)

```
merge_data3<-merge(poblacion,
                   ingresos_sueldos,
                   by=c("folioviv", "foliohog", "numren"),
                   all = F)
dim(merge_data3)
```

```
[1] 99992   198
```

5.5.2 Todos los casos

Si cambiamos la opción "all = TRUE", que nos deja los datos comunes a ambas bases. (como una unión)

```
merge_data3<-merge(poblacion,
                   ingresos_sueldos,
                   by=c("folioviv", "foliohog", "numren"),
                   all = T)
dim(merge_data3)
```

```
[1] 315743    198
```

5.5.3 Casos en la base 1

Si queremos quedarnos con todos los datos que hay en la primera base, x, vamos a usar a opción all.x = TRUE.

```
merge_data3<-merge(poblacion,
                   ingresos_sueldos,
                   by=c("folioviv", "foliohog", "numren"),
                   all.x  = TRUE)
dim(merge_data3)
```

```
[1] 315743    198
```

5.5.4 Casos de la base 2

Notamos que hoy sí tenemos los datos de toda la población y hay missings en las variables aportadas por la base de trabajo

Si queremos lo contrario, quedarnos con los datos aportados por la segunda base, y, vamos a usar la opción all.y=TRUE

```
merge_data3<-merge(poblacion,
                   ingresos_sueldos,
                   by=c("folioviv", "foliohog", "numren"),
                   all.y  = TRUE)
dim(merge_data3)
```

```
[1] 99992   198
```

5.6 Las cuatro formas en `dplyr`

El caso 1:

```
merge_data3<-dplyr::inner_join(poblacion,
                               ingresos_sueldos,
                               by=c("folioviv", "foliohog", "numren"))
dim(merge_data3)
```

```
[1] 99992   198
```

El caso 2:

```
merge_data3<-dplyr::full_join(poblacion,
                              ingresos_sueldos,
                              by=c("folioviv", "foliohog", "numren"))
dim(merge_data3)
```

```
[1] 315743    198
```

El caso 3:

```
merge_data3<-dplyr::left_join(poblacion,
                              ingresos_sueldos,
                              by=c("folioviv", "foliohog", "numren"))
dim(merge_data3)
```

```
[1] 315743    198
```

El caso 4:

```
merge_data3<-dplyr::right_join(poblacion,
                               ingresos_sueldos,
                               by=c("folioviv", "foliohog", "numren"))
dim(merge_data3)
```

```
[1] 99992   198
```

También se puede usar con pipes, cualquier opción de dplyr

```
merge_data3<-poblacion %>% # pongo el conjunto que será la "izquierda
  dplyr::right_join(ingresos_sueldos,
                    by=c("folioviv", "foliohog", "numren"))
dim(merge_data3)
```

```
[1] 99992   198
```

5.7 Práctica

- Pegue a la última base la información de los hogares y las viviendas.

Chapter 6

Funciones, condicionales, bucles y mapeos

6.1 Paquetes

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

```
Loading required package: pacman
```

```
pacman::p_load(tidyverse,
               readxl,
               writexl,
               haven,
               sjlabelled,
               janitor,
               magrittr,
               broom # para limpiar resultados de modelos
)
```

6.2 Datos

```
concentrado2020 <- read_dta("datos/concentrado2020.dta") %>%
  mutate(across(c(sexo_jefe, clase_hog, educa_jefe), as.numeric)) %>% # ojo aquí
  set_labels(sexo_jefe, labels=c("Hombre", "Mujer")) %>%
  set_labels(clase_hog, labels=c("unipersonal","nuclear", "ampliado",
                                 "compuesto","corresidente"))   %>%
```

```
  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")) %>%
  mutate(ent=stringr::str_sub(ubica_geo, start = 1, end = 2)) %>%
  mutate(mun=stringr::str_sub(ubica_geo, start = 3, end = 5))
```

6.3 Mi primera función

Unos de los elementos más poderosos de R es hacer nuestra propias funciones.

La lógica de las funciones es la siguiente:

```
nombre_de_funcion(argumento1, argumento2, ...) {
  operaciones
  salida
}
```

Para ello haremos una función sencilla. Para sumarle un valor un 1

```
mi_funcion<-function(x) {
    resultado<-x+1
    return(resultado)
}

mi_funcion(5)
```

```
[1] 6
```

Vamos a agregar un argumento, podemos agregar un segundo número en lugar de 1

```
mi_funcion<-function(x, a) {
    resultado<-x+a
    return(resultado)
}
```

```
mi_funcion(x=5, a=6)
```

```
[1] 11
```

Los argumentos no necesariamente deben ser un sólo valor

```
mi_funcion(x=1:5, a=6:10)
```

```
[1]  7  9 11 13 15
```

E incluso podríamos llamar variables de nuestra base de concentrado

```
resultado_mi_funcion<-mi_funcion(x=concentrado2020$frutas, a=concentrado2020$azucar)
```

6.4 Una función para hacer edades

Primero un poquito de `pretty()` {base}, es un comando que calcula una secuencia de aproximadamente n+1 valores 'redondos' igualmente espaciados que cubran el rango de los valores en x. Los valores se eligen para que sean 1, 2 o 5 veces una potencia de 10.

```
cortar <- function(x) {
  cut(x,
      breaks = pretty(x),
      right = TRUE,
      include.lowest = TRUE)
}
```

```
#cortar(concentrado2020$edad_jefe)
```

Podemos utilizarla junto con mutate

```
concentrado2020 %>%
  mutate(eda_cut=cortar(edad_jefe)) %>%
  tabyl(eda_cut)
```

```
 eda_cut     n      percent
  [0,20]   539 0.0060557715
 (20,40] 24896 0.2797114801
 (40,60] 38634 0.4340606251
 (60,80] 21301 0.2393209447
(80,100]  3627 0.0407500618
(100,120]    9 0.0001011168
```

6.5 Bucles

6.5.1 `for`

Supongamos que quisiéramos repetir una operación a lo largo de una secuencia, se puede realizar

```
for (i in secuencia) {
  operación 1
  operación 2
  ...
  operación final
}
```

Por ejemplo si quisiéramos que por cada entidad federativa se imprimiera en pantalla el promedio de la edad de los jefes entrevistados

```
unique(concentrado2020$ent) # Nos dan los valores únicos de un vector
```

```
 [1] "01" "02" "03" "04" "05" "06" "07" "08" "09" "10" "11" "12" "13" "14" "15"
[16] "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27" "28" "29" "30"
[31] "31" "32"
```

```
estados<-unique(concentrado2020$ent)
```

Hoy haremos nuestro bucle con "for"

```
for(i in estados){

  x<-concentrado2020 %>%
    filter(ent==i) %>% # aquí ocupamos nuestro indice
    summarise(media=mean(ing_cor))

  assign(paste0("ingreso",i), x)
}
```

Vamos a botar estos objetos

```
rm(list = ls(, pattern = "ingreso"))
rm(x)
```

6.5.2 `while()`

También tenemos `while()`

```
while (expresión a probar) {
  Operaciones
```

}

```
# variable que se cambia
numero = 1

# variable donde se calcula la meida
sum = 0

# Calcular la suma consecutiva hasta que llegue a 30

while(numero <= 30) {

  # calculate sum
  sum = sum + numero

  # increment number by 1
  numero = numero + 1

}
```

6.6 Condicionales

Las operaciones están supeditadas a los elementos que cumplan una condición

```
if (condicion) {
  operación 1
  operación 2
  ...
  operación final
}
```

Supongamos tenemos dos valores

```
a<-45 # un vector entero
b<-5000 #numeros aleatorios que siguen una normal
```

Veamos cómo podemos hacer un condicional muy simple

```
if(a>18){
  print(b)
}
```

```
[1] 5000
```

También se puede combinar con else

```
if(a>18){
  print("Mayor que 18")
} else {
  print("No cumple")
}
```

```
[1] "Mayor que 18"
```

Estos elementos funcionan cuando se programan procesos. Son útiles para cuando computamos modelos y se busca cierto nivel de tolerancia o se hacen procesos sucesivos.

6.7 purrr::map()

Dentro de tidyverse existe el paquete `{purrr}`, es un paquete que tiene muchas funcionalidades parecidas a los `for`.

Por ejemplo y siguiendo los ejemplos del for()

```
1:10 %>%
  map(~.x+1)
```

```
[[1]]
[1] 2

[[2]]
[1] 3

[[3]]
[1] 4

[[4]]
[1] 5

[[5]]
[1] 6

[[6]]
[1] 7

[[7]]
[1] 8

[[8]]
[1] 9
```

```
[[9]]
[1] 10

[[10]]
[1] 11
```

Si guardamos esto en un objeto, vemos que nos da como resultado una lista

```
map1<-1:10 %>%
  map(~.x+1)

class(map1)
```

```
[1] "list"
```

Compliquemos esto un poquito más...

```
names(concentrado2020) %>%
  map_chr(~str_detect(.x,"ing")) # ojo a veces tenemos que hacer explícito el tipo de map
```

```
  [1] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
 [10] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
 [19] "FALSE" "FALSE" "TRUE"  "FALSE" "TRUE"  "TRUE"  "FALSE" "FALSE" "FALSE"
 [28] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
 [37] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
 [46] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
 [55] "FALSE" "TRUE"  "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
 [64] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
 [73] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
 [82] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
 [91] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
[100] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
[109] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
[118] "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE" "FALSE"
[127] "FALSE" "FALSE"
```

Por ejemplo, quizás queremos identificar esta base de datos con 2020

```
names(concentrado2020) %>%
  map_chr(~paste0(.x,"2020"))
```

```
 [1] "folioviv2020"   "foliohog2020"   "ubica_geo2020"  "tam_loc2020"
 [5] "est_socio2020"  "est_dis2020"    "upm2020"        "factor2020"
 [9] "clase_hog2020"  "sexo_jefe2020"  "edad_jefe2020"  "educa_jefe2020"
[13] "tot_integ2020"  "hombres2020"    "mujeres2020"    "mayores2020"
```

```
 [17] "menores2020"    "p12_642020"     "p65mas2020"     "ocupados2020"
 [21] "percep_ing2020" "perc_ocupa2020" "ing_cor2020"    "ingtrab2020"
 [25] "trabajo2020"    "sueldos2020"    "horas_extr2020" "comisiones2020"
 [29] "aguinaldo2020"  "indemtrab2020"  "otra_rem2020"   "remu_espec2020"
 [33] "negocio2020"    "noagrop2020"    "industria2020"  "comercio2020"
 [37] "servicios2020"  "agrope2020"     "agricolas2020"  "pecuarios2020"
 [41] "reproducc2020"  "pesca2020"      "otros_trab2020" "rentas2020"
 [45] "utilidad2020"   "arrenda2020"    "transfer2020"   "jubilacion2020"
 [49] "becas2020"      "donativos2020"  "remesas2020"    "bene_gob2020"
 [53] "transf_hog2020" "trans_inst2020" "estim_alqu2020" "otros_ing2020"
 [57] "gasto_mon2020"  "alimentos2020"  "ali_dentro2020" "cereales2020"
 [61] "carnes2020"     "pescado2020"    "leche2020"      "huevo2020"
 [65] "aceites2020"    "tuberculo2020"  "verduras2020"   "frutas2020"
 [69] "azucar2020"     "cafe2020"       "especias2020"   "otros_alim2020"
 [73] "bebidas2020"    "ali_fuera2020"  "tabaco2020"     "vesti_calz2020"
 [77] "vestido2020"    "calzado2020"    "vivienda2020"   "alquiler2020"
 [81] "pred_cons2020"  "agua2020"       "energia2020"    "limpieza2020"
 [85] "cuidados2020"   "utensilios2020" "enseres2020"    "salud2020"
 [89] "atenc_ambu2020" "hospital2020"   "medicinas2020"  "transporte2020"
 [93] "publico2020"    "foraneo2020"    "adqui_vehi2020" "mantenim2020"
 [97] "refaccion2020"  "combus2020"     "comunica2020"   "educa_espa2020"
[101] "educacion2020"  "esparci2020"    "paq_turist2020" "personales2020"
[105] "cuida_pers2020" "acces_pers2020" "otros_gas2020"  "transf_gas2020"
[109] "percep_tot2020" "retiro_inv2020" "prestamos2020"  "otras_perc2020"
[113] "ero_nm_viv2020" "ero_nm_hog2020" "erogac_tot2020" "cuota_viv2020"
[117] "mater_serv2020" "material2020"   "servicio2020"   "deposito2020"
[121] "prest_terc2020" "pago_tarje2020" "deudas2020"     "balance2020"
[125] "otras_erog2020" "smg2020"        "ent2020"        "mun2020"
```

Con estos nuevos nombres podríamos rápidamente volver a declarar nuestros nombres de la base y todos tienen sufijos.

Combinación con el comando `split()`

```
concentrado2020 %>%
  split(.$ent) %>%
  map(~ mean(.$ing_cor))
```

```
$`01`
[1] 55597.85

$`02`
[1] 62025.77

$`03`
[1] 61035.48
```

```
$`04`
[1] 46077.36

$`05`
[1] 52229.66

$`06`
[1] 51344.63

$`07`
[1] 29010.53

$`08`
[1] 57482.27

$`09`
[1] 60073.04

$`10`
[1] 46701.09

$`11`
[1] 43968.91

$`12`
[1] 31403.23

$`13`
[1] 39114.3

$`14`
[1] 54199.69

$`15`
[1] 42418.68

$`16`
[1] 45058.28

$`17`
[1] 40308.1

$`18`
[1] 52299.53
```

```
$`19`
[1] 57441.94

$`20`
[1] 33983.13

$`21`
[1] 37504.67

$`22`
[1] 53929.64

$`23`
[1] 42746.02

$`24`
[1] 43422.68

$`25`
[1] 57210.55

$`26`
[1] 56762.87

$`27`
[1] 40997.84

$`28`
[1] 46989.76

$`29`
[1] 37484.91

$`30`
[1] 32470.97

$`31`
[1] 39977.85

$`32`
[1] 43811.06
```

Nos da una lista de valores… si hacemos algo más complejo

```
concentrado2020 %>%
  split(.$ent) %>%
  map(~ mean(.$ing_cor)) %>%
  map_dfr(~as.data.frame(.x))
```

```
         .x
1  55597.85
2  62025.77
3  61035.48
4  46077.36
5  52229.66
6  51344.63
7  29010.53
8  57482.27
9  60073.04
10 46701.09
11 43968.91
12 31403.23
13 39114.30
14 54199.69
15 42418.68
16 45058.28
17 40308.10
18 52299.53
19 57441.94
20 33983.13
21 37504.67
22 53929.64
23 42746.02
24 43422.68
25 57210.55
26 56762.87
27 40997.84
28 46989.76
29 37484.91
30 32470.97
31 39977.85
32 43811.06
```

```
concentrado2020 %>%
  split(.$ent) %>%
  map(~ mean(.$ing_cor)) %>%
  map_dfc(~as.data.frame(.x)) %>%
  clean_names()
```

```
New names:
* `.x` -> `.x...1`
* `.x` -> `.x...2`
* `.x` -> `.x...3`
* `.x` -> `.x...4`
* `.x` -> `.x...5`
* `.x` -> `.x...6`
* `.x` -> `.x...7`
* `.x` -> `.x...8`
* `.x` -> `.x...9`
* `.x` -> `.x...10`
* `.x` -> `.x...11`
* `.x` -> `.x...12`
* `.x` -> `.x...13`
* `.x` -> `.x...14`
* `.x` -> `.x...15`
* `.x` -> `.x...16`
* `.x` -> `.x...17`
* `.x` -> `.x...18`
* `.x` -> `.x...19`
* `.x` -> `.x...20`
* `.x` -> `.x...21`
* `.x` -> `.x...22`
* `.x` -> `.x...23`
* `.x` -> `.x...24`
* `.x` -> `.x...25`
* `.x` -> `.x...26`
* `.x` -> `.x...27`
* `.x` -> `.x...28`
* `.x` -> `.x...29`
* `.x` -> `.x...30`
* `.x` -> `.x...31`
* `.x` -> `.x...32`
```

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	55597.85	62025.77	61035.48	46077.36	52229.66	51344.63	29010.53	57482.27

	x_9	x_10	x_11	x_12	x_13	x_14	x_15	x_16
1	60073.04	46701.09	43968.91	31403.23	39114.3	54199.69	42418.68	45058.28

	x_17	x_18	x_19	x_20	x_21	x_22	x_23	x_24
1	40308.1	52299.53	57441.94	33983.13	37504.67	53929.64	42746.02	43422.68

	x_25	x_26	x_27	x_28	x_29	x_30	x_31	x_32
1	57210.55	56762.87	40997.84	46989.76	37484.91	32470.97	39977.85	43811.06

6.8 Combinando funciones con purr::map

```
mi_funcion_summ<-function(x){
  mu<-mean(x, na.rm=T)
  me<-median(x,na.rm=T)
  sd<-sd(x,na.rm=T)
  total<-as.data.frame(cbind(mu,me,sd))
  return(total)
}

mi_funcion_summ(concentrado2020$ing_cor)
```

```
        mu       me       sd
1 47838.49 35172.01 71276.03
```

```
concentrado2020 %>%
  split(.$ent) %>%
  map(~ mi_funcion_summ(.$ing_cor))
```

```
$`01`
        mu       me       sd
1 55597.85 43010.77 47117.57

$`02`
        mu       me       sd
1 62025.77 45830.24 88387.62

$`03`
        mu       me       sd
1 61035.48 46770.46 53076.37

$`04`
        mu       me       sd
1 46077.36 32571.64 44224.74

$`05`
        mu       me       sd
1 52229.66 40902.97 43428.09

$`06`
        mu       me       sd
1 51344.63 40101.63 45501.34

$`07`
        mu      me       sd
```

```
1 29010.53 20903.2 28774.17

$`08`
        mu      me       sd
1 57482.27 38987.7 192053.2

$`09`
        mu       me       sd
1 60073.04 44900.21 57306.15

$`10`
        mu       me       sd
1 46701.09 34080.83 129188.6

$`11`
        mu       me       sd
1 43968.91 34721.14 34355.33

$`12`
        mu       me       sd
1 31403.23 22612.42 29169.38

$`13`
       mu       me       sd
1 39114.3 30099.82 32567.11

$`14`
        mu       me       sd
1 54199.69 41576.07 65377.82

$`15`
        mu       me       sd
1 42418.68 32016.39 53796.92

$`16`
        mu       me      sd
1 45058.28 34874.98 42919.1

$`17`
       mu       me       sd
1 40308.1 30620.45 37761.98

$`18`
        mu       me       sd
1 52299.53 41207.48 47639.84
```

```
$`19`
        mu       me       sd
1 57441.94 41690.93 82230.47

$`20`
        mu       me       sd
1 33983.13 23870.17 35229.34

$`21`
        mu      me       sd
1 37504.67 27817.3 43038.53

$`22`
        mu       me       sd
1 53929.64 41330.98 50918.38

$`23`
        mu       me       sd
1 42746.02 31730.53 50819.97

$`24`
        mu       me       sd
1 43422.68 31982.96 44543.01

$`25`
        mu       me       sd
1 57210.55 44418.12 49915.59

$`26`
        mu       me       sd
1 56762.87 41258.91 58032.84

$`27`
        mu       me       sd
1 40997.84 28915.12 40031.89

$`28`
        mu       me       sd
1 46989.76 34504.18 71032.01

$`29`
        mu       me       sd
1 37484.91 28347.77 31704.41

$`30`
        mu       me       sd
```

```
1 32470.97 24270.49 28210.82

$`31`
        mu       me       sd
1 39977.85 28836.27 48809.08

$`32`
        mu       me       sd
1 43811.06 30948.86 87282.28
```

Lo interesante es que podemos hacer elementos más complejos

```
concentrado2020 %>%
  split(.$ent) %>%
  map(~ lm(ing_cor ~ edad_jefe, data = .))
```

```
$`01`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    50487.3        102.3


$`02`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    50224.7        242.9


$`03`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    47835.8        272.7
```

```
$`04`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   43159.93        58.73


$`05`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   54785.09       -49.81


$`06`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   56096.00       -91.26


$`07`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    20136.7        176.6


$`08`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
```

```
(Intercept)    edad_jefe
   61557.56       -80.51


$`09`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    44275.9        298.1


$`10`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
  46293.749        7.807


$`11`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   45856.40       -36.74


$`12`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   27045.90        83.79


$`13`
```

```
Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   42109.48       -57.72


$`14`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    55515.6        -25.5


$`15`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
      34668          153


$`16`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   48426.93       -65.55


$`17`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    34578.9        108.6
```

```
$`18`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   48185.84        79.67


$`19`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   54543.93        54.53


$`20`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
  33477.567        9.767


$`21`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    29022.5        165.9


$`22`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)
```

```
Coefficients:
(Intercept)    edad_jefe
    47429.7        132.4


$`23`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   42180.99        11.98


$`24`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    49118.0       -107.9


$`25`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   61909.90       -89.92


$`26`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   60217.11       -66.49
```

```
$`27`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    34146.3        137.4


$`28`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    53207.3       -122.3


$`29`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    30845.1        129.9


$`30`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
    28875.7         68.8


$`31`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
```

```
(Intercept)    edad_jefe
   37261.74        53.66


$`32`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Coefficients:
(Intercept)    edad_jefe
   39369.04        84.65
```

Y el mapeo se puede ir agregando...

```
concentrado2020 %>%
  split(.$ent) %>%
  map(~ lm(ing_cor ~ edad_jefe, data = .)) %>% # da solo los coeficientes
  map(summary) # da la parte de inferencia
```

```
$`01`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-53847 -27740 -12184  12656 641581

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 50487.33    3042.99   16.59   <2e-16 ***
edad_jefe     102.33      58.13    1.76   0.0785 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 47100 on 2667 degrees of freedom
Multiple R-squared:  0.00116,   Adjusted R-squared:  0.000786
F-statistic: 3.099 on 1 and 2667 DF,  p-value: 0.07847


$`02`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
 -62786  -32386  -15760   10565 3881875

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 50224.74    4607.58  10.900  < 2e-16 ***
edad_jefe     242.91      90.54   2.683  0.00733 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 88320 on 4140 degrees of freedom
Multiple R-squared:  0.001736,  Adjusted R-squared:  0.001495
F-statistic: 7.198 on 1 and 4140 DF,  p-value: 0.007326


$`03`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-68291 -30773 -13592  13727 639588

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 47835.79    3375.53   14.17  < 2e-16 ***
edad_jefe     272.73      66.52    4.10 4.25e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 52920 on 2715 degrees of freedom
Multiple R-squared:  0.006154,  Adjusted R-squared:  0.005788
F-statistic: 16.81 on 1 and 2715 DF,  p-value: 4.248e-05


$`04`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-45027 -26185 -13443  10044 536634
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 43159.93    3248.67  13.285   <2e-16 ***
edad_jefe      58.73      62.55   0.939    0.348
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 44230 on 2172 degrees of freedom
Multiple R-squared:  0.0004057, Adjusted R-squared:  -5.448e-05
F-statistic: 0.8816 on 1 and 2172 DF,  p-value: 0.3479


$`05`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-50161 -27149 -11519  11967 527608

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 54785.09    2348.34  23.329   <2e-16 ***
edad_jefe     -49.81      43.73  -1.139    0.255
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 43430 on 3920 degrees of freedom
Multiple R-squared:  0.0003308, Adjusted R-squared:  7.581e-05
F-statistic: 1.297 on 1 and 3920 DF,  p-value: 0.2548


$`06`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-50026 -26169 -11367  11829 673149

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 56096.00    2654.84  21.130   <2e-16 ***
edad_jefe     -91.26      48.66  -1.876   0.0608 .
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 45480 on 3280 degrees of freedom
Multiple R-squared:  0.001071,  Adjusted R-squared:  0.0007668
F-statistic: 3.518 on 1 and 3280 DF,  p-value: 0.06081


$`07`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-30674 -15430  -7650   4600 358554

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 20136.69    2038.61   9.878  < 2e-16 ***
edad_jefe     176.55      38.63   4.571 5.14e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 28640 on 2121 degrees of freedom
Multiple R-squared:  0.009753,  Adjusted R-squared:  0.009286
F-statistic: 20.89 on 1 and 2121 DF,  p-value: 5.145e-06


$`08`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
     Min       1Q   Median       3Q      Max
  -56304   -33730   -18662     6192 10645058

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 61557.56    9397.99   6.550 6.39e-11 ***
edad_jefe     -80.51     176.98  -0.455    0.649
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 192100 on 4570 degrees of freedom
```

```
Multiple R-squared:  4.528e-05, Adjusted R-squared:  -0.0001735
F-statistic: 0.2069 on 1 and 4570 DF,  p-value: 0.6492


$`09`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
 -65068  -31879  -15111   13622 1140796

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 44275.87    4045.96  10.943  < 2e-16 ***
edad_jefe     298.14      73.34   4.065 4.94e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 57130 on 2568 degrees of freedom
Multiple R-squared:  0.006395,  Adjusted R-squared:  0.006008
F-statistic: 16.53 on 1 and 2568 DF,  p-value: 4.941e-05


$`10`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
 -45690  -25672  -12659    7365 6360100

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 46293.749   8213.038   5.637 1.91e-08 ***
edad_jefe       7.807    150.147   0.052    0.959
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 129200 on 2744 degrees of freedom
Multiple R-squared:  9.852e-07, Adjusted R-squared:  -0.0003634
F-statistic: 0.002703 on 1 and 2744 DF,  p-value: 0.9585
```

```
$`11`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-41277 -21865  -9269  10680 305297

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 45856.40    2038.40  22.496   <2e-16 ***
edad_jefe     -36.74      37.80  -0.972    0.331
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 34360 on 3081 degrees of freedom
Multiple R-squared:  0.0003064, Adjusted R-squared:  -1.803e-05
F-statistic: 0.9444 on 1 and 3081 DF,  p-value: 0.3312


$`12`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-32748 -17449  -8748   7167 233275

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 27045.90    1905.72  14.192   <2e-16 ***
edad_jefe      83.79      34.88   2.402   0.0164 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 29140 on 2488 degrees of freedom
Multiple R-squared:  0.002314,  Adjusted R-squared:  0.001913
F-statistic:  5.77 on 1 and 2488 DF,  p-value: 0.01638


$`13`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)
```

```
Residuals:
   Min     1Q Median     3Q    Max
-38141 -19608  -9123   8407 380207

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 42109.48    2307.56  18.249   <2e-16 ***
edad_jefe     -57.72      42.42  -1.361    0.174
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 32560 on 2211 degrees of freedom
Multiple R-squared:  0.0008366, Adjusted R-squared:  0.0003847
F-statistic: 1.851 on 1 and 2211 DF,  p-value: 0.1738


$`14`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
 -51479  -26801  -12529   10973 1695965

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  55515.6     4069.4   13.64   <2e-16 ***
edad_jefe      -25.5       75.1   -0.34    0.734
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 65390 on 2777 degrees of freedom
Multiple R-squared:  4.151e-05, Adjusted R-squared:  -0.0003186
F-statistic: 0.1153 on 1 and 2777 DF,  p-value: 0.7342


$`15`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
 -44553  -21374  -10179    7495 2137811
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 34668.17    3179.52  10.904   <2e-16 ***
edad_jefe     152.98      60.19   2.542   0.0111 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 53760 on 3566 degrees of freedom
Multiple R-squared:  0.001808,  Adjusted R-squared:  0.001528
F-statistic:  6.46 on 1 and 3566 DF,  p-value: 0.01108


$`16`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-45469 -23107 -10431   9913 699833

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 48426.93    3050.94  15.873   <2e-16 ***
edad_jefe     -65.55      56.42  -1.162    0.245
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 42920 on 2045 degrees of freedom
Multiple R-squared:  0.0006595, Adjusted R-squared:  0.0001708
F-statistic:  1.35 on 1 and 2045 DF,  p-value: 0.2455


$`17`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-40444 -21067  -9525   8578 422100

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 34578.88    2498.64  13.839   <2e-16 ***
```

```
edad_jefe     108.61      45.21   2.402   0.0164 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 37730 on 2562 degrees of freedom
Multiple R-squared:  0.002247,  Adjusted R-squared:  0.001858
F-statistic: 5.771 on 1 and 2562 DF,  p-value: 0.01637


$`18`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-51372 -26788 -11200  10938 906914

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 48185.84    3461.34  13.921   <2e-16 ***
edad_jefe      79.67      63.95   1.246    0.213
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 47630 on 2101 degrees of freedom
Multiple R-squared:  0.0007383, Adjusted R-squared:  0.0002626
F-statistic: 1.552 on 1 and 2101 DF,  p-value: 0.2129


$`19`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
 -56829  -31122  -15660    8669 3248057

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 54543.93    4696.75  11.613   <2e-16 ***
edad_jefe      54.53      84.42   0.646    0.518
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 82240 on 3500 degrees of freedom
Multiple R-squared:  0.0001192, Adjusted R-squared:  -0.0001665
F-statistic: 0.4172 on 1 and 3500 DF,  p-value: 0.5184


$`20`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-32265 -19866 -10036   7954 618037

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 33477.567   2275.058  14.715   <2e-16 ***
edad_jefe       9.767     41.872   0.233    0.816
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 35240 on 2594 degrees of freedom
Multiple R-squared:  2.097e-05, Adjusted R-squared:  -0.0003645
F-statistic: 0.05441 on 1 and 2594 DF,  p-value: 0.8156


$`21`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
 -37459  -19716   -9087    6812 1298468

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 29022.51    3120.66   9.300  < 2e-16 ***
edad_jefe     165.88      58.26   2.847  0.00446 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 42970 on 2139 degrees of freedom
Multiple R-squared:  0.003775,  Adjusted R-squared:  0.003309
F-statistic: 8.106 on 1 and 2139 DF,  p-value: 0.004455
```

```
$`22`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
 -55242  -27383  -11991   11751 1049250

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 47429.67    2745.77  17.274   <2e-16 ***
edad_jefe     132.41      53.32   2.483   0.0131 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 50880 on 3767 degrees of freedom
Multiple R-squared:  0.001634,  Adjusted R-squared:  0.001369
F-statistic: 6.166 on 1 and 3767 DF,  p-value: 0.01307


$`23`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
 -42718  -23815  -10997    9207 1415545

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 42180.99    3707.87  11.376   <2e-16 ***
edad_jefe      11.98      75.20   0.159    0.873
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 50830 on 2194 degrees of freedom
Multiple R-squared:  1.157e-05, Adjusted R-squared:  -0.0004442
F-statistic: 0.0254 on 1 and 2194 DF,  p-value: 0.8734


$`24`

Call:
```

```
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-41965 -25250 -11448  10590 898064

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 49118.03    2980.85  16.478   <2e-16 ***
edad_jefe    -107.90      53.92  -2.001   0.0455 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 44520 on 2519 degrees of freedom
Multiple R-squared:  0.001587,  Adjusted R-squared:  0.001191
F-statistic: 4.005 on 1 and 2519 DF,  p-value: 0.04548


$`25`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-55695 -28466 -12659  12536 923717

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 61909.90    2993.68  20.680   <2e-16 ***
edad_jefe     -89.92      54.92  -1.638    0.102
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 49900 on 3427 degrees of freedom
Multiple R-squared:  0.0007818, Adjusted R-squared:  0.0004903
F-statistic: 2.681 on 1 and 3427 DF,  p-value: 0.1016


$`26`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
```

```
-55392 -31717 -15980  11905 931996

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 60217.11    4011.12  15.013   <2e-16 ***
edad_jefe     -66.49      73.80  -0.901    0.368
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 58040 on 2418 degrees of freedom
Multiple R-squared:  0.0003356, Adjusted R-squared:  -7.779e-05
F-statistic: 0.8118 on 1 and 2418 DF,  p-value: 0.3677


$`27`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-41568 -22640 -11534   8209 420041

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  34146.3     2993.3  11.407   <2e-16 ***
edad_jefe      137.4       57.4   2.393   0.0168 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 39990 on 2086 degrees of freedom
Multiple R-squared:  0.002739,  Adjusted R-squared:  0.002261
F-statistic: 5.729 on 1 and 2086 DF,  p-value: 0.01678


$`28`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
 -49294  -25381  -12421    7864 2384864

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)  53207.3     4963.6  10.719   <2e-16 ***
edad_jefe     -122.3       93.2  -1.312     0.19
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 71020 on 2309 degrees of freedom
Multiple R-squared:  0.000745,  Adjusted R-squared:  0.0003123
F-statistic: 1.722 on 1 and 2309 DF,  p-value: 0.1896


$`29`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-36585 -18878  -8947   9074 290867

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 30845.14    2334.54  13.213  < 2e-16 ***
edad_jefe     129.93      43.69   2.974  0.00298 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 31650 on 2157 degrees of freedom
Multiple R-squared:  0.004082,  Adjusted R-squared:  0.003621
F-statistic: 8.842 on 1 and 2157 DF,  p-value: 0.002977


$`30`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
   Min     1Q Median     3Q    Max
-31570 -17104  -7854   6713 250230

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  28875.7     1842.2  15.675   <2e-16 ***
edad_jefe       68.8       33.7   2.042   0.0413 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 28190 on 2715 degrees of freedom
Multiple R-squared:  0.001533,  Adjusted R-squared:  0.001165
F-statistic: 4.168 on 1 and 2715 DF,  p-value: 0.04128


$`31`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
 -39137  -22567  -11025    6332 1454038

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 37261.74    3028.89   12.30   <2e-16 ***
edad_jefe      53.66      57.08    0.94    0.347
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 48810 on 2887 degrees of freedom
Multiple R-squared:  0.000306,  Adjusted R-squared:  -4.032e-05
F-statistic: 0.8836 on 1 and 2887 DF,  p-value: 0.3473


$`32`

Call:
lm(formula = ing_cor ~ edad_jefe, data = .)

Residuals:
    Min      1Q  Median      3Q     Max
 -42383  -24463  -12794    7267 3189966

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 39369.04    5738.28   6.861  8.6e-12 ***
edad_jefe      84.65     104.17   0.813    0.417
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 87290 on 2502 degrees of freedom
Multiple R-squared:  0.0002638, Adjusted R-squared:  -0.0001358
F-statistic: 0.6602 on 1 and 2502 DF,  p-value: 0.4166
```

Si queremos esto en una sola base de datos

```
concentrado2020 %>%
  split(.$ent) %>%
  map(~ lm(ing_cor ~ edad_jefe, data = .)) %>% # da solo los coeficientes
  map(~ broom::tidy(.x))
```

```
$`01`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   50487.    3043.      16.6  6.53e-59
2 edad_jefe       102.      58.1      1.76 7.85e- 2

$`02`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   50225.    4608.      10.9  2.69e-27
2 edad_jefe       243.      90.5      2.68 7.33e- 3

$`03`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   47836.    3376.      14.2  4.92e-44
2 edad_jefe       273.      66.5      4.10 4.25e- 5

$`04`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  43160.     3249.     13.3   8.78e-39
2 edad_jefe       58.7      62.6     0.939 3.48e- 1

$`05`
# A tibble: 2 x 5
  term        estimate std.error statistic   p.value
  <chr>          <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)  54785.     2348.      23.3  7.83e-113
2 edad_jefe      -49.8      43.7     -1.14 2.55e-  1

$`06`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
```

```
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  56096.     2655.      21.1  5.13e-93
2 edad_jefe      -91.3      48.7     -1.88 6.08e- 2

$`07`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   20137.    2039.       9.88 1.58e-22
2 edad_jefe       177.      38.6      4.57 5.14e- 6

$`08`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  61558.      9398.     6.55  6.39e-11
2 edad_jefe      -80.5      177.    -0.455 6.49e- 1

$`09`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   44276.    4046.      10.9  2.84e-27
2 edad_jefe       298.      73.3      4.07 4.94e- 5

$`10`
# A tibble: 2 x 5
  term        estimate std.error statistic      p.value
  <chr>          <dbl>     <dbl>     <dbl>        <dbl>
1 (Intercept) 46294.       8213.    5.64   0.0000000191
2 edad_jefe       7.81      150.    0.0520 0.959

$`11`
# A tibble: 2 x 5
  term        estimate std.error statistic   p.value
  <chr>          <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)  45856.     2038.     22.5   6.80e-104
2 edad_jefe      -36.7      37.8    -0.972 3.31e-  1

$`12`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  27046.     1906.      14.2  5.12e-44
2 edad_jefe       83.8      34.9      2.40 1.64e- 2
```

```
$`13`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  42109.     2308.      18.2  2.05e-69
2 edad_jefe      -57.7      42.4     -1.36 1.74e- 1

$`14`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  55516.     4069.     13.6   4.59e-41
2 edad_jefe      -25.5      75.1    -0.340 7.34e- 1

$`15`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   34668.    3180.      10.9  2.97e-27
2 edad_jefe       153.      60.2      2.54 1.11e- 2

$`16`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  48427.     3051.      15.9  1.36e-53
2 edad_jefe      -65.5      56.4     -1.16 2.45e- 1

$`17`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   34579.    2499.      13.8  4.65e-42
2 edad_jefe       109.      45.2      2.40 1.64e- 2

$`18`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  48186.     3461.      13.9  3.32e-42
2 edad_jefe       79.7      63.9      1.25 2.13e- 1

$`19`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
```

```
1 (Intercept)  54544.     4697.     11.6   1.28e-30
2 edad_jefe       54.5      84.4     0.646 5.18e- 1

$`20`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept) 33478.      2275.     14.7   3.89e-47
2 edad_jefe       9.77      41.9     0.233 8.16e- 1

$`21`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   29023.    3121.       9.30 3.35e-20
2 edad_jefe       166.      58.3      2.85 4.46e- 3

$`22`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   47430.    2746.      17.3  2.11e-64
2 edad_jefe       132.      53.3      2.48 1.31e- 2

$`23`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  42181.     3708.     11.4   3.55e-29
2 edad_jefe       12.0      75.2     0.159 8.73e- 1

$`24`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   49118.    2981.      16.5  5.16e-58
2 edad_jefe      -108.      53.9     -2.00 4.55e- 2

$`25`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  61910.     2994.      20.7  1.25e-89
2 edad_jefe      -89.9      54.9     -1.64 1.02e- 1

$`26`
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  60217.     4011.     15.0   8.94e-49
2 edad_jefe      -66.5      73.8    -0.901 3.68e- 1

$`27`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   34146.    2993.      11.4  2.78e-29
2 edad_jefe       137.      57.4      2.39 1.68e- 2

$`28`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   53207.    4964.      10.7  3.37e-26
2 edad_jefe      -122.      93.2     -1.31 1.90e- 1

$`29`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   30845.    2335.      13.2  2.21e-38
2 edad_jefe       130.      43.7      2.97 2.98e- 3

$`30`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  28876.     1842.      15.7  4.44e-53
2 edad_jefe       68.8      33.7      2.04 4.13e- 2

$`31`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  37262.     3029.     12.3   6.15e-34
2 edad_jefe       53.7      57.1     0.940 3.47e- 1

$`32`
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)  39369.     5738.     6.86  8.60e-12
```

```
2 edad_jefe       84.6      104.     0.813 4.17e- 1
```

Como broom::tidy los volvio tibble, lo podemos guardar en un excelito

```
concentrado2020 %>%
  split(.$ent) %>%
  map(~ lm(ing_cor ~ edad_jefe, data = .)) %>% # da solo los coeficientes
  map(~ broom::tidy(.x)) %>%
  write_xlsx(path="modelos.xlsx")
```

6.9 Una aplicación para exportar los resultados de una base

Veamos...

```
names(concentrado2020)
```

```
  [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"        "ent"        "mun"
```

```r
vars<-c("clase_hog",  "sexo_jefe",  "edad_jefe" , "educa_jefe")

tabs<- vars %>%
  map(~ count(x=concentrado2020,
              !!as.name(.x), # para que ponga la variable a tabular
              wt=factor) %>% # para que use el pes
        mutate(pct = round((n / sum(n) * 100), 2)) %>% # Para que ponga el %
        adorn_totals()
    )

tabs # tabulados expandidos para mandar a un excel
```

```
[[1]]
 clase_hog        n    pct
         1  4233047  11.84
         2 22093441  61.80
         3  8999787  25.17
         4   278353   0.78
         5   145031   0.41
     Total 35749659 100.00

[[2]]
 sexo_jefe        n    pct
         1 25072652  70.13
         2 10677007  29.87
     Total 35749659 100.00

[[3]]
 edad_jefe        n    pct
        14      206   0.00
        15      801   0.00
        16     7563   0.02
        17     9373   0.03
        18    32313   0.09
        19    49173   0.14
        20    87861   0.25
        21   119741   0.33
        22   160659   0.45
        23   234934   0.66
        24   245194   0.69
        25   302740   0.85
        26   358181   1.00
        27   393797   1.10
        28   453394   1.27
        29   480245   1.34
```

30	595832	1.67
31	477389	1.34
32	594044	1.66
33	538747	1.51
34	585142	1.64
35	679411	1.90
36	640705	1.79
37	623343	1.74
38	771090	2.16
39	676548	1.89
40	850750	2.38
41	633938	1.77
42	910197	2.55
43	811414	2.27
44	743078	2.08
45	858338	2.40
46	793630	2.22
47	839846	2.35
48	895754	2.51
49	802282	2.24
50	955486	2.67
51	633909	1.77
52	873950	2.44
53	791650	2.21
54	763664	2.14
55	792306	2.22
56	789150	2.21
57	689864	1.93
58	699728	1.96
59	633661	1.77
60	815146	2.28
61	531612	1.49
62	638709	1.79
63	657153	1.84
64	566999	1.59
65	606264	1.70
66	459693	1.29
67	488873	1.37
68	554800	1.55
69	427911	1.20
70	519709	1.45
71	348224	0.97
72	438795	1.23
73	369007	1.03
74	362156	1.01
75	371473	1.04

```
        76   279436   0.78
        77   259322   0.73
        78   296529   0.83
        79   199962   0.56
        80   287766   0.80
        81   138220   0.39
        82   176749   0.49
        83   165610   0.46
        84   177493   0.50
        85   173183   0.48
        86   117048   0.33
        87    93482   0.26
        88    75593   0.21
        89    54120   0.15
        90    63175   0.18
        91    27289   0.08
        92    33386   0.09
        93    24338   0.07
        94    22829   0.06
        95    14664   0.04
        96     9677   0.03
        97     9276   0.03
        98     4954   0.01
        99     3278   0.01
       100     1057   0.00
       101     1296   0.00
       102      863   0.00
       103      607   0.00
       104      621   0.00
       105      146   0.00
       107      145   0.00
     Total 35749659 100.01

[[4]]
 educa_jefe        n    pct
          1  2287415   6.40
          2     4946   0.01
          3  5111202  14.30
          4  5743095  16.06
          5  1175737   3.29
          6  9244410  25.86
          7  1321912   3.70
          8  4598279  12.86
          9  1228591   3.44
         10  4144117  11.59
         11   889955   2.49
```

```
   Total 35749659 100.00
```

```r
names(tabs)<-vars # para que las hojas del excel se llamen como la variables

write_xlsx(tabs, path="tabs.xlsx")
```

Chapter 7

Visualización de datos (I)

7.1 Paquetes y datos

Ahora cargaremos nuestros paquetes para hoy.

```
if(!require("pacman")) install.packages("pacman")
```

```
Loading required package: pacman
```

```
pacman::p_load(tidyverse, readxl, writexl, haven, sjlabelled, foreign, janitor,
               esquisse, RColorBrewer, wesanderson)
```

Y cargaremos la base de datos de concentrado.

```
concentrado2020 <- haven::read_dta("datos/concentrado2020.dta")
```

7.2 Visualización de datos: introducción

Cheatsheet en español: https://diegokoz.github.io/intro_ds/fuentes/ggplot2-cheatsheet-2.1-Spanish.pdf

El ggplot2 se basa en la construcción de gráficos a partir de tres componentes:

1) Datos,
2) Coordenadas y
3) Objetos geométricos

Esto será nuestra "gramática de gráficas"

Para visualizar los resultados, nosotres asignamos variables a las propiedades visuales o estéticas

Por ejemplo: los tamaños, colores y posiciones.

De manera genérica, podríamos pensar que el código para el ggplot será de la siguiente manera:

```
ggplot(datos) + (geometria) + (esteticas)
```

Esta semana, haremos gráficas para una sola variable, cuantitativa o cualitativa.

La próxima semana haremos gráficas para dos variables.

7.3 Variables cuantitativas

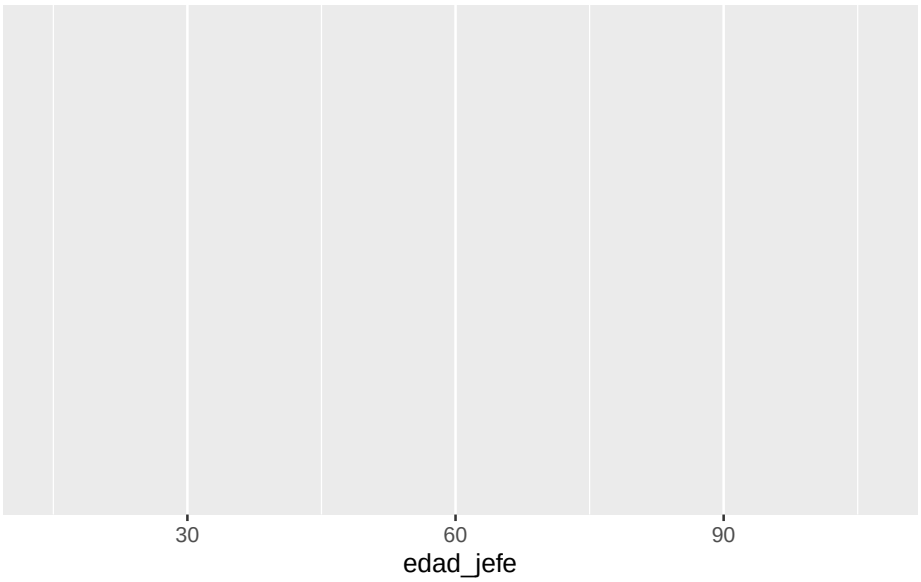
Para las variables cuantitativas, las gráficas más utilizadas son los histogramas, gráficos de densidad.

Menos utilizados: de área y polígonos de frecuencias (ver https://r-graph-gallery.com/)

Bueno, en series de tiempo también se utilizan los gráficos de líneas.

Veamos primero los componentes de nuestra gramática. En los datos incluimos la variable que queremos y la base de datos que ocuparemos. En este caso es la edad del jefe del hogar.

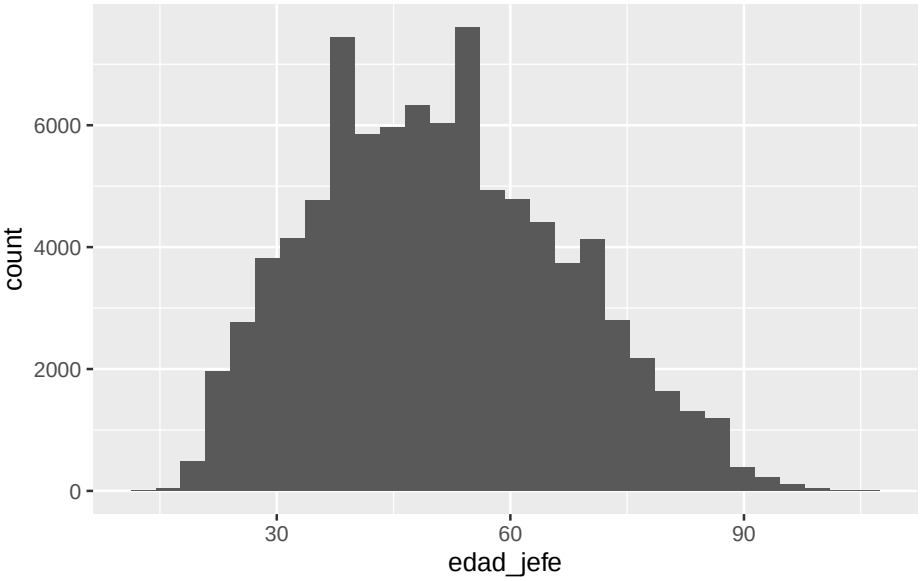
```
concentrado2020 %>%
  ggplot(aes(x=edad_jefe))
```



Ahora agregaremos la geometría.

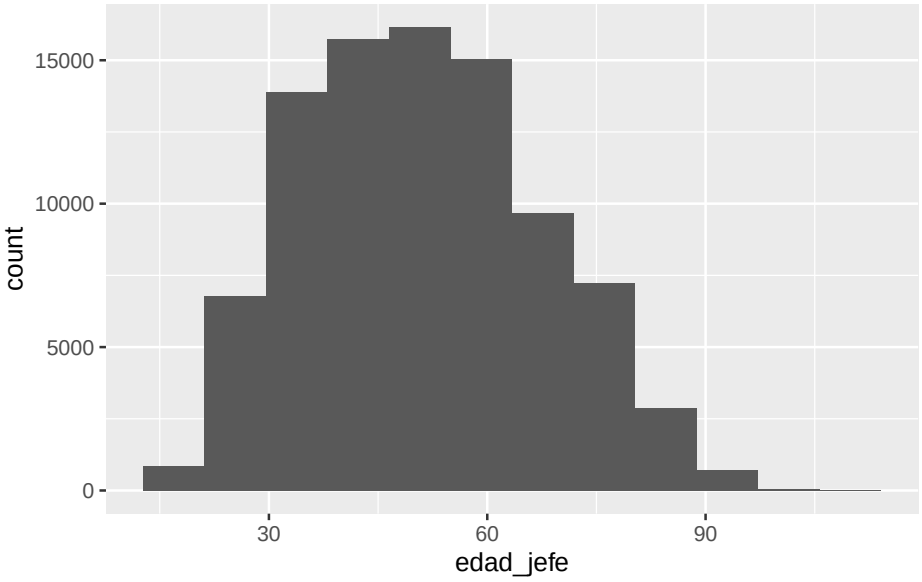
```
concentrado2020 %>%
  ggplot(aes(x=edad_jefe)) +
  geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



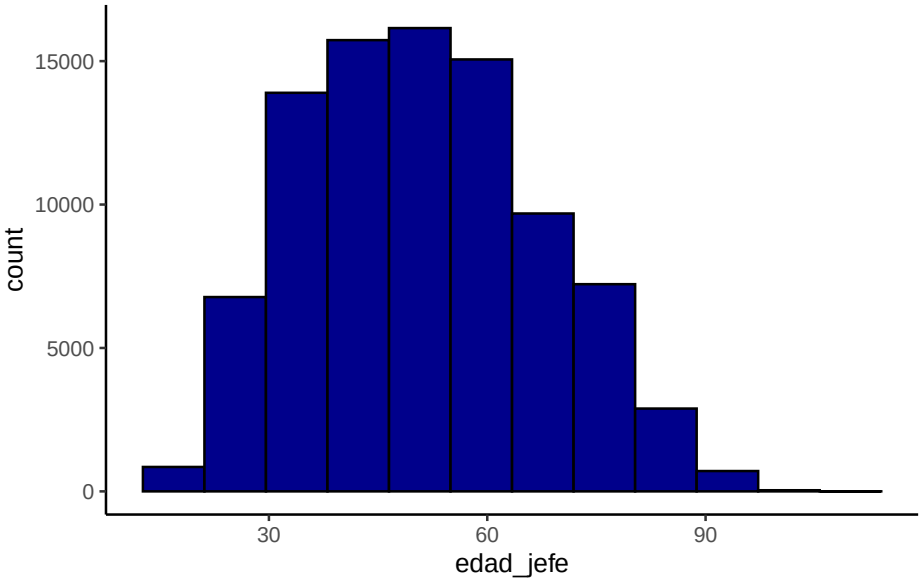
Vemos que el software nos avisa cuántas clases/intervalos está utilizando. Podemos cambiarlas.

```
concentrado2020 %>%
  ggplot(aes(x=edad_jefe)) +
  geom_histogram(bins=12)
```



Una vez que tenemos nuestros datos y geometría, vamos a editar: primero le cambiamos el color y le quitamos el fondo gris

```
concentrado2020 %>%
  ggplot(aes(x=edad_jefe)) +
  geom_histogram(bins=12, color="#000000", fill="darkblue")+
  theme_classic()
```



7.3.1 Sobre los colores en R:

Podemos agregar manualmente los colores, como lo hicimos anteriormente.

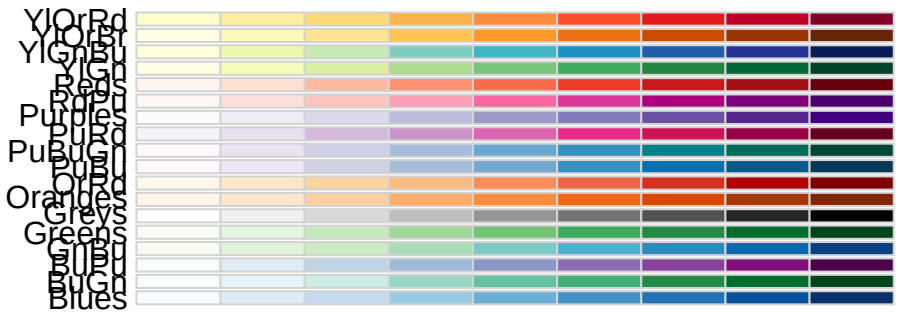
Sin embargo, existen paquetes que ya traen paletas cargadas y que se pueden utilizar dependiendo de los datos.

Una paleta es la de RColorBrewer: https://www.geeksforgeeks.org/introduction-to-color-palettes-in-r-with-rcolorbrewer/

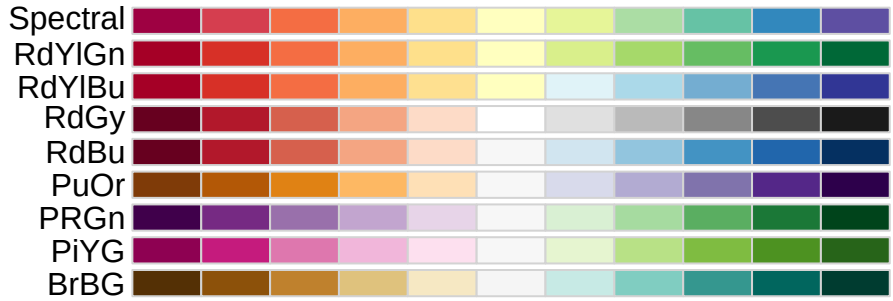
Esta paleta de colores distingue si los datos son secuenciales, divergentes o cualitativos.

Por ejemplo:

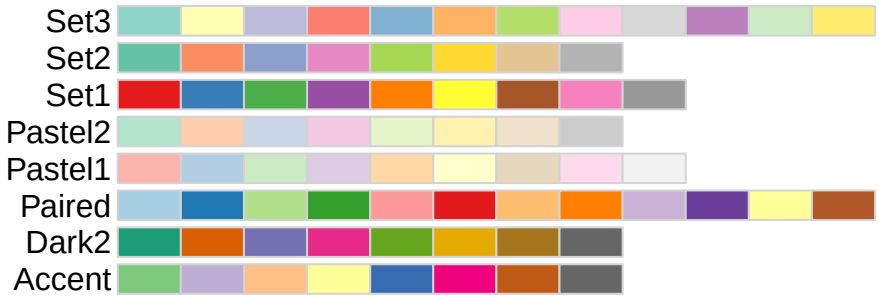
```
display.brewer.all(type="seq") #secuenciales
```



```
display.brewer.all(type="div") #divergentes
```




```
display.brewer.all(type="qual") #datos cualitativos
```

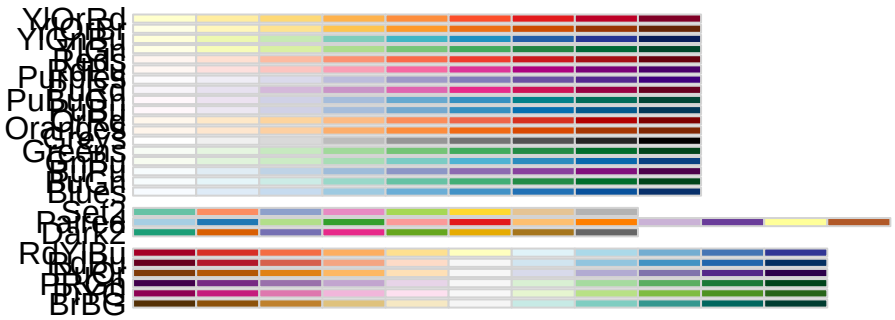



```
brewer.pal.info #Podemos enlistar todos los colores
```

```
          maxcolors category colorblind
BrBG             11      div       TRUE
PiYG             11      div       TRUE
PRGn             11      div       TRUE
PuOr             11      div       TRUE
RdBu             11      div       TRUE
RdGy             11      div      FALSE
RdYlBu           11      div       TRUE
RdYlGn           11      div      FALSE
Spectral         11      div      FALSE
Accent            8     qual      FALSE
Dark2             8     qual       TRUE
Paired           12     qual       TRUE
Pastel1           9     qual      FALSE
Pastel2           8     qual      FALSE
```

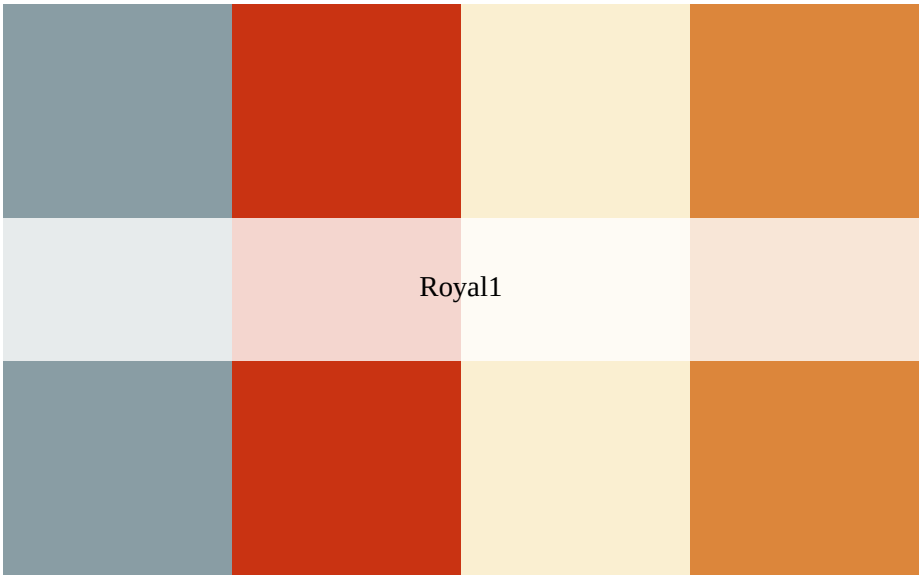
```
Set1          9     qual      FALSE
Set2          8     qual       TRUE
Set3         12     qual      FALSE
Blues         9      seq       TRUE
BuGn          9      seq       TRUE
BuPu          9      seq       TRUE
GnBu          9      seq       TRUE
Greens        9      seq       TRUE
Greys         9      seq       TRUE
Oranges       9      seq       TRUE
OrRd          9      seq       TRUE
PuBu          9      seq       TRUE
PuBuGn        9      seq       TRUE
PuRd          9      seq       TRUE
Purples       9      seq       TRUE
RdPu          9      seq       TRUE
Reds          9      seq       TRUE
YlGn          9      seq       TRUE
YlGnBu        9      seq       TRUE
YlOrBr        9      seq       TRUE
YlOrRd        9      seq       TRUE
```

```
display.brewer.all(colorblindFriendly=T) #La última columna nos dice si alguien con probl
```

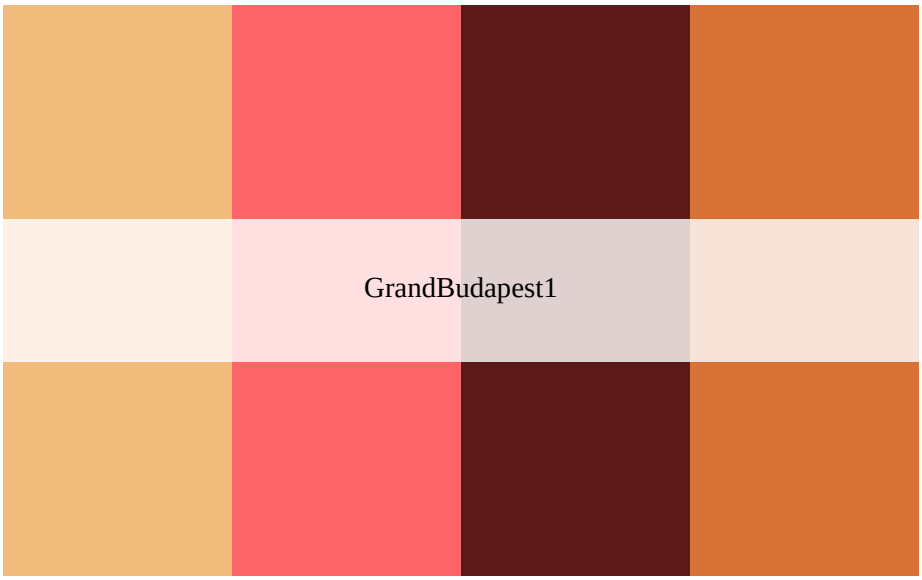


Otra paleta es la de Wesanderson, se inspira en sus peliculas: https://rforpoliticalscience.com/2020/07/26/make-wes-anderson-themed-graphs-with-wesanderson-package-in-r/ Debes escoger el nombre de la paleta y cuántos colores vas a usar.

```
wes_palette("Royal1")
```




```
wes_palette("GrandBudapest1")
```




```
wes_palette("Cavalcanti1")
```

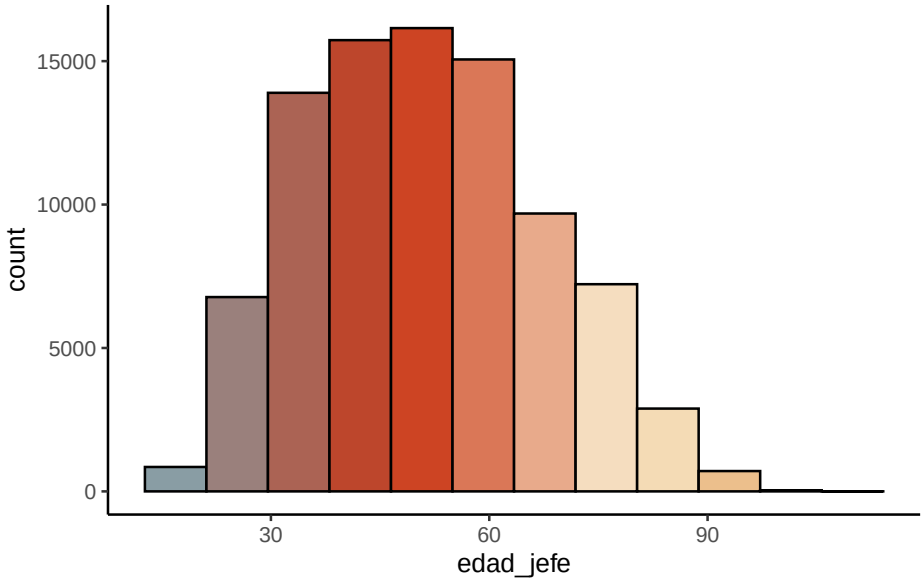

```
wes_palette("Cavalcanti1", 3)
```

La lógica de estos paquetes es crear nuestra paleta de colores pensando en cuántos vamos a tener que utilizar. Entonces, haremos la nuestra.

```
pal <- wes_palette(12, name = "Royal1", type = "continuous")
```

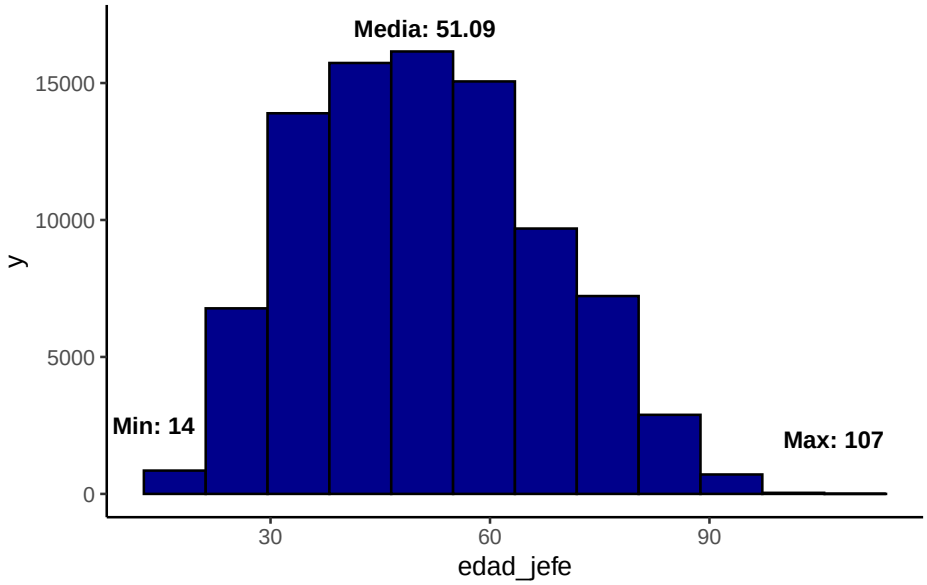
Entonces, podemos volver a hacer nuestro gráfico escogiendo alguno de ellos.

```
concentrado2020 %>%
  ggplot(aes(x=edad_jefe)) +
  geom_histogram(bins=12, color="#000000", fill=pal)+
  theme_classic()
```



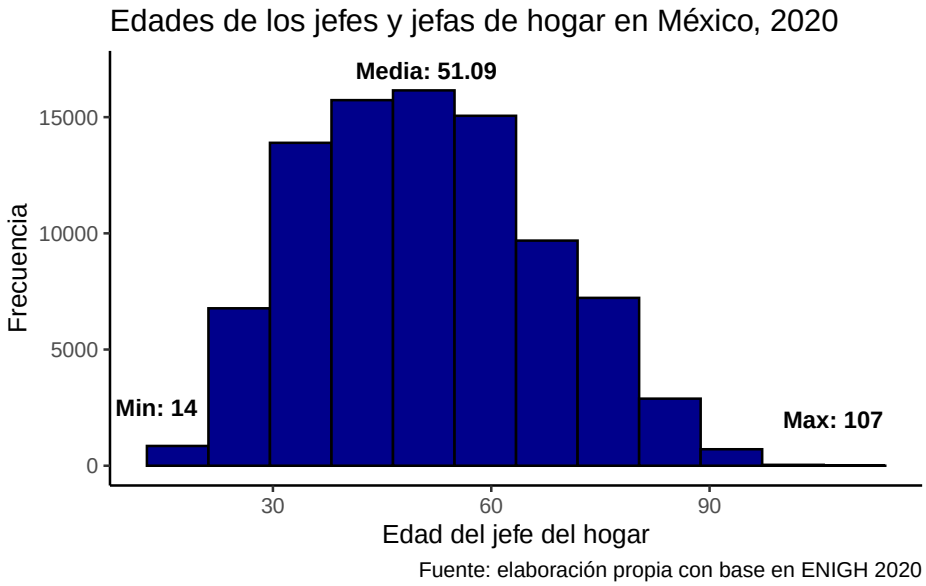
Podemos agregarle el valor mínimo, máximo y la media, pero para eso tenemos que hacer un pequeño dataframe

```
anotaciones <- data.frame(
  x = c(round(min(concentrado2020$edad_jefe), 2), round(mean(concentrado2020$edad_jefe),
  y = c(2500, 17000, 2000),
  label = c("Min:", "Media:", "Max:")
)

concentrado2020 %>%
  ggplot(aes(x=edad_jefe)) +
  geom_histogram(bins=12, color="#000000", fill="darkblue")+
  theme_classic()+
  geom_text(data = anotaciones, aes(x = x, y = y, label = paste(label, x)), size = 3.5, f
```



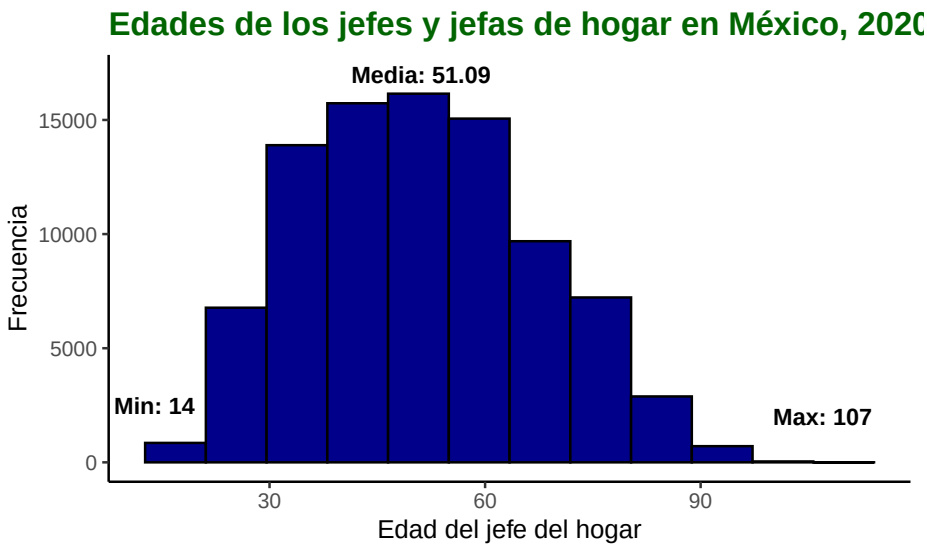
También le vamos a agregar el título, subtítulo y fuente

```
concentrado2020 %>%
  ggplot(aes(x=edad_jefe)) +
  geom_histogram(bins=12, color="#000000", fill="darkblue")+
  theme_classic()+
  geom_text(data = anotaciones, aes(x = x, y = y, label = paste(label, x)), size = 3.5, f
  labs(
    x = "Edad del jefe del hogar",
    y = "Frecuencia",
    title = "Edades de los jefes y jefas de hogar en México, 2020",
    caption = "Fuente: elaboración propia con base en ENIGH 2020"
  )
```



Al título y fuente también podemos agregarle los tipos de letra

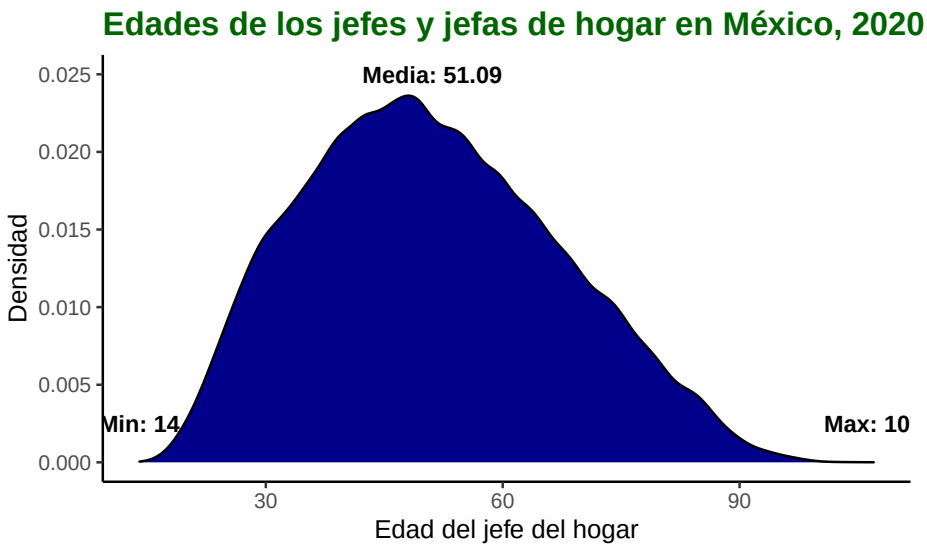
```
concentrado2020 %>%
  ggplot(aes(x=edad_jefe)) +
  geom_histogram(bins=12, color="#000000", fill="darkblue")+
  theme_classic()+
  geom_text(data = anotaciones, aes(x = x, y = y, label = paste(label, x)), size = 3.5, f
  labs(
    x = "Edad del jefe del hogar",
    y = "Frecuencia",
    title = "Edades de los jefes y jefas de hogar en México, 2020",
    caption = "Fuente: elaboración propia con base en ENIGH 2020"
  )+
theme(
  plot.title = element_text(color = "darkgreen", size = 14, face = "bold"),
  plot.caption = element_text(face = "italic")
)
```



Fuente: elaboración propia con base en ENIGH 2020

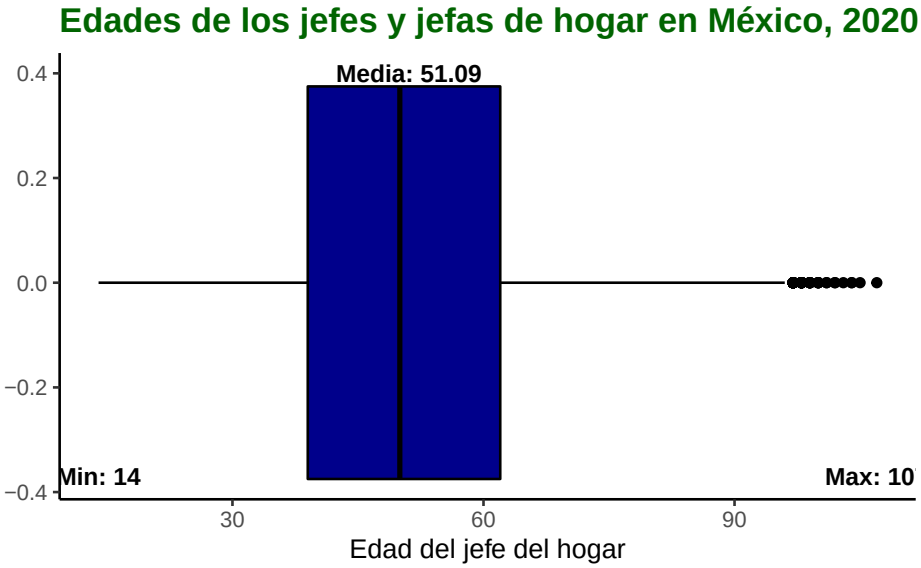
Podemos cambiar el histograma por un gráfico de densidad, para ello cambiamos nuestra base pequeña de anotaciones.

```
anotaciones <- data.frame(
  x = c(round(min(concentrado2020$edad_jefe), 2), round(mean(concentrado2020$edad_jefe),
  y = c(0.0025, 0.025, 0.0025),
  label = c("Min:", "Media:", "Max:")
)

concentrado2020 %>%
  ggplot(aes(x=edad_jefe)) +
  geom_density(adjust = 1L, color="#000000", fill="darkblue")+
  theme_classic()+
  geom_text(data = anotaciones, aes(x = x, y = y, label = paste(label, x)), size = 3.5, f
  labs(
    x = "Edad del jefe del hogar",
    y = "Densidad",
    title = "Edades de los jefes y jefas de hogar en México, 2020",
    caption = "Fuente: elaboración propia con base en ENIGH 2020"
  )+
  theme(
    plot.title = element_text(color = "darkgreen", size = 14, face = "bold"),
    plot.caption = element_text(face = "italic")
  )
```



Por último, también podemos hacer un boxplot

```
anotaciones <- data.frame(
  x = c(round(min(concentrado2020$edad_jefe), 2), round(mean(concentrado2020$edad_jefe),
  y = c(-0.37, 0.4, -0.37),
  label = c("Min:", "Media:", "Max:")
)

concentrado2020 %>%
  ggplot(aes(x=edad_jefe)) +
  geom_boxplot(color="#000000", fill="darkblue") +
  theme_classic()+
  geom_text(data = anotaciones, aes(x = x, y = y, label = paste(label, x)), size = 3.5, f
  labs(
    x = "Edad del jefe del hogar",
    y = "",
    title = "Edades de los jefes y jefas de hogar en México, 2020",
    caption = "Fuente: elaboración propia con base en ENIGH 2020"
  )+
  theme(
    plot.title = element_text(color = "darkgreen", size = 14, face = "bold"),
    plot.caption = element_text(face = "italic")
  )
```



Fuente: elaboración propia con base en ENIGH 2020

El paquete `{esquisse}` es una forma de graficar de forma más sencilla

7.4 Práctica en clase:

elaboren un histograma, diagrama de caja y brazos o un diagrama de densidad con una variable cuantitativa que seleccionen. Modifica los colores, etiquetas, eje y, eje x, título, fuente, etc.

Chapter 8

Visualización de datos (II)

8.1 Paquetes y datos

```
#Paquetería #
if(!require("pacman")) install.packages("pacman")
```

```
Loading required package: pacman
```

```
pacman::p_load(tidyverse, readxl, writexl, haven, sjlabelled, foreign, janitor, esquisse,
```

```
#Importar #
concentrado2020 <- haven::read_dta("datos/concentrado2020.dta")
```

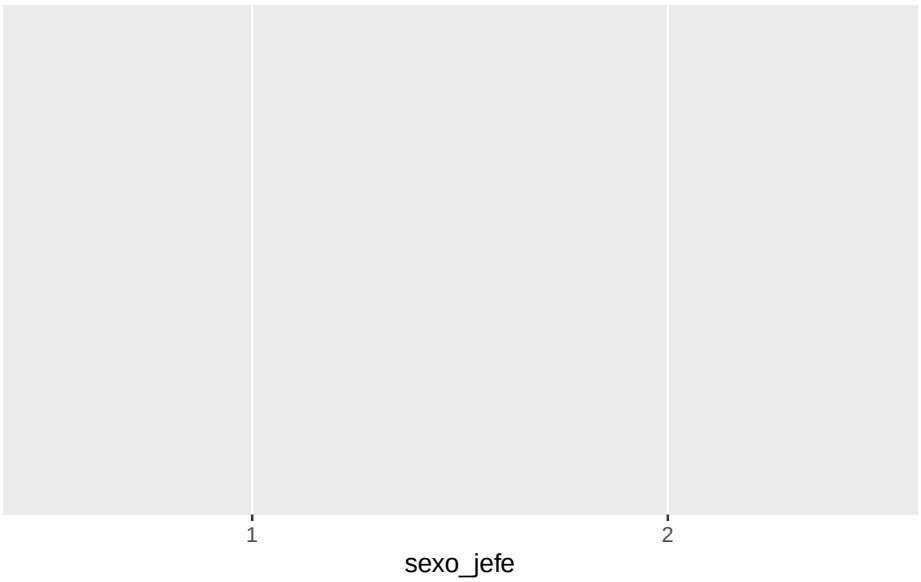
8.2 Variables cualitativas

Los gráficos más utilizados son los gráficos de barras. Primero vamos a hacer nuestro primer gráfico de barras o columnas.

Al igual que en la sesión anterior, vamos agregando capas a nuestros gráficos.

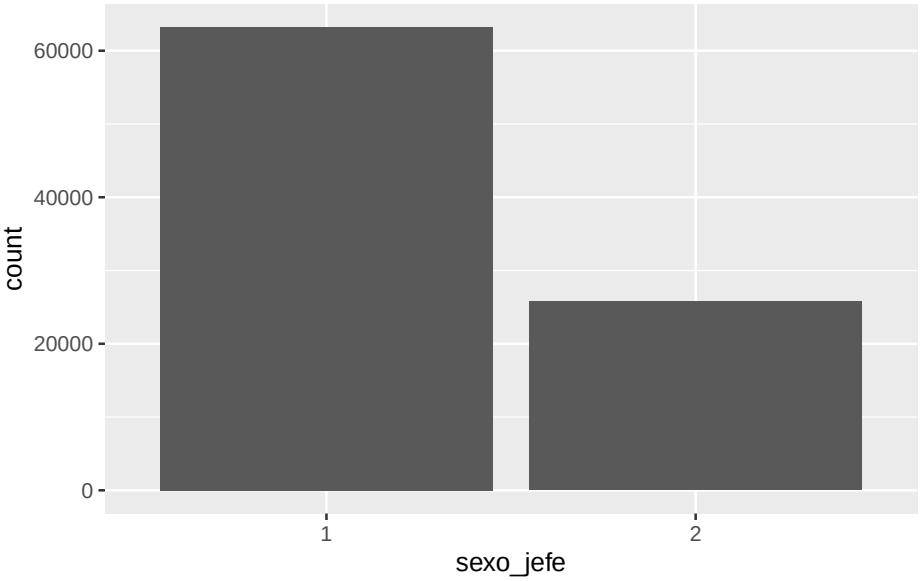
En los datos incluiremos la variable que queremos graficar, en este caso la jefatura de hogar. Lo guardaremos como un objeto.

```
g<- concentrado2020 %>%
  ggplot(aes(x=sexo_jefe))
g
```



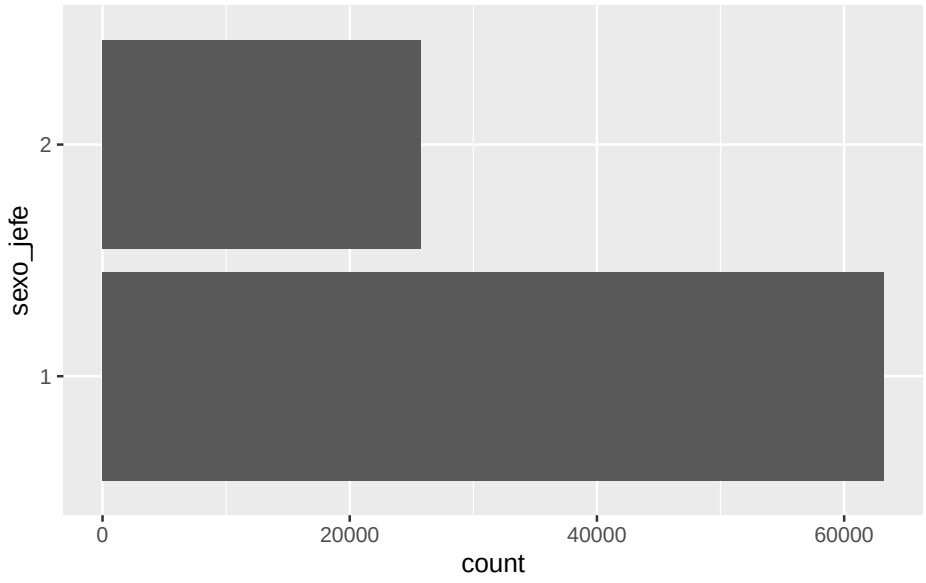
Vemos que graficamos nuestra variable, sin embargo, todavía no hemos especificado la geometría así que la vamos a agregar.

```
g+geom_bar()
```



También podemos voltear nuestras barras.

```
g +
geom_bar()+
coord_flip()
```



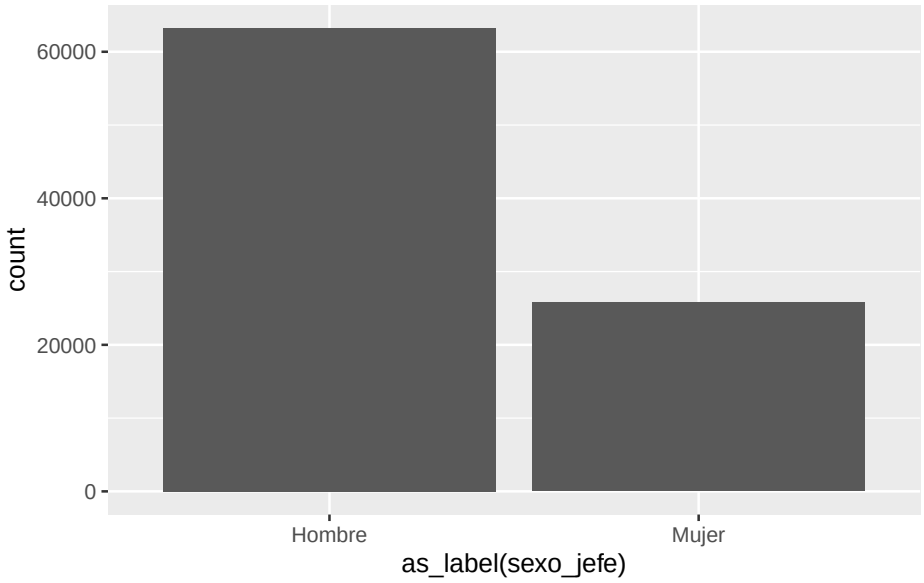
Aunque, se ve mejor como estaba antes.

Para hacer nuestra gráfica de barras y que se vean las etiquetas con facilidad, vamos a etiquetar nuestra variable.

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

concentrado2020<-concentrado2020 %>%
  mutate(sexo_jefe=as_numeric(sexo_jefe)) %>% # para quitar el "string"
  sjlabelled::set_labels(sexo_jefe, labels=etiqueta_sex)
```

Ahora sí podemos ver las etiquetas en nuestro gráfico

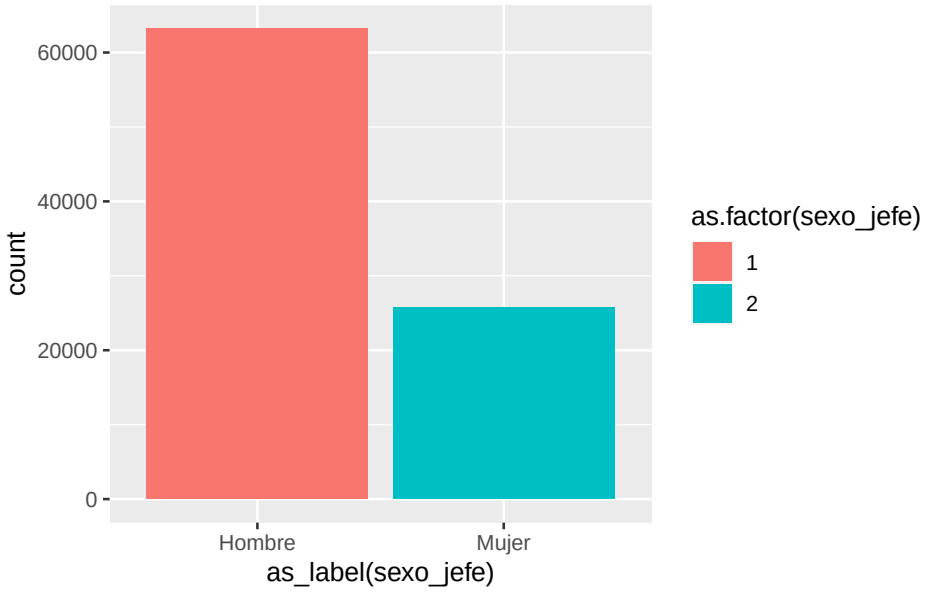
```
g<-concentrado2020 %>%
  ggplot(aes(x=as_label(sexo_jefe))) +
  geom_bar()
g
```



Vamos a ponerle un color a hombre y otro a mujer y a quitar el fondo gris.

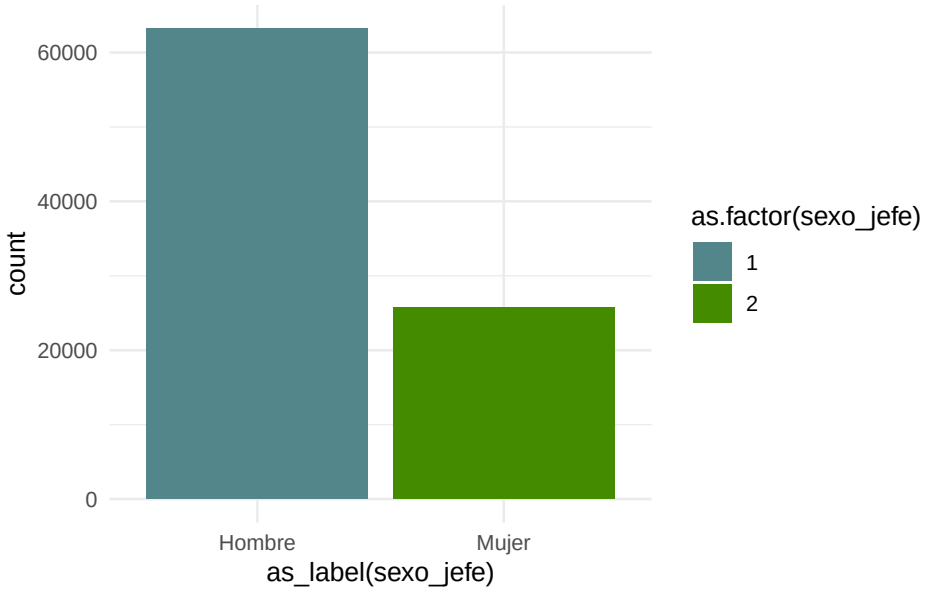
Opción 1. Ponerle color manualmente

```
g<- concentrado2020 %>%
  ggplot(aes(x = as_label(sexo_jefe), fill = as.factor(sexo_jefe))) + #debo rellenar como
  geom_bar()
g
```



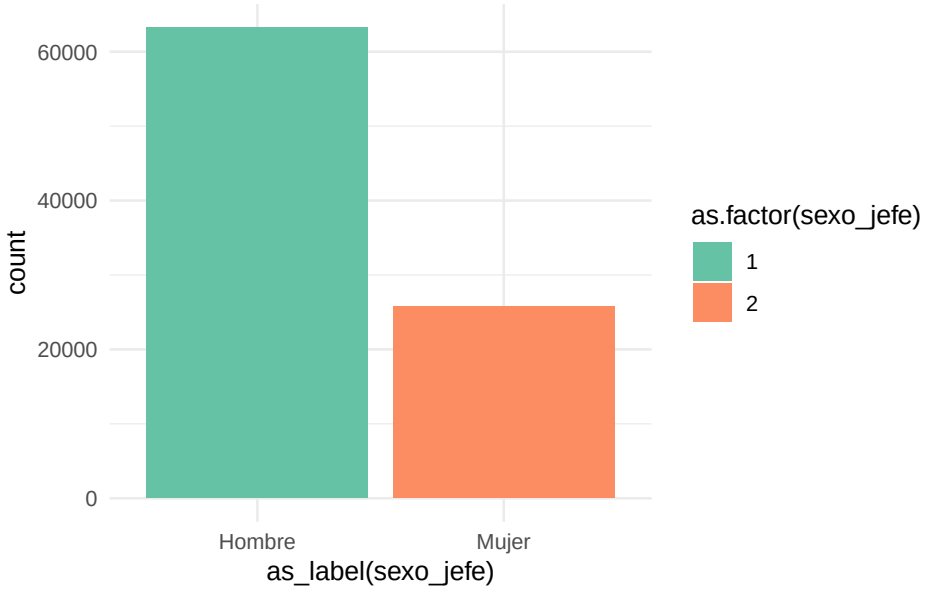
Esos son los colores por default. Ahora especificamos los colores.

```
g + scale_fill_manual(
    values = c("1" = "cadetblue4",
               "2" = "chartreuse4")
  ) +
  theme_minimal()
```



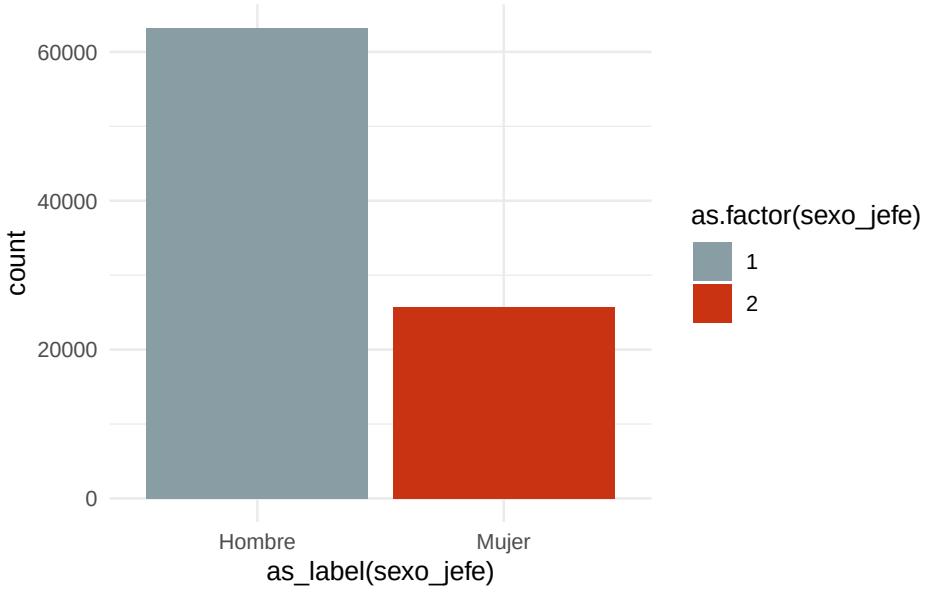
2. Opción 2, rellenamos el gráfico con RColorBrewer

```
g +
  scale_fill_brewer(palette="Set2") +
  theme_minimal()
```



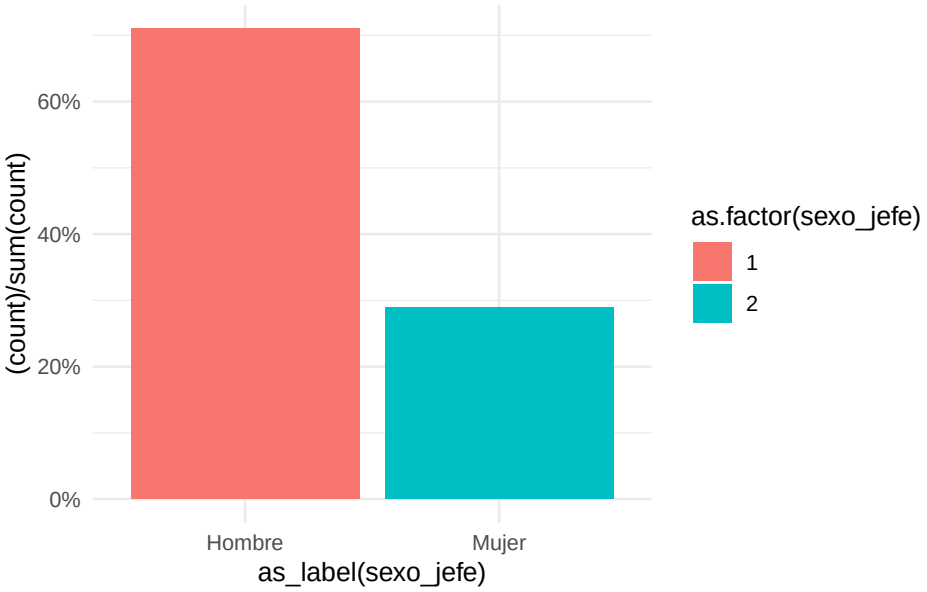
Opción 3. Ponerle color con Wesanderson.

```
pal <- wes_palette(2, name = "Royal1", type = "discrete")

g +
  scale_fill_manual(values=pal) +
  theme_minimal()
```



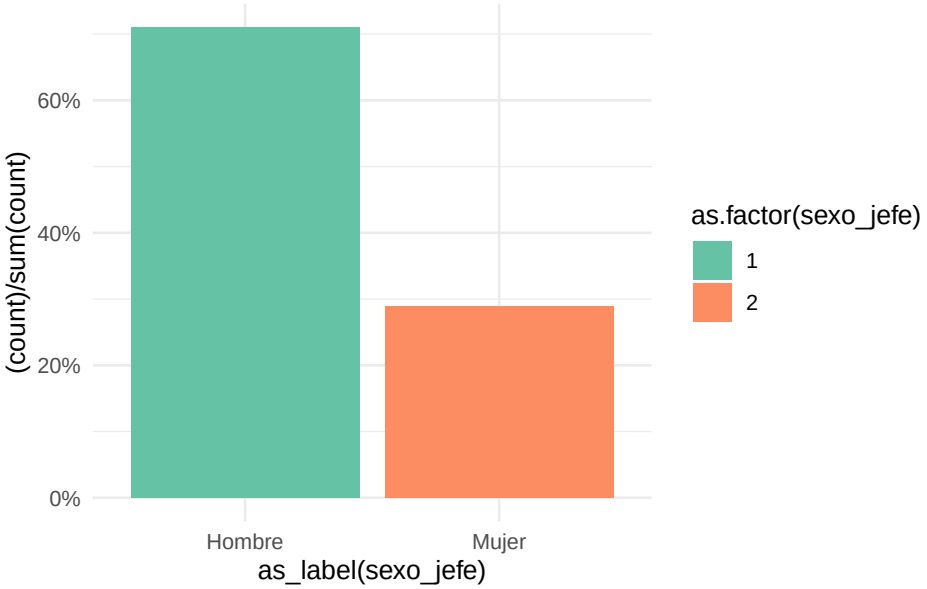
Pero, no queremos el conteo, sino los porcentajes

```
g<- concentrado2020 %>%
  ggplot(aes(as_label(sexo_jefe), fill = as.factor(sexo_jefe))) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  scale_y_continuous(labels=scales::percent) +
  theme_minimal()
g
```



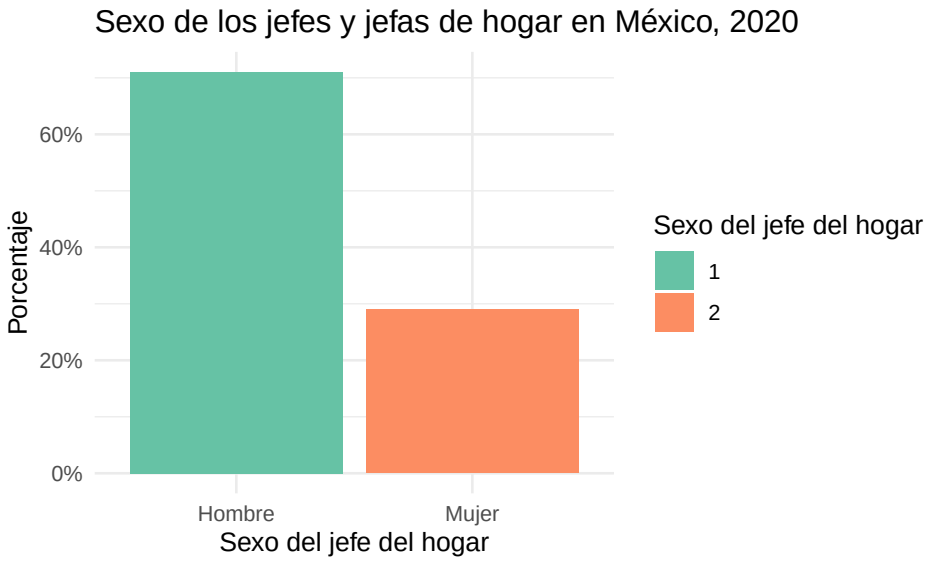
Al gráfico de porcentajes, le agregaremos el color.

```
g +
  scale_fill_brewer(palette="Set2")
```



Ahora, le agregamos los títulos

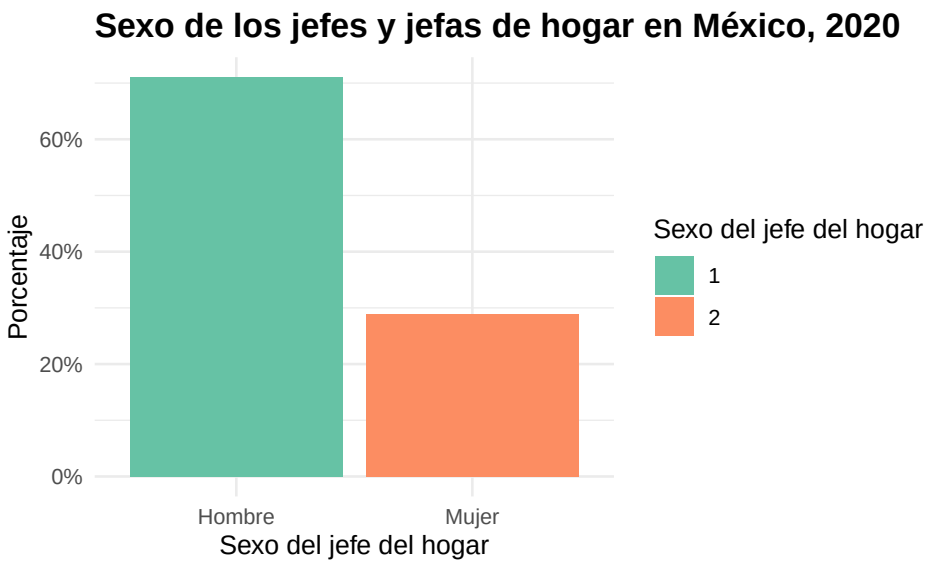
```
g +
  scale_fill_brewer(palette="Set2") +
   labs(
    x = "Sexo del jefe del hogar",
    y = "Porcentaje",
    title = "Sexo de los jefes y jefas de hogar en México, 2020",
    caption = "Fuente: elaboración propia con base en ENIGH 2020",
    fill = "Sexo del jefe del hogar"
  )
```



Y detalles en la fuente de los títulos.

```
concentrado2020 %>%
  ggplot(aes(as_label(sexo_jefe), fill = as.factor(sexo_jefe))) +
  geom_bar(aes(y = (..count..)/sum(..count..))) +
  scale_y_continuous(labels=scales::percent) +
  scale_fill_brewer(palette="Set2") +
   labs(x = "Sexo del jefe del hogar",
    y = "Porcentaje",
    title = "Sexo de los jefes y jefas de hogar en México, 2020",
    caption = "Fuente: elaboración propia con base en ENIGH 2020",
    fill = "Sexo del jefe del hogar") +
```

```
  theme_minimal()+
  theme(plot.title = element_text(color = "black", size = 14, face = "bold"),
    plot.caption = element_text(face = "italic"))
```

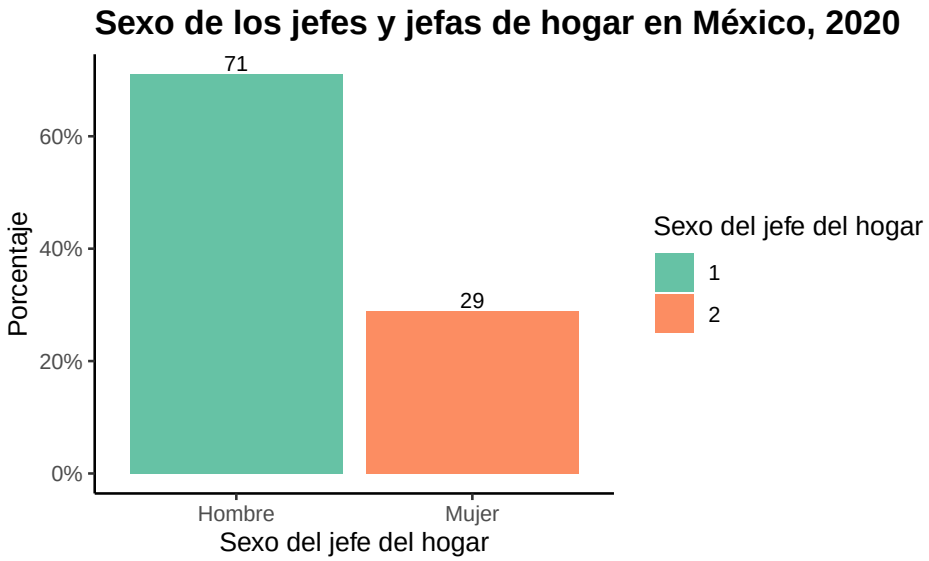


Fuente: elaboración propia con base en ENIGH 2020

Por último, agregaremos las etiquetas de nuestros datos.

```
concentrado2020 %>%
  count(sexo_jefe) %>%
  mutate(pct = prop.table(n)) %>%
  ggplot(aes(x = as_label(sexo_jefe), y = pct, fill = as.factor(sexo_jefe))) +
  geom_col(position = 'dodge') +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette="Set2") +
  theme_classic()+
  geom_text(aes(label = round(pct*100)),
            vjust = -0.2,
            size = 3,
            colour = "black") +
  labs(
    x = "Sexo del jefe del hogar",
    y = "Porcentaje",
    title = "Sexo de los jefes y jefas de hogar en México, 2020",
    caption = "Fuente: elaboración propia con base en ENIGH 2020",
```

```
    fill = "Sexo del jefe del hogar"
  ) +
  theme(
    plot.title = element_text(color = "black", size = 14, face = "bold"),
    plot.caption = element_text(face = "italic")
  )
```



Fuente: elaboración propia con base en ENIGH 2020

8.3 Práctica

Escoge una variable cualitativa y elabora tu gráfico.