2. Procesamiento

La idea de este apartado es entender más a fondo cómo trabajar tablas para poder hacer análisis y visualización. Será necesario ver algunas cuestiones de ejecución básica de `R` y algunas funciones típicas de `Tidyverse` para limpiar y transformar nuestra base de datos. Un pequeño punteo:

- tipos de objetos
- operadores
- funciones de lectura y escritura de archivos
- funciones de resumen y exploración de una base de datos
- funciones de transformación de columnas y valores
- formas típicas de ordenar una base de datos (*wide*, *long*)
- unión y consolidación de distintas tablas

Empecemos.

2.1. Pregunta-problema

Se mantendrá la tradición: definiremos una guía analítica para que las operaciones que hagamos tengan un sentido claro y sea más sencillo entender cuál es la información relevante.

En este caso nos vamos a seguir preguntando por el voto en blanco en las elecciones argentinas. ¿Qué particularidades presenta en las distintas provincias?

2.2. Fuentes de datos

Los datos electorales son una fuente compleja de abordar por el volumen de información que conlleva. El dataset completo está disponible en https://www.argentina.gob.ar/dine/resultados-electorales/elecciones-2023 y vamos a estar utilizando el de las elecciones generales de 2023. No olviden descomprimir el archivo luego de descargarlo. Lo interesante de tomar este tipo de datos es el nivel de desagregación: tenemos resultados por mesa. En los comicios locales, cada mesa pertenece a un circuito, cada circuito pertenence a una sección y cada sección pertenece a un distrito. Por poner un ejemplo: tenemos la mesa **12** en el circuito **12A** pertenenciente a la sección **Comuna 12** del distrito **Capital Federal**. Salvo las mesas, todos los niveles de

agregación tienen un código y un nombre. Por ejemplo, la Provincia de Buenos Aires es el distrito n° 2 y Luján es la sección n° 71.

Vamos primero a cargar la librería de `Tidyverse`. En general, la carga de todas las librerías necesarias se realiza al principio del *script*, por una cuestión de estructura general (sirve para saber si tenemos instaladas todas las necesarias, qué tipo de funciones vamos a utilizar, etc.).

```
library(tidyverse)
```

2.3. Objetos y operadores

En la programación hay distintas lógicas de construcción de lenguajes. `R` se inscribe dentro de la programación orientada a objetos, es decir, cada elemento es tratado como un objeto. No vamos a profundizar, pero nos va a servir saber que una de las características esenciales de los objetos es que pertenencen a una **clase**. ¿Qué implica? La clase del objeto va a determinar un conjunto de propiedades y operaciones que podemos aplicar en dicho objeto. Los tipos o clases de objetos que vamos a estar utilizando son:

Simples

- **character**: texto.
- **numeric**: números.
- **logical** o **bool**: puede ser verdadero (TRUE) o falso (FALSE).

Compuestos

- **vector**: lista de elementos *del mismo tipo*; unidimensional.
- **dataframe**: lo que usualmente llamamos tablas. Tienen filas y columnas.

Hay ciertas palabras y caracteres en los lenguajes de programación que están *reservados* para funciones específicas; en general cambian de color en `Rstudio`. Esas palabras reservadas no se pueden usar para nombrar objetos nuevos, por ejemplo.

```
# veamos las diferencias entre escribir la misma palabra de distintas formas
class(FALSE)
```

```
[1] "logical"
```

```
#class(False)
class("False")
```

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

Un repaso rápido por los operadores más comunes nos muestra los siguientes.

```
# operadores aritméticos
10 + 10 # suma
```

```
[1] 20
```

```
10 - 10 # resta
```

```
[1] 0
```

```
10 * 2 # multiplicación
```

```
[1] 20
```

```
10 / 2 # división
```

```
[1] 5
```

```
10**2 # elevación
```

```
[1] 100
```

```
# operadores de relación
2 > 10 # menor
```

```
[1] FALSE
```

```
10 > 10 # mayor
```

```
[1] FALSE
```

```
10 >= 10 # mayor
```

```
[1] TRUE
```

```
"karl" == "marx" # es igual
```

```
[1] FALSE
```

```
"karl" != "marx" # es distinto
```

```
[1] TRUE
```

```
"marx" %in% c("marx", "durkheim", "weber") # está presente en
```

```
[1] TRUE
```

```
# operadores lógicos
TRUE & FALSE # operador lógico "Y"
```

```
[1] FALSE
```

```
TRUE | FALSE # operador lógico "O"
```

```
[1] TRUE
```

```
# de asignación
a = 2 #asignamos lo que está a la derecha a lo que está a la izquierda del operador
a
```

```
[1] 2
```

```
b <- 3 # idem
b
```

```
[1] 3
```

Carguemos nuestra base de datos para arrancar.

2.4. Procesar resultados electorales

Carguemos sólo 6000 filas, sabiendo que es un dataset muy grande y que quizás se demore mucho cargar toda la base de primera. Veamos una parte para entender la estructura.

```r
ruta <- "data/encuentro_2/2023_Generales/ResultadosElectorales_2023.csv"
data <- read_csv(ruta, n_max=6000) # cargamos sólo 6000 filas
data
```

```
# A tibble: 6,000 x 23
     año eleccion_tipo recuento_tipo padron_tipo distrito_id distrito_nombre
   <dbl> <chr>         <chr>         <chr>              <dbl> <chr>
 1  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 2  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 3  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 4  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 5  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 6  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 7  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 8  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 9  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
10  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
# i 5,990 more rows
# i 17 more variables: seccionprovincial_id <dbl>,
#   seccionprovincial_nombre <chr>, seccion_id <dbl>, seccion_nombre <chr>,
#   circuito_id <chr>, circuito_nombre <chr>, mesa_id <dbl>, mesa_tipo <chr>,
#   mesa_electores <dbl>, cargo_id <dbl>, cargo_nombre <chr>,
#   agrupacion_id <dbl>, agrupacion_nombre <chr>, lista_numero <dbl>,
#   lista_nombre <lgl>, votos_tipo <chr>, votos_cantidad <dbl>
```

La mayor particularidad de los resultados electorales es tener formato **long** (largo). En el formato **long**, cada fila de la tabla representa una observación única para una combinación específica de variables. Es útil para trabajar con datos que necesitan ser agrupados o resumidos fácilmente. Cada variable está dividida en dos columnas: una para el nombre de la variable y otra para su valor. El formato alternativo es **wide**, donde cada fila representa una observación única y cada columna representa una variable diferente. Es útil cuando se necesita acceder rápidamente a las variables individuales sin necesidad de realizar transformaciones adicionales.

LONG

Distrito	Partido	Votos
CABA	Juntos por el Cambio	767.367
CABA	Unión por la Patria	600.832
CABA	La Libertad Avanza	369.424
CABA	FIT-Unidad	66.145
CABA	Hacemos por Nuestro País	57.607

WIDE

Distrito	Juntos por el Cambio	Unión por la Patria	La Libertad Avanza	FIT-Unidad	Hacemos por Nuestro País
CABA	767.367	600.832	369.424	66.145	57.607

Figure 1: formatos de tabla

En la base hay un gran número de columnas, que podemos explorar con la función `names()`. También hay elecciones a distintos cargos, como legislativos.

```
names(data)
```

```
 [1] "año"                       "eleccion_tipo"
 [3] "recuento_tipo"             "padron_tipo"
 [5] "distrito_id"               "distrito_nombre"
 [7] "seccionprovincial_id"      "seccionprovincial_nombre"
 [9] "seccion_id"                "seccion_nombre"
[11] "circuito_id"               "circuito_nombre"
[13] "mesa_id"                   "mesa_tipo"
[15] "mesa_electores"            "cargo_id"
[17] "cargo_nombre"              "agrupacion_id"
[19] "agrupacion_nombre"         "lista_numero"
[21] "lista_nombre"              "votos_tipo"
[23] "votos_cantidad"
```

```
table(data$cargo_nombre)
```

```
PRESIDENTE Y VICE  SENADOR NACIONAL
             5844               156
```

Vamos a cargar la base completa pero usando una función especial para ir filtrando los casos *antes* de cargar la base, para que nos pese menos el objeto y podamos manipularlo con comodidad.

```
f <- function(x, pos){
  filter(x,(cargo_nombre == "PRESIDENTE Y VICE"))
}
data <- read_csv_chunked(ruta, DataFrameCallback$new(f), chunk_size=10000)
dim(data) # vemos cuántas filas y columnas tiene
```

```
[1] 1045200      23
```

```
head(data) # vemos las primeras 5 filas
```

```
# A tibble: 6 x 23
    año eleccion_tipo recuento_tipo padron_tipo distrito_id distrito_nombre
  <dbl> <chr>         <chr>         <chr>              <dbl> <chr>
1  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de ~
2  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de ~
3  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de ~
4  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de ~
5  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de ~
6  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de ~
# i 17 more variables: seccionprovincial_id <dbl>,
#   seccionprovincial_nombre <chr>, seccion_id <dbl>, seccion_nombre <chr>,
#   circuito_id <chr>, circuito_nombre <chr>, mesa_id <dbl>, mesa_tipo <chr>,
#   mesa_electores <dbl>, cargo_id <dbl>, cargo_nombre <chr>,
#   agrupacion_id <dbl>, agrupacion_nombre <chr>, lista_numero <dbl>,
#   lista_nombre <lgl>, votos_tipo <chr>, votos_cantidad <dbl>
```

2.5. Exploración y recorte

Probablemente en la mayoría de los casos una base de datos venga con más información de la necesaria. Una de nuestras primeras tareas en el procesamiento de una base de datos va a ser recortar para quedarnos sólo con las filas y columnas que nos sirven. Tomar esas definiciones va a requerir un primer paneo general de la información que contiene la base de datos.

Existen funciones que nos van a resumir, a grandes rasgos, la totalidad de la base.

```
glimpse(data) # ¡hay otras de este tipo en el primer encuentro!
```

```
Rows: 1,045,200
Columns: 23
$ año                     <dbl> 2023, 2023, 2023, 2023, 2023, 2023, 2023, 202~
$ eleccion_tipo           <chr> "GENERAL", "GENERAL", "GENERAL", "GENERAL", "~
$ recuento_tipo           <chr> "PROVISORIO", "PROVISORIO", "PROVISORIO", "PR~
$ padron_tipo             <chr> "NORMAL", "NORMAL", "NORMAL", "NORMAL", "NORM~
$ distrito_id             <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ distrito_nombre         <chr> "Ciudad Autónoma de Buenos Aires", "Ciudad Au~
$ seccionprovincial_id    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ seccionprovincial_nombre <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
$ seccion_id              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ seccion_nombre          <chr> "Comuna 01", "Comuna 01", "Comuna 01", "Comun~
$ circuito_id             <chr> "00018", "00018", "00018", "00018", "00018", ~
$ circuito_nombre         <chr> "00018", "00018", "00018", "00018", "00018", ~
```

```
$ mesa_id                 <dbl> 474, 474, 474, 474, 475, 475, 475, 475, 475, ~
$ mesa_tipo               <chr> "NATIVOS", "NATIVOS", "NATIVOS", "NATIVOS", "~
$ mesa_electores          <dbl> 343, 343, 343, 343, 349, 349, 349, 349, 349, ~
$ cargo_id                <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ cargo_nombre            <chr> "PRESIDENTE Y VICE", "PRESIDENTE Y VICE", "PR~
$ agrupacion_id           <dbl> 0, 0, 0, 0, 134, 132, 135, 136, 133, 0, 0, 0,~
$ agrupacion_nombre       <chr> NA, NA, NA, NA, "UNION POR LA PATRIA", "JUNTO~
$ lista_numero            <dbl> 0, 0, 0, 0, NA, NA, NA, NA, NA, 0, 0, 0, 0, 0~
$ lista_nombre            <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
$ votos_tipo              <chr> "NULO", "IMPUGNADO", "RECURRIDO", "COMANDO", ~
$ votos_cantidad          <dbl> 0, 0, 0, 0, 95, 59, 57, 9, 4, 4, 3, 0, 0, 0, ~
```

Quizás queremos ver algunas filas completas.

```
head(data, 10) # un top N de filas
```

```
# A tibble: 10 x 23
     año eleccion_tipo recuento_tipo padron_tipo distrito_id distrito_nombre
   <dbl> <chr>         <chr>         <chr>             <dbl> <chr>
 1  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 2  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 3  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 4  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 5  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 6  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 7  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 8  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 9  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
10  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
# i 17 more variables: seccionprovincial_id <dbl>,
#   seccionprovincial_nombre <chr>, seccion_id <dbl>, seccion_nombre <chr>,
#   circuito_id <chr>, circuito_nombre <chr>, mesa_id <dbl>, mesa_tipo <chr>,
#   mesa_electores <dbl>, cargo_id <dbl>, cargo_nombre <chr>,
#   agrupacion_id <dbl>, agrupacion_nombre <chr>, lista_numero <dbl>,
#   lista_nombre <lgl>, votos_tipo <chr>, votos_cantidad <dbl>
```

```
tail(data, 10) # las últimas N filas
```

```
# A tibble: 10 x 23
     año eleccion_tipo recuento_tipo padron_tipo distrito_id distrito_nombre
   <dbl> <chr>         <chr>         <chr>             <dbl> <chr>
```

```
 1  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 2  2023 GENERAL       PROVISORIO    NORMAL               17 Salta
 3  2023 GENERAL       PROVISORIO    NORMAL               23 Tucumán
 4  2023 GENERAL       PROVISORIO    NORMAL               10 Jujuy
 5  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 6  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 7  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 8  2023 GENERAL       PROVISORIO    NORMAL               22 Santiago del Estero
 9  2023 GENERAL       PROVISORIO    NORMAL               14 Misiones
10  2023 GENERAL       PROVISORIO    NORMAL               18 San Juan
# i 17 more variables: seccionprovincial_id <dbl>,
#   seccionprovincial_nombre <chr>, seccion_id <dbl>, seccion_nombre <chr>,
#   circuito_id <chr>, circuito_nombre <chr>, mesa_id <dbl>, mesa_tipo <chr>,
#   mesa_electores <dbl>, cargo_id <dbl>, cargo_nombre <chr>,
#   agrupacion_id <dbl>, agrupacion_nombre <chr>, lista_numero <dbl>,
#   lista_nombre <lgl>, votos_tipo <chr>, votos_cantidad <dbl>
```

```r
sample_n(data, 10) # 10 filas al azar
```

```
# A tibble: 10 x 23
     año eleccion_tipo recuento_tipo padron_tipo distrito_id distrito_nombre
   <dbl> <chr>         <chr>         <chr>             <dbl> <chr>
 1  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 2  2023 GENERAL       PROVISORIO    NORMAL               13 Mendoza
 3  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 4  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 5  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 6  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 7  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 8  2023 GENERAL       PROVISORIO    NORMAL               18 San Juan
 9  2023 GENERAL       PROVISORIO    NORMAL                4 Córdoba
10  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
# i 17 more variables: seccionprovincial_id <dbl>,
#   seccionprovincial_nombre <chr>, seccion_id <dbl>, seccion_nombre <chr>,
#   circuito_id <chr>, circuito_nombre <chr>, mesa_id <dbl>, mesa_tipo <chr>,
#   mesa_electores <dbl>, cargo_id <dbl>, cargo_nombre <chr>,
#   agrupacion_id <dbl>, agrupacion_nombre <chr>, lista_numero <dbl>,
#   lista_nombre <lgl>, votos_tipo <chr>, votos_cantidad <dbl>
```

Quizás queremos *contar* apariciones de algunos valores.

```r
data$votos_tipo %>% # contamos apariciones de tipo de votos hay, en absolutos
  table()
```

.

```
  COMANDO EN BLANCO IMPUGNADO      NULO  POSITIVO RECURRIDO
   104520    104520    104520    104520    522600    104520
```

```r
data$distrito_nombre %>% # contamos % de apariciones de distritos
  table() %>%
  prop.table() %>%
  round(digits=4)*100
```

.

```
                                    Buenos Aires
                                           36.43
                                       Catamarca
                                            1.01
                                           Chaco
                                            2.82
                                          Chubut
                                            1.35
                         Ciudad Autónoma de Buenos Aires
                                            7.01
                                         Córdoba
                                            8.70
                                       Corrientes
                                            2.64
                                       Entre Ríos
                                            3.28
                                         Formosa
                                            1.42
                                           Jujuy
                                            1.75
                                      La Pampa
                                            0.86
                                      La Rioja
                                            0.91
                                       Mendoza
                                            4.17
                                      Misiones
                                            2.75
```

```
                                        Neuquén
                                           1.60
                                      Río Negro
                                           1.71
                                          Salta
                                           3.15
                                       San Juan
                                           1.73
                                       San Luis
                                           1.25
                                     Santa Cruz
                                           0.87
                                       Santa Fe
                                           7.97
                              Santiago del Estero
                                           2.36
Tierra del Fuego, Antártida e Islas del Atlántico Sur
                                           0.49
                                      Tucumán
                                           3.75
```

Quizás queremos saber, de manera más general, cuántos valores únicos tiene cada una de las columnas.

```
sapply(data, n_distinct) # sapply sirve para aplicar una función a todas las columnas
```

```
                  año            eleccion_tipo            recuento_tipo
                    1                        1                        1
         padron_tipo              distrito_id          distrito_nombre
                    1                       24                       24
seccionprovincial_id seccionprovincial_nombre               seccion_id
                    9                       12                      135
      seccion_nombre              circuito_id          circuito_nombre
                  446                     2335                     2335
            mesa_id                mesa_tipo           mesa_electores
                 9097                        1                      378
          cargo_id              cargo_nombre            agrupacion_id
                    1                        1                        6
   agrupacion_nombre             lista_numero             lista_nombre
                    6                        2                        1
         votos_tipo           votos_cantidad
                    6                      264
```

Puntos interesantes hasta este punto: hay más de un millón de filas; cada una representa un tipo de voto en una mesa en particular; cada nivel de geográfico (distrito, sección, circuito), el cargo y la agrupación tiene id y nombre; al menos 8 columnas tienen un sólo valor, por lo que no nos sirven para esta base de datos.

Para seleccionar filas y columnas vamos a utilizar las funciones `filter()` y `select()`, dos estrellas de `Tidyverse`.

```
# filter recibe un dataframe y una condición
data %>%
  filter(distrito_nombre == "Buenos Aires") # filtramos sólo los casos de Buenos Aires
```

```
# A tibble: 380,740 x 23
     año eleccion_tipo recuento_tipo padron_tipo distrito_id distrito_nombre
   <dbl> <chr>         <chr>         <chr>              <dbl> <chr>
 1  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
 2  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
 3  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
 4  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
 5  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
 6  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
 7  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
 8  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
 9  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
10  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
# i 380,730 more rows
# i 17 more variables: seccionprovincial_id <dbl>,
#   seccionprovincial_nombre <chr>, seccion_id <dbl>, seccion_nombre <chr>,
#   circuito_id <chr>, circuito_nombre <chr>, mesa_id <dbl>, mesa_tipo <chr>,
#   mesa_electores <dbl>, cargo_id <dbl>, cargo_nombre <chr>,
#   agrupacion_id <dbl>, agrupacion_nombre <chr>, lista_numero <dbl>,
#   lista_nombre <lgl>, votos_tipo <chr>, votos_cantidad <dbl>
```

```
# podríamos querer filtrar por más de una condición
data %>%
  filter(distrito_nombre == "Buenos Aires" & votos_cantidad > 200) # ¿por qué haríamos eso? s
```

```
# A tibble: 54 x 23
     año eleccion_tipo recuento_tipo padron_tipo distrito_id distrito_nombre
   <dbl> <chr>         <chr>         <chr>              <dbl> <chr>
 1  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
 2  2023 GENERAL       PROVISORIO    NORMAL                 2 Buenos Aires
```

```
 3  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 4  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 5  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 6  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 7  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 8  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
 9  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
10  2023 GENERAL       PROVISORIO    NORMAL                2 Buenos Aires
# i 44 more rows
# i 17 more variables: seccionprovincial_id <dbl>,
#   seccionprovincial_nombre <chr>, seccion_id <dbl>, seccion_nombre <chr>,
#   circuito_id <chr>, circuito_nombre <chr>, mesa_id <dbl>, mesa_tipo <chr>,
#   mesa_electores <dbl>, cargo_id <dbl>, cargo_nombre <chr>,
#   agrupacion_id <dbl>, agrupacion_nombre <chr>, lista_numero <dbl>,
#   lista_nombre <lgl>, votos_tipo <chr>, votos_cantidad <dbl>
```

```r
# podemos jugar con otros operadores lógicos
data %>%
  filter(distrito_nombre=="Buenos Aires" | votos_tipo != "POSITIVO")
```

```
# A tibble: 712,970 x 23
     año eleccion_tipo recuento_tipo padron_tipo distrito_id distrito_nombre
   <dbl> <chr>         <chr>         <chr>             <dbl> <chr>
 1  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 2  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 3  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 4  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 5  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 6  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 7  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 8  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
 9  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
10  2023 GENERAL       PROVISORIO    NORMAL                1 Ciudad Autónoma de~
# i 712,960 more rows
# i 17 more variables: seccionprovincial_id <dbl>,
#   seccionprovincial_nombre <chr>, seccion_id <dbl>, seccion_nombre <chr>,
#   circuito_id <chr>, circuito_nombre <chr>, mesa_id <dbl>, mesa_tipo <chr>,
#   mesa_electores <dbl>, cargo_id <dbl>, cargo_nombre <chr>,
#   agrupacion_id <dbl>, agrupacion_nombre <chr>, lista_numero <dbl>,
#   lista_nombre <lgl>, votos_tipo <chr>, votos_cantidad <dbl>
```

```r
data %>%
  filter(distrito_nombre %in% c("Buenos Aires","Ciudad Autónoma de Buenos Aires"))
```

```
# A tibble: 454,000 x 23
     año eleccion_tipo recuento_tipo padron_tipo distrito_id distrito_nombre
   <dbl> <chr>         <chr>         <chr>              <dbl> <chr>
 1  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 2  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 3  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 4  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 5  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 6  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 7  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 8  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
 9  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
10  2023 GENERAL       PROVISORIO    NORMAL                 1 Ciudad Autónoma de~
# i 453,990 more rows
# i 17 more variables: seccionprovincial_id <dbl>,
#   seccionprovincial_nombre <chr>, seccion_id <dbl>, seccion_nombre <chr>,
#   circuito_id <chr>, circuito_nombre <chr>, mesa_id <dbl>, mesa_tipo <chr>,
#   mesa_electores <dbl>, cargo_id <dbl>, cargo_nombre <chr>,
#   agrupacion_id <dbl>, agrupacion_nombre <chr>, lista_numero <dbl>,
#   lista_nombre <lgl>, votos_tipo <chr>, votos_cantidad <dbl>
```

Para seleccionar, `select()` nos da una diversidad muy interesante e intuitiva de opciones.

```r
# podemos querer seleccionar una columna específica
data %>%
  select(distrito_id)
```

```
# A tibble: 1,045,200 x 1
   distrito_id
         <dbl>
 1           1
 2           1
 3           1
 4           1
 5           1
 6           1
 7           1
 8           1
```

```
 9           1
10           1
# i 1,045,190 more rows
```

```
# o varias columnas
data %>%
  select(distrito_id, distrito_nombre)
```

```
# A tibble: 1,045,200 x 2
   distrito_id distrito_nombre
         <dbl> <chr>
 1           1 Ciudad Autónoma de Buenos Aires
 2           1 Ciudad Autónoma de Buenos Aires
 3           1 Ciudad Autónoma de Buenos Aires
 4           1 Ciudad Autónoma de Buenos Aires
 5           1 Ciudad Autónoma de Buenos Aires
 6           1 Ciudad Autónoma de Buenos Aires
 7           1 Ciudad Autónoma de Buenos Aires
 8           1 Ciudad Autónoma de Buenos Aires
 9           1 Ciudad Autónoma de Buenos Aires
10           1 Ciudad Autónoma de Buenos Aires
# i 1,045,190 more rows
```

```
# si lo pasamos como vector, hay que incluir las comillas
data %>%
  select(c("distrito_id", "distrito_nombre"))
```

```
# A tibble: 1,045,200 x 2
   distrito_id distrito_nombre
         <dbl> <chr>
 1           1 Ciudad Autónoma de Buenos Aires
 2           1 Ciudad Autónoma de Buenos Aires
 3           1 Ciudad Autónoma de Buenos Aires
 4           1 Ciudad Autónoma de Buenos Aires
 5           1 Ciudad Autónoma de Buenos Aires
 6           1 Ciudad Autónoma de Buenos Aires
 7           1 Ciudad Autónoma de Buenos Aires
 8           1 Ciudad Autónoma de Buenos Aires
 9           1 Ciudad Autónoma de Buenos Aires
10           1 Ciudad Autónoma de Buenos Aires
# i 1,045,190 more rows
```

```r
# podemos querer filtrar un subconjunto continuo de columnas
data %>%
  select(distrito_id:lista_nombre)
```

```
# A tibble: 1,045,200 x 17
   distrito_id distrito_nombre       seccionprovincial_id seccionprovincial_no~1
         <dbl> <chr>                                <dbl> <chr>
 1           1 Ciudad Autónoma de B~                    0 <NA>
 2           1 Ciudad Autónoma de B~                    0 <NA>
 3           1 Ciudad Autónoma de B~                    0 <NA>
 4           1 Ciudad Autónoma de B~                    0 <NA>
 5           1 Ciudad Autónoma de B~                    0 <NA>
 6           1 Ciudad Autónoma de B~                    0 <NA>
 7           1 Ciudad Autónoma de B~                    0 <NA>
 8           1 Ciudad Autónoma de B~                    0 <NA>
 9           1 Ciudad Autónoma de B~                    0 <NA>
10           1 Ciudad Autónoma de B~                    0 <NA>
# i 1,045,190 more rows
# i abbreviated name: 1: seccionprovincial_nombre
# i 13 more variables: seccion_id <dbl>, seccion_nombre <chr>,
#   circuito_id <chr>, circuito_nombre <chr>, mesa_id <dbl>, mesa_tipo <chr>,
#   mesa_electores <dbl>, cargo_id <dbl>, cargo_nombre <chr>,
#   agrupacion_id <dbl>, agrupacion_nombre <chr>, lista_numero <dbl>,
#   lista_nombre <lgl>
```

```r
# o podemos querer negar una condición
data %>%
  select(!distrito_id:lista_nombre)
```

```
# A tibble: 1,045,200 x 6
     año eleccion_tipo recuento_tipo padron_tipo votos_tipo votos_cantidad
   <dbl> <chr>         <chr>         <chr>       <chr>                <dbl>
 1  2023 GENERAL       PROVISORIO    NORMAL      NULO                     0
 2  2023 GENERAL       PROVISORIO    NORMAL      IMPUGNADO                0
 3  2023 GENERAL       PROVISORIO    NORMAL      RECURRIDO                0
 4  2023 GENERAL       PROVISORIO    NORMAL      COMANDO                  0
 5  2023 GENERAL       PROVISORIO    NORMAL      POSITIVO                95
 6  2023 GENERAL       PROVISORIO    NORMAL      POSITIVO                59
 7  2023 GENERAL       PROVISORIO    NORMAL      POSITIVO                57
 8  2023 GENERAL       PROVISORIO    NORMAL      POSITIVO                 9
 9  2023 GENERAL       PROVISORIO    NORMAL      POSITIVO                 4
```

```
10  2023 GENERAL       PROVISORIO    NORMAL      EN BLANCO                4
# i 1,045,190 more rows
```

```
# podemos filtrar según el nombre de la columna
data %>%
  select(ends_with("nombre"))
```

```
# A tibble: 1,045,200 x 7
   distrito_nombre         seccionprovincial_no~1 seccion_nombre circuito_nombre
   <chr>                   <chr>                  <chr>          <chr>
 1 Ciudad Autónoma de Bue~ <NA>                   Comuna 01      00018
 2 Ciudad Autónoma de Bue~ <NA>                   Comuna 01      00018
 3 Ciudad Autónoma de Bue~ <NA>                   Comuna 01      00018
 4 Ciudad Autónoma de Bue~ <NA>                   Comuna 01      00018
 5 Ciudad Autónoma de Bue~ <NA>                   Comuna 01      00018
 6 Ciudad Autónoma de Bue~ <NA>                   Comuna 01      00018
 7 Ciudad Autónoma de Bue~ <NA>                   Comuna 01      00018
 8 Ciudad Autónoma de Bue~ <NA>                   Comuna 01      00018
 9 Ciudad Autónoma de Bue~ <NA>                   Comuna 01      00018
10 Ciudad Autónoma de Bue~ <NA>                   Comuna 01      00018
# i 1,045,190 more rows
# i abbreviated name: 1: seccionprovincial_nombre
# i 3 more variables: cargo_nombre <chr>, agrupacion_nombre <chr>,
#   lista_nombre <lgl>
```

```
# y agregar condiciones
data %>%
  select(starts_with("distrito") | starts_with("seccion"))
```

```
# A tibble: 1,045,200 x 6
   distrito_id distrito_nombre       seccionprovincial_id seccionprovincial_no~1
         <dbl> <chr>                                <dbl> <chr>
 1           1 Ciudad Autónoma de B~                    0 <NA>
 2           1 Ciudad Autónoma de B~                    0 <NA>
 3           1 Ciudad Autónoma de B~                    0 <NA>
 4           1 Ciudad Autónoma de B~                    0 <NA>
 5           1 Ciudad Autónoma de B~                    0 <NA>
 6           1 Ciudad Autónoma de B~                    0 <NA>
 7           1 Ciudad Autónoma de B~                    0 <NA>
 8           1 Ciudad Autónoma de B~                    0 <NA>
 9           1 Ciudad Autónoma de B~                    0 <NA>
```

```
10           1 Ciudad Autónoma de B~                    0 <NA>
# i 1,045,190 more rows
# i abbreviated name: 1: seccionprovincial_nombre
# i 2 more variables: seccion_id <dbl>, seccion_nombre <chr>
```

En este caso, vamos a quedarnos con las columnas que tienen más de 1 valor único. No hará falta filtrar todavía.

```r
cols_sin_valores <- data %>%
  sapply(n_distinct) %>%
  as.data.frame() %>%
  rename(n_distinct_values = 1) %>% # asigno nombre
  filter(n_distinct_values == 1) %>% # filtro
  rownames()

cols_sin_valores
```

```
[1] "año"           "eleccion_tipo" "recuento_tipo" "padron_tipo"
[5] "mesa_tipo"     "cargo_id"      "cargo_nombre"  "lista_nombre"
```

```r
data <- data %>%
  select(!cols_sin_valores) %>%
  select(!ends_with("id")) %>%
  select(!c("lista_numero")) %>%
  select(!(starts_with("seccion") | starts_with("circuito") | starts_with("mesa")))
```

2.6. Limpieza y transformación

Limpiar y transformar una base de datos es un proceso *creativo* que implica adaptar las variables según el uso que se le va a dar. Se dice *creativo* porque, si bien hay ciertas funciones y procesos frecuentes, varía mucho según cada proyecto y origen de la base. `Mutate()` es una de las funciones más importantes para este momento.

`Mutate()` sirve para crear, modificar o eliminar columnas. Es como un gran paragüas donde vamos a insertar las modificaciones.

```r
# podríamos querer crear una variable con un valor único de texto o numérico
data %>%
  mutate(nueva_variable = "valor único",
         nueva_variable_num = 42)
```

```
# A tibble: 1,045,200 x 6
   distrito_nombre    agrupacion_nombre votos_tipo votos_cantidad nueva_variable
   <chr>              <chr>             <chr>               <dbl> <chr>
 1 Ciudad Autónoma d~ <NA>              NULO                    0 valor único
 2 Ciudad Autónoma d~ <NA>              IMPUGNADO               0 valor único
 3 Ciudad Autónoma d~ <NA>              RECURRIDO               0 valor único
 4 Ciudad Autónoma d~ <NA>              COMANDO                 0 valor único
 5 Ciudad Autónoma d~ UNION POR LA PAT~ POSITIVO               95 valor único
 6 Ciudad Autónoma d~ JUNTOS POR EL CA~ POSITIVO               59 valor único
 7 Ciudad Autónoma d~ LA LIBERTAD AVAN~ POSITIVO               57 valor único
 8 Ciudad Autónoma d~ FRENTE DE IZQUIE~ POSITIVO                9 valor único
 9 Ciudad Autónoma d~ HACEMOS POR NUES~ POSITIVO                4 valor único
10 Ciudad Autónoma d~ <NA>              EN BLANCO               4 valor único
# i 1,045,190 more rows
# i 1 more variable: nueva_variable_num <dbl>
```

```
# podríamos querer crear una variable que se desprenda de otras columnas o modificar las exis
data %>%
  mutate(max_votos = max(votos_cantidad),
         votos_div = votos_cantidad / max_votos,
         distrito_nombre_minusc = tolower(distrito_nombre),
         distrito_nombre = toupper(distrito_nombre))
```

```
# A tibble: 1,045,200 x 7
   distrito_nombre         agrupacion_nombre votos_tipo votos_cantidad max_votos
   <chr>                   <chr>             <chr>               <dbl>     <dbl>
 1 CIUDAD AUTÓNOMA DE BUE~ <NA>              NULO                    0       300
 2 CIUDAD AUTÓNOMA DE BUE~ <NA>              IMPUGNADO               0       300
 3 CIUDAD AUTÓNOMA DE BUE~ <NA>              RECURRIDO               0       300
 4 CIUDAD AUTÓNOMA DE BUE~ <NA>              COMANDO                 0       300
 5 CIUDAD AUTÓNOMA DE BUE~ UNION POR LA PAT~ POSITIVO               95       300
 6 CIUDAD AUTÓNOMA DE BUE~ JUNTOS POR EL CA~ POSITIVO               59       300
 7 CIUDAD AUTÓNOMA DE BUE~ LA LIBERTAD AVAN~ POSITIVO               57       300
 8 CIUDAD AUTÓNOMA DE BUE~ FRENTE DE IZQUIE~ POSITIVO                9       300
 9 CIUDAD AUTÓNOMA DE BUE~ HACEMOS POR NUES~ POSITIVO                4       300
10 CIUDAD AUTÓNOMA DE BUE~ <NA>              EN BLANCO               4       300
# i 1,045,190 more rows
# i 2 more variables: votos_div <dbl>, distrito_nombre_minusc <chr>
```

```
# podríamos querer recategorizar una variable
data %>%
```

```
  mutate(votos_cantidad_grupo = case_when(votos_cantidad > 60 ~ "mayor a 60 votos",
                                          votos_cantidad <= 60 ~ "menor o igual a 60 votos"))
```

```
# A tibble: 1,045,200 x 5
   distrito_nombre                 agrupacion_nombre   votos_tipo votos_cantidad
   <chr>                           <chr>               <chr>                <dbl>
 1 Ciudad Autónoma de Buenos Aires <NA>                NULO                     0
 2 Ciudad Autónoma de Buenos Aires <NA>                IMPUGNADO                0
 3 Ciudad Autónoma de Buenos Aires <NA>                RECURRIDO                0
 4 Ciudad Autónoma de Buenos Aires <NA>                COMANDO                  0
 5 Ciudad Autónoma de Buenos Aires UNION POR LA PATRIA POSITIVO                95
 6 Ciudad Autónoma de Buenos Aires JUNTOS POR EL CAMB~ POSITIVO                59
 7 Ciudad Autónoma de Buenos Aires LA LIBERTAD AVANZA  POSITIVO                57
 8 Ciudad Autónoma de Buenos Aires FRENTE DE IZQUIERD~ POSITIVO                 9
 9 Ciudad Autónoma de Buenos Aires HACEMOS POR NUESTR~ POSITIVO                 4
10 Ciudad Autónoma de Buenos Aires <NA>                EN BLANCO                4
# i 1,045,190 more rows
# i 1 more variable: votos_cantidad_grupo <chr>
```

```
data %>%
  mutate(votos_tipo_recat = case_when(votos_tipo %in% c("NULO","IMPUGNADO","RECURRIDO","COMA
                                      .default = as.character(votos_tipo)))
```

```
# A tibble: 1,045,200 x 5
   distrito_nombre  agrupacion_nombre votos_tipo votos_cantidad votos_tipo_recat
   <chr>            <chr>             <chr>               <dbl> <chr>
 1 Ciudad Autónoma~ <NA>              NULO                    0 NEGATIVO
 2 Ciudad Autónoma~ <NA>              IMPUGNADO               0 NEGATIVO
 3 Ciudad Autónoma~ <NA>              RECURRIDO               0 NEGATIVO
 4 Ciudad Autónoma~ <NA>              COMANDO                 0 NEGATIVO
 5 Ciudad Autónoma~ UNION POR LA PAT~ POSITIVO               95 POSITIVO
 6 Ciudad Autónoma~ JUNTOS POR EL CA~ POSITIVO               59 POSITIVO
 7 Ciudad Autónoma~ LA LIBERTAD AVAN~ POSITIVO               57 POSITIVO
 8 Ciudad Autónoma~ FRENTE DE IZQUIE~ POSITIVO                9 POSITIVO
 9 Ciudad Autónoma~ HACEMOS POR NUES~ POSITIVO                4 POSITIVO
10 Ciudad Autónoma~ <NA>              EN BLANCO               4 NEGATIVO
# i 1,045,190 more rows
```

```
data %>%
  mutate(votos_agrupado = case_when(votos_tipo %in% c("NULO","IMPUGNADO","RECURRIDO","COMANDO
```

```
                                      .default = as.character(agrupacion_nombre)),
         votos_agrupado_v2 = case_when(is.na(agrupacion_nombre) ~ "NEGATIVO",
                                       .default = as.character(agrupacion_nombre)))
```

```
# A tibble: 1,045,200 x 6
   distrito_nombre    agrupacion_nombre votos_tipo votos_cantidad votos_agrupado
   <chr>              <chr>             <chr>               <dbl> <chr>
 1 Ciudad Autónoma d~ <NA>              NULO                    0 NEGATIVO
 2 Ciudad Autónoma d~ <NA>              IMPUGNADO               0 NEGATIVO
 3 Ciudad Autónoma d~ <NA>              RECURRIDO               0 NEGATIVO
 4 Ciudad Autónoma d~ <NA>              COMANDO                 0 NEGATIVO
 5 Ciudad Autónoma d~ UNION POR LA PAT~ POSITIVO               95 UNION POR LA ~
 6 Ciudad Autónoma d~ JUNTOS POR EL CA~ POSITIVO               59 JUNTOS POR EL~
 7 Ciudad Autónoma d~ LA LIBERTAD AVAN~ POSITIVO               57 LA LIBERTAD A~
 8 Ciudad Autónoma d~ FRENTE DE IZQUIE~ POSITIVO                9 FRENTE DE IZQ~
 9 Ciudad Autónoma d~ HACEMOS POR NUES~ POSITIVO                4 HACEMOS POR N~
10 Ciudad Autónoma d~ <NA>              EN BLANCO               4 NEGATIVO
# i 1,045,190 more rows
# i 1 more variable: votos_agrupado_v2 <chr>
```

```
# vamos a hacer una columna con nombres reducidos para utilizar con comodidad
data <- data %>%
  mutate(votos_agrupado = case_when(is.na(agrupacion_nombre) ~ "NEGATIVO",
                                    .default = as.character(agrupacion_nombre)),
         votos_agrupado_red = case_when(votos_agrupado == "NEGATIVO" ~ "NEG",
                                        votos_agrupado == "UNION POR LA PATRIA" ~ "UXP",
                                        votos_agrupado == "JUNTOS POR EL CAMBIO" ~ "JXC",
                                        votos_agrupado == "LA LIBERTAD AVANZA" ~ "LLA",
                                        votos_agrupado == "FRENTE DE IZQUIERDA Y DE TRABAJADO
                                        votos_agrupado == "HACEMOS POR NUESTRO PAIS" ~ "HNP")
         distrito_nombre_red = stringi::stri_trans_general(distrito_nombre, id="Latin-ASCII")
         distrito_nombre_red = tolower(distrito_nombre_red),
         distrito_nombre_red = gsub(" ", "_", distrito_nombre_red))

data
```

```
# A tibble: 1,045,200 x 7
   distrito_nombre    agrupacion_nombre votos_tipo votos_cantidad votos_agrupado
   <chr>              <chr>             <chr>               <dbl> <chr>
 1 Ciudad Autónoma d~ <NA>              NULO                    0 NEGATIVO
 2 Ciudad Autónoma d~ <NA>              IMPUGNADO               0 NEGATIVO
```

```
 3 Ciudad Autónoma d~ <NA>              RECURRIDO              0 NEGATIVO
 4 Ciudad Autónoma d~ <NA>              COMANDO                0 NEGATIVO
 5 Ciudad Autónoma d~ UNION POR LA PAT~ POSITIVO              95 UNION POR LA ~
 6 Ciudad Autónoma d~ JUNTOS POR EL CA~ POSITIVO              59 JUNTOS POR EL~
 7 Ciudad Autónoma d~ LA LIBERTAD AVAN~ POSITIVO              57 LA LIBERTAD A~
 8 Ciudad Autónoma d~ FRENTE DE IZQUIE~ POSITIVO              9 FRENTE DE IZQ~
 9 Ciudad Autónoma d~ HACEMOS POR NUES~ POSITIVO              4 HACEMOS POR N~
10 Ciudad Autónoma d~ <NA>              EN BLANCO              4 NEGATIVO
# i 1,045,190 more rows
# i 2 more variables: votos_agrupado_red <chr>, distrito_nombre_red <chr>
```

2.7. Formatos

Quizás sea más cómodo alternar entre formato *long* y *wide*. Para aquellos casos, `pivot_longer()` y `pivot_wider()` nos van a servir.

```
data_agrupada <- data %>%
  group_by(distrito_nombre, votos_agrupado_red) %>% # como tenemos una fila por mesa/agrupaci
  summarise(votos_cantidad = sum(votos_cantidad))
data_agrupada
```

```
# A tibble: 144 x 3
# Groups:   distrito_nombre [24]
   distrito_nombre votos_agrupado_red votos_cantidad
   <chr>           <chr>                       <dbl>
 1 Buenos Aires    FIT                        352790
 2 Buenos Aires    HNP                        367457
 3 Buenos Aires    JXC                       2374023
 4 Buenos Aires    LLA                       2533633
 5 Buenos Aires    NEG                        346808
 6 Buenos Aires    UXP                       4224688
 7 Catamarca       FIT                          3666
 8 Catamarca       HNP                         14967
 9 Catamarca       JXC                         39960
10 Catamarca       LLA                         74570
# i 134 more rows
```

```
data_wider <- data_agrupada %>%
  pivot_wider(names_from=votos_agrupado_red, values_from=votos_cantidad)

data_wider
```

```
# A tibble: 24 x 7
# Groups:   distrito_nombre [24]
   distrito_nombre                    FIT    HNP     JXC     LLA    NEG     UXP
   <chr>                            <dbl>  <dbl>   <dbl>   <dbl>  <dbl>   <dbl>
 1 Buenos Aires                    352790 367457 2374023 2533633 346808 4224688
 2 Catamarca                         3666  14967   39960   74570  15211   99612
 3 Chaco                             5482  25804  171756  197596  12966  310962
 4 Chubut                           15122  26555   70717  120297  10314  110820
 5 Ciudad Autónoma de Buenos Aires  66145  57607  767367  369424  50056  600832
 6 Corrientes                        7315  19118  224519  187916  15535  260040
 7 Córdoba                          31895 665717  519252  769847  37521  308016
 8 Entre Ríos                       12760  44720  250512  247640  47171  276850
 9 Formosa                           2772   8793   55097  103911   6953  187229
10 Jujuy                            16006  30775   90095  168921  11193  146027
# i 14 more rows
```

Y si quisiéramos volver al formato anterior.

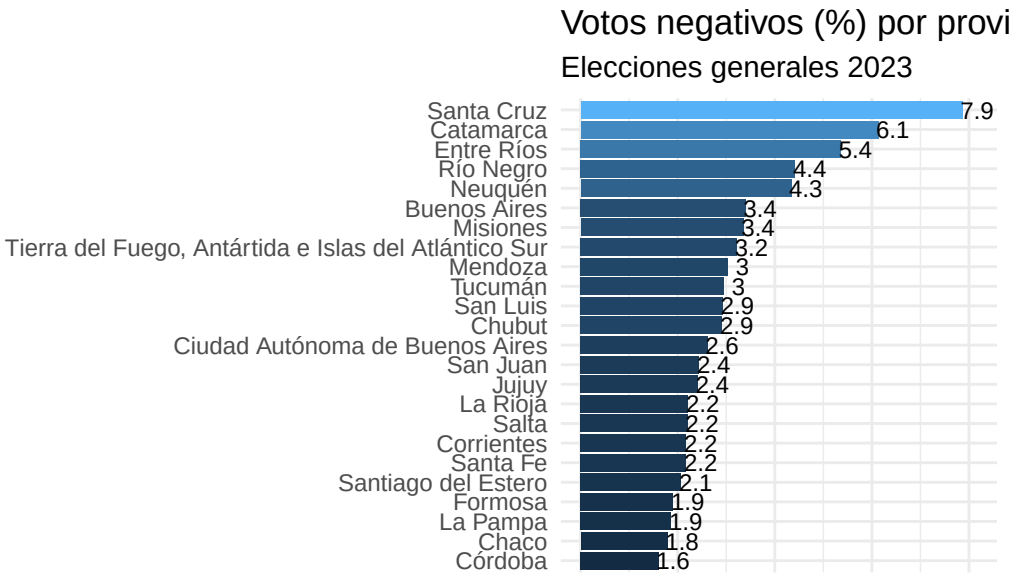
```
data_longer <- data_wider %>%
  pivot_longer(!distrito_nombre)

data_longer
```

```
# A tibble: 144 x 3
# Groups:   distrito_nombre [24]
   distrito_nombre name    value
   <chr>           <chr>   <dbl>
 1 Buenos Aires    FIT    352790
 2 Buenos Aires    HNP    367457
 3 Buenos Aires    JXC   2374023
 4 Buenos Aires    LLA   2533633
 5 Buenos Aires    NEG    346808
 6 Buenos Aires    UXP   4224688
 7 Catamarca       FIT      3666
 8 Catamarca       HNP     14967
 9 Catamarca       JXC     39960
10 Catamarca       LLA     74570
# i 134 more rows
```

Finalmente, veamos el voto negativo (blanco y otros) por provincia.

```r
data_wider %>%
  mutate(TOTAL = rowSums(across()),
         NEG_per = NEG / TOTAL,
         label = round(NEG_per*100,1)) %>%
  ggplot(aes(y=reorder(distrito_nombre, NEG_per), x=NEG_per, fill=NEG_per, label=label)) +
  geom_col(show.legend = FALSE) +
  geom_text(aes(x=NEG_per + 0.003), size=3)+
  labs(x="", y="",
       title="Votos negativos (%) por provincia",
       subtitle="Elecciones generales 2023",
       caption="Elaboración propia según resultados provisorios (DINE)")+
  theme_minimal()+
  theme(axis.text.x=element_blank())
```



2.8. Para seguir practicando

Graficar:

- El voto negativo en **las secciones de Santa Cruz** u otra provincia de preferencia.
- El voto negativo a **en regiones (NEA, NOA, CUYO, PAMPEANA, PATAGONIA)**.

- El voto de **LLA por provincia en absolutos**.
- El voto de **UXP por provincia en porcentaje sobre votos afirmativos (sin contar los negativos)**.