Tidyverse (parte 1)

José Luis Texcalac Sangrador

Procesamiento y visualización de datos espaciales en R

Organizar nuestro trabajo en proyectos

- Los "proyectos en **R**" (.Rproj) constituyen una herramienta útil para construir entornos cerrados de trabajo.
- Un proyecto se genera en RStudio siguiendo la ruta:

```
File  >  New Project
```

- Podemos generar el proyecto en un nuevo directorio o en uno ya existente.

Organizar nuestro trabajo en proyectos

- Los "proyectos en R" (.Rproj) constituyen una herramienta útil para construir entornos cerrados de trabajo.

- Un proyecto se genera en RStudio siguiendo la ruta:

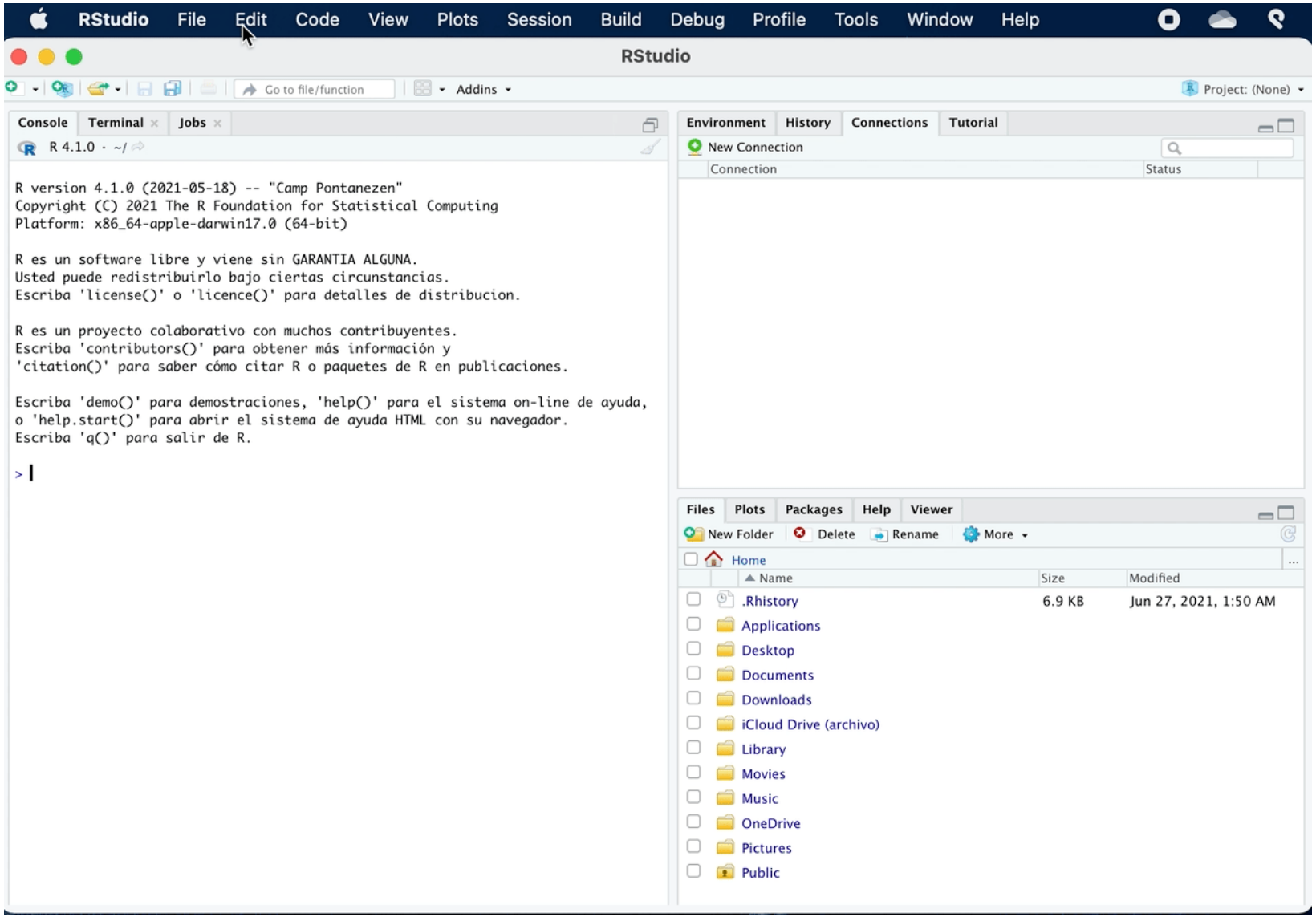
  File > New Project

- Podemos generar el proyecto en un nuevo directorio o en uno ya existente

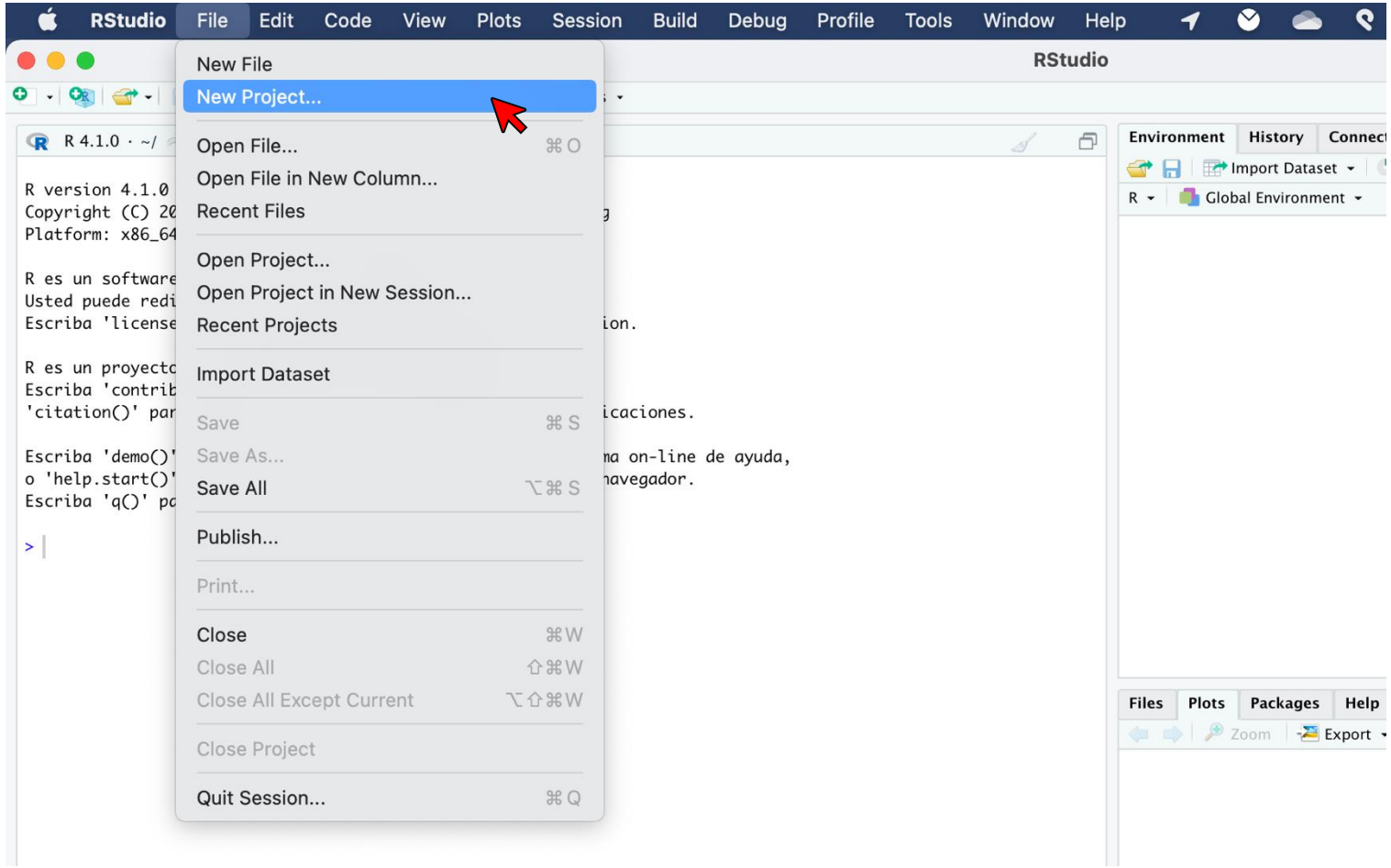
INSP/ESPM
Escuela de Salud Pública de México

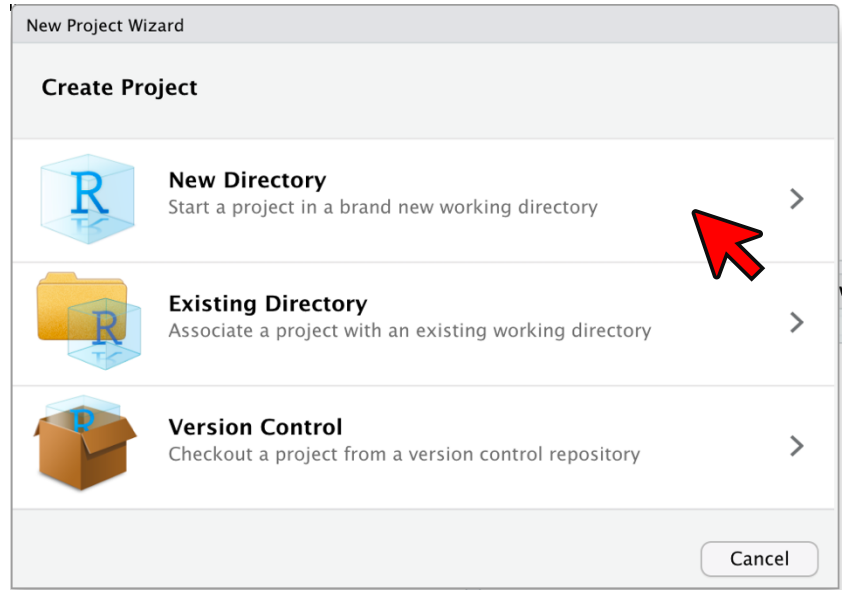
Creando un proyecto en RStudio

En el directorio se creará un archivo con extensión .RProj

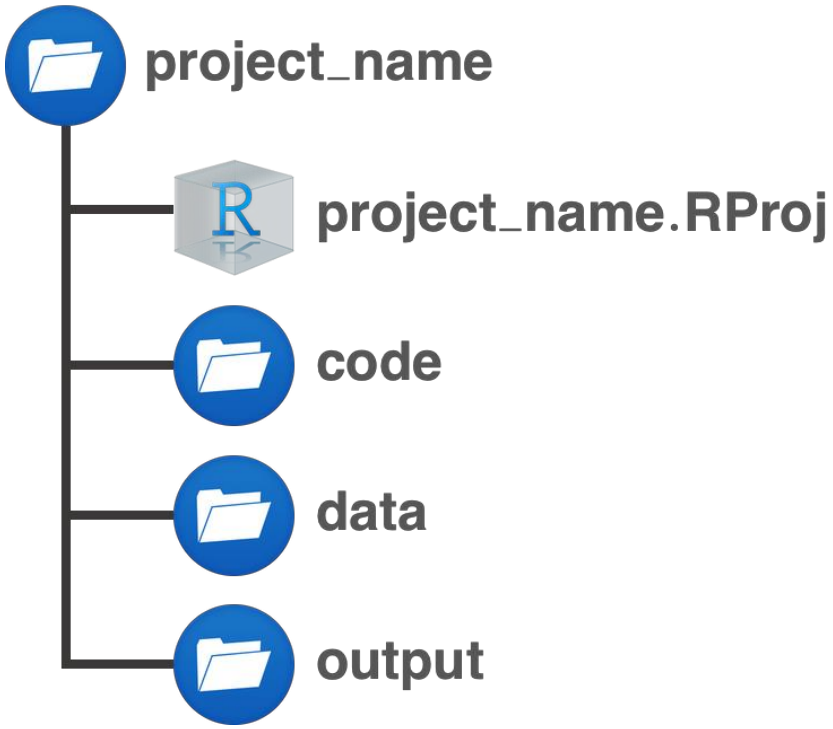


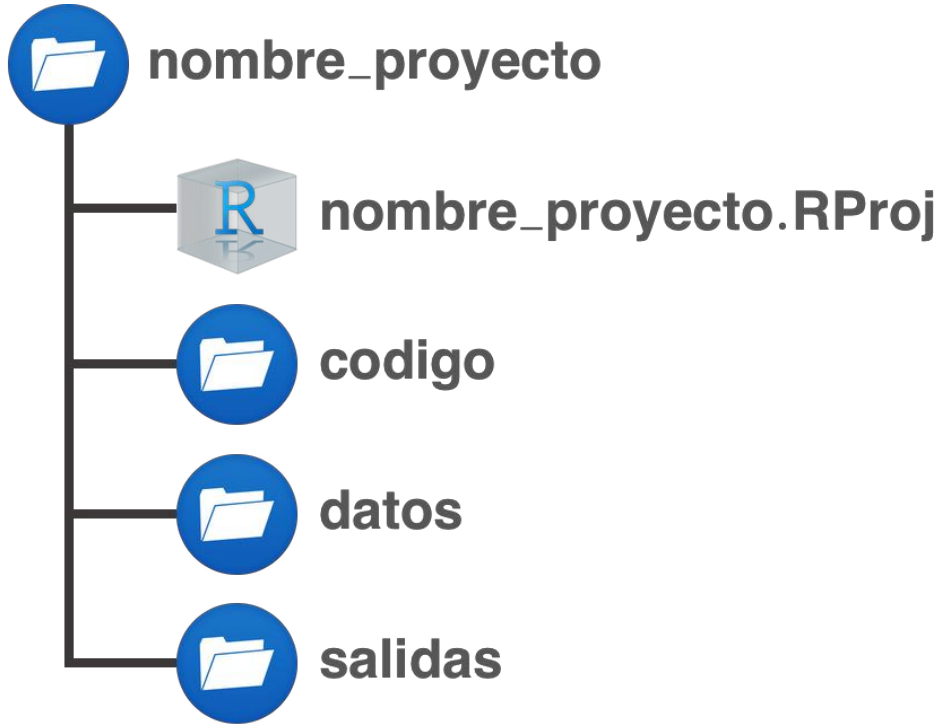
Creando un proyecto en RStudio



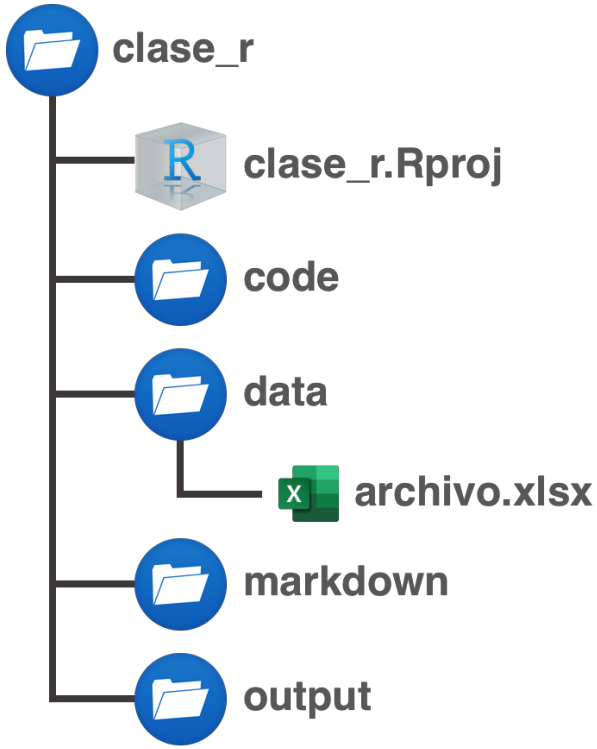


Configuración básica de un proyecto





Rutas de archivo dentro de un proyecto



Llamar un archivo a una sesión sin un proyecto…

```
> read_xlsx("/Users/tex/Documents/insp/clase_r/data/archivo.xlsx")
```

Llamar un archivo a una sesión dentro un proyecto…

```
> read_xlsx("./data/archivo.xlsx")
```

Compartir proyectos

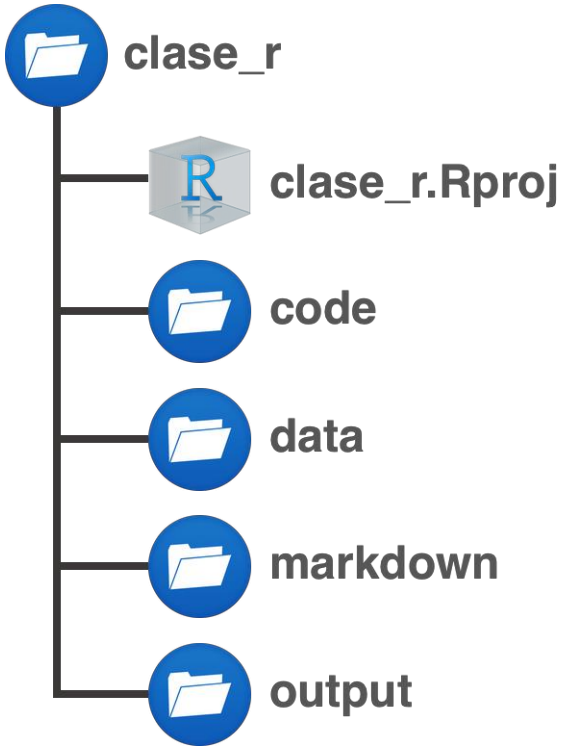
- El directorio es muteable o sea que usted sólo debe compartir la carpeta del proyecto y no habrá necesidad de modificar el código para redirigir rutas de lectura y/o salida.

User > Ana > Maestria > Procesam > clase_r

User > Juan > Docs > Clases > R > clase_r

Su turno...

- Genere una carpeta de nombre clase_r con los siguientes subdirectorios
  - code, data, markdown, output
- Abra RStudio y genere un nuevo proyecto en la carpeta clase_r
- Note que automáticamente se genera un archivo de nombre similar al del directorio raíz "clase_r", ese será entonces el nombre de su proyecto, revise en el explorador.



Importar datos a R

Es posible importar datos en diferentes formatos

- .csv (comma-separated values)
- .dta (Stata-format dataset)
- .dbf (data base format)
- .xlsx (microsoft excel file format)

Su turno

- Descargue los archivos que se le indican
- Archivo iris.dta
  https://www.stata-press.com/data/r10/mvmain.html
- Archivo SUN_2018.csv (información sociodemográfica del SUN 2018)
  https://www.gob.mx/conapo/documentos/sistema-urbano-nacional-2018
- Descargue de google classroom los archivos: 09mun.dbf y datos_pob_2020.xlsx
- Mueva los archivos a la carpeta **data** de su proyecto
- Genere un nuevo script
- Active el paquete **tidyverse**, **haven** y **foreign** en su sesión

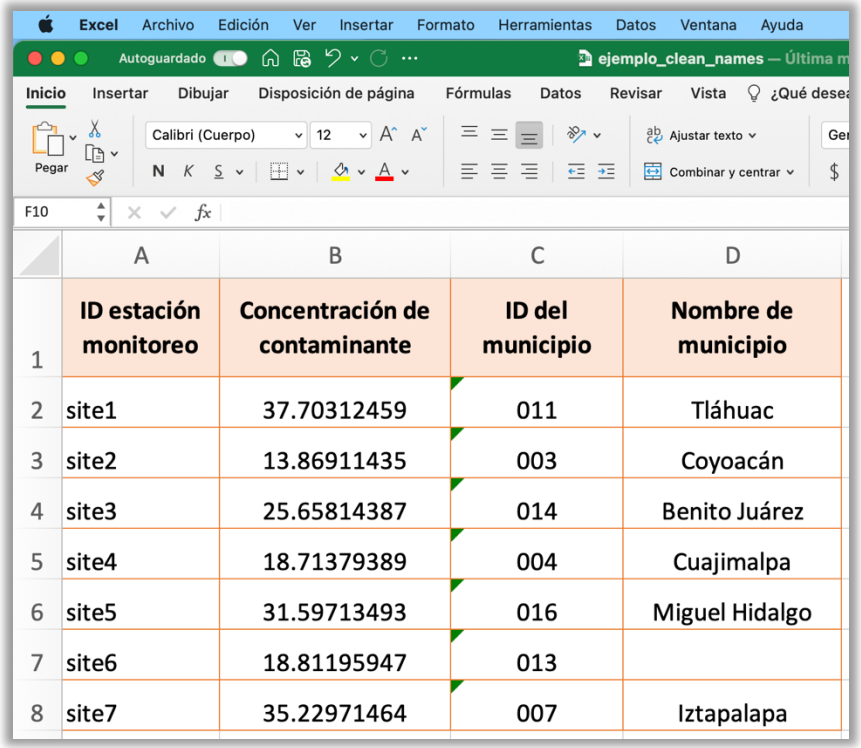

```
cdmx_dbf <-
   read.dbf("./data/09mun.dbf") %>%
   print()
```

```
iris_dta <-
   read_dta("./data/iris.dta") %>%
   print()
```

```
sun_csv <-
   read_csv("./data/SUN_2018.csv") %>%
   print()
```

```
pob_xlsx <-
   read_xlsx("./data/datos_pob_2020.xlsx",
             sheet = "pob_max_mun",
             skip = 2) %>%
   print()
```

Formato de nombre de columnas



	A	B	C	D
1	**ID estación monitoreo**	**Concentración de contaminante**	**ID del municipio**	**Nombre de municipio**
2	site1	37.70312459	011	Tláhuac
3	site2	13.86911435	003	Coyoacán
4	site3	25.65814387	014	Benito Juárez
5	site4	18.71379389	004	Cuajimalpa
6	site5	31.59713493	016	Miguel Hidalgo
7	site6	18.81195947	013	
8	site7	35.22971464	007	Iztapalapa

```
read_xlsx("./data/datos_pob_2020.xlsx",
          sheet = "contam")
```

```
# A tibble: 7 x 4
  `ID estación monitoreo` `Concentración de contaminante` `ID del municipio` `Nombre de municipio`
  <chr>                                             <dbl> <chr>              <chr>
1 site1                                              37.7 011                Tláhuac
2 site2                                              13.9 003                Coyoacán
3 site3                                              25.7 014                Benito Juárez
4 site4                                              18.7 004                Cuajimalpa
5 site5                                              31.6 016                Miguel Hidalgo
6 site6                                              18.8 013                NA
7 site7                                              35.2 007                Iztapalapa
```

```
read_xlsx("./data/datos_pob_2020.xlsx",
          sheet = "contam") %>%
clean_names( ) %>%
print( )
```

janitor:

clean_names( )

```
# A tibble: 7 x 4
  id_estacion_monitoreo concentracion_de_contaminante id_del_municipio nombre_de_municipio
  <chr>                                         <dbl> <chr>            <chr>
1 site1                                          37.7 011              Tláhuac
2 site2                                          13.9 003              Coyoacán
3 site3                                          25.7 014              Benito Juárez
4 site4                                          18.7 004              Cuajimalpa
5 site5                                          31.6 016              Miguel Hidalgo
6 site6                                          18.8 013              NA
7 site7                                          35.2 007              Iztapalapa
```

Snake case: nombre_de_variable, nom_var, n_var, name_var, ~~Nombre_Var, NOM_VAR, NAME~~

https://geeks.ms/jorge/2019/03/24/la-importancia-de-las-convenciones-de-codificacion-pascalcase-camelcase-snake_case-y-kebab-case/

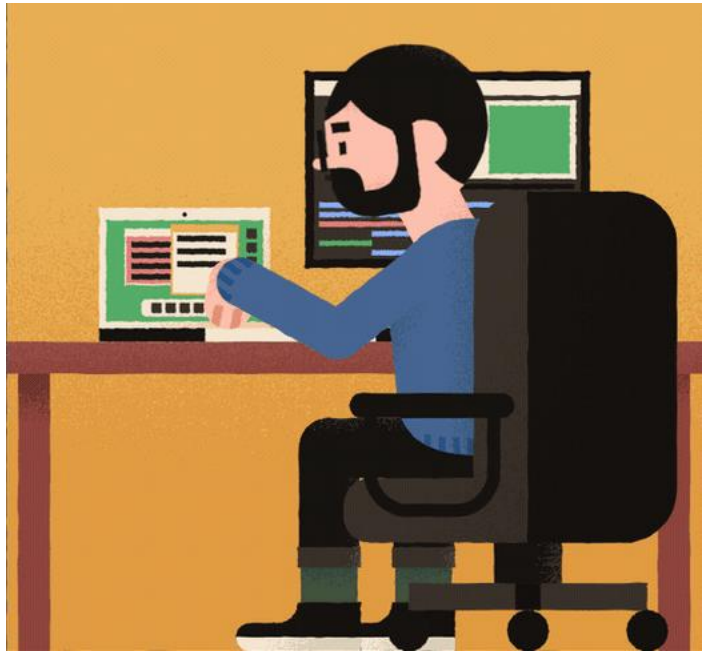
http://programacion.jias.es/2017/09/estandares-de-nomenclatura-snake-case-kebab-case-camel-case/

https://github.com/Tazinho/snakecase

Convención codificación

Tipo codificación	Resultado
camelCase	firstName
PascalCase	FirstName
SnakeCase	first_name
KebabCase	first-name
UpperCase + SnakeCase	FIRST_NAME
lowercase	firstname

Procesamiento de datos



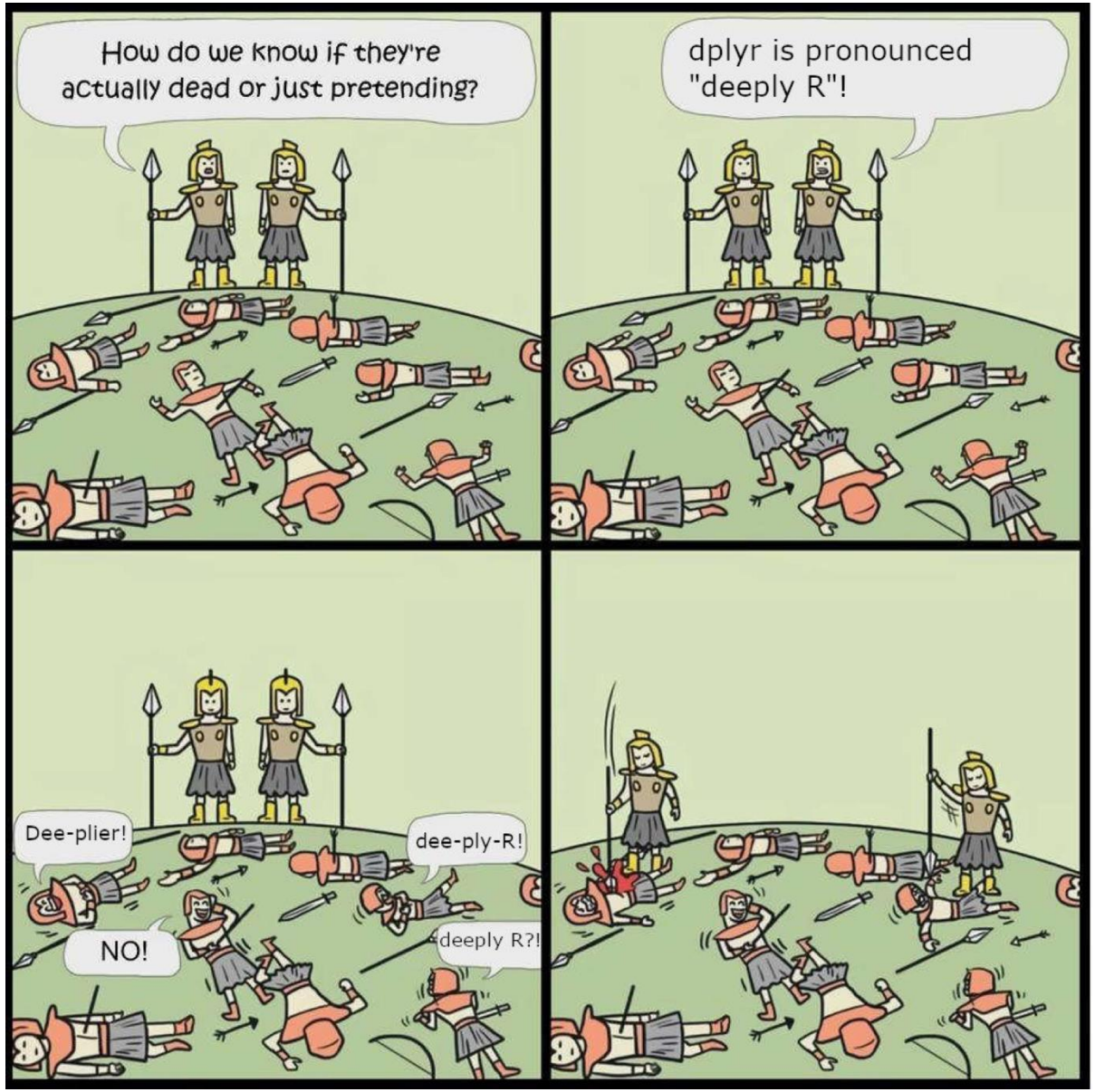
Tidyverse - verbos

Las funciones en Tidyverse están consistentemente diseñadas para trabajar con pipes %>% dónde el nombre del dataset es el primer argumento

```
nombre_obj <-
  malla %>%
  filter(filas) %>%
  select(columnas) %>%
  print()
```

INSP/ESPM
Escuela de Salud Pública de México

- dplyr es un paquete que se encuentra dentro de la colección de paquetes de tidyverse.
- dplyr es un paquete que transforma datos.
- dplyr implementa una gramática para la manipulación de datos ttabulares, proporciona un conjunto de verbos para resolver los desafíos más comunes en la manipulación de datos.



dplyr - manipulación datos

mutate()

select()

filter()

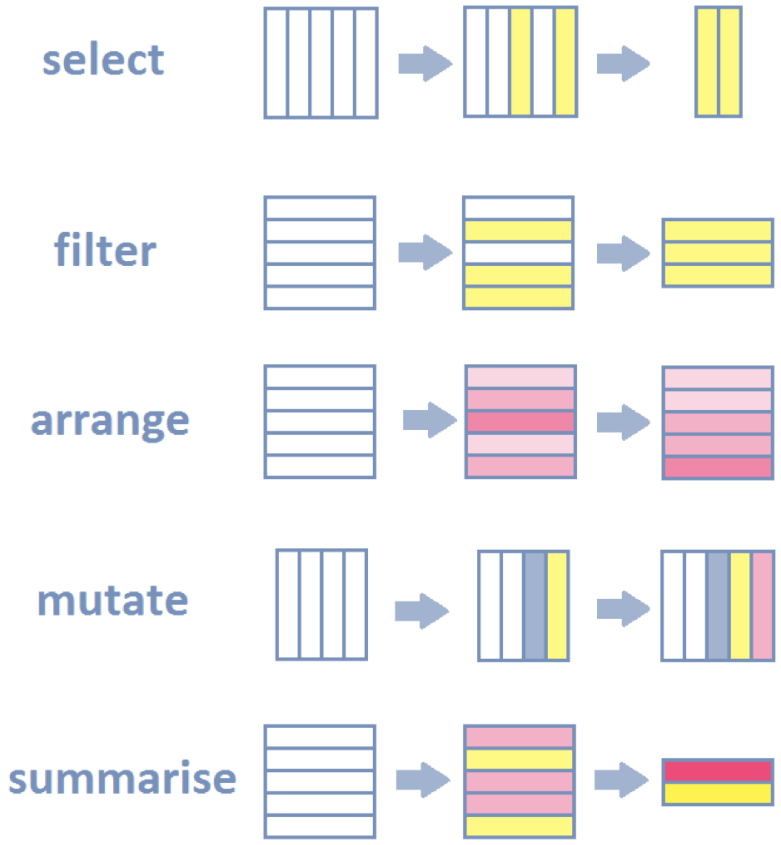
group_by()

summarise()

arrange()

pivot_wide() y pivot_long()

join…

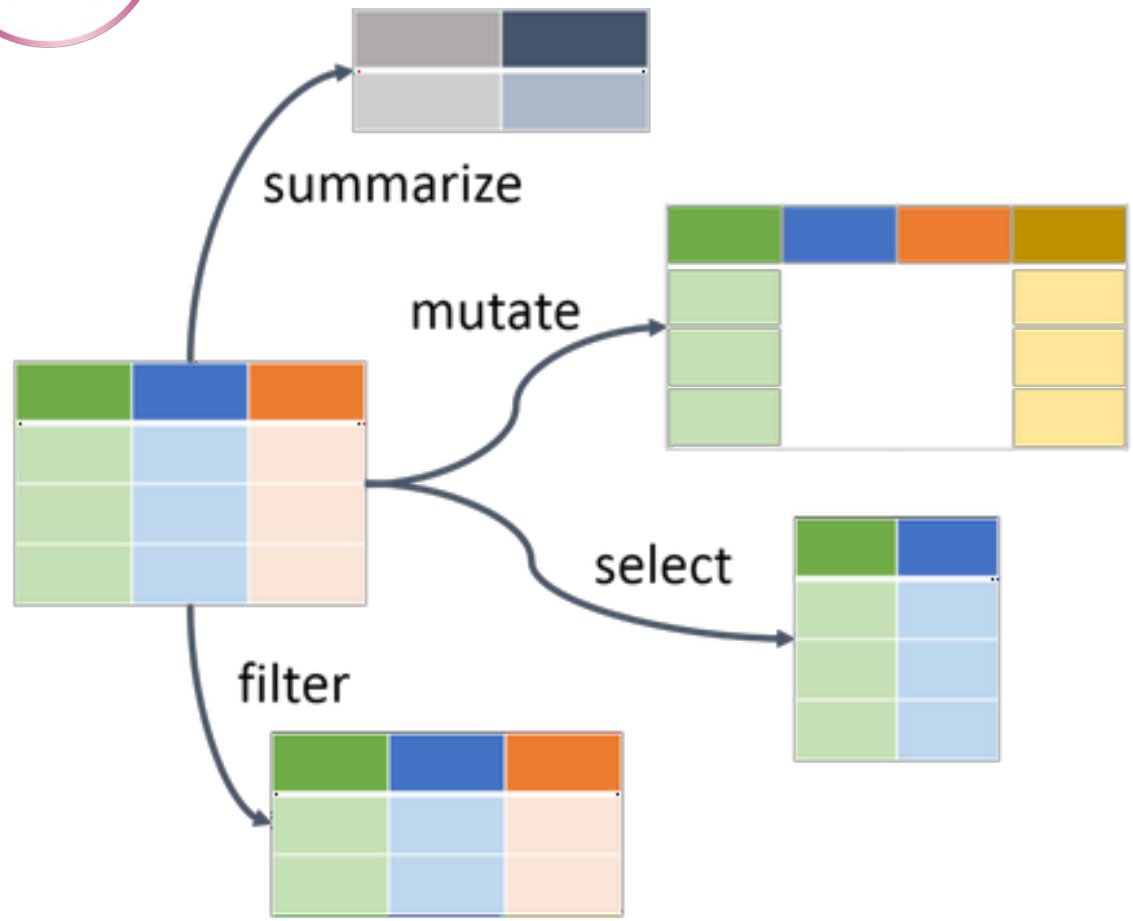


Focalizando la información

filter( ) - Filtrar valores (filas)

select( ) - Seleccionar variables (columnas)

arrange( ) - Ordenar valores



wide

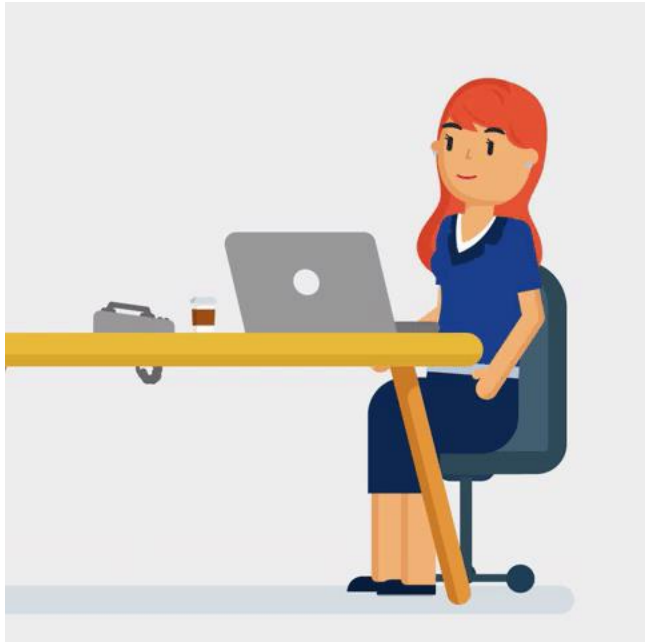
id	x	y	z
1	a	c	e
2	b	d	f

long

id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f

Su turno...

- Navegue a: http://www.aire.cdmx.gob.mx
  - Clic en Estadísticas/Concentraciones
  - Seleccione el reporte de promedios horarios
  - Parámetro: O3
  - Año: 2023
  - Mes: Septiembre
  - Descargue el archivo y mueva a carpeta data de su proyecto
- Importe el archivo a su sesión, nombre al objeto como ozono




```r
ozono <-
  read_csv("./data/Promedio horarios de o3.csv",
           skip = 1,
           na = "nr",
           n_max = 500) %>%
  print()
```

```
New names:
• `` -> `...48`
Rows: 500 Columns: 48
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (1): Fecha
dbl (31): Hora, ACO, AJM, AJU, ATI, BJU, CAM, CCA, MON, CHO, CUA, CUT, FAC, FAR, IZT, LLA, LPR, MER, MGH, MPA, NEZ, PED, SAC, SAG, TAH, TLA,...
lgl (16): AZC, CES, COY, GAM, HGM, INN, LAG, PLA, SFE, SJA, SUR, TAC, TAX, TEC, TPN, ...48

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
# A tibble: 500 × 48
   Fecha      Hora   ACO   AJM   AJU   ATI AZC     BJU   CAM   CCA CES     MON   CHO COY     CUA   CUT   FAC   FAR GAM   HGM   INN     IZT LAG
   <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <lgl> <dbl> <dbl> <dbl> <lgl> <dbl> <dbl> <lgl> <dbl> <dbl> <dbl> <dbl> <lgl> <lgl> <lgl> <dbl> <lgl>
 1 01-09-20…     1    NA    NA    NA    NA NA       NA    NA    NA NA       NA    NA NA       NA    NA    NA    NA NA    NA    NA    NA NA
 2 01-09-20…     2    NA    NA    NA    NA NA       NA    NA    NA NA       NA    NA NA       NA    NA    NA    NA NA    NA    NA    NA NA
 3 01-09-20…     3    NA    NA    NA    NA NA       NA    NA    NA NA       NA    NA NA       NA    NA    NA    NA NA    NA    NA    NA NA
 4 01-09-20…     4    19    NA    25    20 NA       28    22    29 NA       20    33 NA       14    24    16    25 NA    NA    NA    29 NA
 5 01-09-20…     5    11    NA    21    10 NA       27    15    25 NA       18    20 NA        7    18    13    25 NA    NA    NA    25 NA
 6 01-09-20…     6     2    NA    18     4 NA       23     5    23 NA        7    11 NA       16     1    10    16 NA    NA    NA    10 NA
 7 01-09-20…     7     3    NA    16     1 NA       15     3    15 NA        4     8 NA        6     1     2    11 NA    NA    NA     6 NA
 8 01-09-20…     8     6    NA    14     1 NA       10     4    11 NA        5     8 NA        5     3     1     7 NA    NA    NA     8 NA
 9 01-09-20…     9    23    NA    25    13 NA       20    12    15 NA       18    22 NA       14     6    11    23 NA    NA    NA    16 NA
10 01-09-20…    10    40    NA    40    19 NA       29    27    27 NA       37    34 NA       28    15    16    32 NA    NA    NA    27 NA
# ℹ 490 more rows
# ℹ 25 more variables: LLA <dbl>, LPR <dbl>, MER <dbl>, MGH <dbl>, MPA <dbl>, NEZ <dbl>, PED <dbl>, PLA <lgl>, SAC <dbl>, SAG <dbl>, SFE <lgl>,
#   SJA <lgl>, SUR <lgl>, TAC <lgl>, TAH <dbl>, TAX <lgl>, TEC <lgl>, TLA <dbl>, TLI <dbl>, TPN <lgl>, UAX <dbl>, UIZ <dbl>, VIF <dbl>,
#   XAL <dbl>, ...48 <lgl>
# ℹ Use `print(n = ...)` to see more rows
Warning message:
One or more parsing issues, call `problems()` on your data frame for details, e.g.:
  dat <- vroom(...)
  problems(dat)
```

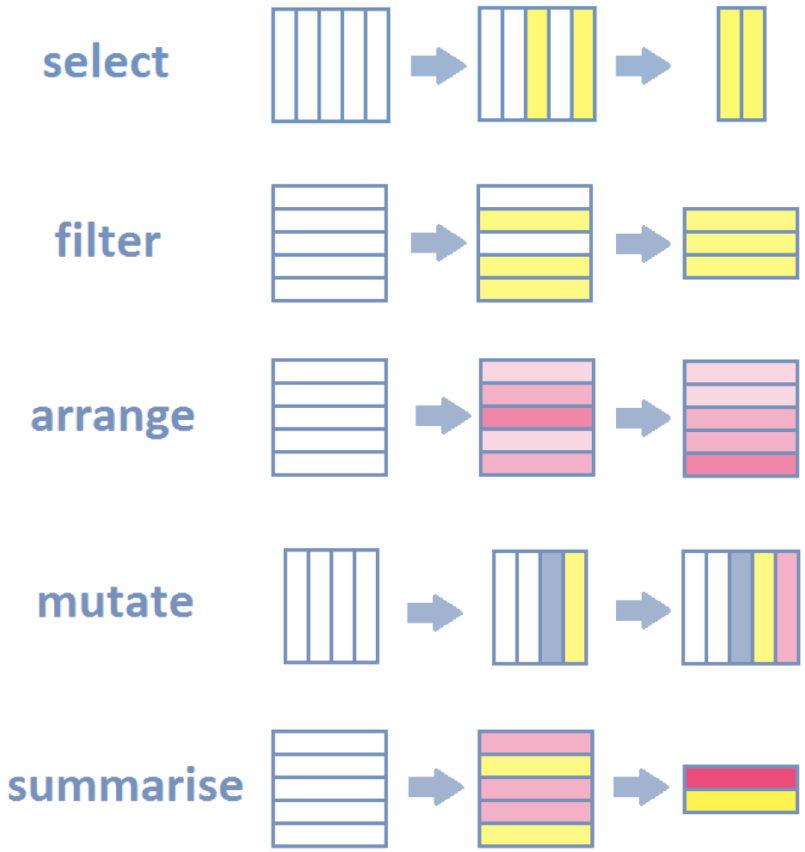

```r
ozono <-
  read_csv("./data/Promedio horarios de o3.csv",
           skip = 1,
           na = "nr",
           n_max = 500) %>%
  clean_names() %>%
  print()
```

```
New names:
• `` -> `...48`
Rows: 500 Columns: 48
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr  (1): Fecha
dbl (31): Hora, ACO, AJM, AJU, ATI, BJU, CAM, CCA, MON, CHO, CUA, CUT, FAC, FAR, IZT, LLA, LPR, MER, MGH, MPA, NEZ, PED, SAC, SAG, TAH, TLA,...
lgl (16): AZC, CES, COY, GAM, HGM, INN, LAG, PLA, SFE, SJA, SUR, TAC, TAX, TEC, TPN, ...48

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
# A tibble: 500 × 48
   fecha      hora   aco   ajm   aju   ati azc     bju   cam   cca ces     mon   cho coy     cua   cut   fac   far gam   hgm   inn     izt lag
   <chr>     <dbl> <dbl> <dbl> <dbl> <dbl> <lgl> <dbl> <dbl> <dbl> <lgl> <dbl> <dbl> <lgl> <dbl> <dbl> <dbl> <dbl> <lgl> <lgl> <lgl> <dbl> <lgl>
 1 01-09-20…     1    NA    NA    NA    NA NA       NA    NA    NA NA       NA    NA NA       NA    NA    NA    NA NA    NA    NA    NA NA
 2 01-09-20…     2    NA    NA    NA    NA NA       NA    NA    NA NA       NA    NA NA       NA    NA    NA    NA NA    NA    NA    NA NA
 3 01-09-20…     3    NA    NA    NA    NA NA       NA    NA    NA NA       NA    NA NA       NA    NA    NA    NA NA    NA    NA    NA NA
 4 01-09-20…     4    19    NA    25    20 NA       28    22    29 NA       20    33 NA       14    24    16    25 NA    NA    NA    29 NA
 5 01-09-20…     5    11    NA    21    10 NA       27    15    25 NA       18    20 NA        7    18    13    25 NA    NA    NA    25 NA
 6 01-09-20…     6     2    NA    18     4 NA       23     5    23 NA        7    11 NA       16     1    10    16 NA    NA    NA    10 NA
 7 01-09-20…     7     3    NA    16     1 NA       15     3    15 NA        4     8 NA        6     1     2    11 NA    NA    NA     6 NA
 8 01-09-20…     8     6    NA    14     1 NA       10     4    11 NA        5     8 NA        5     3     1     7 NA    NA    NA     8 NA
 9 01-09-20…     9    23    NA    25    13 NA       20    12    15 NA       18    22 NA       14     6    11    23 NA    NA    NA    16 NA
10 01-09-20…    10    40    NA    40    19 NA       29    27    27 NA       37    34 NA       28    15    16    32 NA    NA    NA    27 NA
# ℹ 490 more rows
# ℹ 25 more variables: lla <dbl>, lpr <dbl>, mer <dbl>, mgh <dbl>, mpa <dbl>, nez <dbl>, ped <dbl>, pla <lgl>, sac <dbl>, sag <dbl>, sfe <lgl>,
#   sja <lgl>, sur <lgl>, tac <lgl>, tah <dbl>, tax <lgl>, tec <lgl>, tla <dbl>, tli <dbl>, tpn <lgl>, uax <dbl>, uiz <dbl>, vif <dbl>,
#   xal <dbl>, x48 <lgl>
# ℹ Use `print(n = ...)` to see more rows
Warning message:
One or more parsing issues, call `problems()` on your data frame for details, e.g.:
  dat <- vroom(...)
  problems(dat)
```

INSP/ESPM
Escuela de Salud Pública de México

select( )



Seleccionar columnas – select( )

```
malla %>% select(...)
```

dataset

Argumentos de selección

Seleccionar un rango de columnas

```
malla %>% select(col_1:col_8)
```

Seleccionar columnas a excluir

```
malla %>% select(-c(col_1:col_3, col_5, col_7:col_9))
```

Seleccionar columnas cuyo nombre inicia con

```
malla %>% select(starts_with("y")
```

Seleccionar columnas cuyo nombre finaliza con

```
malla %>% select(ends_with("y"))
```

https://dplyr.tidyverse.org/reference/select.html

select()

```
malla
```

```
# A tibble: 8 × 3
  nombre programa  edad
  <chr>  <chr>    <int>
1 Juan   MC          30
2 Eva    MC          25
3 Ana    MSP         25
4 Sofía  MSP         29
5 Pedro  MSP         26
6 Olivia MC          30
7 Mario  MSP         29
8 David  MSP         25
```

```
malla %>%
  select(nombre, edad)
```

```
# A tibble: 8 × 2
  nombre  edad
  <chr>  <dbl>
1 Juan      30
2 Eva       25
3 Ana       25
4 Sofía     29
5 Pedro     26
6 Olivia    30
7 Mario     29
8 David     25
```

select()

```
malla
```

```
# A tibble: 8 × 3
  nombre programa  edad
  <chr>  <chr>    <int>
1 Juan   MC          30
2 Eva    MC          25
3 Ana    MSP         25
4 Sofía  MSP         29
5 Pedro  MSP         26
6 Olivia MC          30
7 Mario  MSP         29
8 David  MSP         25
```

```
malla %>%
   select(-programa)
```

```
# A tibble: 8 × 2
  nombre  edad
  <chr>  <dbl>
1 Juan      30
2 Eva       25
3 Ana       25
4 Sofía     29
5 Pedro     26
6 Olivia    30
7 Mario     29
8 David     25
```

select()

```
malla
```

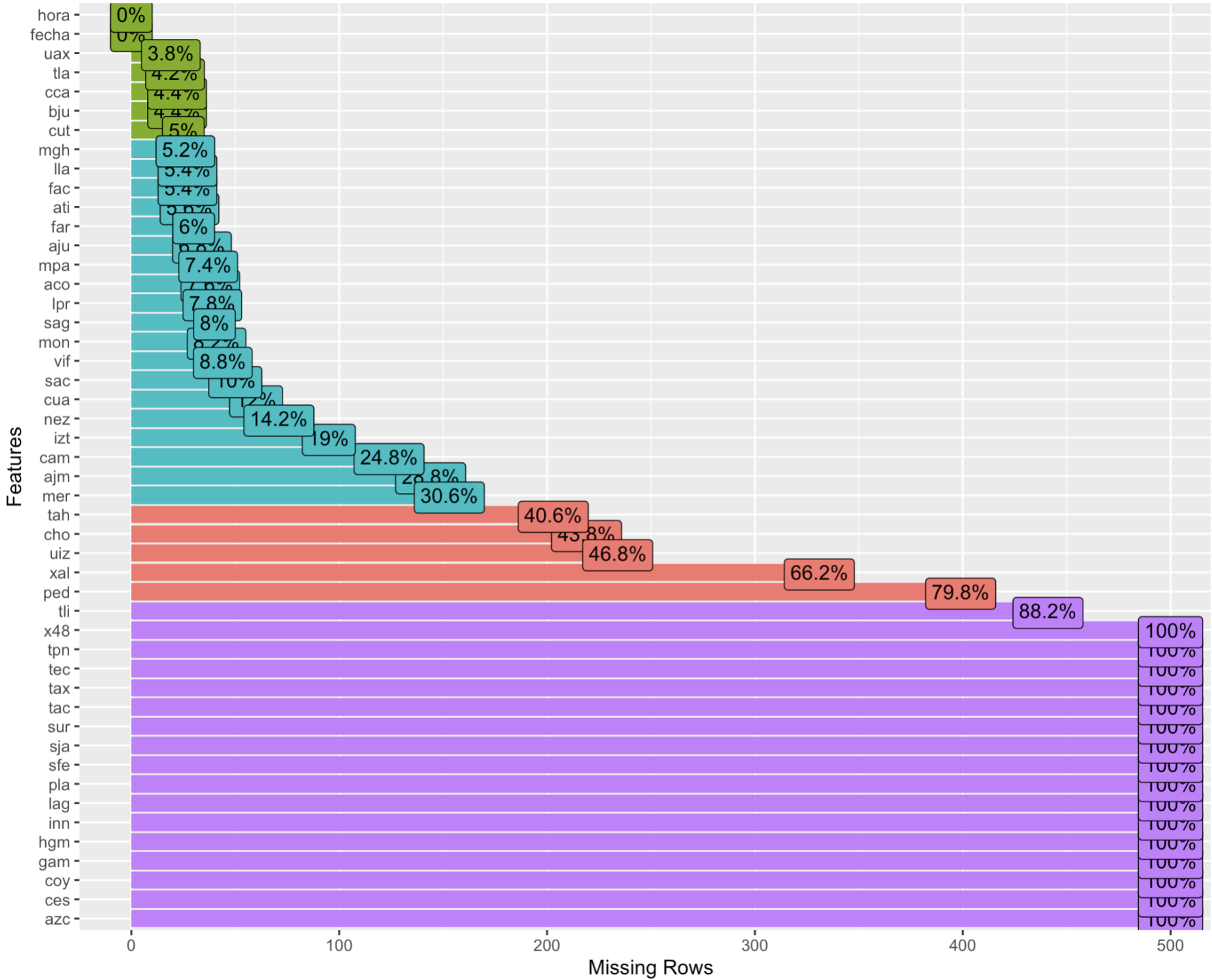
```
# A tibble: 8 × 3
  nombre programa  edad
  <chr>  <chr>    <int>
1 Juan   MC          30
2 Eva    MC          25
3 Ana    MSP         25
4 Sofía  MSP         29
5 Pedro  MSP         26
6 Olivia MC          30
7 Mario  MSP         29
8 David  MSP         25
```

```
malla %>%
   select(-c(nombre, edad))
```

```
# A tibble: 8 × 1
  programa
  <chr>
1 MC
2 MC
3 MSP
4 MSP
5 MSP
6 MC
7 MSP
8 MSP
```

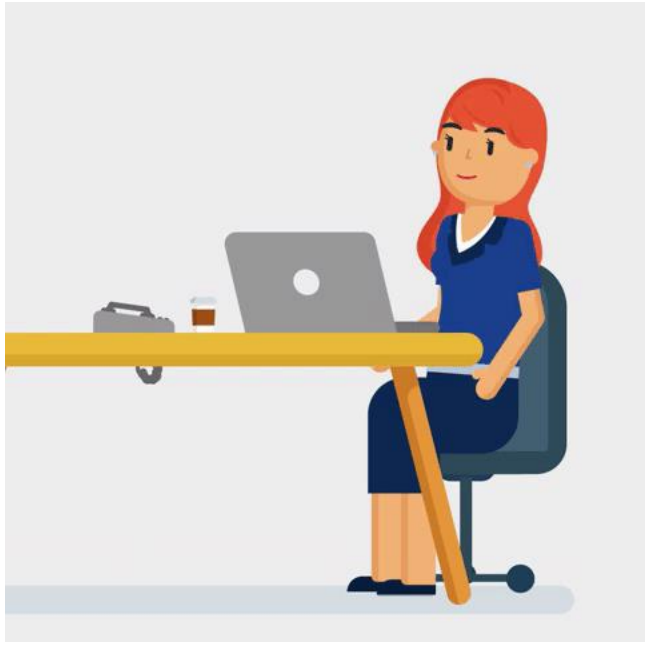
INSP/ESPM
Escuela de Salud Pública de México

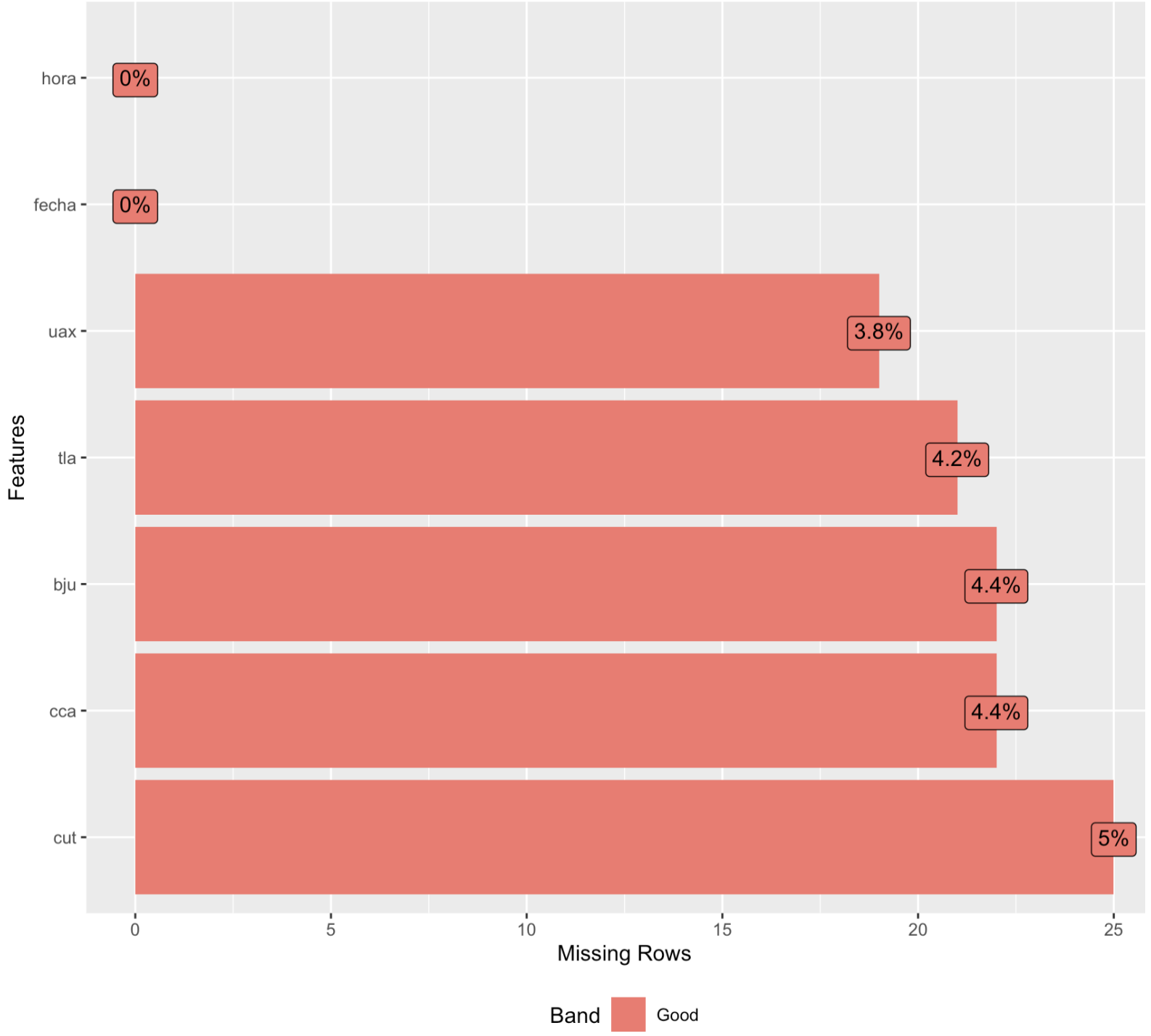
```
library(DataExplorer)
plot_missing(ozono)
```



Su turno...

- Seleccione variables marcadas en color verde en el gráfico generado con el paquete DataExplorer:
- Guarde el resultado como ozono
- Visualize nuevamente el gráfico usando el comando plot_missing( )




hora
fecha
uax
tla
bju
cca
cut
0%
0%
3.8%
4.2%
4.4%
4.4%
5%
Features
0
5
10
15
20
25
Missing Rows
Band
Good