Procesamiento y visualización de datos en R

Sesión 05 - Tidyverse

José Luis Texcalac Sangrador

Septiembre 2021

Editar/generar archivo .CSS

```
h1{
  color: red;
}

h2{
  color: black;
}

body{
  font-family: 'Avenir';
  color: blue;
  font-size;
  line-height: 2em;
}
```

```
.vestColor{
    fill:#449E0A;
}

.jackhammerMainColor{
    fill:#FAA31C;
}

.pantsColor{
    fill:#566275;
}
```

tidydata

Dataset: Colección de valores (tabla, malla de datos, data frame)

Variable: Contiene todos los valores que miden el mismo atributo entre todas las unidades de observación

Observación: Contiene todos los valores medidos, en todos los atributos, en la misma unidad de observación

Valor: Intersección entre la variable y la observación

nombre	edad	peso
Kevin	17	64.3
Brayan	16	61.7
Kimberly	15	51.9
Britany	16	59.3
Brandon	17	69.1
Melany	16	61.6

messydata

Los encabezados son valores, no atributos

nombre	edad	peso	lacio	rizado	ondulado
Kevin	17	64.3	0	1	0
Brayan	16	61.7	0	0	0
Kimberly	15	51.9	1	0	0
Britany	16	59.3	0	0	0
Brandon	17	69.1	0	0	1
Melany	16	61.6	0	0	0

Datos wide y long

Site	2013	2014	2015
CAM	51.0	42.8	39.9
FAC	48.3	39.0	36.6
IZT	44.6	39.3	35.0

Site	Year	PM10
CAM	2013	51.0
FAC	2013	48.3
IZT	2013	44.6
CAM	2014	42.8
FAC	2014	39.0
IZT	2014	39.3
CAM	2015	39.9
FAC	2015	36.6
IZT	2015	35.0

Core Tidyverse

```r
install.packages("tidyverse")
```

es equivalente a

```r
install.packages("ggplot2")
install.packages("tibble")
install.packages("tidyr")
install.packages("readr")
install.packages("purrr")
install.packages("dplyr")
install.packages("stringr")
install.packages("forcats")
```

```r
library("tidyverse")
```

es equivalente a

```r
library("ggplot2")
library("tibble")
library("tidyr")
library("readr")
library("purrr")
library("dplyr")
library("stringr")
library("forcats")
```

Cargar el paquete Tidyverse a la sesión

```r
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.5     ✓ purrr   0.3.4
## ✓ tibble  3.1.5     ✓ dplyr   1.0.7
## ✓ tidyr   1.1.4     ✓ stringr 1.4.0
## ✓ readr   2.0.1     ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Pipe

El operador %>% simplifica y concatena múltiples funciones

```
mi_dia <- veo_tv(paseo_perro(regreso(chamba(traslado(despierto(🙂))))))
```

```
🙂 %>%
  despierto %>%
  traslado %>%
  chamba %>%
  regreso %>%
  paseo_perro %>%
  veo_tv
```

```
malla_datos %>%
  filtro %>%
  genero_variables %>%
  agrupo %>%
  paso_a_wide %>%
  genero_variables %>%
  selecciono_columnas
```

Importar datos a R

readr

readxl

haven

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...

Instale los siguientes paquetes

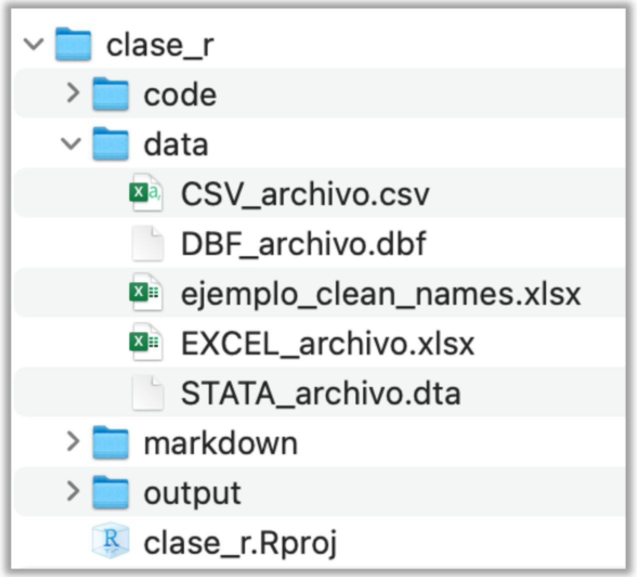
- Desde el **CRAN**:
    - tidyverse, lubridate, readxl, haven, janitor
- Desde **Github**:
    - `kableExtra`: "haozhu233/kableExtra"
    - `pander`: "Rapporter/pander"
    - `summarytools`: "dcomtois/summarytools"
    - `visdat`: "ropensci/visdat"

```
install.packages("tidyverse")
```

```
remotes::install_github("Rapporter/pander")
```

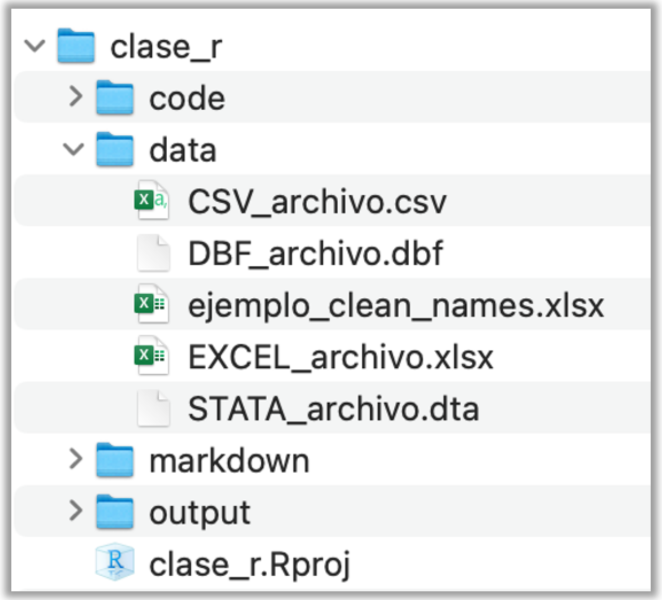
Su turno...

- Descargue los archivios que se le indiquen de Google Classroom
- Confirme que tiene en su carpeta data los siguientes archivos:
  - CSV_archivo.csv
  - DBF_archivo.dbf
  - STATA_archivo.dta
  - EXCEL_archivo.xlsx
  - ejemplo_clean_names.xlsx
- Genere un nuevo **sript** de nombre "Script_05_Nombre"
- Active en su script al paquete tidyverse



```
clase_r
  code
  data
    CSV_archivo.csv
    DBF_archivo.dbf
    ejemplo_clean_names.xlsx
    EXCEL_archivo.xlsx
    STATA_archivo.dta
  markdown
  output
  clase_r.Rproj
```

Gestión de directorios en un proyecto



```
clase_r
  code
  data
    CSV_archivo.csv
    DBF_archivo.dbf
    ejemplo_clean_names.xlsx
    EXCEL_archivo.xlsx
    STATA_archivo.dta
  markdown
  output
  clase_r.Rproj
```

- Nuestra carpeta principal es clase_r
- Para acceder (en un chunk) a uno de los directorios colocamos un punto, el punto indica que todo es dentro de la carpeta principal de nuestro proyecto

  `"./data"`

- Si queremos acceder a un archivo específico

  `"./data/DBF_archivo.dbf"`

Importar archivos - .DBF

- El paquete foreign permite importar archivos con extensión .dbf
- El comando read.dbf( ) nos permite leer archivos con extensión .dbf

```
library(foreign)
tabla_dbf <- read.dbf("./data/DBF_archivo.dbf")
```

- Se guarda el objeto pero no se imprime en pantalla
- Se requiere llamar a la malla o a una parte de ella con el comando **head(*x*, *n*)**

```
head(tabla_dbf)
```

```
##   MONTH ESTACION   ZONA      Y2007      Y2008      Y2009      Y2010      Y2011      Y2012      Y2013      Y2014
## 1   m07      ACO   ZMVM 0.01884852 0.02532318 0.02966288 0.02261605 0.02474085 0.02520094 0.02980886 0.02510226
## 2   m07      ACO   ZMVM 0.01884852 0.02532318 0.02966288 0.02261605 0.02474085 0.02520094 0.02980886 0.02510226
## 3   m07      AGU ZM_Gdl 0.03924623 0.02776190 0.02936694 0.02775538 0.02652604 0.02786606 0.02555833 0.02449194
## 4   m07      AJM   ZMVM 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 5   m07      AJU   ZMVM 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
## 6   m07      AJU   ZMVM 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
```

Revisemos el objeto *`tabla-dbf`*

- Veamos el tipo de objeto

```
class(tabla_dbf)
```

```
## [1] "data.frame"
```

- Reviso los nombres de columnas

```
names(tabla_dbf)
```

```
##  [1] "MONTH"    "ESTACION" "ZONA"     "Y2007"    "Y2008"    "Y2009"
##  [7] "Y2010"    "Y2011"    "Y2012"    "Y2013"    "Y2014"    "Y2015"
```

```
dim(tabla_dbf)
```

```
## [1] 68 12
```

Imprimo mi objeto *(data frame)* en pantalla

```r
tabla_dbf
```

```
##    MONTH ESTACION      ZONA      Y2007      Y2008      Y2009      Y2010      Y2011      Y2012      Y2013      Y2
## 1    m07      ACO      ZMVM 0.01884852 0.02532318 0.02966288 0.02261605 0.02474085 0.02520094 0.02980886 0.0251
## 2    m07      ACO      ZMVM 0.01884852 0.02532318 0.02966288 0.02261605 0.02474085 0.02520094 0.02980886 0.0251
## 3    m07      AGU    ZM_Gdl 0.03924623 0.02776190 0.02936694 0.02775538 0.02652604 0.02786606 0.02555833 0.0244
## 4    m07      AJM      ZMVM 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.0000
## 5    m07      AJU      ZMVM 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.0000
## 6    m07      AJU      ZMVM 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.0000
## 7    m07       AP ZM_Toluca 0.00000000 0.00000000 0.00000000 0.00000000 0.02033721 0.02358477 0.02793113 0.0199
## 8    m07      ATI      ZMVM 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.02652964 0.0231
## 9    m07      ATM    ZM_Gdl 0.03733179 0.02548223 0.02757735 0.02286939 0.02916225 0.00000000 0.03041052 0.0245
## 10   m07      ATM    ZM_Gdl 0.03733179 0.02548223 0.02757735 0.02286939 0.02916225 0.00000000 0.03041052 0.0245
## 11   m07      AZC      ZMVM 0.02740734 0.02770051 0.02830638 0.02035509 0.00000000 0.00000000 0.00000000 0.0000
## 12   m07      CAM      ZMVM 0.00000000 0.00000000 0.00000000 0.00000000 0.01873655 0.01988024 0.02338102 0.0225
## 13   m07       CB ZM_Toluca 0.00000000 0.00000000 0.00000000 0.00000000 0.02606044 0.00000000 0.00000000 0.0288
## 14   m07      CCA      ZMVM 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.0000
## 15   m07   CE_Mty    ZM_Mty 0.02246026 0.02048039 0.02405338 0.01661003 0.02260133 0.02347037 0.02669415 0.0264
## 16   m07   CE_Tol ZM_Toluca 0.02641941 0.00000000 0.00000000 0.00000000 0.01942101 0.02645037 0.01999982 0.0192
## 17   m07      CEN    ZM_Gdl 0.03753904 0.02120899 0.03116512 0.01957423 0.02350000 0.02691667 0.02298457 0.0253
## 18   m07      CES      ZMVM 0.02600196 0.01952489 0.02543040 0.01663381 0.00000000 0.00000000 0.00000000 0.0000
## 19   m07      CHO      ZMVM 0.02126449 0.02802959 0.03111654 0.02173535 0.02148589 0.02442070 0.02757090 0.0255
## 20   m07      COY      ZMVM 0.03554655 0.02605470 0.03203012 0.01888026 0.02161158 0.02371116 0.02655446 0.0255
```

Importando archivos .CSV

- El paquete readr permite importar archivos con extensión .csv
- El comando read_csv( ) nos permite leer archivos con extensión .csv

```
tabla_csv <-
  read_csv("./data/CSV_archivo.csv") %>%
  print(n = 3)
```

```
## Rows: 68 Columns: 12

## ── Column specification ────────────────────────────────────────────
## Delimiter: ","
## chr (3): month, estacion, zona
## dbl (9): y2007, y2008, y2009, y2010, y2011, y2012, y2013, y2014, y2015

##
## 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: 68 × 12
##   month estacion zona    y2007  y2008  y2009  y2010  y2011  y2012  y2013  y2014
##   <chr> <chr>    <chr>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 m07   ACO      ZMVM   0.0188 0.0253 0.0297 0.0226 0.0247 0.0252 0.0298 0.0251
## 2 m07   ACO      ZMVM   0.0188 0.0253 0.0297 0.0226 0.0247 0.0252 0.0298 0.0251
## 3 m07   AGU      ZM_Gdl 0.0392 0.0278 0.0294 0.0278 0.0265 0.0279 0.0256 0.0245
## # … with 65 more rows, and 1 more variable: y2015 <dbl>
```

Importando archivos de Stata .DTA

- El paquete haven permite importar archivos con extensión .dta
- El comando read_dta( ) nos permite leer archivos con extensión .dta

```r
library(haven)

tabla_stata <-
  read_dta("./data/STATA_archivo.dta") %>%
  print()
```

```
## # A tibble: 68 × 12
##    month estacion zona     y2007  y2008  y2009  y2010  y2011  y2012  y2013
##    <chr> <chr>    <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1 m07   ACO      ZMVM    0.0188 0.0253 0.0297 0.0226 0.0247 0.0252 0.0298
##  2 m07   ACO      ZMVM    0.0188 0.0253 0.0297 0.0226 0.0247 0.0252 0.0298
##  3 m07   AGU      ZM_Gdl  0.0392 0.0278 0.0294 0.0278 0.0265 0.0279 0.0256
##  4 m07   AJM      ZMVM   NA     NA     NA     NA     NA     NA     NA
##  5 m07   AJU      ZMVM   NA     NA     NA     NA     NA     NA     NA
##  6 m07   AJU      ZMVM   NA     NA     NA     NA     NA     NA     NA
##  7 m07   AP       ZM_To… NA     NA     NA     NA      0.0203 0.0236 0.0279
##  8 m07   ATI      ZMVM   NA     NA     NA     NA     NA     NA      0.0265
##  9 m07   ATM      ZM_Gdl  0.0373 0.0255 0.0276 0.0229 0.0292 NA      0.0304
## 10 m07   ATM      ZM_Gdl  0.0373 0.0255 0.0276 0.0229 0.0292 NA      0.0304
## # … with 58 more rows, and 2 more variables: y2014 <dbl>, y2015 <dbl>
```

Importando archivos de Excel .XLSX

- El paquete readxl permite importar archivos con extensión .xlsx
- El comando read_xlsx( ) nos permite leer archivos con extensión .xlsx

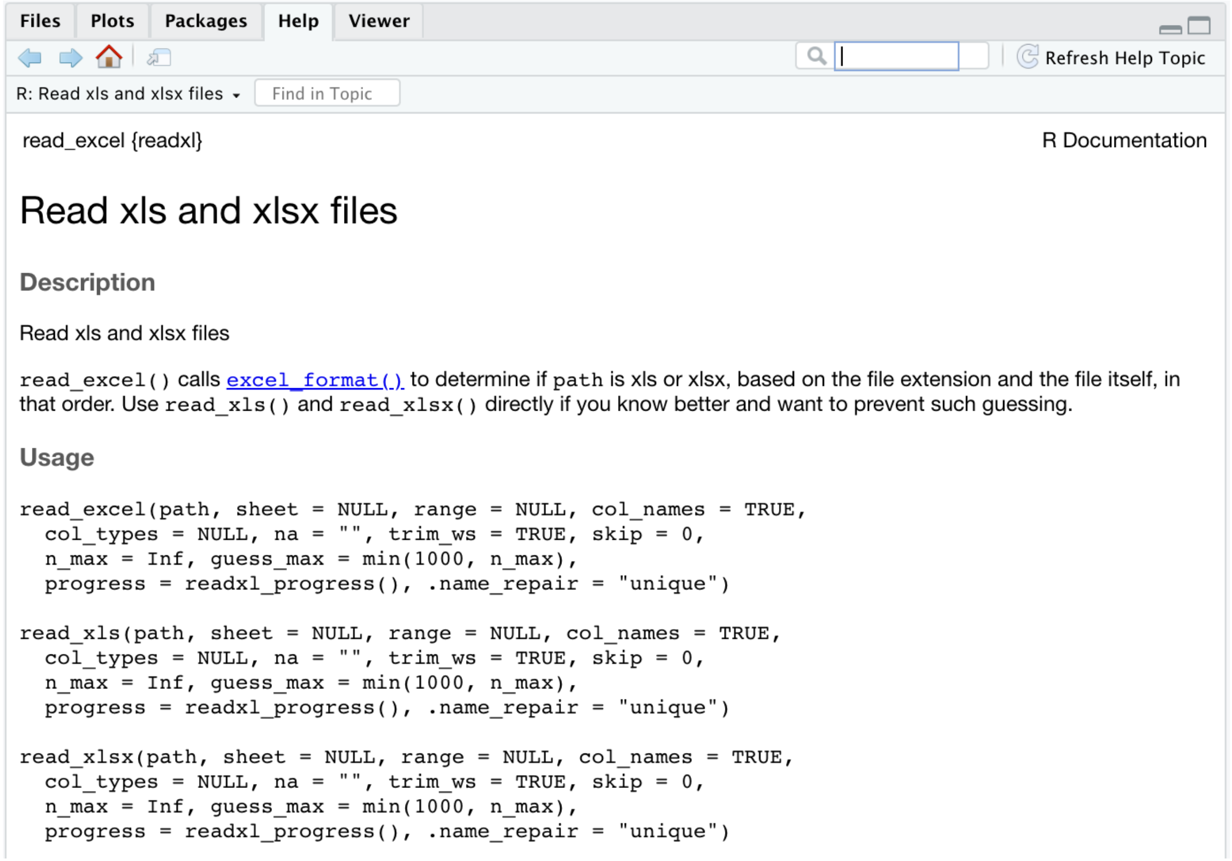
```r
library(readxl)

tabla_excel <-
  read_xlsx("./data/EXCEL_archivo.xlsx", sheet = "datos_wide") %>%
  print()
```

```
## # A tibble: 68 × 12
##    month estacion zona     y2007   y2008   y2009   y2010   y2011   y2012   y2013
##    <chr> <chr>    <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 m07   ACO      ZMVM    0.0188  0.0253  0.0297  0.0226  0.0247  0.0252  0.0298
##  2 m07   ACO      ZMVM    0.0188  0.0253  0.0297  0.0226  0.0247  0.0252  0.0298
##  3 m07   AGU      ZM_Gdl  0.0392  0.0278  0.0294  0.0278  0.0265  0.0279  0.0256
##  4 m07   AJM      ZMVM   NA      NA      NA      NA      NA      NA      NA
##  5 m07   AJU      ZMVM   NA      NA      NA      NA      NA      NA      NA
##  6 m07   AJU      ZMVM   NA      NA      NA      NA      NA      NA      NA
##  7 m07   AP       ZM_To… NA      NA      NA      NA       0.0203  0.0236  0.0279
##  8 m07   ATI      ZMVM   NA      NA      NA      NA      NA      NA       0.0265
##  9 m07   ATM      ZM_Gdl  0.0373  0.0255  0.0276  0.0229  0.0292 NA       0.0304
## 10 m07   ATM      ZM_Gdl  0.0373  0.0255  0.0276  0.0229  0.0292 NA       0.0304
## # … with 58 more rows, and 2 more variables: y2014 <dbl>, y2015 <dbl>
```

Buscando ayuda de un comando

```
?read_xlsx
```



read_excel {readxl} R Documentation

Read xls and xlsx files

Description

Read xls and xlsx files

`read_excel()` calls `excel_format()` to determine if `path` is xls or xlsx, based on the file extension and the file itself, in that order. Use `read_xls()` and `read_xlsx()` directly if you know better and want to prevent such guessing.

Usage

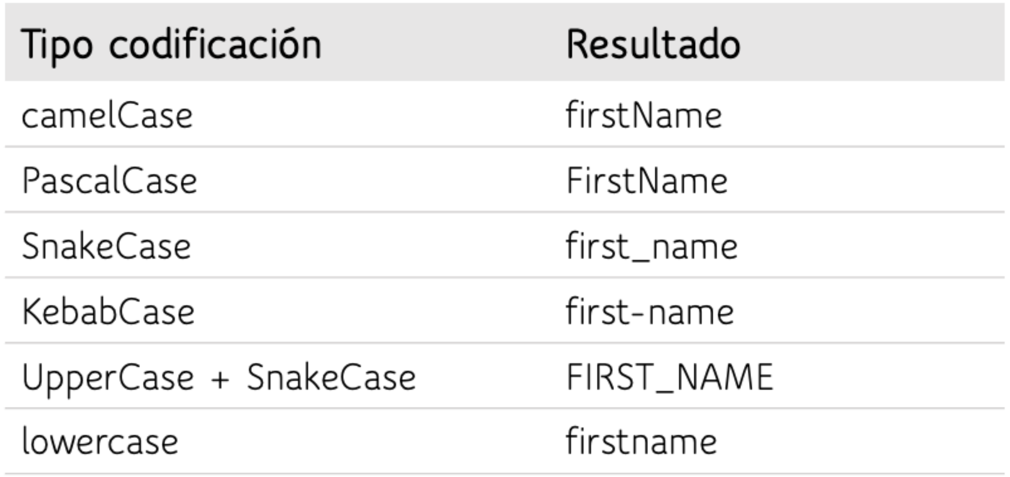
```
read_excel(path, sheet = NULL, range = NULL, col_names = TRUE,
  col_types = NULL, na = "", trim_ws = TRUE, skip = 0,
  n_max = Inf, guess_max = min(1000, n_max),
  progress = readxl_progress(), .name_repair = "unique")

read_xls(path, sheet = NULL, range = NULL, col_names = TRUE,
  col_types = NULL, na = "", trim_ws = TRUE, skip = 0,
  n_max = Inf, guess_max = min(1000, n_max),
  progress = readxl_progress(), .name_repair = "unique")

read_xlsx(path, sheet = NULL, range = NULL, col_names = TRUE,
  col_types = NULL, na = "", trim_ws = TRUE, skip = 0,
  n_max = Inf, guess_max = min(1000, n_max),
  progress = readxl_progress(), .name_repair = "unique")
```

Convenciones de codificación

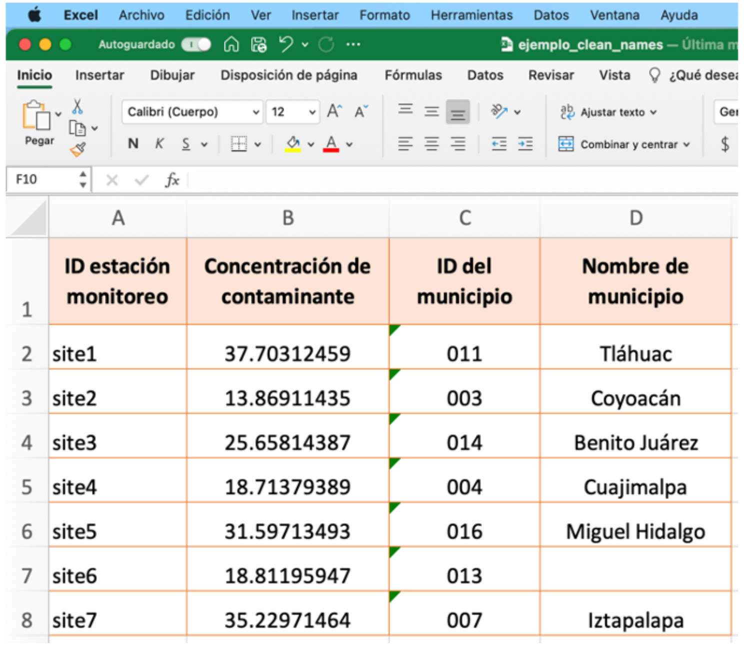
- Son un conjunto de normas que estandarizan el estilo de redacción de los lenguajes de programación.
- Cada lenguaje de programación tiene (más o menos) sus propias convenciones *(naming)*.
- Recomiendan estilos de programación, buenas prácticas y métodos para mantener el aspecto del código fuente por ejemplo: la organización de archivos, la indentación, nombres de columnas, los comentarios, las declaraciones los espacios en blanco, las llaves de apertura y cerrado…



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

Importancia de convenciones de codificación
Desde el CamelCase hasta el kebab-case

Ejemplo de columnas sin formato de origen


Excel Archivo Edición Ver Insertar Formato Herramientas Datos Ventana Ayuda
Autoguardado
ejemplo_clean_names — Última m
Inicio Insertar Dibujar Disposición de página Fórmulas Datos Revisar Vista ¿Qué desea
Calibri (Cuerpo) 12
Ajustar texto
Combinar y centrar
Pegar
F10

	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

```r
read_xlsx("./data/ejemplo_clean_names.xlsx") %>%
  print()
```

```
## # A tibble: 7 × 4
##   `ID estación monitoreo` `Concentración de …` `ID del municipi… `Nombre de muni…
##   <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
```

Formato snake_case usando el paquete {Janitor}

```r
library(janitor)

read_xlsx("./data/ejemplo_clean_names.xlsx") %>%
  clean_names() %>%
  print()
```

```
## # A tibble: 7 × 4
##   id_estacion_monitoreo concentracion_de_con… id_del_municipio nombre_de_munici…
##   <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
```

Su turno

- Importe a su sesión el archivo pm10_2014_salamanca.csv
- Llame a su objeto pm10

```r
pm10 <-
  read_csv("./data/pm10_2014_salamanca.csv") %>%
  print()
```

```
## Rows: 8764 Columns: 5

## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (3): Fecha, Cruz Roja, Nativitas
## dbl (2): hora, DIF

##
## 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: 8,764 × 5
##    Fecha     hora   DIF `Cruz Roja` Nativitas
##    <chr>    <dbl> <dbl> <chr>       <chr>
##  1 01/01/14     0    NA <NA>        <NA>
##  2 01/01/14     1    68 57          224
##  3 01/01/14     2    74 52          118
##  4 01/01/14     3    58 45          59
##  5 01/01/14     4    50 51          40
##  6 01/01/14     5    48 39          43
##  7 01/01/14     6    49 25          39
##  8 01/01/14     7    41 42          43
##  9 01/01/14     8    49 48          46
## 10 01/01/14     9    60 50          47
## # … with 8,754 more rows
```

Revisemos la columna **Fecha**

- El comando unique nos muestra los valores únicos en una variable
- El resultado es un vector

```
unique(pm10$Fecha)
```

```
##   [1] "01/01/14" "02/01/14" "03/01/14" "04/01/14" "05/01/14" "06/01/14" "07/01/14" "08/01/14" "09/01/14" "10/01,
##  [23] "23/01/14" "24/01/14" "25/01/14" "26/01/14" "27/01/14" "28/01/14" "29/01/14" "30/01/14" "31/01/14" "01/02,
##  [45] "14/02/14" "15/02/14" "16/02/14" "17/02/14" "18/02/14" "19/02/14" "20/02/14" "21/02/14" "22/02/14" "23/02,
##  [67] "08/03/14" "09/03/14" "10/03/14" "11/03/14" "12/03/14" "13/03/14" "14/03/14" "15/03/14" "16/03/14" "17/03,
##  [89] "30/03/14" "31/03/14" "01/04/14" "02/04/14" "03/04/14" "04/04/14" "05/04/14" "06/04/14" "07/04/14" "08/04,
## [111] "21/04/14" "22/04/14" "23/04/14" "24/04/14" "25/04/14" "26/04/14" "27/04/14" "28/04/14" "29/04/14" "30/04,
## [133] "13/05/14" "14/05/14" "15/05/14" "16/05/14" "17/05/14" "18/05/14" "19/05/14" "20/05/14" "21/05/14" "22/05,
## [155] "04/06/14" "05/06/14" "06/06/14" "07/06/14" "08/06/14" "09/06/14" "10/06/14" "11/06/14" "12/06/14" "13/06,
## [177] "26/06/14" "27/06/14" "28/06/14" "29/06/14" "30/06/14" "01/07/14" "02/07/14" "03/07/14" "04/07/14" "05/07,
## [199] "18/07/14" "19/07/14" "20/07/14" "21/07/14" "22/07/14" "23/07/14" "24/07/14" "25/07/14" "26/07/14" "27/07,
## [221] "09/08/14" "10/08/14" "11/08/14" "12/08/14" "13/08/14" "14/08/14" "15/08/14" "16/08/14" "17/08/14" "18/08,
## [243] "31/08/14" "01/09/14" "02/09/14" "03/09/14" "04/09/14" "05/09/14" "06/09/14" "07/09/14" "08/09/14" "09/09,
## [265] "22/09/14" "23/09/14" "24/09/14" "25/09/14" "26/09/14" "27/09/14" "28/09/14" "29/09/14" "30/09/14" "01/10,
## [287] "14/10/14" "15/10/14" "16/10/14" "17/10/14" "18/10/14" "19/10/14" "20/10/14" "21/10/14" "22/10/14" "23/10,
## [309] "05/11/14" "06/11/14" "07/11/14" "08/11/14" "09/11/14" "10/11/14" "11/11/14" "12/11/14" "13/11/14" "14/11,
## [331] "27/11/14" "28/11/14" "29/11/14" "30/11/14" "01/12/14" "02/12/14" "03/12/14" "04/12/14" "05/12/14" "06/12,
## [353] "19/12/14" "20/12/14" "21/12/14" "22/12/14" "23/12/14" "24/12/14" "25/12/14" "26/12/14" "27/12/14" "28/12,
```

Columna **hora**

```r
unique(pm10$hora)
```

```
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
```

Columna **DIF**

```
unique(pm10$DIF)
```

```
##   [1]  NA  68  74  58  50  48  49  41  60  33  23  26  45  25  19  31  43  56
##  [19]  69  34  20  22  28  44  38  30  51  61  40  37  27  16  17  12  15  14
##  [37]  29  42 108 146 189  97  70  52  64  36  47  82 133  98  67  54  81 120
##  [55] 149  89  88 115  75  53  59 122  87  63 109 123  55  66  72  18  21  32
##  [73]  39  35  46 148 160  93  62  77  96  83  71  24  76 105 116 144 100  84
##  [91] 170 159  80  65 110 129 161 114  57  90  92 126  78  85 101  73 118 113
## [109] 151  94 124 139 175 153 121 106 166 192 191 107  79 195  99 130 140 141
## [127] 135  91 174 221 331 316 154 103 134 183 162 102  86 119 206 212  95 251
## [145] 279 111 178 177 230 227 128 205 242 247 104 214 171 125 216 194 280 252
## [163] 203 209 138 225 143 132 219 249 112 193 155 187 235 150 259 312 145 156
## [181] 290 286 217 165 117 142 218 202 190 152 127   9 137 131 136 287 282 288
## [199] 222 199 158 182 261 147 186 250 169 173 208 172 168 157 318  11 268 334
## [217] 220 180  13   8   6   7 256  10 188 167   5 246 179 200   4 211 198 229
## [235] 176 234 231 309 164 332 439 308 263 423 303 201 387 365 380 265 358 213
## [253] 184 239 163 304 260
```

Columna 'Cruz Roja'

- El comando sort( ) nos permite ordenar los datos de forma ascendente

```
sort(unique(pm10$`Cruz Roja`))
```

```
##   [1] "10"  "100" "101" "102" "103" "104" "105" "106" "107" "108" "109" "11"  "110" "111" "112" "113" "114" "115
##  [19] "116" "117" "118" "119" "12"  "120" "121" "122" "123" "124" "125" "126" "127" "128" "129" "13"  "130" "131
##  [37] "132" "133" "134" "135" "136" "137" "138" "139" "14"  "140" "141" "142" "143" "144" "145" "146" "147" "148
##  [55] "149" "15"  "150" "151" "152" "153" "155" "156" "157" "158" "159" "16"  "160" "161" "163" "164" "165" "166
##  [73] "167" "168" "169" "17"  "170" "171" "173" "174" "175" "176" "177" "178" "179" "18"  "180" "181" "183" "184
##  [91] "187" "188" "19"  "192" "196" "197" "199" "20"  "200" "201" "203" "207" "209" "21"  "210" "211" "215" "216
## [109] "22"  "221" "222" "226" "23"  "24"  "247" "25"  "252" "26"  "260" "27"  "271" "275" "28"  "29"  "3"   "30"
## [127] "309" "31"  "32"  "33"  "34"  "35"  "36"  "37"  "38"  "39"  "4"   "40"  "41"  "42"  "43"  "44"  "45"  "46"
## [145] "47"  "48"  "49"  "5"   "50"  "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"  "59"  "6"   "60"  "61"  "62"
## [163] "63"  "64"  "65"  "66"  "67"  "68"  "69"  "7"   "70"  "71"  "72"  "73"  "74"  "75"  "76"  "77"  "78"  "79"
## [181] "8"   "80"  "81"  "82"  "83"  "84"  "85"  "86"  "87"  "88"  "89"  "9"   "90"  "91"  "92"  "93"  "94"  "95"
## [199] "96"  "97"  "98"  "99"  "nd"  "ND"
```

Columna Nativitas

```
sort(unique(pm10$Nativitas))
```

```
##   [1] "10"      "100"     "101"     "102"     "103"     "104"     "105"     "106"     "107"     "108"
##  [11] "109"     "11"      "110"     "111"     "112"     "113"     "114"     "115"     "116"     "117"
##  [21] "118"     "119"     "12"      "120"     "121"     "122"     "123"     "124"     "125"     "126"
##  [31] "127"     "128"     "129"     "13"      "130"     "131"     "132"     "133"     "134"     "135"
##  [41] "136"     "137"     "138"     "139"     "14"      "140"     "141"     "142"     "143"     "144"
##  [51] "145"     "146"     "147"     "148"     "149"     "15"      "150"     "151"     "152"     "153"
##  [61] "154"     "155"     "156"     "157"     "158"     "159"     "16"      "160"     "161"     "162"
##  [71] "163"     "164"     "165"     "166"     "167"     "168"     "169"     "17"      "170"     "171"
##  [81] "172"     "173"     "174"     "175"     "176"     "177"     "178"     "179"     "18"      "180"
##  [91] "181"     "182"     "183"     "184"     "185"     "186"     "187"     "188"     "189"     "19"
## [101] "190"     "191"     "192"     "193"     "194"     "195"     "196"     "197"     "198"     "199"
## [111] "2"       "20"      "200"     "201"     "202"     "203"     "204"     "205"     "206"     "207"
## [121] "208"     "209"     "21"      "212"     "213"     "214"     "215"     "216"     "217"     "218"
## [131] "219"     "22"      "220"     "221"     "222"     "224"     "225"     "226"     "227"     "228"
## [141] "229"     "23"      "230"     "231"     "232"     "234"     "235"     "236"     "238"     "239"
## [151] "24"      "240"     "241"     "242"     "245"     "247"     "248"     "249"     "25"      "250"
## [161] "252"     "253"     "257"     "258"     "26"      "260"     "261"     "263"     "264"     "267"
## [171] "268"     "269"     "27"      "273"     "276"     "278"     "28"      "281"     "285"     "286"
## [181] "288"     "29"      "291"     "294"     "296"     "298"     "30"      "300"     "301"     "302"
## [191] "307"     "308"     "309"     "31"      "311"     "317"     "319"     "32"      "320"     "321"
## [201] "327"     "33"      "330"     "339"     "34"      "346"     "35"      "354"     "358"     "36"
```

```r
pm10 <-
  read_csv("./data/pm10_2014_salamanca.csv",
           # Identifico en un vector a los valores a codificar como NA
           na = c("ND", "nd", "sin dato"),
           # Selecciono de la columna 1 a la 5
           col_select = 1:5) %>%
  clean_names() %>%
  print(n = 4)
```

```
## Rows: 8764 Columns: 5

## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): Fecha
## dbl (4): hora, DIF, Cruz Roja, Nativitas

##
## 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.

## Warning: One or more parsing issues, see `problems()` for details

## # A tibble: 8,764 × 5
##   fecha     hora   dif cruz_roja nativitas
##   <chr>    <dbl> <dbl>     <dbl>     <dbl>
## 1 01/01/14     0    NA        NA        NA
## 2 01/01/14     1    68        57       224
## 3 01/01/14     2    74        52       118
## 4 01/01/14     3    58        45        59
## # … with 8,760 more rows
```

Análisis exploratorio de la malla de datos

```
names(pm10)
```

```
## [1] "fecha"     "hora"      "dif"       "cruz_roja" "nativitas"
```

```
glimpse(pm10)
```

```
## Rows: 8,764
## Columns: 5
## $ fecha     <chr> "01/01/14", "01/01/14", "01/01/14", "01/01/14", "01/01/14", …
## $ hora      <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17…
## $ dif       <dbl> NA, 68, 74, 58, 50, 48, 49, 41, 49, 60, 33, 23, 26, 45, 41, …
## $ cruz_roja <dbl> NA, 57, 52, 45, 51, 39, 25, 42, 48, 50, 40, 76, 47, 44, 46, …
## $ nativitas <dbl> NA, 224, 118, 59, 40, 43, 39, 43, 46, 47, 37, 34, 43, 48, 50…
```

```
summary(pm10)
```

```
##     fecha                hora            dif            cruz_roja        nativitas
##  Length:8764        Min.   : 0.00   Min.   :  4.00   Min.   :  3.00   Min.   :  2.00
##  Class :character   1st Qu.: 5.75   1st Qu.: 33.00   1st Qu.: 28.00   1st Qu.: 37.00
##  Mode  :character   Median :11.50   Median : 47.00   Median : 40.00   Median : 53.00
##                     Mean   :11.50   Mean   : 55.54   Mean   : 46.81   Mean   : 62.86
##                     3rd Qu.:17.00   3rd Qu.: 66.00   3rd Qu.: 57.00   3rd Qu.: 76.00
##                     Max.   :23.00   Max.   :439.00   Max.   :309.00   Max.   :547.00
##                                     NA's   :18       NA's   :57       NA's   :62
```