



Clase 2

Cargar datos

Miriam Lerma

Febrero 2021

1. Intro

- Teoría.
Paquetes, Directorios, Cargar datos.
- Practica.
Cargar datos desde csv, txt, excel y desde una url (internet).

Quienes son ustedes?

- Saben abrir e identificar partes de R Studio
- Quieren cargar sus datos a RStudio y poder inspeccionarlos.

Objetivo de hoy

- Puedan cargar datos y verlos en su computadora.

Créditos

-El material esta basado en el libro:

📖 R4DS, editado por Riva Quiroga

-El libro de:

📖 Bookdown, por Sonia Mendizabal

-Y materiales de RLadies

🐱 Zero to Hero

🐱 Recursos en español

🐱 RdesdeCero por Yanina Bellini

🐱 Primeros pasos en R por Riva Quiroga

Imágenes adicionales

🖼️ Unsplash

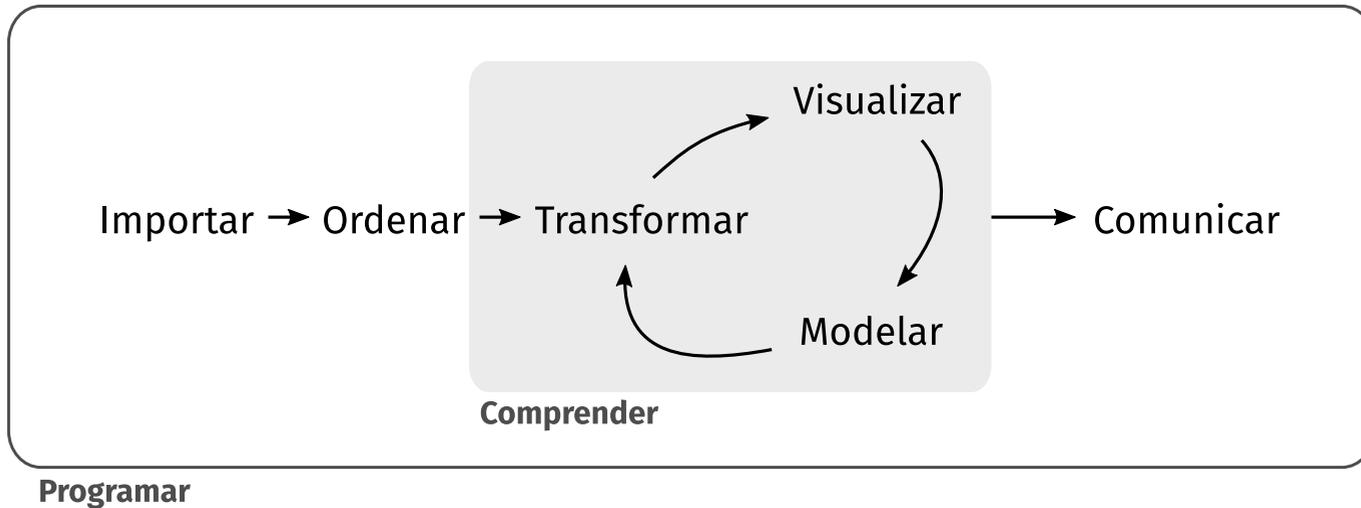
🖼️ Portada por Fredy Jacob



Empecemos!

1. Intro

Un proyecto típico de ciencia de datos se ve así:



Fuente: R4DS

Hoy queremos pasar el primer paso que es **importar nuestros datos**.

1.1. Paquetes

- Son colecciones de funciones que realizan operaciones.
- Se instalan una única vez pero deben cargarse cada vez que se utilizan.
Para instalar hay que escribir: `install.packages("")`.
Para cargar hay que escribir: `library("")`.

Si han bajado apps a su cel, `install.packages()` es descargar la app y se hace una sola vez y `library()` es abrirla. Ahora mismo abran la consola, escriban el siguiente código y denle enter

```
install.packages("cowsay")
```

```
install.packages("here")
```

⌚ Tarda un poquito en cargarse...

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

Nota: nos da un warning, solo para **advertirnos** o **informarnos** de que contiene el paquete. No es un mensaje de error.

Otra opción es ir al **workspace** Packages>Install>Escribir el nombre.

1.1. Paquetes

```
say(what = "Ya sabes cargar paquetes!", by = "cat")
```

No sale nada? Hay que cargar la libreria primero.

```
library("cowsay")  
say(what = "Ya sabes cargar paquetes!", by = "cat")
```

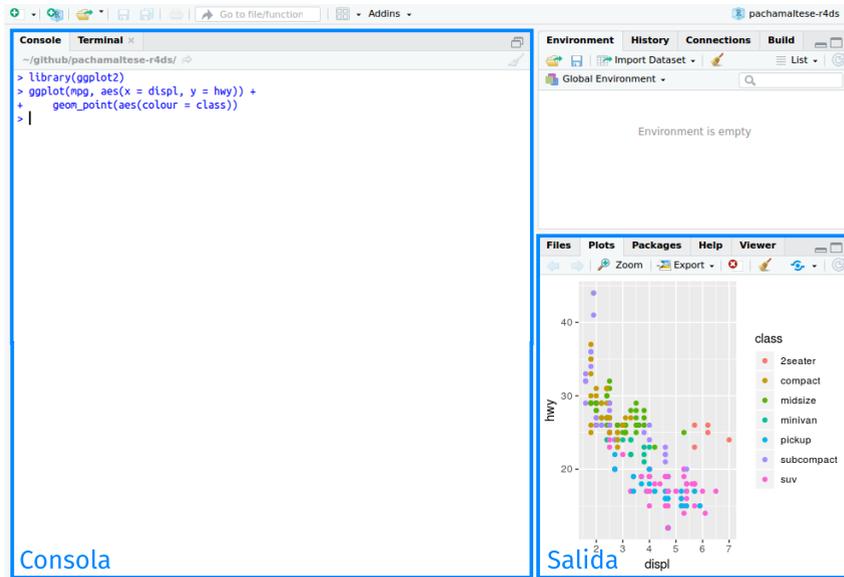
```
-----  
Ya sabes cargar paquetes!  
-----
```

```
  \  \  \  
    | \_--\_--/ |  
  ==)  ^Y^  (==  
    \  ^  /  
      )=*=(  
    /      \  
    |      |  
  /| | | | \  
 \ | | | - | / \  
  \ | | | - | / \  
    |      |
```

1.2. Directorios

Donde están los archivos en tu computadora.

Manualmente: Esquina inferior derecha: **Salida** > Files > More > Set working directory



Pero recuerden que no queremos hacer esto cada vez que entramos a R, queremos que sea automático y rápido.

1.2. Directorios

Donde están tus datos?

Opción 1

```
getwd()
```

```
setwd()
```

Se tiene que copiar y pegar completo y cada quien va a tener diferentes carpetas en sus computadoras.

1.2. Directorios

Donde están tus datos?

```
library(here)
```

```
here()
```

Ventajas:

- Usa la carpeta donde tenemos guardado el script para encontrar el archivo que necesitamos.
- Se puede compartir con colaboradores.
- Se adapta si mueves la carpeta de lugar.

1.3. Cargar datos

Cargar el paquete tidyverse

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

```
## -- Attaching packages ----- tidyverse 1.3.0

## v ggplot2 3.3.3      v purrr   0.3.4
## v tibble  3.1.0      v dplyr   1.0.5
## v tidyr   1.1.2      v stringr 1.4.0
## v readr   1.4.0      v forcats 0.5.1

## Warning: package 'tibble' was built under R version 4.0.4

## Warning: package 'dplyr' was built under R version 4.0.4

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

1.4. Formato CSV

Manualmente: esquina inferior derecha

Files> Doble clic en el archivo de **Pingus_1.csv**

Esta separado por comas, entonces:

`read_csv` se usa cuando los valores están separados por **comas**.

```
Pingus_csv<-read_csv(here("Pingus_1.csv"))
```

```
head(Pingus_csv)
```

1.4. Formato CSV

Doble clic en el archivo de **Pingus_2.csv**

Esta separado por punto y coma.

`read_csv2` se usa para cuando están separados con **punto y coma**.

```
Pingus_csv2<-read_csv(here("Pingus_2.csv"))
```

```
head(Pingus_csv2)
```

Funciono?

Ah! tengo que usar `read_csv2` para ";"

```
Pingus_csv2<-read_csv2(here("Pingus_2.csv"))
```

```
head(Pingus_csv2)
```

1.5. Formato tab

Doble clic en el archivo de **Pingus_3.txt**.

Esta separado por espacios.

`read_tsv` se usa para documentos que tengan tab (espacio) separated values.

```
Pingus_txt<-read_tsv(here("Pingus_3.txt"))
```

```
head(Pingus_txt)
```

1.6. Formato Excel

Hay que cargar la librería readxl

```
library("readxl")
```

read_excel se usa para cargar archivos de excel.

```
Pingus_excel<- read_excel(here("Pingus_4.xlsx"))
```

```
head(Pingus_excel)
```

Hay otros tipos de formatos pero no los vamos a ver hoy, ni creo que los vayamos a necesitar durante el curso.

1.7. Desde Internet

Fuentes de datos en Internet para practicar y que vamos a usar durante el curso:

- [TidyTuesday](#)
- [DatosDeMiercoles](#)

Base de datos de plásticos. Datos liberados el 26 de enero de 2021.

```
plastics <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidydata/master/plastics.csv')
```

Datos de plásticos

```
head(plastics, 5)
```

1.8. Paquetes que contienen datos

```
install.packages("remotes")  
remotes::install_github("cienciadedatos/datos")
```

```
library(datos)  
Pingus<-pinguinos
```

```
head(Pingus)
```

```
## # A tibble: 6 x 8  
##   especie isla   largo_pico_mm alto_pico_mm largo_aleta_mm masa_corporal_g  
##   <fct>   <fct>         <dbl>         <dbl>         <int>         <int>  
## 1 Adelia Torge~         39.1          18.7          181          3750  
## 2 Adelia Torge~         39.5          17.4          186          3800  
## 3 Adelia Torge~         40.3           18           195          3250  
## 4 Adelia Torge~         NA            NA            NA            NA  
## 5 Adelia Torge~         36.7          19.3          193          3450  
## 6 Adelia Torge~         39.3          20.6          190          3650  
## # ... with 1 more variable: anio <int>
```

1.9. Unir documentos

Se pueden unir muchos documentos (En este ejemplo .xlsx) Pero **ojo** el numero de columnas y nombres de las columnas debe ser lo mismo.

```
MainFolder<-here()
FilesTracking <- list.files(MainFolder,pattern="*.xlsx",full.names=TRUE)
ListTracking <- lapply(FilesTracking,read_excel)
ListTracking #Te muestra todos los documentos de la carpeta
AllTracks <- do.call("rbind",ListTracking)
```

1.10. Ejercicios

Carga los datos de pingüinos

Usando `read_csv`

Usando `read_csv2`

Usando `read_tsv`

Usando `read_excel`

Recapitulando

Esta clase:

- Instalar y cargar paquetes
- Directorios
- Cargar csv y excel

Siguiente clase:

- Visualización de datos
- Usar ggplot2



Contacto

Para dudas, comentarios y sugerencias:
Escríbeme a miriamjlerma@gmail.com

Este material esta accesible
y se encuentra en mi [github](#) y mi [página](#)

