Actividad 2

Logo UOC

Lectura y escritura de ficheros de datos

Autor: Xavier Duran Albareda Coordinación: Julià Minguillón

PID\_00233252

Introducción

La exploración, manipulación, visualización y análisis de datos empieza con la carfa de los datos des de diferentes formatos de ficheros. En esta actividad veremos cómo leer com ficheros de tipo CSV y Excel, cargarlos en un DataFrame y volverlos a guardar en otro fichero.

usaremos las librerías [readr](https://readr.tidyverse.org) y [readxl](https://readxl.tidyverse.org), que forman parte de la colección de paquetes [tidyverse](https://tidyverse.org/).





Ficheros CSV

El acrónimo CSV corresponde a *Comma Separated Values*, es decir, ficheros separados por comas. De hecho, veremos que la función de dplyr que usaremos para leer este tipo de ficheros también sirve para leer ficheros separados por otros carácteres, como los TSV o *Tab-Separated Values*, o los ficheros separados por punto y coma.

El fichero CSV que leeremos corresponde al dataset de la competición [Titanic: Machine Learning from Disaster](https://www.kaggle.com/c/titanic) de [Kaggle](https://www.kaggle.com/).

Para leer un fichero CSV usaremos la función [read\_csv](https://readr.tidyverse.org/reference/read_delim.html) de readr.

library(readr)  
df <- read\_csv("data/titanic.csv")

## Parsed with column specification:  
## cols(  
## PassengerId = col\_double(),  
## Survived = col\_double(),  
## Pclass = col\_double(),  
## Name = col\_character(),  
## Sex = col\_character(),  
## Age = col\_double(),  
## SibSp = col\_double(),  
## Parch = col\_double(),  
## Ticket = col\_character(),  
## Fare = col\_double(),  
## Cabin = col\_character(),  
## Embarked = col\_character()  
## )

df

## # A tibble: 891 x 12  
## PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare Cabin  
## <dbl> <dbl> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr> <dbl> <chr>  
## 1 1 0 3 Brau~ male 22 1 0 A/5 2~ 7.25 <NA>   
## 2 2 1 1 Cumi~ fema~ 38 1 0 PC 17~ 71.3 C85   
## 3 3 1 3 Heik~ fema~ 26 0 0 STON/~ 7.92 <NA>   
## 4 4 1 1 Futr~ fema~ 35 1 0 113803 53.1 C123   
## 5 5 0 3 Alle~ male 35 0 0 373450 8.05 <NA>   
## 6 6 0 3 Mora~ male NA 0 0 330877 8.46 <NA>   
## 7 7 0 1 McCa~ male 54 0 0 17463 51.9 E46   
## 8 8 0 3 Pals~ male 2 3 1 349909 21.1 <NA>   
## 9 9 1 3 John~ fema~ 27 0 2 347742 11.1 <NA>   
## 10 10 1 2 Nass~ fema~ 14 1 0 237736 30.1 <NA>   
## # ... with 881 more rows, and 1 more variable: Embarked <chr>

Si queremos leer un fichero que no está separado por comas, usaremos la función [read\_delim](https://readr.tidyverse.org/reference/read_delim.html) y especificaremos el separador con el parámetro delim.

df <- read\_delim("data/titanic.tsv", delim='\t')

## Parsed with column specification:  
## cols(  
## PassengerId = col\_double(),  
## Survived = col\_double(),  
## Pclass = col\_double(),  
## Name = col\_character(),  
## Sex = col\_character(),  
## Age = col\_double(),  
## SibSp = col\_double(),  
## Parch = col\_double(),  
## Ticket = col\_character(),  
## Fare = col\_double(),  
## Cabin = col\_character(),  
## Embarked = col\_character()  
## )

df

## # A tibble: 891 x 12  
## PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare Cabin  
## <dbl> <dbl> <dbl> <chr> <chr> <dbl> <dbl> <dbl> <chr> <dbl> <chr>  
## 1 1 0 3 Brau~ male 22 1 0 A/5 2~ 7.25 <NA>   
## 2 2 1 1 Cumi~ fema~ 38 1 0 PC 17~ 71.3 C85   
## 3 3 1 3 Heik~ fema~ 26 0 0 STON/~ 7.92 <NA>   
## 4 4 1 1 Futr~ fema~ 35 1 0 113803 53.1 C123   
## 5 5 0 3 Alle~ male 35 0 0 373450 8.05 <NA>   
## 6 6 0 3 Mora~ male NA 0 0 330877 8.46 <NA>   
## 7 7 0 1 McCa~ male 54 0 0 17463 51.9 E46   
## 8 8 0 3 Pals~ male 2 3 1 349909 21.1 <NA>   
## 9 9 1 3 John~ fema~ 27 0 2 347742 11.1 <NA>   
## 10 10 1 2 Nass~ fema~ 14 1 0 237736 30.1 <NA>   
## # ... with 881 more rows, and 1 more variable: Embarked <chr>

Para volverlo a guardar en otro fichero, por ejemplo separado por punto y coma, usaremos la función [write\_csv](https://readr.tidyverse.org/reference/write_delim.html) de readr.

write\_csv(df, 'data/titanic\_semicolon.csv')

Ficheros Excel

Aún y ser un software propietario, Excel es una de las herramientas más populares y usadas en el tratamiento de datos. [readxl](https://readxl.tidyverse.org/) es capaz de leer los ficheros que genera este software en sus diferentes versiones.

El fichero Excel que leeremos corresponde al dataset de la competición [TMDB 5000 Movie Dataset](https://www.kaggle.com/tmdb/tmdb-movie-metadata) de [Kaggle](https://www.kaggle.com/). Este Excel tiene tres sheets que corresponden a las películas de los años 1900s, 2000s y 2010s.

Para leer un fichero Excel usaremos la función [read\_excel](https://readxl.tidyverse.org/reference/read_excel.html) de [readxl](https://readxl.tidyverse.org/).

library(readxl)  
xls\_file <- read\_excel("data/movies.xls")  
xls\_file

## # A tibble: 1,338 x 25  
## Title Year Genres Language Country `Content Rating` Duration `Aspect Ratio`  
## <chr> <dbl> <chr> <chr> <chr> <chr> <dbl> <dbl>  
## 1 Into~ 1916 Drama~ <NA> USA Not Rated 123 1.33  
## 2 Over~ 1920 Crime~ <NA> USA <NA> 110 1.33  
## 3 The ~ 1925 Drama~ <NA> USA Not Rated 151 1.33  
## 4 Metr~ 1927 Drama~ German Germany Not Rated 145 1.33  
## 5 Pand~ 1929 Crime~ German Germany Not Rated 110 1.33  
## 6 The ~ 1929 Music~ English USA Passed 100 1.37  
## 7 Hell~ 1930 Drama~ English USA Passed 96 1.2   
## 8 A Fa~ 1932 Drama~ English USA Unrated 79 1.37  
## 9 42nd~ 1933 Comed~ English USA Unrated 89 1.37  
## 10 She ~ 1933 Comed~ English USA Approved 66 1.37  
## # ... with 1,328 more rows, and 17 more variables: Budget <dbl>, `Gross  
## # Earnings` <dbl>, Director <chr>, `Actor 1` <chr>, `Actor 2` <chr>, `Actor  
## # 3` <chr>, `Facebook Likes - Director` <dbl>, `Facebook Likes - Actor  
## # 1` <dbl>, `Facebook Likes - Actor 2` <dbl>, `Facebook Likes - Actor  
## # 3` <dbl>, `Facebook Likes - cast Total` <dbl>, `Facebook likes -  
## # Movie` <dbl>, `Facenumber in posters` <dbl>, `User Votes` <dbl>, `Reviews  
## # by Users` <dbl>, `Reviews by Crtiics` <dbl>, `IMDB Score` <dbl>

Nuestro Excel tiene tres sheets. Si no le decimos nada, por defecto la función [read\_excel](https://readxl.tidyverse.org/reference/read_excel.html) nos carga el primero. Si quermos ver qué sheets tiene, podemos usar la función [excel\_sheets](https://readxl.tidyverse.org/reference/excel_sheets.html).

excel\_sheets("data/movies.xls")

## [1] "1900s" "2000s" "2010s"

Y a continuación podemos especificar que abra otro con el parámetro sheet.

xls\_file <- read\_excel("data/movies.xls", sheet = "2010s")  
xls\_file

## # A tibble: 1,604 x 25  
## Title Year Genres Language Country `Content Rating` Duration `Aspect Ratio`  
## <chr> <dbl> <chr> <chr> <chr> <chr> <dbl> <dbl>  
## 1 127 ~ 2010 Adven~ English USA R 94 1.85  
## 2 3 Ba~ 2010 Drama English USA R 88 NA   
## 3 3  2010 Comed~ German Germany Unrated 119 2.35  
## 4 8: T~ 2010 Docum~ English USA R 80 1.78  
## 5 A Tu~ 2010 Adven~ English France PG 88 2.35  
## 6 Alic~ 2010 Adven~ English USA PG 108 1.85  
## 7 Alic~ 2010 Adven~ English USA PG 108 1.85  
## 8 All ~ 2010 Crime~ English USA R 101 1.85  
## 9 Alph~ 2010 Adven~ English USA PG 90 1.85  
## 10 Amig~ 2010 Drama~ English USA R 124 NA   
## # ... with 1,594 more rows, and 17 more variables: Budget <dbl>, `Gross  
## # Earnings` <dbl>, Director <chr>, `Actor 1` <chr>, `Actor 2` <chr>, `Actor  
## # 3` <chr>, `Facebook Likes - Director` <dbl>, `Facebook Likes - Actor  
## # 1` <dbl>, `Facebook Likes - Actor 2` <dbl>, `Facebook Likes - Actor  
## # 3` <dbl>, `Facebook Likes - cast Total` <dbl>, `Facebook likes -  
## # Movie` <dbl>, `Facenumber in posters` <dbl>, `User Votes` <dbl>, `Reviews  
## # by Users` <dbl>, `Reviews by Crtiics` <dbl>, `IMDB Score` <dbl>

Para volverlo a guardar en otro fichero Excel usaremos la función [write.xlsx](https://rdrr.io/cran/openxlsx/man/write.xlsx.html) de openxlsx.

openxlsx::write.xlsx(xls\_file, file = "data/movies\_2010.xlsx")

Logo UOC