



Universidad del
Rosario

Facultad de
Economía

Laboratorio de
Finanzas

Escuela de Ingeniería,
Ciencia y Tecnología

TALLER DE **DATA SCIENCE** EN **R**



Sueño**SER**

Potencializa tu perfil
académico y profesional.

6:00 p.m a 8:00 p.m

10 al **12** de octubre

Trasciende,
construye
y Lidera

Rosarios
con
Propósito

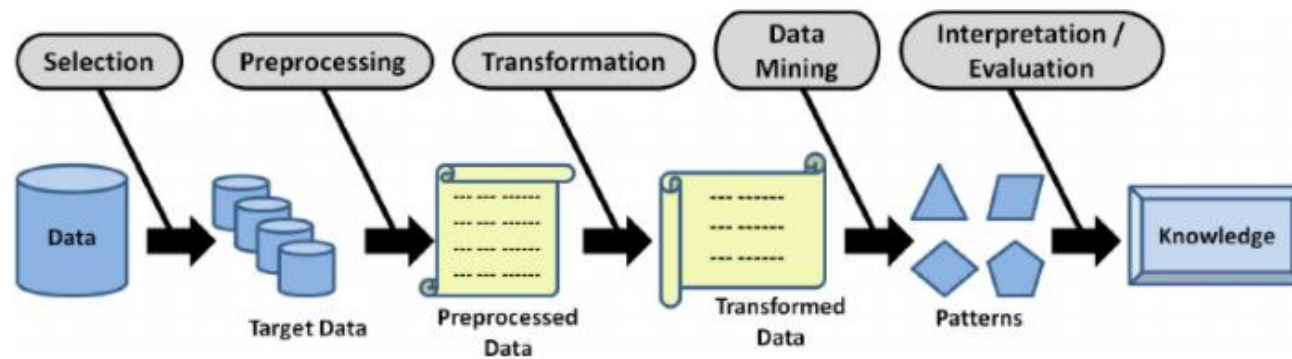
Contenido

- Introducción
- Diferentes formatos de datos
- Lectura de archivos planos
- La librería readr
- Lectura de archivos de Excel



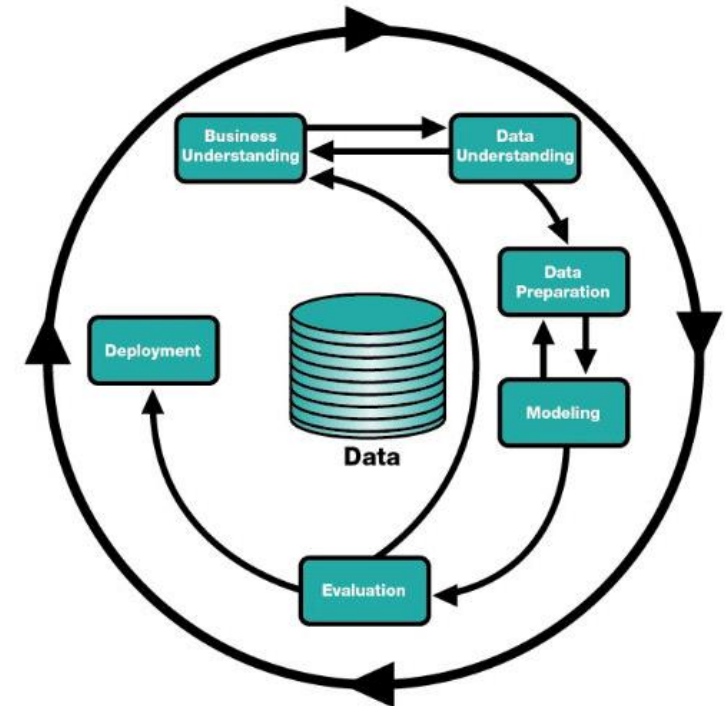
Introducción

Metodologías de analítica famosas



Knowledge Discovery on Databases (KDD)

Introducción



Cross Industry Standard Process for Data Mining (CRISP-DM)

Algunas preguntas

- ¿Qué es la extensión de un archivo?

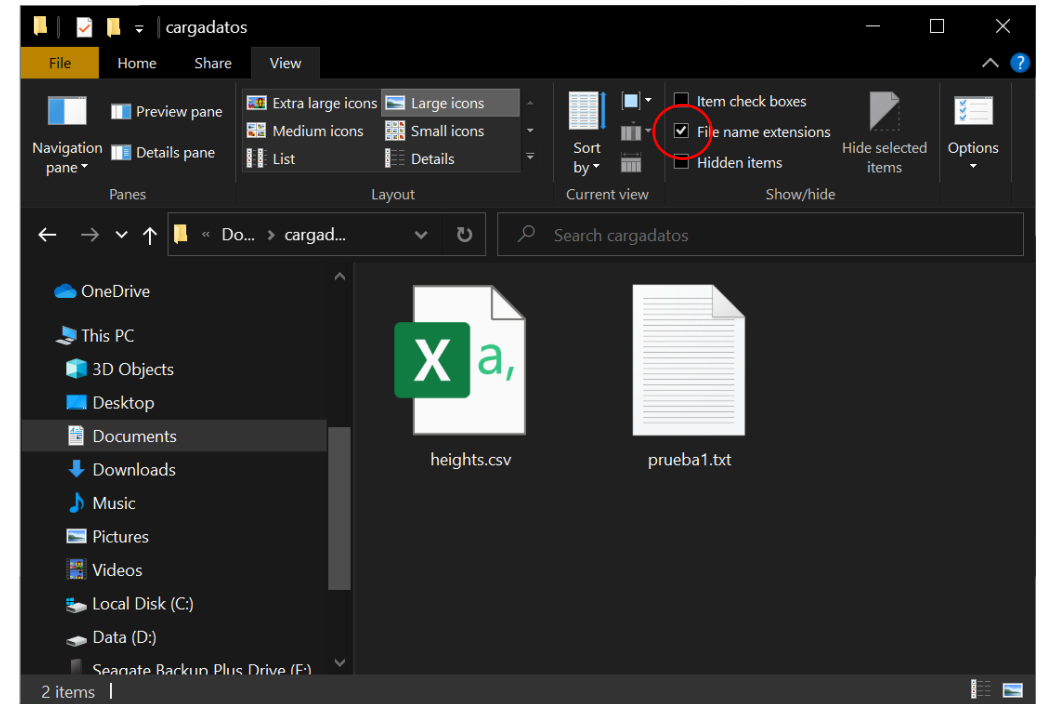
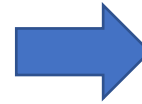
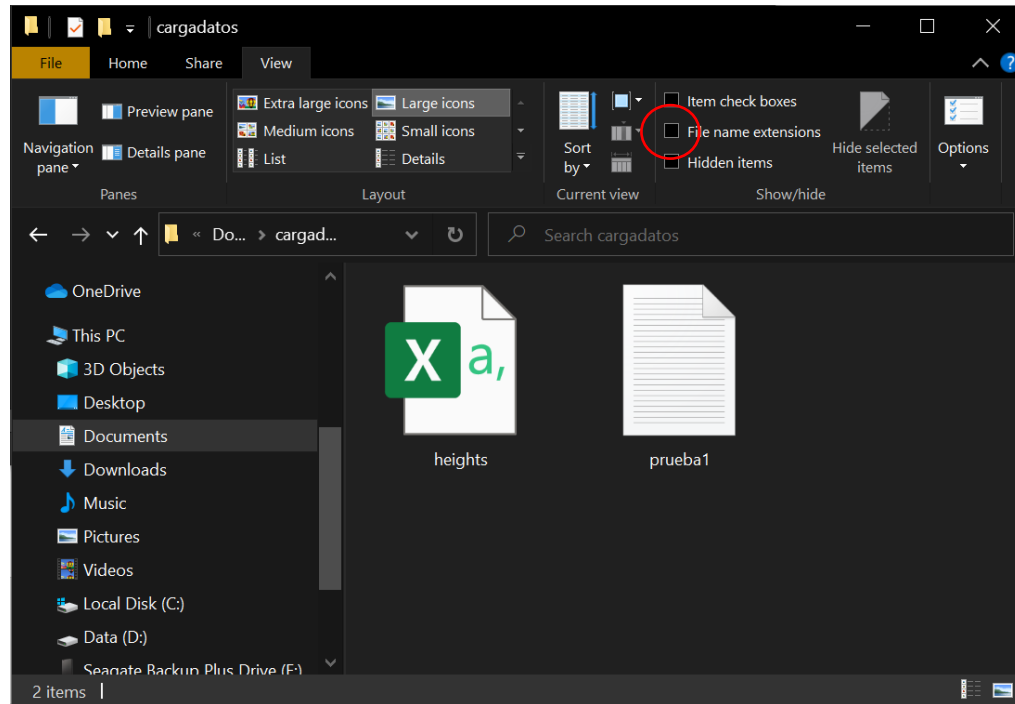


- ¿Qué significan las extensiones csv,.xlsx, txt?
- ¿Dónde están los datos que usualmente cargamos para análisis?

Una recomendación

Introducción

Activar la opción para ver las extensiones de archivos





Diferentes formatos de datos

Según su estructura

Diferentes formatos de datos

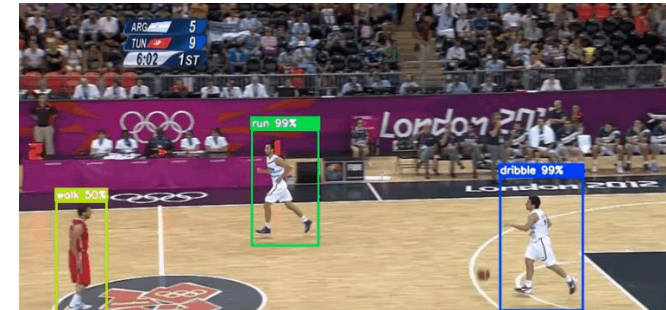
Datos Estructurados

| | nombre | color | edad | altura | peso | puntuacion |
|----|---------|----------|------|--------|------|------------|
| 1: | Paco | Rojo | 24 | 182 | 74.8 | 83 |
| 2: | Juan | Green | 30 | 170 | 70.1 | 500 |
| 3: | Andres | Amarillo | 41 | 169 | 60.0 | 20 |
| 4: | Natalia | Green | 22 | 183 | 75.0 | 865 |
| 5: | Vanesa | Verde | 31 | 178 | 83.9 | 221 |
| 6: | Miriam | Rojo | 35 | 172 | 76.2 | 413 |
| 7: | Juan | Amarillo | 22 | 164 | 68.0 | 902 |

Datos Semi-Estructurados

```
{  
  "marcadores": [  
    {  
      "latitude": 40.416875,  
      "longitude": -3.703308,  
      "city": "Madrid",  
      "description": "Puerta del Sol"  
    },  
    {  
      "latitude": 40.417438,  
      "longitude": -3.693363,  
      "city": "Madrid",  
      "description": "Paseo del Prado"  
    }  
  ]  
}
```

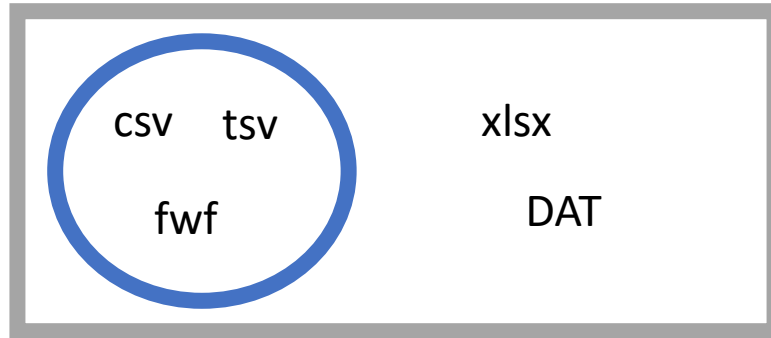
Datos NO Estructurados



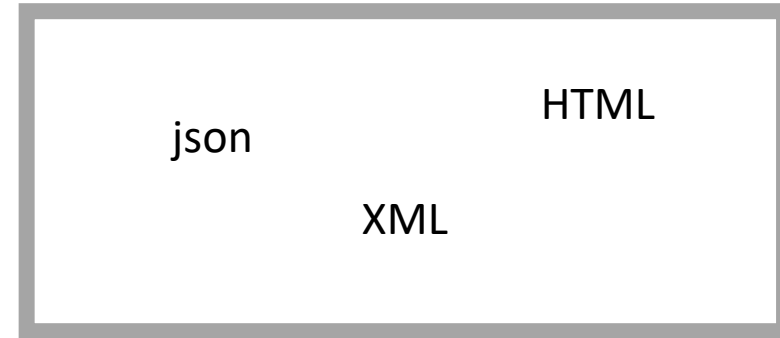
Según su ubicación

Diferentes formatos de datos

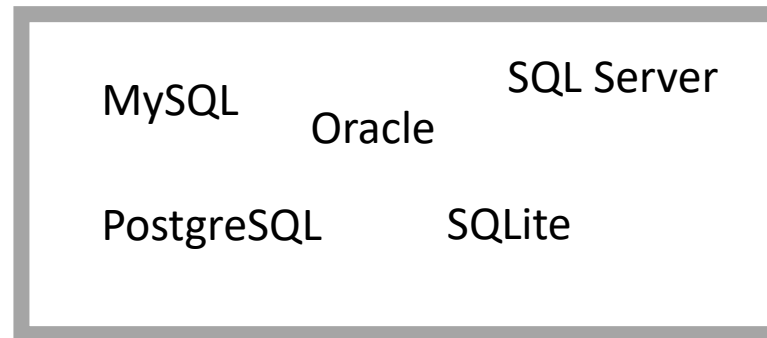
En el disco duro



En la web



En una base de datos





Lectura de archivos planos

Algunas herramientas básicas

Editor de archivos de texto plano

- Notepad de Windows
- Sublime Text
- Notepad++
- Visual Studio Code
- Atom
- Vim
- ...

Librerías de carga de datos de R

- utils
- readr
- readxl

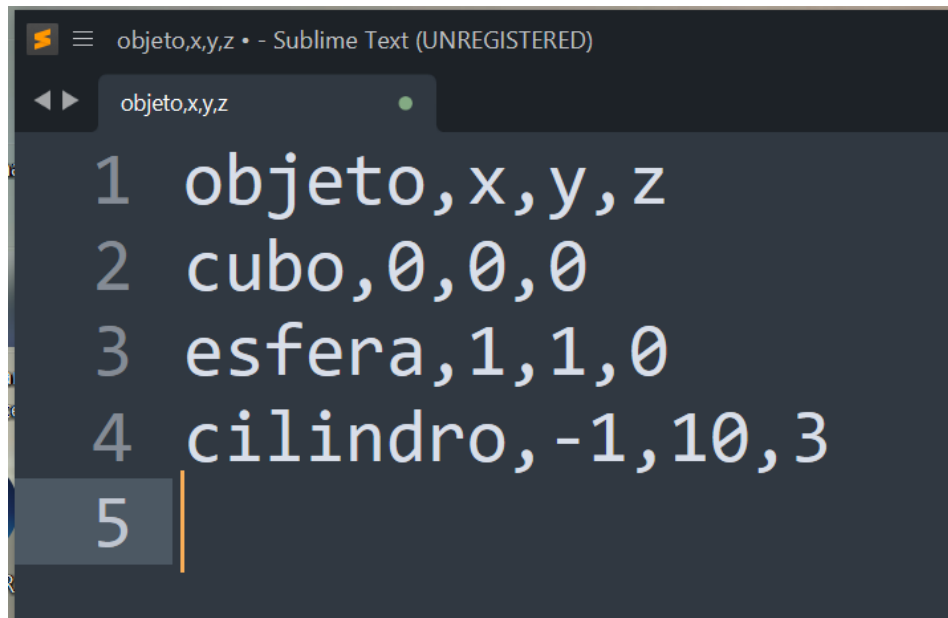
Parámetros de una función de carga

- Archivo (file): el nombre de un archivo o una conexión
- Encabezado (header): Valor lógico que indica si el archivo tiene una línea de encabezado
- Separador: una cadena que indica cómo se separan las columnas (p.ej, por comas, por tabs, por algún símbolo raro)
- Clases de columnas: Datos que indica la clase de cada columna en el conjunto de datos.
- Filas a saltar: el número de líneas para saltar desde el principio
- ...

Archivo de prueba # 1

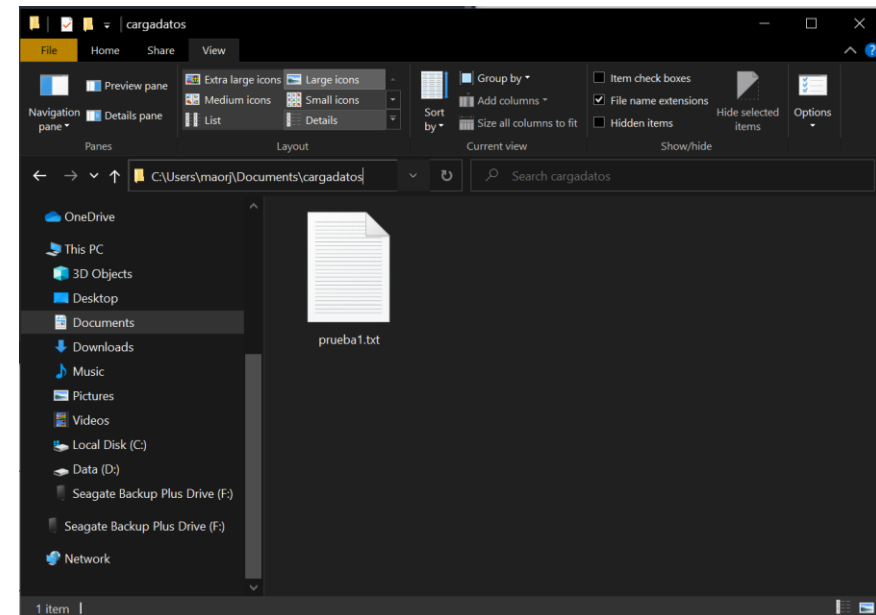
Lectura de archivos planos

1. Abrir el editor de texto de preferencia y escribir lo siguiente:



```
objeto,x,y,z
cubo,0,0,0
esfera,1,1,0
cilindro,-1,10,3
5
```

2. Guardar el archivo en una carpeta



Carga del archivo con librería utils

Las funciones read.* permiten cargar datos

```
x <- read.csv("cargadatos/prueba1.txt")
```

```
x <- read.table("cargadatos/prueba1.txt")
```

Existen varias formas de usar las funciones read.*

- Revisar la documentación (F1)

Archivo de prueba # 2

Lectura de archivos planos

Descarga el siguiente archivo

<https://raw.githubusercontent.com/hadley/r4ds/master/data/heights.csv>

Comprueba el separador de columnas en el archivo con un editor de texto

■ ¿Son comas? ¿Son puntos y comas?
¿Son tabs? ¿Son caracteres raros?

Selecciona el método adecuado para cargar el archivo (Ver documentación)

```
read.table(file, header = FALSE, sep = "", quote = "\"'",  
  dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),  
  row.names, col.names, as.is = !stringsAsFactors,  
  na.strings = "NA", colClasses = NA, nrows = -1,  
  skip = 0, check.names = TRUE, fill = !blank.lines.skip,  
  strip.white = FALSE, blank.lines.skip = TRUE,  
  comment.char = "#",  
  allowEscapes = FALSE, flush = FALSE,  
  stringsAsFactors = FALSE,  
  fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)  
  
read.csv(file, header = TRUE, sep = ",", quote = "\"",  
  dec = ".", fill = TRUE, comment.char = "", ...)  
  
read.csv2(file, header = TRUE, sep = ";", quote = "\"",  
  dec = ",", fill = TRUE, comment.char = "", ...)  
  
read.delim(file, header = TRUE, sep = "\t", quote = "\"",  
  dec = ".", fill = TRUE, comment.char = "", ...)  
  
read.delim2(file, header = TRUE, sep = "\t", quote = "\"",  
  dec = ",", fill = TRUE, comment.char = "", ...)
```

```
x <- read.csv("cargadatos/heights.csv")
```

Codificación de caracteres (encoding)

Lectura de archivos planos

Proceso de asignación de números a caracteres gráficos

USASCII code chart

| b7 b6 b5 b4 b3 b2 b1 b0 | | | | | Column | | | | | | | |
|-------------------------|----|----|----|----|--------|-----|----|---|---|---|---|-----|
| Bits | | | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| b4 | b3 | b2 | b1 | b0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 0 | 0 | 0 | 0 | NUL | DLE | SP | 0 | @ | P | ` | p |
| 0 | 0 | 0 | 1 | 0 | SOH | DC1 | ! | 1 | A | Q | a | q |
| 0 | 0 | 1 | 0 | 0 | STX | DC2 | " | 2 | B | R | b | r |
| 0 | 0 | 1 | 1 | 0 | ETX | DC3 | # | 3 | C | S | c | s |
| 0 | 1 | 0 | 0 | 0 | EOT | DC4 | \$ | 4 | D | T | d | t |
| 0 | 1 | 0 | 1 | 0 | ENQ | NAK | % | 5 | E | U | e | u |
| 0 | 1 | 1 | 0 | 0 | ACK | SYN | & | 6 | F | V | f | v |
| 0 | 1 | 1 | 1 | 0 | BEL | ETB | ' | 7 | G | W | g | w |
| 1 | 0 | 0 | 0 | 0 | BS | CAN | (| 8 | H | X | h | x |
| 1 | 0 | 0 | 1 | 0 | HT | EM |) | 9 | I | Y | i | y |
| 1 | 0 | 1 | 0 | 0 | LF | SUB | * | : | J | Z | j | z |
| 1 | 0 | 1 | 1 | 0 | VT | ESC | + | ; | K | [| k | { |
| 1 | 1 | 0 | 0 | 0 | FF | FS | , | < | L | \ | l | |
| 1 | 1 | 0 | 1 | 0 | CR | GS | - | = | M |] | m | } |
| 1 | 1 | 1 | 0 | 0 | SO | RS | . | > | N | ^ | n | ~ |
| 1 | 1 | 1 | 1 | 0 | SI | US | / | ? | O | _ | o | DEL |

Text to Binary Converter

Enter ASCII/Unicode text string and press the Convert to get "01000101 01111000 01100001 01101101 0111

From

To

Text

Binary

Open File

Paste text or drop text file

A

Character encoding (optional)

ASCII/UTF-8

Output delimiter string (optional)

Space

Convert

Reset

Swap

01000001

<https://cs61.seas.harvard.edu/site/2020/Unicode/>
<https://www.rapidtables.com/convert/number/ascii-to-binary.html>

Codificaciones comunes

Lectura de archivos planos

UTF-8

UTF-16 LE

UTF-16 BE

Western (Windows 1252)

Western (ISO 8859-1)

Western (ISO 8859-3)

Western (ISO 8859-15)

Western (Mac Roman)

DOS (CP 437)

Arabic (Windows 1256)

Arabic (ISO 8859-6)

Baltic (Windows 1257)

Baltic (ISO 8859-4)

Celtic (ISO 8859-14)

Central European (Windows 1250)

Central European (ISO 8859-2)

Central European (Mac)

Cyrillic (Windows 1251)

Cyrillic (Windows 866)

Table for Debugging Common UTF-8 Character Encoding Problems.

| Code Point | | Characters | | UTF-8 Bytes | Code Point | | Characters | | UTF-8 Bytes |
|------------|--------------|------------|--------|-------------|------------|--------------|------------|--------|-------------|
| Unicode | Windows 1252 | Expected | Actual | | Unicode | Windows 1252 | Expected | Actual | |
| U+20AC | 0x80 | € | â , ¬ | %E2 %82 %AC | U+00C0 | 0xC0 | À | Ã € | %C3 %80 |
| | 0x81 | | | | U+00C1 | 0xC1 | Á | Ã Á | %C3 %81 |
| U+201A | 0x82 | , | â € š | %E2 %80 %9A | U+00C2 | 0xC2 | Â | Ã , | %C3 %82 |
| U+0192 | 0x83 | f | Æ ’ | %C6 %92 | U+00C3 | 0xC3 | Ã | Ã f | %C3 %83 |
| U+201E | 0x84 | „ | â € ž | %E2 %80 %9E | U+00C4 | 0xC4 | Ä | Ã „ | %C3 %84 |
| U+2026 | 0x85 | ... | â € | %E2 %80 %A6 | U+00C5 | 0xC5 | Å | Ã ... | %C3 %85 |
| U+2020 | 0x86 | † | â € | %E2 %80 %A0 | U+00C6 | 0xC6 | Æ | Ã † | %C3 %86 |
| U+2021 | 0x87 | ‡ | â € ‡ | %E2 %80 %A1 | U+00C7 | 0xC7 | Ç | Ã ‡ | %C3 %87 |
| U+02C6 | 0x88 | ˆ | Ë † | %CB %86 | U+00C8 | 0xC8 | È | Ã ˆ | %C3 %88 |
| U+2030 | 0x89 | %o | â € ° | %E2 %80 %B0 | U+00C9 | 0xC9 | É | Ã %o | %C3 %89 |
| U+0160 | 0x8A | Š | Å | %C5 %A0 | U+00CA | 0xCA | Ê | Ã Š | %C3 %8A |
| U+2039 | 0x8B | ‹ | â € ’ | %E2 %80 %B9 | U+00CB | 0xCB | Ë | Ã ‹ | %C3 %8B |
| U+0152 | 0x8C | Œ | Å ’ | %C5 %92 | U+00CC | 0xCC | Ì | Ã Œ | %C3 %8C |
| | 0x8D | | | | U+00CD | 0xCD | Í | Ã | %C3 %8D |
| U+017D | 0x8E | Ž | Å ½ | %C5 %BD | U+00CE | 0xCE | Î | Ã Ž | %C3 %8E |
| | 0x8F | | | | U+00CF | 0xCF | Ï | Ã | %C3 %8F |
| | 0x90 | | | | U+00D0 | 0xD0 | Ð | Ã | %C3 %90 |
| U+2018 | 0x91 | ‘ | â € ~ | %E2 %80 %98 | U+00D1 | 0xD1 | Ñ | Ã ‘ | %C3 %91 |
| U+2019 | 0x92 | ’ | â € ™ | %E2 %80 %99 | U+00D2 | 0xD2 | Ò | Ã ’ | %C3 %92 |
| U+201C | 0x93 | “ | â € œ | %E2 %80 %9C | U+00D3 | 0xD3 | Ó | Ã “ | %C3 %93 |
| U+201D | 0x94 | ” | â € | %E2 %80 %9D | U+00D4 | 0xD4 | Ô | Ã ” | %C3 %94 |
| U+2022 | 0x95 | • | â € ¢ | %E2 %80 %A2 | U+00D5 | 0xD5 | Õ | Ã • | %C3 %95 |

<https://www.i18nqa.com/debug/utf8-debug.html>



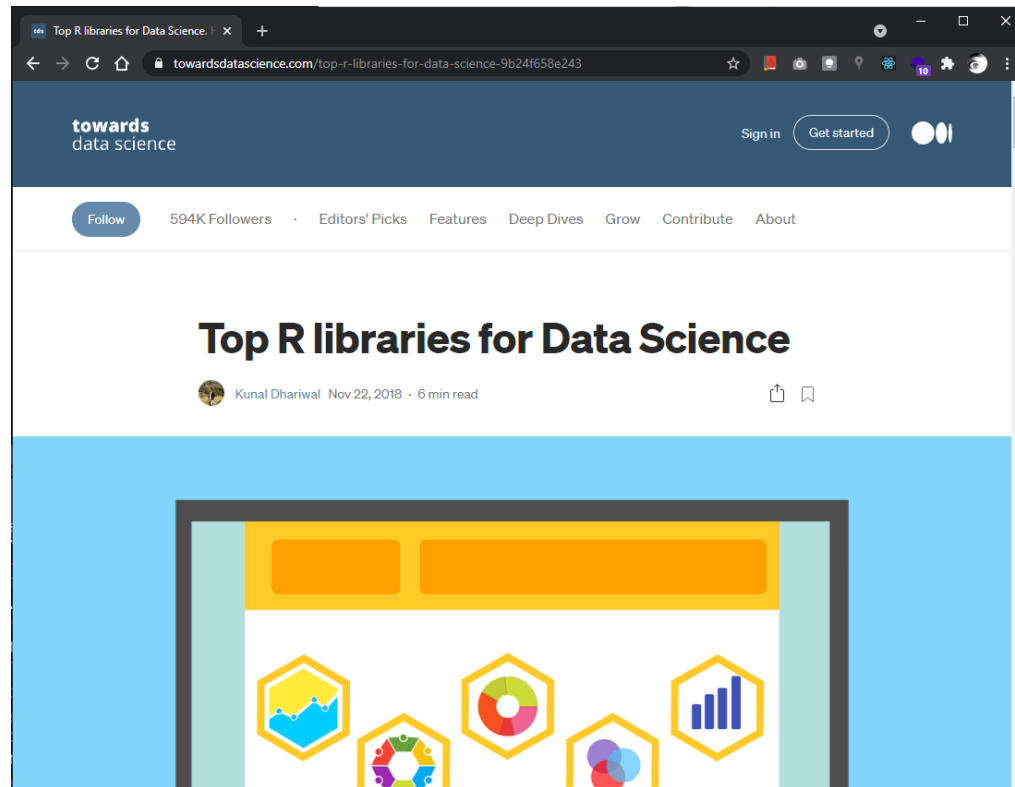
La librería readr

¿Qué es?

- Es una librería más para cargar datos
- Proporcionar una forma rápida y sencilla de leer datos rectangulares (como csv, tsv y fwf).
- Diseñada para analizar de manera flexible muchos tipos de datos que se encuentran en la naturaleza
- Muestra claramente cuando los datos fallan en cargar

Instalación de paquetes

La librería readr



```
# Se necesitan instalar una vez  
# por computador
```

```
install.packages("nombrepaquete")
```

```
# pero hay que cargarlo cada vez  
# que se inicia R
```

```
library(nombredelpaquete)
```

Instalación de paquetes

La librería readr

librerías importantes

library(readr)



library(tidyr)

library(dplyr)

library(ggplot2)

Paquete **tidyverse**

La librería readr

`library(tidyverse)` las incluye todas

```
> library(tidyverse)
-- Attaching packages ----- tidyverse 1.3.1 --
v ggplot2 3.3.3      v purrr   0.3.4
v tibble  3.1.1      v dplyr   1.0.5
v tidyr   1.1.3      v stringr 1.4.0
v readr   1.4.0      v forcats 0.5.1
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
```

Modo de uso

La librería readr

Tiene funciones muy parecidas a las funciones de utils

```
read_csv()  
read_csv2()  
read_tsv()  
read_delim()  
read_fwf()  
read_table()  
read_log()
```

Ejemplos:

```
x <- read_csv("cargadatos/prueba1.txt")
```

```
x <- read_csv("cargadatos/heights.csv ")
```

Especificación de tipos de datos

La librería readr

readr usa heurísticas para detectar el tipo de dato de cada columna

En algunos casos no detecta correctamente el tipo de dato, lo cual podremos hacer manualmente

Ejemplo:

- Descarga el archivo trickydata.csv
https://drive.google.com/file/d/1ABfwmOXoMAdGz_c9g_pUjX-93KX9JdOg/view?usp=sharing
- ¿Qué tipo de datos tiene el archivo?
- ¿Encuentras problemas de calidad en los datos?

Especificación de tipos de datos

Carga los datos en R y revisa el formato de los tipos de datos

- Sin especificar el tipo de dato

```
tricky <- read_csv("cargadatos/trickydata.csv")
```

- Especificando el tipo de dato

```
tricky <- read_csv("cargadatos/trickydata.csv",  
  col_types = list(  
    x = col_double()  
  )  
)
```

Nombre de la
variable

Tipo de variable

Tipos de datos

La librería readr

| Type | Function | Expects |
|------------|------------------------------|-------------------------------------|
| Logical | <code>col_logical()</code> | T, F, TRUE, FALSE |
| Integer | <code>col_integer()</code> | Integers |
| Double | <code>col_double()</code> | Real numbers, include scientific |
| Character | <code>col_character()</code> | Anything |
| Dates | <code>col_date()</code> | 2010-10-20 |
| Date times | <code>col_datetime()</code> | 2010-10-20 20:15 |

Sobre las fechas

La librería readr

<https://xkcd.com/1179/>


PUBLIC SERVICE ANNOUNCEMENT:

OUR DIFFERENT WAYS OF WRITING DATES AS NUMBERS CAN LEAD TO ONLINE CONFUSION. THAT'S WHY IN 1988 ISO SET A GLOBAL STANDARD NUMERIC DATE FORMAT.

THIS IS ***THE*** CORRECT WAY TO WRITE NUMERIC DATES:

2013-02-27

THE FOLLOWING FORMATS ARE THEREFORE DISCOURAGED:

02/27/2013 02/27/13 27/02/2013 27/02/13
20130227 2013.02.27 27.02.13 27-02-13
27.2.13 2013.II.27. $27\frac{1}{2}$ -13 2013.158904109
MMXIII-II-XXVII MMXIII $\frac{\text{LVII}}{\text{CCCLXV}}$ 1330300800
 $((3+3) \times (111+1) - 1) \times 3 / 3 - 1 / 3^3$ 2013 miss
10/11011/1101 02/27/20/13 $\begin{matrix} 2 & 3 & 1 & 4 \\ 0 & 1 & 2 & 3 & 7 \\ & 5 & 6 & 7 & 8 \end{matrix}$ 

Archivo de prueba # 3

- Descarga el archivo weather.csv
<https://drive.google.com/file/d/1YLmgmoDREMAVcXcFiJEp-6cKqGFBvfNq/view?usp=sharing>
- Revisa los tipos de datos de cada columna mediante un editor de texto, identifica las columnas que son
 - Datos de texto
 - Números enteros
 - Números decimales
 - Fechas
- Carga el archivo en R usando readr ¿la heurística detectó correctamente las columnas?



Lectura de archivos de Excel

Actividad

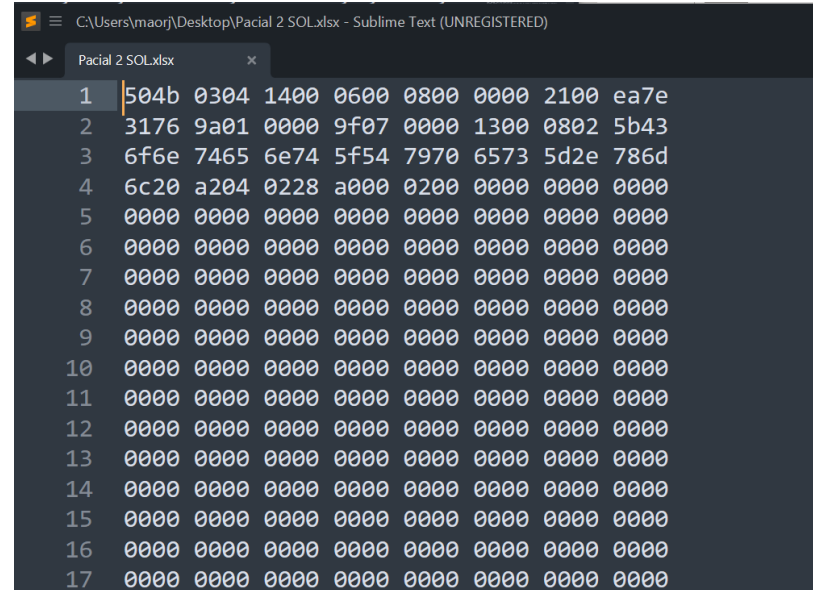
Lectura de archivos de Excel

- Intenta abrir un archivo de Excel con un editor de texto



Archivos de Excel

Lectura de archivos de Excel



- Un archivo de Excel es diferente al archivo de texto plano
- No se puede abrir con un editor de texto
- Se requiere conocer la lógica de codificación del archivo (o tener una librería que lo sepa)

```
library(readxl)
```

Habilita la función `read_excel()`

- Permite definir parámetros propios de un archivo de Excel:
 - Número o nombre de hoja
 - Rango de celdas con los datos (p.je C1:E4)
 - Filas o columnas específicas
 - Qué se entiende como un dato nulo (NA)

Instalación de paquetes

```
library(readxls)
```

Habilita la función `read_excel()`

- Permite definir parámetros propios de un archivo de Excel:
 - Número o nombre de hoja
 - Rango de celdas con los datos (p.je C1:E4)
 - Filas o columnas específicas
 - Qué se entiende como un dato nulo (NA)

Gracias por tu asistencia y participación 😊

Contacto



miguela.orjuela@urosario.edu.co



<https://www.linkedin.com/in/miguel-orjuela/>



<https://github.com/maorjuela73>