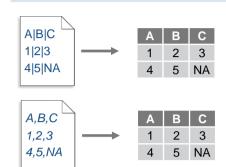
# Importación de datos con tidyverse : : guía rápida



## Leer Datos Tabulares con readr

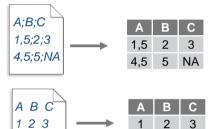
read\_\*(file, col\_names = TRUE, col\_types = NULL, col\_select = NULL, id = NULL, locale, n\_max = Inf, skip = 0, na = c("", "NA"), guess\_max = min(1000, n\_max), show\_col\_types = TRUE) Vea ?read\_delim



**read\_delim(**"file.txt", delim = "|") Leer archivos con cualquier delimitador. Si no se especifica ningún delimitador, adivinará automáticamente.Para crear file.txt, ejecute: write\_file("A|B|C\n1|2|3\n4|5|NA", file = "file.txt")

read\_csv("file.csv") Leer un archivo delimitado por comas con puntos como decimales.

 $write_file("A,B,C\n1,2,3\n4,5,NA",file = "file.csv")$ 



**read\_csv2(**"file2.csv") Leer archivos delimitados por punto y coma con comas como decimales.

write\_file("A;B;C\n1,5;2;3\n4,5;5;NA", file = "file2.csv")

**read\_tsv(**"file.tsv") Leer un archivo delimitado por tabulaciones. También **read\_table()**. **read\_fwf(**"file.tsv", fwf\_widths(c(2, 2, NA))) lee un archivo de ancho fijo.

write file("A\tB\tC\n1\t2\t3\n4\t5\tNA\n", file = "file.tsv")

#### **ARGUMENTOS ÚTILES**

4 5 NA

Α	В	С	Sin encabezado
1	2	3	read_csv("file.csv", col_names = FALS
4	5	NA	
	_		Dranavajanav anasharada
Х	У	Z	Proporcionar encabezado
Α	В	С	read_csv("file.csv",
1	2	3	col_names = c("x", "y", "z"))
4	5	NA	
	<b>→</b>		Leer varios archivos en una sola tab read_csv(c("f1.csv", "f2.csv", "f3.csv"), id = "origin_file")

4 5 NA



#### Saltar líneas

read\_csv("file.csv", skip = 1)

Leer un subconjunto de líneas read\_csv("file.csv", n\_max = 1)



1 2 3

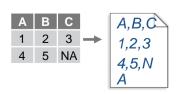
Leer valores como faltantes read\_csv("file.csv", na = c("1"))



Especificar marcas decimales
read\_delim("file2.csv", locale =
 locale(decimal mark = ","))

## Guarde Datos con readr

write\_\*(x, file, na = "NA", append, col\_names, quote, escape, eol, num\_threads, progress)



write\_delim(x, file, delim = " ") Escribir archivos con cualquier delimitador.

write\_csv(x, file) Escribir un archivo delimitado por comas.

write\_csv2(x, file) Escribir un archivo delimitado por punto y coma.

write\_tsv(x, file) Escribir un archivo delimitado por tabulaciones.

Uno de los primeros pasos de un proyecto es importar datos externos a R. Los datos a menudo se almacenan en formatos tabulares, como archivos csv u hojas de cálculo.



La primera página de esta hoja muestra cómo importar y guardar archivos de texto en R usando **readr**.



La última página muestra cóm importar datos de hojas de cálculo desde archivos de Excel usando readxl o Google Sheets usando googlesheets4.

#### **OTROS TIPOS DE DATOS**

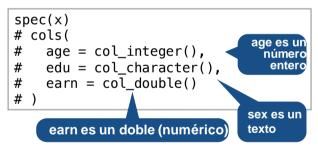
Pruebe uno de los siguientes paquetes para importar otros tipos de archivos:

- haven archivos SPSS, Stata, y SAS
- **DBI** bases de datos
- **isonlite** ison
- xml2 XML
- httr APIs
- rvest HTML
- readr::read lines() datos de texto

# Especificación de columna con readr

Las especificaciones de columna definen el tipo de datos con el que se importará cada columna de un archivo. De forma predeterminada, readr generará una especificación de columna cuando se lea un archivo y generará un resumen.

**spec(x)** Extre la especificación completa de la columna para el marco de datos importado dado.



#### **TIPOS DE COLUMNAS**

Cada tipo de columna tiene una función y la abreviatura correspondiente.

- col\_logical() "I"
- col\_integer() "i"
- col\_double() "d"col\_number() "n"
- col\_character() "c"
- col factor(levels, ordered = FALSE) "f"
- col datetime(format = "") "T"
- **col date(**format = "") "D"
- **col time(**format = "") "t"
- col\_skip() "-", " "
- col\_guess() "?"

#### ARGUMENTOS DE COLUMNA ÚTILES

Ocultar el mensaje de especificación de col read\_\*(file, show\_col\_types = FALSE)

Seleccionar las columnas que se van a importar Usar ayudantes de nombres, posición o selección. read\_\*(file, col\_select = c(age, earn))

#### Adivinar tipos de columnas

Para adivinar un tipo de columna, read\_ \*() Examina las primeras 1000 filas de datos. Aumenta con **guess\_max**.

read\_\*(file, guess\_max = Inf)

# DEFINIR ESPECIFICACIÓN DE COLUMNA Establecer un tipo predeterminado

# read\_csv( file, col\_type = list(.default = col\_double()) )

## Usar el tipo de columna o la abreviatura read\_csv(

```
file,
col_type = list(x = col_double(), y = "l", z = "_")
```

#### Usa una sola abreviatura

```
# col types: skip, guess, integer, logical,
character
read_csv(
    file,
    col_type = "_?ilc"
)
```



## Importar hojas de cálculo

## con readxl

#### LEER ARCHIVOS DE EXCEL



read\_excel(path, sheet = NULL, range = NULL) Leer un archivo .xls o .xlsx en función de la extensión del archivo. Consulte la primera página para conocer más sobre otros argumentos de read. Además read\_xls() y read\_xlsx(). read\_excel("excel\_file.xlsx")

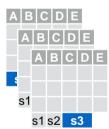
#### **LEER HOJAS**



read\_excel(path, sheet = NULL) Especifique qué hoja leer por posición o nombre.  $read_excel(path, sheet = 1)$ read\_excel(path, sheet = "s1")



excel\_sheets(path) Obtener un vector de nombres de hoja. excel\_sheets("excel\_file.xlsx")



#### Para leer múltiples hojas:

- 1. Obtener un vector de nombres de hoja a partir de la ruta del archivo.
- 2. Establezca los nombres de los vectores para que sean los nombres de las hojas.
- 3. Utilice purrr::map() y purrr::list rbind() para leer varios archivos en un marco de datos.

path <- "your\_file\_path.xlsx" path |> excel\_sheets() |> set\_names() |> map(read\_excel, path = path) |> list\_rbind()

#### OTROS PAQUETES UTILES DE EXCEL

Para funciones escriban datos en archivos de Excel. consulte:

- openxisx
- writexl

Para trabajar con datos de Excel no tabulares, consulte:

tidyxl



#### ESPECIFICACIÓN DE COLUMNAS READXL

Las especificaciones de columna definen el tipo de datos con el que se importará cada columna de un archivo.

Utilice el argumento col types de read\_excel() para establecer la especificación de la columna

#### Adivinar tipos de columna

Para adivinar un tipo de columna, read excel() mira las primeras 1000 filas de datos. Aumenta con el argumento guess\_max. read\_excel(path, guess\_max = Inf)

Establezca todas las columnas en el mismo tipo, por ejemplo, carácter read excel(path, col types = "text")

#### Establecer cada columna individualmente read excel(

path, col\_types = c("text", "quess". 'quess", "numeric")

#### **TIPOS DE COLUMNAS**

logical	numeric	text	date	list
TRUE	2	hello	1947-01-08	hello
FALSE	3,45	world	1956-10-21	1

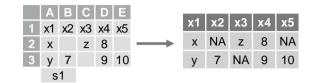
- skip
- logical
- date

- quess
- numeric list
  - - text

Utilice la lista para las columnas que incluyen varios tipos de datos. Consulte tidvr v purrr para ver los datos de las columnas de lista.

# con googlesheets4

#### **LEER HOJAS**



read\_sheet(ss, sheet = NULL, range = NULL) Lee una hoja de una URL, un ID de hoja o un dribble del paquete de googledrive. Consulte la primera página para conocer más argumentos de read. Iqual que range read().

#### METADATOS DE HOJAS DE CÁLCULO

URLs están en la forma:

https://docs.google.com/spreadsheets/d/ SPREADSHEET ID/edit#gid=SHEET ID

gs4 get(ss). Obtener metadatos de la hoja de

gs4\_find(...) Obtener datos en todos los archivos de hojas de cálculo.

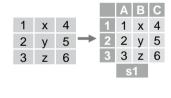
sheet\_properties(ss) Obtenga una serie de propiedades para cada hoja de cálculo. Además sheet names().

#### ESCRIBA HOJAS DE CÁLCULO

ABC

1 x1 x2 x3

2 1 x 4



ABCD

2

2 y 5

3 z 6

write\_sheet(data, ss = NULL, sheet = NULL) Escribir un marco de datos en una hoja nueva o existente.

gs4\_create(name, ..., sheets = NULL) Crear una nueva hoia con un vector de nombres, un marco de datos o una lista (con nombre) de marcos de datos.

sheet\_append(ss, data, sheet = 1) Agregar filas al final de una hoja de cálculo.

### read\_sheet(path, col\_types = "c") Establecer cada columna individualmente

**COLUMNAS** 

de un archivo.

# tipos: skip, guess, integer, logical, character read\_sheets(ss, col\_types = "\_?ilc")

GOOGLESHEETS4 ESPECIFICACIÓN DE

Las especificaciones de columna definen el tipo

de datos con el que se importará cada columna

read\_sheet()/range\_read() para establecer la

read\_sheet()/range\_read() mira las primeras

1000 filas de datos. Aumenta con guess\_max.

Establezca todas las columnas en el mismo

Utilice el argumento col\_types de

read sheet(path, guess max = Inf)

especificación de columna.

Adivinar tipos de columna

Para adivinar una columna.

tipo, por ejemplo, carácter

googlesheets

#### **COLUMN TYPES**

1	n	С	D	L
TRUE	2	hello	1947-01-08	hello
FALSE	3,45	world	1956-10-21	1

· datetime - "T"

· character - "c"

• list-column - "L"

list of raw cell

data.

• cell - "C" Returns

- skip " " or "-"
- guess "?"
- logical "I"
- integer "i"
- double "d"
- numeric "n" • date - "D"

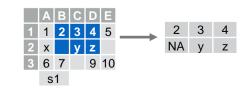
Utilice la lista para las columnas que incluyen varios tipos de datos. Ver tidyr y purrr para columnas de lista.

#### **OPERACIONES A NIVEL DE ARCHIVO**

googlesheets4 también ofrece formas de modificar otros aspectos de las hojas (por ejemplo, congelar filas, establecer el ancho de la columna, administrar hojas (de trabajo)). Ir a googlesheets4.tidyverse.org para leer más.

Para operaciones de archivo completo (por ejemplo, cambiar el nombre, compartir, colocar dentro de una carpeta), consulte el paquete tidyverse googledrive en googledrive.tidyverse.org

#### ESPECIFICACIÓN DE CELDAS PARA READXL Y GOOGLESHEETS4



Utilice el argumento range de readxl::read excel() o qooglesheets4::read sheet() para leer un subconjunto de celdas de una hoja.

read\_excel(path, range = "Sheet1!B1:D2") read sheet(ss, range = "B1:D2")

Utilice también el argumento range con funciones de especificación de celda cell\_limits(), cell\_rows(), cell\_cols(), y anchored().

