

# 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](#)

A B C
1 2 3
4 5 NA

A	B	C
1	2	3
4	5	NA

**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")`

A,B,C
1,2,3
4,5,NA

A	B	C
1	2	3
4	5	NA

**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")`

A;B;C
1,5;2;3
4,5;5;NA

A	B	C
1,5	2	3
4,5	5	NA

**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")`

A B C
1 2 3
4 5 NA

A	B	C
1	2	3
4	5	NA

**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

A	B	C
1	2	3
4	5	NA

#### Sin encabezado

`read_csv("file.csv", col_names = FALSE)`

1	2	3
4	5	NA

#### Saltar líneas

`read_csv("file.csv", skip = 1)`

A	B	C
1	2	3

#### Leer un subconjunto de líneas

`read_csv("file.csv", n_max = 1)`

A	B	C
NA	2	3
4	5	NA

#### Leer valores como faltantes

`read_csv("file.csv", na = c("1"))`

A;B;C
1,5;2;3,0

#### Especificar marcas decimales

`read_delim("file2.csv", locale = locale(decimal_mark = ","))`

x	y	z
A	B	C
1	2	3
4	5	NA

#### Proporcionar encabezado

`read_csv("file.csv", col_names = c("x", "y", "z"))`



#### Leer varios archivos en una sola tabla

`read_csv(c("f1.csv", "f2.csv", "f3.csv"), id = "origin_file")`

## Guarde Datos con readr

`write_*(x, file, na = "NA", append, col_names, quote, escape, eol, num_threads, progress)`

A	B	C
1	2	3
4	5	NA

**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 cómo importar datos de hojas de cálculo desde archivos de Excel usando **readxl** o Google Sheets usando **googlesheets4**.

## 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)** Extrae la especificación completa de la columna para el marco de datos importado dado.

```
spec(x)
# cols(
#   age = col_integer(),
#   edu = col_character(),
#   earn = col_double()
# )
```

age es un número entero  
sex es un texto  
earn es un doble (numérico)

### TIPOS DE COLUMNAS

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

- col\_logical()** - "l"
- 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()** - "?"

### 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
- jsonlite** - json
- xml2** - XML
- httr** - APIs
- rvest** - HTML
- readr::read\_lines()** - datos de texto

### 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

	A	B	C	D	E
1	x1	x2	x3	x4	x5
2	x		z	8	
3	y	7		9	10

s1

x1	x2	x3	x4	x5
x	NA	z	8	NA
y	7	NA	9	10

**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

A	B	C	D	E

s1 s2 s3

**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")`

A	B	C	D	E

s1 s2 s3

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 ÚTILES DE EXCEL

Para funciones escriban datos en archivos de Excel, consulte:

- **openxlsx**
- **writexl**

Para trabajar con datos de Excel no tabulares, consulte:

- **tidyxl**



## con googlesheets4

### LEER HOJAS

	A	B	C	D	E
1	x1	x2	x3	x4	x5
2	x		z	8	
3	y	7		9	10

s1

x1	x2	x3	x4	x5
x	NA	z	8	NA
y	7	NA	9	10

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

**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

1	x	4
2	y	5
3	z	6

s1

A	B	C
1	x	4
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 hoja 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.

### ESPECIFICACIÓN DE CELDAS PARA READXL Y GOOGLESHEETS4

### 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", "guess", "guess", "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
- guess
- logical
- numeric
- text
- date
- list

Utilice la lista para las columnas que incluyen varios tipos de datos. Consulte **tidyr** y **purrr** para ver los datos de las columnas de lista.

A	B	C	D	E
1	1	2	3	4
2	x	y	z	
3	6	7		9

s1

2	3	4
NA	y	z

Utilice el argumento **range** de **readxl::read\_excel()** o **googlesheets4::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()**.



### GOOGLESHEETS4 ESPECIFICACIÓN DE COLUMNAS

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\_sheet()/range\_read()** para establecer la especificación de columna.

### Adivinar tipos de columna

Para adivinar una columna, **read\_sheet()/range\_read()** mira las primeras 1000 filas de datos. Aumenta con **guess\_max**.  
`read_sheet(path, guess_max = Inf)`

**Establezca todas las columnas en el mismo tipo, por ejemplo, carácter**  
`read_sheet(path, col_types = "c")`

**Establecer cada columna individualmente**  
# tipos: skip, guess, integer, logical, character  
`read_sheets(ss, col_types = "?ilc")`

### COLUMN TYPES

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

- skip - "?" or "-"
- guess - "?"
- logical - "l"
- integer - "i"
- double - "d"
- numeric - "n"
- date - "D"
- datetime - "T"
- character - "c"
- list-column - "L"
- cell - "C" Returns list of raw cell data.

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](https://tidyverse.org/guides/googlesheets4/) 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](https://tidyverse.org/guides/googledrive/).