Importación de datos

En esta lectura, aprenderás los conceptos básicos de la importación de datos en R. Viene con conjuntos de datos integrados que son excelentes herramientas para aprender a usar R y practicar el análisis de datos. Explorarás la función **data()** y aprenderás a cargar conjuntos de datos de muestra en RStudio. A continuación, aprenderás a usar dos paquetes de tidyverse (readr y readxl) para importar archivos de otras fuentes en R. Aprenderás a utilizar readr para leer un archivo .csv, y a utilizar readxl para leer un archivo .xlsx.

# La función data()

La instalación predeterminada de R viene con un número de conjuntos de datos precargados con los que puedes practicar. Esta es una excelente manera de desarrollar tus habilidades en R y aprender algunas funciones importantes sobre el análisis de datos. Además, muchos recursos y tutoriales en línea utilizan estos conjuntos de datos de muestra para enseñar conceptos de codificación en R.

Puedes usar la función **data()** para cargar estos conjuntos de datos en R. Si ejecutas la función data sin un argumento, R mostrará una lista de los conjuntos de datos disponibles.

data()

Incluye la lista de conjuntos de datos precargados del paquete de conjuntos de datos "datasets".

Si deseas cargar un conjunto de datos específico, solo tienes que introducir su nombre en el paréntesis de la función **data()**. Por ejemplo, carguemos el conjunto de datos "mtcars", que contiene información sobre los automóviles que aparecieron en números anteriores de la revista *Motor Trend*.

data(mtcars)

Cuando ejecutes la función, R cargará el conjunto de datos. El conjunto de datos también aparecerá en el panel "entorno" de tu RStudio. El panel de entorno muestra los nombres de los objetos de datos, por ejemplo, marcos de datos y variables, que tienes en tu espacio de trabajo actual. En esta imagen, "mtcars" aparece en la quinta fila del panel. R nos dice que contiene 32 observaciones y 11 variables.

Ahora que el conjunto de datos está cargado, puedes obtener una vista previa del mismo en el panel de la consola de R. Solo tienes que escribir su nombre...

mtcars

...y luego presionar ctrl (o cmnd) y la tecla Enter.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear | carb |
| Mazda RX4 | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.620 | 16.46 | 0 | 1 | 4 | 4 |
| Mazda Rx4 Wag | 21.0 | 6 | 160.0 | 110 | 3.90 | 2.875 | 17.02 | 0 | 1 | 4 | 4 |
| Datsun 710 | 22.8 | 4 | 108.0 | 93 | 3.85 | 2.320 | 18.62 | 1 | 0 | 4 | 1 |
| Hornet 4 Drive | 21.4 | 6 | 258.0 | 110 | 3.08 | 3.215 | 19.44 | 1 | 0 | 3 | 1 |
| Hornet Sportabout | 18.7 | 8 | 360.0 | 175 | 3.15 | 3.440 | 17.02 | 0 | 0 | 3 | 2 |
| Valiant | 18.1 | 6 | 225.0 | 105 | 2.76 | 3.460 | 20.22 | 0 | 0 | 3 | 1 |
| Duster 360 | 14.3 | 8 | 360.0 | 245 | 3.21 | 3.570 | 15.84 | 0 | 0 | 3 | 4 |
| Merc 250D | 24.4 | 4 | 146.7 | 62 | 3.69 | 3.190 | 20.00 | 0 | 0 | 4 | 2 |
| Merc 230 | 22.8 | 4 | 140.8 | 95 | 3.92 | 3.150 | 22.90 | 0 | 0 | 4 | 2 |
| Merc 280 | 19.2 | 6 | 167.6 | 123 | 3.2 | 3.440 | 18.30 | 0 | 0 | 4 | 4 |

También puedes ver el conjunto de datos haciendo clic directamente sobre el nombre del conjunto de datos en el panel Entorno. Así que, si haces clic en **mtcars**

en el panel de entorno, R ejecuta automáticamente la función View() y muestra el conjunto de datos en el visor de datos de RStudio.

Intenta experimentar con otros conjuntos de datos de la lista si quieres practicar un poco más.

# readr

Además de usar los conjuntos de datos integrados de R, también es útil importar datos de otras fuentes para utilizarlos en la práctica o en el análisis. El paquete readr en R es una excelente herramienta para leer datos rectangulares. Los datos rectangulares son datos que encajan perfectamente en un rectángulo de filas y columnas, en el que cada columna se refiere a una única variable y cada fila a una única observación.

Estos son algunos ejemplos de tipos de archivos que almacenan datos rectangulares:

* **.csv (valores separados por comas):** Un archivo .csv es un archivo de texto sin formato que contiene una lista de datos. La mayoría de las veces usan comas para separar (o delimitar) los datos, pero a veces utilizan otros caracteres, como el punto y coma.
* **.tsv (valores separados por tabulaciones):** Un archivo .tsv almacena una tabla de datos en la que las columnas de datos están separadas por tabulaciones. Por ejemplo, una tabla de una base de datos o datos de una hoja de cálculo.
* **.fwf (archivos de ancho fijo):** Un archivo .fwf tiene un formato específico que permite guardar datos textuales de forma organizada.
* **.log:** Un archivo .log es un archivo generado por computadora que registra eventos de sistemas operativos y otros programas de software.

Base R también tiene funciones para leer archivos, pero las funciones equivalentes en readr suelen ser *mucho* más rápidas. Además, producen tibbles, que son fáciles de usar y leer.

El paquete readr forma parte del núcleo de tidyverse. Así que, si ya instalaste tidyverse, tienes lo que necesitas para empezar

a trabajar con readr. Si no lo has hecho, puedes instalar tidyverse ahora.

## Funciones readr

El objetivo de readr es proporcionar una forma rápida y amigable de leer datos rectangulares. La función readr soporta varias funciones read\_. Cada función se refiere a un formato específico de archivo.

* read\_csv() : Archivos separados por comas (CSV)
* read\_tsv() : Archivos separados por tabulaciones
* read\_delim(): Archivos delimitados en general
* read\_fwf(): Archivos de ancho fijo
* read\_table() : Archivos tabulares cuyas columnas están separadas por espacios en blanco
* read\_log(): Archivos de registro de la web

Todas estas funciones tienen una sintaxis similar, por lo que una vez que aprendes a usar una de ellas, podrás aplicar tus conocimientos a las demás. Esta lectura se centrará en la función read\_csv(), ya que los archivos .csv son una de las formas más comunes de almacenamiento de datos y trabajarás con ellos frecuentemente.

En la mayoría de los casos, estas funciones funcionarán de forma automática: provees la ruta de acceso a un archivo, ejecutas la función y obtienes un tibble que muestra los datos del archivo. Detrás de escena, readr analiza todo el archivo y especifica cómo debe convertirse cada columna de un vector de caracteres al tipo de datos más apropiado.

## Leer un archivo csv con readr

El paquete readr viene con algunos archivos de muestra de conjuntos de datos integrados que puedes usar para el código de ejemplo. Para enumerar los archivos de muestra, puedes ejecutar la función readr\_example() sin argumentos.

|  |  |  |
| --- | --- | --- |
| readr\_example() |  |  |
| [1] "challenge.csv" | "epa78.txt" | "example.log" |
| [4] "fwf-sample.txt" | "massey-rating.txt" "mtcars.csv" | |
| [7] "mtcars.csv.bz2" | "mtcars.csv.zip" |  |

El archivo "mtcars.csv" hace referencia al conjunto de datos "mtcars" mencionado anteriormente. Usemos la función **read\_csv()** para leer el archivo "mtcars.csv", como ejemplo. En el paréntesis, debes indicar la ruta del archivo. En este caso, es “readr\_example(“mtcars.csv”).

read\_csv(readr\_example("mtcars.csv"))

Cuando ejecutas la función, R imprime una especificación de columna que da el nombre y el tipo de cada columna.

── **Especificación de columna** ─────────────────────────────────────────────────── cols(

mpg = col\_double(),

cyl = col\_double(),

disp = col\_double(),

hp = col\_double(),

drat = col\_double(),

wt = col\_double(),

qsec = col\_double(),

vs = col\_double(),

am = col\_double(),

gear = col\_double(),

carb = col\_double()

)

R también imprime un tibble.

# A tibble: 32 x 11

mpg cyl disp hp drat wt qsec vs am gear carb

*<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>*

1 21 6 160 110 3.9 2.62 16.5 0 1 4 4

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 18.1 | 6 | 225 | 105 | 2.76 | 3.46 | 20.2 | 1 | 0 | 3 | 1 |
| 7 | 14.3 | 8 | 360 | 245 | 3.21 | 3.57 | 15.8 | 0 | 0 | 3 | 4 |
| 8 | 24.4 | 4 | 147. | 62 | 3.69 | 3.19 | 20 | 1 | 0 | 4 | 2 |
| 9 | 22.8 | 4 | 141. | 95 | 3.92 | 3.15 | 22.9 | 1 | 0 | 4 | 2 |
| 10 | 19.2 | 6 | 168. | 123 | 3.92 | 3.44 | 18.3 | 1 | 0 | 4 | 4 |

# … con 22 filas más

# readxl

Para importar datos de hojas de cálculo a R, puedes utilizar el paquete readxl. El paquete readxl facilita la transferencia de datos de Excel a R. Readxl admite tanto el formato de archivo .xls heredado como el moderno formato de archivo .xlsx basado en xml.

El paquete readxl forma parte de tidyverse, pero no es un paquete *núcleo* de tidyverse, por lo que es necesario cargar readxl en R mediante el uso de la función library().

library(readxl)

## Leer un archivo .csv con readxl

Igual que el paquete readr, el paquete readxl viene con algunos archivos de muestra de conjuntos de datos integrados que puedes usar para practicar. Puedes ejecutar el código readxl\_example() para ver la lista.

Puedes utilizar la función **read\_excel()** para leer un archivo de hoja de cálculo al igual que usaste la función read\_csv() para leer un archivo .csv. El código para leer el archivo de ejemplo "type-me.xlsx" incluye la ruta del archivo en el paréntesis de la función.

read\_excel(readxl\_example("type-me.xlsx"))

Puedes usar la función [excel\_sheets()](https://readxl.tidyverse.org/reference/excel_sheets.html) para listar los nombres de las hojas individuales.

excel\_sheets(readxl\_example("type-me.xlsx"))

[1] "logical\_coercion" "numeric\_coercion" "date\_coercion" "text\_coercion"

Puedes usar la función [excel\_sheets()](https://readxl.tidyverse.org/reference/excel_sheets.html) para listar los nombres de las hojas individuales.

excel\_sheets(readxl\_example("type-me.xlsx"))

1. "logical\_coercion" "numeric\_coercion" "date\_coercion" "text\_coercion"

También puedes especificar una hoja por nombre o número. Basta con escribir "sheet =" seguido del nombre o número de la hoja. Por ejemplo, puedes usar la hoja denominada "numeric\_coercion" de la lista anterior.

read\_excel(readxl\_example("type-me.xlsx"), sheet = "numeric\_coercion")

Cuando se ejecuta la función, R devuelve un tibble de la hoja.

# A tibble: 7 x 2

`maybe numeric?` explanation

|  |  |
| --- | --- |
| <chr> | <chr> |
| 1 NA | "empty" |
| 2 TRUE | "boolean true" |
| 3 FALSE | "boolean false" |
| 4 40534 | "datetime" |
| 5 123456 | "the string \"123456\"" |
| 6 123456 | "the number 123456" |
| 7 cabbage | "\"cabbage\"" |

## Recursos

* + Si quieres aprender a usar las funciones de readr para trabajar con archivos más complejos, consulta el capítulo ["Importación de datos" en R](https://r4ds.had.co.nz/data-import.html)  [en Ciencia de análisis de datos](https://r4ds.had.co.nz/data-import.html). Explora algunos de los problemas comunes que puedes encontrar al leer archivos, y cómo usar readr para solucionarlos.
  + La entrada de [readxl](https://readxl.tidyverse.org/) en la documentación de tidyverse ofrece una buena descripción general de las funciones básicas de readxl, proporciona una explicación detallada de cómo funciona el paquete y los conceptos de codificación ocultos, y ofrece enlaces a otros recursos útiles.
  + El paquete "datasets" de R contiene muchos conjuntos de datos útiles precargados. Consulta la lista de conjuntos de datos en el sitio web del paquete [R Datasets.](https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html) La lista incluye enlaces a descripciones detalladas de cada conjunto de datos.