

Guatemala 31 de marzo del 2023.

Investigue cómo leer un archivo CSV desde R.

¿Cómo leer un CSV en R?

En esta sección aprenderás a importar datos CSV en R o RStudio con las funciones `read.csv` y `read.csv2`. Puedes ver la sintaxis básica de las funciones con los argumentos más comunes en el siguiente bloque de código. Para obtener detalles adicionales, recuerda escribir? `read.csv` o ? `read.csv2`.

Sintaxis básica



```
# Por defecto coma (,) como separador y punto (.) como separador decimal
read.csv(file,                # Nombre del archivo o ruta completa del archivo
  header = TRUE,              # Leer el encabezado (TRUE) o no (FALSE)
  sep = ",",                  # Separador de los valores
  quote = "\"",               # Caracter de citaciones
  dec = ".",                  # Punto decimal
  fill = TRUE,                # Rellenar celdas vacías (TRUE) o no (FALSE)
  comment.char = "",          # Carácter de los comentarios o cadenas vacías
  encoding = "unknown",       # Codificación del archivo
  ...)                        # Argumentos adicionales

# Por defecto punto y coma (;) como separador y coma (,) como separador decimal
read.csv2(file, header = TRUE, sep = ";", quote = "\"", dec = ",",
  fill = TRUE, comment.char = "", encoding = "unknown", ...)
```

Es posible que hayas notado que la única diferencia entre las funciones es el separador de valores y el separador decimal, debido a que en algunos países usan comas como separador decimal.

La siguiente tabla resume los tres argumentos predeterminados principales:

Función	Header	Sep	Dec
<code>read.csv</code>	TRUE	","	."
<code>read.csv2</code>	TRUE	";"	","

Para cargar un archivo CSV en R con los argumentos predeterminados puedes pasar el archivo como cadena de caracteres a la función correspondiente. La salida será de clase `data.frame`.

```
read.csv("mi_archivo.csv")
```

Si tan solo ejecutas el código anterior imprimirás el data frame, pero no se almacenará en memoria, ya que no lo estás asignado a ninguna variable. Si lo guardas, por ejemplo, en un variable llamada `mi_archivo`, podrás acceder a las variables o a los datos que quieras.

```
mi_archivo <- read.csv("mi_archivo.csv")
```

Preparación de datos con dplyr

Introducción

`dplyr` es un paquete de R que permite realizar la preparación necesaria de los datos para su posterior análisis estadístico. Contiene una serie de funciones bastante sencillas a la lectura que gramaticalmente tienen bastantes similitudes al lenguaje SQL, proveyendo una serie de verbos que ayudan a preparar los datos para su análisis.

Instalación

Para instalar y cargar la librería al ambiente de trabajo, solo es necesario utilizar las siguientes funciones:

```
install.packages('dplyr')  
library(dplyr)
```

Primero, para leer y cargar los datos de un archivo con formato `.csv` hacemos uso del paquete `readr`.

```
iris <- readr::read_csv('data/iris.csv')
```

Esto almacena en nuestro ambiente una variable llamada "iris" que tendrá una estructura de data frame que es un arreglo que emula el comportamiento de una matriz de datos en el que las variables se representan por columnas y las observaciones por filas. Podemos echar un vistazo a los datos al ejecutar en la consola el nombre del data frame.

```
iris
```

```
## # A tibble: 150 x 5  
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
##         <dbl>         <dbl>         <dbl>         <dbl> <chr>  
## 1         5.1         3.5         1.4         0.2 setosa  
## 2         4.9         3         1.4         0.2 setosa  
## 3         4.7         3.2         1.3         0.2 setosa  
## 4         4.6         3.1         1.5         0.2 setosa  
## 5         5         3.6         1.4         0.2 setosa  
## 6         5.4         3.9         1.7         0.4 setosa  
## 7         4.6         3.4         1.4         0.3 setosa  
## 8         5         3.4         1.5         0.2 setosa  
## 9         4.4         2.9         1.4         0.2 setosa  
## 10        4.9         3.1         1.5         0.1 setosa  
## # ... with 140 more rows
```

Verbos principales

Los verbos principales de dplyr permiten manejar las estructuras de datos con una sintaxis fácilmente comprensible que busca asemejarse a la descripción del procesamiento de los datos antes de convertirse en macrodatos. Los verbos o funciones más importantes son los siguientes:

Función select():

Permite seleccionar o dejar seleccionar columnas específicas de acuerdo a sus nombres en el orden en el que sean expresados los argumentos de la función inmediatamente siguientes al data frame objetivo que es expresado como primer argumento. Podría establecerse una equivalencia en SQL con la sentencia SELECT.

```
library(dplyr)
select(iris, Species, Sepal.Length)
```

```
## # A tibble: 150 x 2
##   Species Sepal.Length
##   <chr>      <dbl>
## 1 setosa      5.1
## 2 setosa      4.9
## 3 setosa      4.7
## 4 setosa      4.6
## 5 setosa      5
## 6 setosa      5.4
## 7 setosa      4.6
## 8 setosa      5
## 9 setosa      4.4
## 10 setosa     4.9
## # ... with 140 more rows
```

También, es posible seleccionar todas las columnas excepto las que sean especificadas agregando un - en las variables que se desean excluir. Por ejemplo:

```
library(dplyr)
select(iris, -Species, -Sepal.Length)
```

```
## # A tibble: 150 x 3
##   Sepal.Width Petal.Length Petal.Width
##   <dbl>      <dbl>      <dbl>
## 1 3.5        1.4        0.2
## 2 3          1.4        0.2
## 3 3.2        1.3        0.2
## 4 3.1        1.5        0.2
## 5 3.6        1.4        0.2
## 6 3.9        1.7        0.4
## 7 3.4        1.4        0.3
## 8 3.4        1.5        0.2
## 9 2.9        1.4        0.2
## 10 3.1        1.5        0.1
## # ... with 140 more rows
```

BIBLIOGRAFIA:

- Leer CSV en R. <https://r-coder.com/leer-csv-r/>
- Preparación de datos con dplyr. Carlos Figueroa. Junio 2020. <https://rpubs.com/carligueroa/633581>