



# Carga de datos

En términos prácticos, generalmente encontraremos la información que deseamos analizar mediante algoritmos y almacenada en diferentes tipos de archivos. Para poder utilizar esta información en nuestros procesos, será necesario importarla desde R. Asimismo, es posible que la información esté ubicada en la web, en el portapapeles o en otros recursos similares.



## Importar, exportar, guardar y lectura de datos

### Datos en formato CSV

El formato CSV, conocido como Valores Separados por Comas (*Comma Separated Values*) es ampliamente utilizado para intercambiar datos entre distintas aplicaciones. De acuerdo con Charte (2014), este tipo de archivo es un archivo de texto en el cual cada fila representa una muestra o evento, y cada columna representa una variable. Los datos en cada fila se separan entre sí mediante comas, lo que le da su nombre al formato.

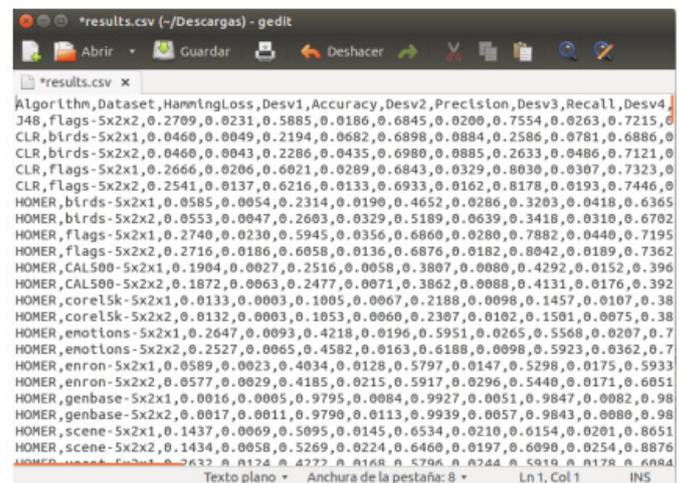
En un archivo CSV, es posible que haya una línea inicial opcional que sirva como encabezado donde se indiquen los nombres de cada columna de datos. Si el separador decimal de los números en el archivo es una coma

en lugar de un punto, los datos de cada fila se separan utilizando puntos y comas. Por otro lado, los datos no numéricos, como cadenas de caracteres, pueden o no estar delimitados por comillas.

En resumen, aunque el formato CSV puede tener cierta variabilidad, su estructura básica siempre se mantiene. Debido a esto, R proporciona varias funciones diferentes para trabajar con este tipo de archivos.

### Lectura de archivos CSV

En la mayoría de los casos, será necesario importar datos almacenados en formato CSV, lo que nos dará como resultado un data frame en R. Para lograr esto, podemos utilizar funciones como `read.table()`, `read.csv()` y `read.csv2()`. Estas dos últimas son versiones especializadas de la primera, en las que se asigna un valor predeterminado específico a algunos de los parámetros.



La figura anterior muestra una vista parcial de un archivo CSV importado a R.

## Sintaxis de la sentencia

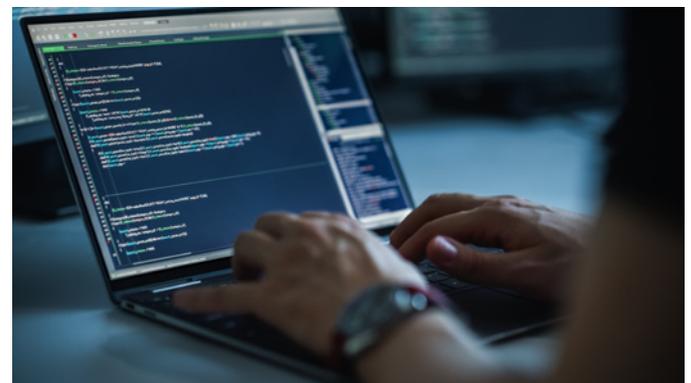
```
read.table(file = archivo[, header = TRUE|FALSE,  
sep = separadorDatos, dec = separadorDecimal,  
quote = delimitadorCadenas, stringsAsFactors = TRUE|FALSE])
```

La función genérica antes presentada permite leer datos en formato CSV. El único argumento requerido es "file", que se utiliza para proporcionar el nombre del archivo que se desea leer. Este nombre debe incluir la extensión del archivo y, si es necesario, la ruta completa del archivo. Según Charte (2014), los demás parámetros tienen los siguientes propósitos:

- **header:** De tipo lógico e indica a la función si el archivo contiene una cabecera o no.
- **Sep:** Establece cuál es el separador de datos en cada fila.
- **dec:** Para datos numéricos, establece cuál es el separador entre parte entera y decimal.
- **quote:** Para datos alfanuméricos, determina el carácter que se usa como delimitador.
- **stringsAsFactors:** Tiene la misma finalidad que en la función `data.frame()`.
- **read.csv y read.csv2 (archivo):** Es una implementación especializada de `read.table()` en la que se asume que los parámetros `header`, `sep` y `dec` toman los valores `TRUE`, `"` y `."`, respectivamente. Acepta los mismos parámetros que `read.table()`.

## Importar datos desde Excel

Microsoft Excel es una de las aplicaciones más eficaces y eficientes para analizar datos y diseñar gráficos. Por tanto, es común que necesitemos importar información de una hoja de cálculo de Excel a R. Para realizar esta tarea, existen varios paquetes disponibles en CRAN. En esta sección, se explicará cómo utilizar uno de ellos.



### XLConnect

Dentro de este paquete, se encuentran varias funciones que tienen la capacidad de leer el contenido de un archivo de Excel y crear un data frame utilizando solo una parte o todo el contenido de una de sus hojas. Una de estas funciones es `readWorksheetFromFile()`:

```
readWorksheetFromFile(archivo, sheet = hojaALeer[,  
header = TRUE|FALSE, region = rango])
```

Estas líneas de código abren el archivo Excel especificado en el primer parámetro, recuperan el contenido de la hoja indicada por "sheet" y generan un data frame con esa información. Opcionalmente, se puede indicar si los datos tienen encabezado o no

mediante el parámetro "header". Además, se puede especificar un rango de celdas a leer utilizando el parámetro "region", el cual debe ser una cadena con un rango en formato "B12:D18".

El paquete XLConnect permite trabajar con archivos de Excel tanto en el formato antiguo (.xls) como en el nuevo formato (.xlsx).

Es fundamental obtener información precisa y de fuentes confiables, pues de lo contrario, no será útil para tomar decisiones. Como analistas de datos, es de gran importancia preservar el valor de los datos a lo largo del tiempo.





# Referencias bibliográficas

- Charte, F. (2014). *Análisis exploratorio y visualización de los datos con R*. Recuperado de <http://www.fcharte.com/libros/ExploraVisualizaConR-Fcharte.pdf>