

# DATA FRAME

Ramon Ceballos

22/1/2021

## 1. Cargar o importar Datos (ficheros) con `read.table()`

Para cargar data frames en R tenemos diversas funciones. Ahora nos vamos a centrar en:

- `read.table()`: para definir un data frame a partir de una tabla de datos contenida en un fichero
  - Este fichero puede estar guardado en nuestro ordenador o bien podemos conocer su url. Sea cual sea el caso, se aplica la función al nombre del fichero o a la dirección entre comillas. El fichero debe ser simple (tabla de datos).

Lo suyo para trabajar con datos es organizar en dos subcarpetas diferentes los script y los datos que empleemos.

La función `read.table()` lleva asociada una serie de parámetros:

- `header = TRUE`: para indicar si la tabla que importamos tiene una primera fila con los nombres de las columnas. El valor por defecto es `FALSE`
- `col.names = c(...)`: para especificar el nombre de las columnas. No olvidéis que cada nombre debe ir entre comillas
- `sep`: para especificar las separaciones entre columnas en el fichero (si no es un espacio en blanco). Si es así, hay que introducir el parámetro pertinente entre comillas
- `dec`: para especificar el signo que separa la parte entera de la decimal (si no es un punto). Si es así, hay que introducir el parámetro pertinente entre comillas. Por defecto es un punto

En ocasiones los datos no se cargan en el formato adecuado. Para ello utilizamos parámetros adicionales de la función `read.table()`.

- `stringsAsFactors`: para prohibir la transformación de las columnas de palabras en factores debemos usar `stringsAsFactors=FALSE` (ya que por defecto, R realiza dicha transformación). Cuidado con esto.
- Para importar un fichero de una página web segura (cuyo url empiece con `https`), no podemos entrar directamente la dirección en `read.table()`; una solución es instalar y cargar el paquete `Rcurl` y entonces usar la instrucción `read.table (textConnection(getURL("url ")),...)`.

### 1.1. Carga DT Local

Se utiliza: `read.table("path_fichero")`.

## 1.2. Carga DT a través de URL

Se utiliza la url donde esté el fichero en la web: `read.table("url")`.

## 2. Cargar datos con otros comandos

Se pueden leer diferentes tipos de ficheros empleando las siguientes instrucciones.

- `read.csv()`: para importar ficheros en formato CSV. Formato más usado
- `read.xls()` o `read.xlsx()`: para importar hojas de cálculo tipo Excel u OpenOffice en formato XLS o XLSX, respectivamente. Se necesita el paquete `xlsx` (porque tiene Java detrás)
- `read.mtb()`: para importar tablas de datos Minitab. Se necesita el paquete `foreign`
- `read.spss()`: para importar tablas de datos SPSS. Se necesita el paquete `foreign`

Para buscar más funciones `read` escribe lo siguiente: `help.search("read")`

## 2. Guardar o exportar un fichero con `write.table()`

- `write.table(df, file = "")`: para exportar un data frame a un fichero
  - `file = ""`: es donde indicaremos el nombre que queremos darle al fichero
  - Podemos usar el parámetro `sep` para indicar el símbolo de separación de columnas. Siempre entre comillas
  - También podemos utilizar el parámetro `dec` para indicar la separación entre la parte entera y decimal de los datos

```
df_iris = iris
head(df_iris,3)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1         5.1         3.5         1.4         0.2   setosa
## 2         4.9         3.0         1.4         0.2   setosa
## 3         4.7         3.2         1.3         0.2   setosa
```

```
#Ejemplo si tuviésemos un data frame definido como df_iris
write.table(df_iris, file = ".././../data/iris2.txt", dec = ".")

df_iris2 = read.table(".././../data/iris2.txt", header = TRUE, dec = ".")
head(df_iris2,3)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1         5.1         3.5         1.4         0.2   setosa
## 2         4.9         3.0         1.4         0.2   setosa
## 3         4.7         3.2         1.3         0.2   setosa
```