

[000Z] DEFAD: Introducción a R y Rstudio

Lectura conjuntos de datos

000R Team

2015–16

1 read.table()

```
read.table()
```

Leer datos de ejemplo PIB

```
df <- read.table(  
  "http://ares.inf.um.es/00Rteam/datos/pibCcAaEj.dat",  
  sep=";")  
head( df )
```

##	ciudad	actividad	anho	valor
## 1	Andaluc	Agric	2008	6467.357
## 2	Andaluc	Const	2008	21477.597
## 3	Andaluc	Host	2008	10076.699
## 4	Arag	Agric	2008	1197.806
## 5	Arag	Const	2008	4678.884
## 6	Arag	Host	2008	1905.278

Seleccionar subconjuntos de un dataframe

Forma general

$$\text{nuevo_df} \leftarrow \text{df}[\text{índicesFilas}, \text{índicesColumnas}]$$

- **Ojo:** la regla es “*filas por columnas*”.

Seleccionar variables (columnas. . .)

$$\text{nuevo_df} \leftarrow \text{df}[\text{ , } \text{índices}]$$

Observar que dejamos en blanco la primera posición entre los corchetes para referirnos a la columnas

Acceder a las variables de un data frame

```
options( width = 80)
df[ , 4 ]
```

```
## [1] 6467.357 21477.597 10076.699 1197.806 4678.884 1905.278 2278.151
## [8] 7901.498 2710.660 224.709 20320.981 10418.601 1377.749 4110.927
## [15] 1505.280 6025.496 19223.889 1005.749 1188.230 4447.108 1919.271
## [22] 2244.530 7507.649 2927.641 195.667 19171.700 10861.179 1233.257
## [29] 3526.090 1489.960
```

```
df$valor
```

```
## [1] 6467.357 21477.597 10076.699 1197.806 4678.884 1905.278 2278.151
## [8] 7901.498 2710.660 224.709 20320.981 10418.601 1377.749 4110.927
## [15] 1505.280 6025.496 19223.889 1005.749 1188.230 4447.108 1919.271
## [22] 2244.530 7507.649 2927.641 195.667 19171.700 10861.179 1233.257
## [29] 3526.090 1489.960
```

```
df [, "valor" ]
```

```
## [1] 6467.357 21477.597 10076.699 1197.806 4678.884 1905.278 2278.151
## [8] 7901.498 2710.660 224.709 20320.981 10418.601 1377.749 4110.927
## [15] 1505.280 6025.496 19223.889 1005.749 1188.230 4447.108 1919.271
## [22] 2244.530 7507.649 2927.641 195.667 19171.700 10861.179 1233.257
## [29] 3526.090 1489.960
```


Acceder a las variables de un data frame

```
df[ 4 ]
```

```
##      valor
## 1  6467.357
## 2 21477.597
## 3  10076.699
## 4   1197.806
## 5   4678.884
## 6   1905.278
## 7   2278.151
## 8   7901.498
## 9   2710.660
## 10  224.709
## 11 20320.981
## 12 10418.601
## 13  1377.749
## 14  4110.927
## 15  1505.280
## 16  6025.496
## 17 19223.889
## 18  1005.749
## 19  1188.230
## 20  4447.108
## 21  1919.271
## 22  2244.530
## 23  7507.649
## 24  2927.641
## 25   195.667
## 26 19171.700
## 27 10861.179
## 28  1233.257
```

Ejercicio

- Selecciona el valor del PIB para Andalucía
- Selecciona el valor del PIB en Andalucía correspondiente a la agricultura
- Haz un boxplot para valor del PIB en general (utiliza la ayuda y google para saber como hacer un boxplot)
- Haz un boxplot para valor del PIB para el año 2008 y otro para el año 2009
- Intenta hacer lo anterior (dos boxplots) en una única línea de código (y la misma pantalla)

read.table()

La función más importante para *importar* datos: `read.table()` - automáticamente convierte los datos en un *dataframe*.

```
df <- read.table(file,  
                  header = valor_logico,  
                  sep = "delimitador",  
                  dec = "signo del decimal")
```

- header TRUE/FALSE según la primera fila del fichero

- sep es el delimitador: el separador de campos
 - sep = ";", sep=".", sep=",", sep="\t" (tabulación)...
- dec es el indicador del signo decimal
 - dec=".", dec=";"...

Ejercicio

Prueba a leer el fichero que está en la nube pero con distintas configuraciones de los argumentos de la función `read.table()`, ¿qué sucede?

```
df <- read.table(  
  "http://ares.inf.um.es/00Rteam/datos/pibCcAaEj.dat",  
  sep="\t", dec=",", header= F)  
head( df )
```

Factores

Podemos tener un factor codificado, por ejemplo sexo, donde 1 es '*Masculino*' y 2 '*Femenino*'.

Creamos un data frame de atributos sexo y peso y le asignamos a sexo el valor que tiene (es una variable CUALITATIVA, no cuantitativa)

```
sexo <- c( 1, 1, 1, 1, 2, 2, 2 )  
peso <- c(60,65,70,66,80,60,76)  
df <- data.frame( sexo, peso )  
df$sexo <- factor( df$sexo )
```

Factores: etiquetas de valores

Creamos las etiquetas

```
df$sexo <- factor( df$sexo,  
                    levels = c(1, 2),  
                    labels = c("masculino", "femenino")  
                    )
```

¿Hay algún *factor* en nuestro conjunto de datos?

```
df <- read.table(  
  "http://ares.inf.um.es/00Rteam/datos/pibCcAaEj.dat",  
  sep=";")  
head( df )
```

##	ciudad	actividad	anho	valor
## 1	Andaluc	Agric	2008	6467.357
## 2	Andaluc	Const	2008	21477.597
## 3	Andaluc	Host	2008	10076.699
## 4	Arag	Agric	2008	1197.806
## 5	Arag	Const	2008	4678.884
## 6	Arag	Host	2008	1905.278

Variables cualitativas: dosis de una sustancia...

read.table() ii

```
df <- read.table(  
  "http://ares.inf.um.es/00Rteam/datos/pibCcAaEj.dat",  
  sep = ";", dec = ".", head = T,  
  stringsAsFactors = T)
```

- variables de tipo carácter se convierten a factores
 - stringsAsFactor = FALSE
- help(read.table)
 - na.strings = "NA"
 - dec = "."

read.table() iii

Lo hacemos juntos:

Guarda en tu working directory el .xls (fichero de excell) que tienes en recursos `pibCcAaEj.xls`. A continuación, conviértelo en csv especificando los campos para poder leerlo con R.

Algunas funciones útiles

Table 1: Algunas funciones

Función	Acción
<code>length(obj)</code>	Número de componentes, elementos
<code>dim(obj)</code>	Dimensión de un objeto
<code>str(obj)</code>	Estructura de un objeto
<code>class(obj)</code>	Clase (class) o tipo de objeto
<code>names(obj)</code>	Nombres de los componentes de un objeto
<code>c(obj,obj,...)</code>	Combina objetos en un vector
<code>head(obj)</code>	Lista la primera parte de un objeto
<code>tail(obj)</code>	Lista la última parte (cola) de un objeto
<code>ls()</code>	Lista los objetos actuales
<code>rm(obj)</code>	Borra un objeto
<code>newobj <- edit(obj)</code>	Edita un objeto y lo guarda
<code>fix(obj)</code>	Edita sobre un objeto ya creado