Leyendo Data Frame

Daniel

07-10-2019

# Leyendo tablas de datos

* **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 pc o podemos conocer su url.
* [Lista de dataset](https://maitra.public.iastate.edu/stat501/datasets.html)

# Carga de ficheros local

## Parámetros de read.table()

* **header = TRUE**: indicar si la tabla que importamos tiene una primera fila con los nombres de las columnas. El valor por defectos es FALSE.
* **col.names = c(..)**: para especificar el nombre de las columnas.
* **sep**: para especificar separaciones entre columnas en el fichero. Si es así hay que introducir el parámetro pertinente entre comillas.
* **dec**: para especificar el signo que separa la parte entera de la decimal
* **stringASFactors**: para prohibir la transformación de las columnas de palabras en factores debemos usar *stringsAsFactors=FALSE* (por defector R realiza la trasnformación)
* Para importar un fichero de una página web segura (URL comienza 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”)),…)*

Breed (1/5/8) Sale Price Yearling height at shoulder (in.) Fat Free Body (lbs.), Percent Fat-free body, Frame – scale from 1 (small) to 8 (large), Back fat (in.), sale height at shoulder (in.) sale weight (lbs.) of three breeds of bulls,

df = read.table("../data/bulls.dat",   
 header = FALSE,   
 col.names = c("breed","sale\_price", "shoulder",  
 "fat\_free", "percent\_ff", "frame\_scale",   
 "back\_fat", "sale\_height", "sale\_weight"),  
 sep= "", dec = ".")  
head(df)

## breed sale\_price shoulder fat\_free percent\_ff frame\_scale back\_fat  
## 1 1 2200 51.0 1128 70.9 7 0.25  
## 2 1 2250 51.9 1108 72.1 7 0.25  
## 3 1 1625 49.9 1011 71.6 6 0.15  
## 4 1 4600 53.1 993 68.9 8 0.35  
## 5 1 2150 51.2 996 68.6 7 0.25  
## 6 1 1225 49.2 985 71.4 6 0.15  
## sale\_height sale\_weight  
## 1 54.8 1720  
## 2 55.3 1575  
## 3 53.1 1410  
## 4 56.4 1595  
## 5 55.0 1488  
## 6 51.4 1500

# Carga desde URL

df2 = read.table("https://maitra.public.iastate.edu/stat501/datasets/bulls.dat",   
 header = FALSE,   
 col.names = c("breed","sale\_price", "shoulder",  
 "fat\_free", "percent\_ff", "frame\_scale",   
 "back\_fat", "sale\_height", "sale\_weight"),  
 sep= "", dec = ".")  
head(df2)

## breed sale\_price shoulder fat\_free percent\_ff frame\_scale back\_fat  
## 1 1 2200 51.0 1128 70.9 7 0.25  
## 2 1 2250 51.9 1108 72.1 7 0.25  
## 3 1 1625 49.9 1011 71.6 6 0.15  
## 4 1 4600 53.1 993 68.9 8 0.35  
## 5 1 2150 51.2 996 68.6 7 0.25  
## 6 1 1225 49.2 985 71.4 6 0.15  
## sale\_height sale\_weight  
## 1 54.8 1720  
## 2 55.3 1575  
## 3 53.1 1410  
## 4 56.4 1595  
## 5 55.0 1488  
## 6 51.4 1500

str(df2)

## 'data.frame': 76 obs. of 9 variables:  
## $ breed : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ sale\_price : int 2200 2250 1625 4600 2150 1225 2250 4000 1600 1525 ...  
## $ shoulder : num 51 51.9 49.9 53.1 51.2 49.2 51 51.5 50.1 49.6 ...  
## $ fat\_free : int 1128 1108 1011 993 996 985 959 1060 979 1083 ...  
## $ percent\_ff : num 70.9 72.1 71.6 68.9 68.6 71.4 72.1 69.3 71.2 75.8 ...  
## $ frame\_scale: int 7 7 6 8 7 6 7 7 6 6 ...  
## $ back\_fat : num 0.25 0.25 0.15 0.35 0.25 0.15 0.2 0.3 0.25 0.3 ...  
## $ sale\_height: num 54.8 55.3 53.1 56.4 55 51.4 54 55.6 51.5 54.6 ...  
## $ sale\_weight: int 1720 1575 1410 1595 1488 1500 1522 1765 1365 1640 ...

df3 = read.table("https://maitra.public.iastate.edu/stat501/datasets/olive.dat",  
 stringsAsFactors = FALSE, header =TRUE)  
str(df3)

## 'data.frame': 572 obs. of 9 variables:  
## $ group.id: int 1 1 1 1 1 1 1 1 1 1 ...  
## $ X1 : int 1075 1088 911 966 1051 911 922 1100 1082 1037 ...  
## $ X2 : int 75 73 54 57 67 49 66 61 60 55 ...  
## $ X3 : int 226 224 246 240 259 268 264 235 239 213 ...  
## $ X4 : int 7823 7709 8113 7952 7771 7924 7990 7728 7745 7944 ...  
## $ X5 : int 672 781 549 619 672 678 618 734 709 633 ...  
## $ X6 : int 36 31 31 50 50 51 49 39 46 26 ...  
## $ X7 : int 60 61 63 78 80 70 56 64 83 52 ...  
## $ X8 : int 29 29 29 35 46 44 29 35 33 30 ...

# Leyendo datos de diferentes tipos de ficheros

* **read.csv()**: para importar ficheros en formato CSV
* **read.xls() o read.xlsx()** para importar hojas de cálculo de Excel u OpenOffice en formato XLS o XLSX., respectivamente. Se necesita el paquete xlsx
* **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

# Exportando datos a ficheros

* **write.table(dfm file = "")**: exporta un data frame a un fichero.
* **file = ""**: es donde indicaresmos el nombre que queremos darle al fichero
* Podemos usar el parámetro **sep** para indicar el simbolo de separación de columnas. Siempre entre comillas.
* También podemos utilizar el parámetro **dec** para indicar la separación entre parte entera y decimal.

write.table(df3, file = "../data/olive.txt", dec = ".")  
df4 = read.table("../data/olive.txt", header = TRUE, dec =".")  
head(df4)

## group.id X1 X2 X3 X4 X5 X6 X7 X8  
## 1 1 1075 75 226 7823 672 36 60 29  
## 2 1 1088 73 224 7709 781 31 61 29  
## 3 1 911 54 246 8113 549 31 63 29  
## 4 1 966 57 240 7952 619 50 78 35  
## 5 1 1051 67 259 7771 672 50 80 46  
## 6 1 911 49 268 7924 678 51 70 44

# Construyendo data frames

* **data.frame(vector\_1,…,vector\_n)**: para construir un data frama a partir de vectores introducidos en el orden que queremos disponer las columnas de la tabla.
* R considera del mismo tipo de datos todas las entradas de una columna de un data frame.
* Las variables tomarán nombres de los vectores. Estos nombres se pueden especificar en el argumento de *data.frame* entrando una construccion de la forma *nombre\_variable = vector*.
* **rownames**: para especificar los identificadores de las filas.
* También en esta función podemos hacer uso del parámetro, *stringAsFactors* para evitar la transformación de las columnas de tipo de palabras en factores.

gender = c("H", "M", "M", "M", "H")  
age = c(23, 45, 20, 30, 18)  
family = c(2, 3, 4, 2, 5)  
  
df5 = data.frame(genero = gender, edad = age, familia = family,   
 stringsAsFactors = TRUE)  
row.names(df5) = c("P1", "P2", "P3", "P4", "P5")  
df5

## genero edad familia  
## P1 H 23 2  
## P2 M 45 3  
## P3 M 20 4  
## P4 M 30 2  
## P5 H 18 5

str(df5)

## 'data.frame': 5 obs. of 3 variables:  
## $ genero : Factor w/ 2 levels "H","M": 1 2 2 2 1  
## $ edad : num 23 45 20 30 18  
## $ familia: num 2 3 4 2 5