Carga de Data Frames

Oscar Gerardo Hernández Martínez

14/8/2019

Carga de ficheros local

```
df = read.table("../../Cursos programación/Curso de R básico/data/bulls.dat",
                header = FALSE,
                col.names = c("breed", "sale_price",
                              "shoulder", "fat_free",
                              "percent_ff", "frame_scale",
                              "back_fat", "sale_height",
                              "scale_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 scale_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
```

Cuando colocamos las comillas, R automáticamente puede autocompletar la dirección dentro del ordenador si presionamos la tecla Tab. Si precisamos "subir" un nivel dentro de la estructura del ordenador, colocamos *"../"*

```
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",
                              "scale_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 scale_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 ...
##  $ scale_weight: int  1720 1575 1410 1595 1488 1500 1522 1765 1365 1640 ...
```

Es recomendable terminar la carga con la innstrucción **head()** y la instruncción **str()** para que se pueda apreciar si los datos se cargaron de forma correcta o ha habido algún problema con la carga de los mismos.

Factores en un Data Frame

```
df3 = read.table("https://maitra.public.iastate.edu/stat501/datasets/olive.dat")
str(df3)
```

```
## 'data.frame':    573 obs. of  9 variables:
##  $ V1: Factor w/ 10 levels "1","2","3","4",..: 10 1 1 1 1 1 1 1 1 1 ...
##  $ V2: Factor w/ 310 levels "1010","1020",..: 310 24 32 294 303 12 294 297 40 27 ...
##  $ V3: Factor w/ 176 levels "100","101","102",..: 176 152 150 136 138 145 132 144 142 141 ...
##  $ V4: Factor w/ 139 levels "152","156","158",..: 139 59 57 79 73 92 99 96 68 72 ...
##  $ V5: Factor w/ 390 levels "6300","6367",..: 390 351 327 388 373 341 366 384 332 336 ...
##  $ V6: Factor w/ 345 levels "1000","1002",..: 345 230 265 203 219 230 231 218 248 242 ...
##  $ V7: Factor w/ 46 levels "0","10","15",..: 46 21 16 16 35 35 36 34 24 31 ...
##  $ V8: Factor w/ 77 levels "0","10","100",..: 77 38 39 41 56 58 48 34 42 61 ...
##  $ V9: Factor w/ 45 levels "1","10","11",..: 45 22 22 22 29 40 38 22 29 27 ...
```

En este ejemplo, todo se ha cargado como factores, por lo que, será necesario hacer la pertinente modificación con el comando **stringsAsFactors**

```
df4 = read.table("https://maitra.public.iastate.edu/stat501/datasets/olive.dat",
                 stringsAsFactors = FALSE)
str(df4)
```

```
## 'data.frame':    573 obs. of  9 variables:
##  $ V1: chr  "group.id" "1" "1" "1" ...
##  $ V2: chr  "X1" "1075" "1088" "911" ...
##  $ V3: chr  "X2" "75" "73" "54" ...
##  $ V4: chr  "X3" "226" "224" "246" ...
##  $ V5: chr  "X4" "7823" "7709" "8113" ...
##  $ V6: chr  "X5" "672" "781" "549" ...
##  $ V7: chr  "X6" "36" "31" "31" ...
```

```
##  $ V8: chr  "X7" "60" "61" "63" ...
##  $ V9: chr  "X8" "29" "29" "29" ...
```

Ahora, observamos que también convierte como parte de los datos el título de las columnas, por lo que, será entonces necesario hacer uso del comando **header**

```
df5 = read.table("https://maitra.public.iastate.edu/stat501/datasets/olive.dat",
                 stringsAsFactors = FALSE, header = TRUE)
str(df5)
```

```
## '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 ...
```

Guardar un Data Frame

```
write.table(df5, file = "olive.txt", dec=".")
df6 = read.table(file = "data/olive.txt", header = TRUE, dec=".")
head(df6)
```

```
##   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
```

#Crear un Data Frame

```
gender = c("H", "M", "M", "M", "H")
age = c(23, 45, 20, 30, 18)
family = c(2, 3, 4, 2, 5)
df7 = data.frame(genero = gender, edad = age, familia = family,
                 stringsAsFactors = TRUE)
row.names(df7) = c("P1", "P2", "P3", "P4", "P5")
df7
```

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

```
str(df7)
```

```
## '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
```

```
dimnames(df7) = list(c("Manny", "Paulina", "Victoria", "Aranza", "Govea")
                     , c("Sexo", "Edad", "MiembrosFam")
)
df7
```

```
##          Sexo Edad MiembrosFam
## Manny       H   23           2
## Paulina     M   45           3
## Victoria    M   20           4
## Aranza      M   30           2
## Govea       H   18           5
```

```
df7 = rbind(df7, c("H", 30, 1))
df7
```

```
##          Sexo Edad MiembrosFam
## Manny       H   23           2
## Paulina     M   45           3
## Victoria    M   20           4
## Aranza      M   30           2
## Govea       H   18           5
## 6           H   30           1
```

```
df7$Sexo = as.character(df7$Sexo)
df7$Ingresos = c(10000, 12000, 15000, 20000, 25000, 12000)
row.names(df7) = c("Manny", "Paulina", "Victoria", "Aranza", "Govea", "Gerardo")
df7
```

```
##          Sexo Edad MiembrosFam Ingresos
## Manny       H   23           2    10000
## Paulina     M   45           3    12000
## Victoria    M   20           4    15000
## Aranza      M   30           2    20000
## Govea       H   18           5    25000
## Gerardo     H   30           1    12000
```

#Sub-Data Frames

```
gender = c("H", "M", "M", "M", "H")
age = c(23, 45, 20, 30, 18)
family = c(2, 3, 4, 2, 5)
df7 = data.frame(genero = gender, edad = age, familia = family,
                 stringsAsFactors = TRUE)
df7[df7$genero == "M", ] -> df_m
str(df_m)
```

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

```
df_m = droplevels(df_m)
str(df_m)
```

```
## 'data.frame':    3 obs. of  3 variables:
##  $ genero : Factor w/ 1 level "M": 1 1 1
##  $ edad   : num  45 20 30
```

```
##  $ familia: num  3 4 2
```

Observamos que este nuevo sub-data frame (df_m) hereda la estructura del data frame del cual fue obtenido (df7). Para modificar esta estructura, haremos uso del comando **droplevels()**

Tidyverse

```
library(tidyverse)
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
## -- Attaching packages ---------------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
iris_petal = select(iris, starts_with("Petal"))
head(iris_petal)
```

```
##   Petal.Length Petal.Width
## 1          1.4         0.2
## 2          1.4         0.2
## 3          1.3         0.2
## 4          1.5         0.2
## 5          1.4         0.2
## 6          1.7         0.4
```

```
iris_length = select(iris, ends_with("Length"))
head(iris_length)
```

```
##   Sepal.Length Petal.Length
## 1          5.1          1.4
## 2          4.9          1.4
## 3          4.7          1.3
## 4          4.6          1.5
## 5          5.0          1.4
## 6          5.4          1.7
```

Subset

```
subset(iris, Species == "setosa") -> setosa
head(setosa, 5)
```

```
##   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
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
```

```
str(setosa)
```

```
## 'data.frame':    50 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
setosa = droplevels(setosa)
str(setosa)
```

```
## 'data.frame':    50 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 1 level "setosa": 1 1 1 1 1 1 1 1 1 1 ...
```

```
subset(iris, Species == "versicolor") -> versicolor
head(versicolor, 5)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 51          7.0         3.2          4.7         1.4 versicolor
## 52          6.4         3.2          4.5         1.5 versicolor
## 53          6.9         3.1          4.9         1.5 versicolor
## 54          5.5         2.3          4.0         1.3 versicolor
## 55          6.5         2.8          4.6         1.5 versicolor
```

```
str(versicolor)
```

```
## 'data.frame':    50 obs. of  5 variables:
##  $ Sepal.Length: num  7 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 ...
##  $ Sepal.Width : num  3.2 3.2 3.1 2.3 2.8 2.8 3.3 2.4 2.9 2.7 ...
##  $ Petal.Length: num  4.7 4.5 4.9 4 4.6 4.5 4.7 3.3 4.6 3.9 ...
##  $ Petal.Width : num  1.4 1.5 1.5 1.3 1.5 1.3 1.6 1 1.3 1.4 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 2 2 2 2 2 2 2 2 2 2 ...
```

En este caso para modificar los nombres de las filas, haremos uso de la función **rownames**

```
subset(iris, Species == "versicolor") -> versicolor
rownames(versicolor) = 1:nrow(versicolor)
head(versicolor, 5)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1          7.0         3.2          4.7         1.4 versicolor
## 2          6.4         3.2          4.5         1.5 versicolor
## 3          6.9         3.1          4.9         1.5 versicolor
## 4          5.5         2.3          4.0         1.3 versicolor
## 5          6.5         2.8          4.6         1.5 versicolor
```

```
str(versicolor)
```

```
## 'data.frame':    50 obs. of  5 variables:
##  $ Sepal.Length: num  7 6.4 6.9 5.5 6.5 5.7 6.3 4.9 6.6 5.2 ...
```

```
##  $ Sepal.Width : num  3.2 3.2 3.1 2.3 2.8 2.8 3.3 2.4 2.9 2.7 ...
##  $ Petal.Length: num  4.7 4.5 4.9 4 4.6 4.5 4.7 3.3 4.6 3.9 ...
##  $ Petal.Width : num  1.4 1.5 1.5 1.3 1.5 1.3 1.6 1 1.3 1.4 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 2 2 2 2 2 2 2 2 2 2 ...
```