DATA FRAMES

Ramon Ceballos

22/1/2021

Data Frames

Hay sinónimos de Dta Frame como son Tablas de datos, Data sets, hojas de datos...

1. Introducción

Data frame. Un data frame es una tabla de doble entrada, formada por variables en las columnas y observaciones de estas variables en las filas, de manera que cada fila contiene los valores de las variables (recogidas en las columnas) para un mismo caso o un mismo individuo.

Las columnas pueden contener datos de naturaleza diferente.

- `data()`: para abrir una ventana con la lista de los objetos de datos a los que tenemos acceso en la sesión actual de R (los que lleva la instalación básica de R y los que aportan los paquetes que tengamos cargados. Para carta un paquete específico utiliza data(package = "nombre_paquete").
  - Si entramos `data(package=.packages(all.available = TRUE))` obtendremos la lista de todos los objetos de datos a los que tenemos acceso, incluyendo los de los paquetes que tengamos instalados, pero que no estén cargados en la sesión actual.

Cuando utilizamos un data set es importante documentarse acerca de él (de donde viene y demás). Usa `?nombre_dataset`.

Cuando se utiliza un dataset que no ha sido creado por nosotrso, lo recomendable es guardarlo en una variable denominada `df` de data frame. Esto permite modificar el data frame sin modificar el original.

Para obtener la información de un data frmae se emplea las siguientes funciones:

- `head(d.f,n)`: para mostrar las n primeras filas del data frame. Por defecto se muestran las 6 primeras filas.
- `tail(d.f,n)`: para mostrar las n últimas filas del data frame. Por defecto semuestran las 6 últimas.
- `str(d.f)`: para conocer la estructura global de un data frame.
- `names(d.f)`: para producir un vector con los nombres de las columnas.

```
str(Orange)
```

```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':   35 obs. of  3 variables
##  $ Tree         : Ord.factor w/ 5 levels "3"<"1"<"5"<"2"<..: 2 2 2 2 2 2 2 4 4 4 ...
##  $ age          : num  118 484 664 1004 1231 ...
##  $ circumference: num  30 58 87 115 120 142 145 33 69 111 ...
```

```
##  - attr(*, "formula")=Class 'formula'  language circumference ~ age | Tree
##   .. ..- attr(*, ".Environment")=<environment: R_EmptyEnv>
##  - attr(*, "labels")=List of 2
##   ..$ x: chr "Time since December 31, 1968"
##   ..$ y: chr "Trunk circumference"
##  - attr(*, "units")=List of 2
##   ..$ x: chr "(days)"
##   ..$ y: chr "(mm)"
```

1.1. Data Frame de Iris (ejemplo)

Vamos a investigar un poco con el data frame de Iris.

```
df = iris
#Muestra cinco primero elementos del df
head(df,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
```

```
#Muestra cinco últimos elementos del df
tail(df,5)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

```
#Obtenemos los nombres de las columnas de df
names(df)
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

```
#Conocer la estructura global del df
str(df)
```

```
## 'data.frame':    150 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 ...
```

2. Estructura y filtrado de Data Frames

Algunas funciones aplicadas a los data frame son:

- `rownames(d.f)`: para producir un vector con los identificadores de las filas
  - R entiende siempre que estos identificadores son palabras, aunque sean números, de ahí que los imprima entre comillas
- `colnames(d.f)`: para producir un vector con los identificadores de las columnas
- `dimnames(d.f)`: para producir una list formada por dos vectores (el de los identificadores de las filas y el de los nombres de las columnas)
- `nrow(d.f)`: para consultar el número de filas de un data frame
- `ncol(d.f)`: para consultar el número de columnas de un data frame
- `dim(d.f)`: para producir un vector con el número de filas y el de columnas
- `d.f$nombre_variable (columna)`: para obtener una columna concreta de un dataframe
  - El resultado será un vector o un factor, según cómo esté definida la columna dentro del data frame
  - Las variables de un data frame son internas, no están definidas en el entorno global de trabajo de R

```
df = iris
head(df, 5)
```

Ejemplo

```
##   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
```

```
tail(df, 5)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

```
str(df)
```

```
## 'data.frame':    150 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 ...
```

```r
names(df) #colnames
```

```
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

```r
rownames(df)
```

```
##   [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"  "12"
##  [13] "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"  "23"  "24"
##  [25] "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33"  "34"  "35"  "36"
##  [37] "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44"  "45"  "46"  "47"  "48"
##  [49] "49"  "50"  "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"  "59"  "60"
##  [61] "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"  "71"  "72"
##  [73] "73"  "74"  "75"  "76"  "77"  "78"  "79"  "80"  "81"  "82"  "83"  "84"
##  [85] "85"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"
##  [97] "97"  "98"  "99"  "100" "101" "102" "103" "104" "105" "106" "107" "108"
## [109] "109" "110" "111" "112" "113" "114" "115" "116" "117" "118" "119" "120"
## [121] "121" "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132"
## [133] "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144"
## [145] "145" "146" "147" "148" "149" "150"
```

```r
dimnames(df)
```

```
## [[1]]
##   [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"  "12"
##  [13] "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"  "23"  "24"
##  [25] "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33"  "34"  "35"  "36"
##  [37] "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44"  "45"  "46"  "47"  "48"
##  [49] "49"  "50"  "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"  "59"  "60"
##  [61] "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"  "71"  "72"
##  [73] "73"  "74"  "75"  "76"  "77"  "78"  "79"  "80"  "81"  "82"  "83"  "84"
##  [85] "85"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"
##  [97] "97"  "98"  "99"  "100" "101" "102" "103" "104" "105" "106" "107" "108"
## [109] "109" "110" "111" "112" "113" "114" "115" "116" "117" "118" "119" "120"
## [121] "121" "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132"
## [133] "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144"
## [145] "145" "146" "147" "148" "149" "150"
##
## [[2]]
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
```

```r
dim(df)
```

```
## [1] 150   5
```

```r
#Extraer las 10 primeras filas de la columa Petal.Lenght
df$Petal.Length[1:10] # Da un vector
```

```
##  [1] 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5
```

```r
#Extraer las 10 primeras filas de la columa Species
df$Species[1:10] # Da un factor
```

```
##  [1] setosa setosa setosa setosa setosa setosa setosa setosa setosa setosa
## Levels: setosa versicolor virginica
```

2.1. Sub-data Frames

De un Data Frame se puede extraer un SubData Frame.

- `d.f[n,m]`: para extraer "trozos" del data frame por filas y columnas (funciona exactamente igual que en matrices) donde n y m pueden definirse como:
  - intervalos
  - condiciones
  - números naturales
  - no poner nada
  - Si sólo queremos definir la subtabla quedándonos con algunas variables, basta aplicar el nombre del data frame al vector de variables
  - Estas construcciones se pueden usar también para reordenar las filas o columnas

Ejemplo Acceso al Data Frame

```r
#Da subDF de las filas 1 a la 10
df[1:10, ]
```

```
##    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
## 6           5.4         3.9          1.7         0.4  setosa
## 7           4.6         3.4          1.4         0.3  setosa
## 8           5.0         3.4          1.5         0.2  setosa
## 9           4.4         2.9          1.4         0.2  setosa
## 10          4.9         3.1          1.5         0.1  setosa
```

```r
#Da subDF de las filas 1 a la 10 en las columnas 2, 3 y 4
df[1:10, 2:4]
```

```
##    Sepal.Width Petal.Length Petal.Width
## 1          3.5          1.4         0.2
## 2          3.0          1.4         0.2
## 3          3.2          1.3         0.2
## 4          3.1          1.5         0.2
## 5          3.6          1.4         0.2
## 6          3.9          1.7         0.4
## 7          3.4          1.4         0.3
## 8          3.4          1.5         0.2
## 9          2.9          1.4         0.2
## 10         3.1          1.5         0.1
```

```r
#Establecer una condición sobre la fila extrayendola
df[df$Species == "setosa" & df$Sepal.Width> 4, ]
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 16          5.7         4.4          1.5         0.4  setosa
## 33          5.2         4.1          1.5         0.1  setosa
## 34          5.5         4.2          1.4         0.2  setosa
```

```r
#Además de por condición se pueden seleccionar las columnas y filas que nos queremos quedar de la selec
df[df$Species == "setosa" & df$Sepal.Width> 4, ][c(1,3), c(2,5)]
```

```
##    Sepal.Width Species
## 16         4.4  setosa
## 34         4.2  setosa
```

Otro ejemplo con DF de Orange

```r
dataOrange = Orange
dataOrange[c(10:12),]
```

```
##    Tree  age circumference
## 10    2  664           111
## 11    2 1004           156
## 12    2 1231           172
```

```r
dataOrange[c(2,17),c(1,3)]
```

```
##    Tree circumference
## 2     1            58
## 17    3            75
```

```r
dataOrange[2,3]
```

```
## [1] 58
```

```r
dataOrange[dataOrange$circumference<=50,]
```

```
##    Tree age circumference
## 1     1 118            30
## 8     2 118            33
## 15    3 118            30
## 22    4 118            32
## 29    5 118            30
## 30    5 484            49
```

3. Cargar Data Frames en R

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 datas 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

3.1. Carga DT Local

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

```
#../ me permite subir de nivel en el directorio
#Los datos cargados no tienen nombre de columnas
dbulls=read.table("../../../data/bulls.dat",header = FALSE, col.names = c("breed","sale_price","shoulde
head(dbulls)
```

```
##   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
```

```
#Descargado csv a través de kaggle
dkaggle=read.table("../../../data/country_vaccinations.csv",header=TRUE,sep=",")
head(dkaggle,4)
```

```
##     country iso_code       date total_vaccinations people_vaccinated
## 1 Argentina      ARG 2020-12-29                700                NA
## 2 Argentina      ARG 2020-12-30                 NA                NA
## 3 Argentina      ARG 2020-12-31              32013                NA
## 4 Argentina      ARG 2021-01-01                 NA                NA
##   people_fully_vaccinated daily_vaccinations_raw daily_vaccinations
## 1                      NA                    NA                NA
## 2                      NA                    NA             15656
## 3                      NA                    NA             15656
## 4                      NA                    NA             11070
##   total_vaccinations_per_hundred people_vaccinated_per_hundred
## 1                           0.00                             NA
## 2                             NA                             NA
## 3                           0.07                             NA
## 4                             NA                             NA
##   people_fully_vaccinated_per_hundred daily_vaccinations_per_million  vaccines
## 1                                  NA                              NA Sputnik V
## 2                                  NA                             346 Sputnik V
## 3                                  NA                             346 Sputnik V
## 4                                  NA                             245 Sputnik V
##          source_name
## 1 Ministry of Health
## 2 Ministry of Health
## 3 Ministry of Health
## 4 Ministry of Health
##                                                                                source_websit
## 1 http://datos.salud.gob.ar/dataset/vacunas-contra-covid-19-dosis-aplicadas-en-la-republica-argentina
## 2 http://datos.salud.gob.ar/dataset/vacunas-contra-covid-19-dosis-aplicadas-en-la-republica-argentina
## 3 http://datos.salud.gob.ar/dataset/vacunas-contra-covid-19-dosis-aplicadas-en-la-republica-argentina
## 4 http://datos.salud.gob.ar/dataset/vacunas-contra-covid-19-dosis-aplicadas-en-la-republica-argentina
```

```
str(dkaggle)
```

```
## 'data.frame':    1266 obs. of  15 variables:
##  $ country                           : Factor w/ 57 levels "Argentina","Austria",..: 1 1 1 1 1 1 1
##  $ iso_code                          : Factor w/ 54 levels "","ARE","ARG",..: 3 3 3 3 3 3 3 3 3 3 .
##  $ date                              : Factor w/ 41 levels "2020-12-13","2020-12-14",..: 17 18 19 20
##  $ total_vaccinations                : num  700 NA 32013 NA NA ...
##  $ people_vaccinated                 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ people_fully_vaccinated           : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ daily_vaccinations_raw            : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ daily_vaccinations                : num  NA 15656 15656 11070 8776 ...
##  $ total_vaccinations_per_hundred    : num  0 NA 0.07 NA NA NA 0.09 NA NA NA ...
##  $ people_vaccinated_per_hundred     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ people_fully_vaccinated_per_hundred: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ daily_vaccinations_per_million    : num  NA 346 346 245 194 164 143 177 181 185 ...
##  $ vaccines                          : Factor w/ 9 levels "CNBG, Sinovac",..: 9 9 9 9 9 9 9 9 9 9 .
##  $ source_name                       : Factor w/ 34 levels "Centers for Disease Control and Preventi
##  $ source_website                    : Factor w/ 53 levels "http://datos.salud.gob.ar/dataset/vacuna
```

3.2. Carga DT a través de URL

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

```r
dkaggle2=read.table("https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinatio
head(dkaggle2,4)
```

```
##    location iso_code       date total_vaccinations people_vaccinated
## 1 Argentina      ARG 2020-12-29                700                NA
## 2 Argentina      ARG 2020-12-30                 NA                NA
## 3 Argentina      ARG 2020-12-31              32013                NA
## 4 Argentina      ARG 2021-01-01                 NA                NA
##   people_fully_vaccinated daily_vaccinations_raw daily_vaccinations
## 1                      NA                    NA                NA
## 2                      NA                    NA             15656
## 3                      NA                    NA             15656
## 4                      NA                    NA             11070
##   total_vaccinations_per_hundred people_vaccinated_per_hundred
## 1                           0.00                             NA
## 2                             NA                             NA
## 3                           0.07                             NA
## 4                             NA                             NA
##   people_fully_vaccinated_per_hundred daily_vaccinations_per_million
## 1                                  NA                              NA
## 2                                  NA                             346
## 3                                  NA                             346
## 4                                  NA                             245
```

```r
str(dkaggle2)
```

```
## 'data.frame':    1333 obs. of  12 variables:
##  $ location                          : Factor w/ 59 levels "Argentina","Austria",..: 1 1 1 1 1 1 1
##  $ iso_code                          : Factor w/ 55 levels "","ARE","ARG",..: 3 3 3 3 3 3 3 3 3 3 .
##  $ date                              : Factor w/ 41 levels "2020-12-13","2020-12-14",..: 17 18 19 20
##  $ total_vaccinations                : int  700 NA 32013 NA NA NA 39599 NA NA NA ...
##  $ people_vaccinated                 : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ people_fully_vaccinated           : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ daily_vaccinations_raw            : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ daily_vaccinations                : int  NA 15656 15656 11070 8776 7400 6483 7984 8173 8363 ...
##  $ total_vaccinations_per_hundred    : num  0 NA 0.07 NA NA NA 0.09 NA NA NA ...
##  $ people_vaccinated_per_hundred     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ people_fully_vaccinated_per_hundred: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ daily_vaccinations_per_million    : int  NA 346 346 245 194 164 143 177 181 185 ...
```