

Cargar data frames

Téllez Gerardo Rubén

13/6/2021

Data frames interesantes

<https://archive.ics.uci.edu/ml/datasets/Echocardiogram>

Para leer tablas de datos

- **read.table()**: para definir un DF a partir de una tabla de datos contenida en un fichero. Este fichero puede estar guardado o tener su URL, en ambos casos colocar la direccion en forma de string.
 - **col.names(vector)**: especificar el nombre de las columnas.
 - **sep = str**: especificar las separaciones entre columnas en el fichero.
 - **dec = str**: especificar el signo de separación decimal de no ser el punto.

```
E.coli = read.table("https://archive.ics.uci.edu/ml/machine-learning-databases/ecoli/ecoli.data", header = FALSE)

# Mismo resultado
# Le di los nombres aparte en un nuevo CSV
# nursery = read.table("C:/Programación ExL/R_y_Python/tema7_Data_frames/nursery.csv", header = TRUE, sep = ";")
nursery = read.csv("C:/Programación ExL/R_y_Python/tema7_Data_frames/nursery.csv", header = TRUE)

wine = read.table("wine.data", sep = ",", header = FALSE, col.names = c("Class Distribution", "Alcohol", "Acidity", "Malic acid", "Ash", "Chlorophyll", "Sulfur dioxide", "Total phenols", "Free phenols", "Proline"))

head(E.coli)
```

##	Sequence.Name	mcg	gvh	lip	chg	aac	alm1	alm2	site
## 1	AAT_ECOLI	0.49	0.29	0.48	0.5	0.56	0.24	0.35	cp
## 2	ACEA_ECOLI	0.07	0.40	0.48	0.5	0.54	0.35	0.44	cp
## 3	ACEK_ECOLI	0.56	0.40	0.48	0.5	0.49	0.37	0.46	cp
## 4	ACKA_ECOLI	0.59	0.49	0.48	0.5	0.52	0.45	0.36	cp
## 5	ADI_ECOLI	0.23	0.32	0.48	0.5	0.55	0.25	0.35	cp
## 6	ALKH_ECOLI	0.67	0.39	0.48	0.5	0.36	0.38	0.46	cp

```
dim(E.coli)
```

```
## [1] 336 9
```

```
str(E.coli)
```

```
## 'data.frame': 336 obs. of 9 variables:
## $ Sequence.Name: chr "AAT_ECOLI" "ACEA_ECOLI" "ACEK_ECOLI" "ACKA_ECOLI" ...
## $ mcg : num 0.49 0.07 0.56 0.59 0.23 0.67 0.29 0.21 0.2 0.42 ...
## $ gvh : num 0.29 0.4 0.4 0.49 0.32 0.39 0.28 0.34 0.44 0.4 ...
## $ lip : num 0.48 0.48 0.48 0.48 0.48 0.48 0.48 0.48 0.48 0.48 ...
## $ chg : num 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 0.5 ...
## $ aac : num 0.56 0.54 0.49 0.52 0.55 0.36 0.44 0.51 0.46 0.56 ...
## $ alm1 : num 0.24 0.35 0.37 0.45 0.25 0.38 0.23 0.28 0.51 0.18 ...
## $ alm2 : num 0.35 0.44 0.46 0.36 0.35 0.46 0.34 0.39 0.57 0.3 ...
## $ site : chr "cp" "cp" "cp" "cp" ...
```

```
head(nursery)
```

```
## parents has_nurs form children housing finance social
## 1 usual proper complete 1 convenient convenient nonprob
## 2 usual proper complete 1 convenient convenient nonprob
## 3 usual proper complete 1 convenient convenient nonprob
## 4 usual proper complete 1 convenient convenient slightly_prob
## 5 usual proper complete 1 convenient convenient slightly_prob
## 6 usual proper complete 1 convenient convenient slightly_prob
## health health2
## 1 recommended recommend
## 2 priority priority
## 3 not_recom not_recom
## 4 recommended recommend
## 5 priority priority
## 6 not_recom not_recom
```

```
dim(nursery)
```

```
## [1] 12960 9
```

```
str(nursery)
```

```
## 'data.frame': 12960 obs. of 9 variables:
## $ parents : chr "usual" "usual" "usual" "usual" ...
## $ has_nurs: chr "proper" "proper" "proper" "proper" ...
## $ form : chr "complete" "complete" "complete" "complete" ...
## $ children: chr "1" "1" "1" "1" ...
## $ housing : chr "convenient" "convenient" "convenient" "convenient" ...
## $ finance : chr "convenient" "convenient" "convenient" "convenient" ...
## $ social : chr "nonprob" "nonprob" "nonprob" "slightly_prob" ...
## $ health : chr "recommended" "priority" "not_recom" "recommended" ...
## $ health2 : chr "recommend" "priority" "not_recom" "recommend" ...
```

```
head(wine)
```

```
##   Class.Distribution Alcohol Malic.acid  Ash Alcalinity.of.ash Magnesium
## 1           1      14.23      1.71 2.43           15.6      127
## 2           1      13.20      1.78 2.14           11.2      100
## 3           1      13.16      2.36 2.67           18.6      101
## 4           1      14.37      1.95 2.50           16.8      113
## 5           1      13.24      2.59 2.87           21.0      118
## 6           1      14.20      1.76 2.45           15.2      112
##   Total.phenols Flavanoid Nonflavanoid.phenols Proanthocyanins Color.intensity
## 1           2.80      3.06              0.28      2.29           5.64
## 2           2.65      2.76              0.26      1.28           4.38
## 3           2.80      3.24              0.30      2.81           5.68
## 4           3.85      3.49              0.24      2.18           7.80
## 5           2.80      2.69              0.39      1.82           4.32
## 6           3.27      3.39              0.34      1.97           6.75
##   Hue OD280.OD315.of.diluted.wines Proline
## 1 1.04              3.92      1065
## 2 1.05              3.40      1050
## 3 1.03              3.17      1185
## 4 0.86              3.45      1480
## 5 1.04              2.93       735
## 6 1.05              2.85      1450
```

```
dim(wine)
```

```
## [1] 178  14
```

```
str(wine)
```

```
## 'data.frame':  178 obs. of  14 variables:
## $ Class.Distribution      : int  1 1 1 1 1 1 1 1 1 1 ...
## $ Alcohol                 : num  14.2 13.2 13.2 14.4 13.2 ...
## $ Malic.acid              : num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
## $ Ash                     : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
## $ Alcalinity.of.ash       : num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
## $ Magnesium               : int  127 100 101 113 118 112 96 121 97 98 ...
## $ Total.phenols           : num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
## $ Flavanoid               : num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
## $ Nonflavanoid.phenols    : num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
## $ Proanthocyanins         : num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
## $ Color.intensity         : num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
## $ Hue                     : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
## $ OD280.OD315.of.diluted.wines: num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
## $ Proline                 : int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
```

Más atributos de read table

- **stringsAsFactors = BOOL**: Convierte columnas de strings a factores, por defecto en TRUE.

Derivados de read table estándar

- **read.csv()**: formato CSV.

- `read.xls()` | `read.xlsx()`: tipo Excel u Open Office, requiere librería `xlsx`.
- `read.mtb()`: para importar tablas de datos de Minitab, requiere paquete librería `foregin`.
- `read.spss()`: para importar tablas de datos SPSS, se necesita librería `foregin`.

Escribir ficheros

- `write.table(df, file="")`: para exportar un DF a un fichero.
 - `file = ""`: donde indicamos el nombre del archivo
 - `sep = ""`: especificar el símbolo de separación de columnas.
 - `dec = ""`: especificar el separador decimal.