# Lab Dataframe(tablas) en R

## Objetivo

- Mostrar al participante las diferentes operaciones que se pueden realizar con las estructuras de datos llamadas dataframe, en el entorno R.

## Contenido

Para practicar y como ejemplo, usaremos una tabla (dataframe) que viene de serie con R, iris. Tiene 150 filas que corresponden a otras tantas iris ([una especie de flor](http://es.wikipedia.org/wiki/Iris_%28planta%29)) y sus columnas contienen cuatro características métricas de cada ejemplar: la longitud y la anchura de sus pétalos y sépalos; y la subespecie: setosa, versicolor o virgínica, a la que pertenece.

### Inspección de un Dataframe

1. Inspecciona el dataframe iris.

   Escribir en la consola de R
   ```
   iris
   ```

   es lo mismo que ejecutar
   ```
   print(iris)
   ```

2. Inspecciona iris con las siguientes funciones.

   ```
   plot(iris)          # la representa gráficamente
   summary(iris)   # resumen estadístico de las columnas
   ```

3. Inspecciona las primeras y las ultimas seis filas de iris

   ```
   head(iris)   # primeras seis filas
   ```

   | | 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 |

   ```
   tail(iris)   # últimas seis filas
   ```

```
    Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
145          6.7         3.3          5.7         2.5 virginica
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
```

4. Consulta la ayuda de la función head y averigua cómo mostrar las diez primeras filas de iris en lugar de las seis que aparecen por defecto.

5. Inspección y descripción de iris

```
dim(iris)     # filas x columnas
## [1] 150   5
nrow(iris)    # número de filas
## [1] 150
ncol(iris)    # número de columnas
## [1] 5

colnames(iris)  #nombre de sus columnas
## [1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"
## [5] "Species"

str(iris) # "representación textual" del objeto
'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 ...
```

6. Consulta el tamaño, número de filas y el número y nombre de las columnas del conjunto de datos airquality; muestra también las primeras 13 filas de esa table.

7. Examina el conjunto de datos attenu. Consulta su ayuda (?attenu) para averiguar qué tipo de información contiene. Finalmente, usa summary para ver si contiene algún nulo en alguna columna.

## Selección de Filas y Columnas

8. Selecciona las filas y columnas indicadas.

```
iris[1:10,]       # diez primeras filas
iris[, 3:4]       # columnas 3 y 4
iris[1:10, 3:4]
iris[, "Species"]
iris$Species
iris[iris$Species == "setosa",] #utiliza una condicion logica
```

9. Selecciona las filas de iris cuya longitud del pétalo sea mayor que 4.

10. Selecciona las filas donde cyl sea menor que 6 y gear igual a 4 en mtcars. Nota: el operador AND en R es &.

    **Nota:** La selección de filas mediante condiciones lógicas es muy útil y será muy necesaria posteriormente cuando queramos eliminar sujetos con edades negativas, detectar los pacientes con niveles de glucemia por encima de un determinado umbral, etc.

## Creación y Eliminación de Tablas y Columnas

11. Crea una nueva tabla a partir de la tabla iris

    ```
    mi.iris <- iris  # mi.iris es una copia de iris
    head(mi.iris)
    ```

12. Eliminar tablas

    ```
    ls()        # lista de objetos en memoria
    rm(mi.iris)  # borra el objeto mi.iris
    ls()
    ```

13. Agregando columnas

    ```
    mi.iris <- iris
    mi.iris$Petal.Area <- mi.iris$Petal.Length * mi.iris$Petal.Width
    mi.iris$Petal.Area <- NULL
    ```

    Nota: ten en cuenta que:

- agregar una columna que existe la reemplaza,
- agregar una columna que no existe la crea y
- asignar NULL a una columna existente la elimina.

14. Crea una copia del conjunto de datos airquality. Comprueba con ls que está efectivamente creado y luego añádele una columna nueva llamada temperatura que contenga una copia de Temp. Comprueba que efectivamente está allí y luego, elimínala. Finalmente, borra la tabla.

15. Usando el conjunto de datos CO2, selecciona los valores en los que el tratamiento sea chilled, y el valor de uptake, mayor que 15; devuelve únicamente las 10 primeras filas.

## Ordenamiento

La reordenación de las filas de una tabla es fundamental para analizar su contenido y, frecuentemente, para detectar problemas en los datos.

R no dispone de ninguna función de serie para ordenar por una columna (o varias). En R, ordenar es seleccionar ordenadamente:

16. Selecciona iris de manera ordenada y crea una nueva tabla

    mi.iris <- iris[order(iris$Petal.Length),]

    **Nota:** La función order aplicada a un vector devuelve otro vector de la misma longitud que tiene el valor 1 en el primer elemento del vector, 2 en el segundo, etc. Es decir, en el ejemplo anterior, mi.iris tiene como primera fila aquella que corresponde al valor más pequeño de iris$Petal.Length, etc.

17. Verifica que mi.iris <- iris[order(-iris$Petal.Length),] ordena decrecientemente.

18. Crea una versión de iris ordenando por especie y dentro de cada especie, por Petal.Length. Ten en cuenta que en R se puede ordenar por dos o más columnas porque order admite dos o más argumentos (véase ?order). Por ejemplo, iris[order(iris\(Petal.Length, iris\)Sepal.Length),] deshace los empates en Petal.Length de acuerdo con Sepal.Length.

19. Encuentra el día más frío de los que contiene airquality.

20. Usando el mismo conjunto de datos (airquality), encuentra el día más caluroso del mes de junio.

21. Estudia el conjunto de datos airquality (información meteorológica de ciertos meses de cierto año en Nueva York, que también viene de serie en R) aplicando las funciones anteriores. En particular, responde las preguntas: ¿cuál es la temperatura media de esos días? , ¿cuál es la temperatura media en mayo? Y ¿cuál fue el día más ventoso?

22. Crea una tabla adicional seleccionando todas las columnas menos mes y día; luego haz un plot de ella y trata de encontrar relaciones (cualitativas) entre la temperatura y el viento, o el ozono,…

23. Usando el conjunto de datos mtcars (consulta ?mtcars), averigua: ¿cuál es el modelo que menos consume?, ¿cuál es el consumo medio de los modelos de 4 cilindros?

## Lectura de Datos Externos

Crea la carpeta C:\R\data y copia los archivos de datos

24. Configura el directorio de trabajo

```
getwd()
[1] "C:/Users/simulador/Documents"
setwd("c:/R/data") # ruta absoluta en Windows
dir()          # contenidos del directorio "de trabajo"
[1] "BPF1_12022019173128564.csv"
```

25. Lee el archivo BPF1_12022019173128564.csv, utilizando la función read.table()

```
bankprofit <- read.table("BPF1_12022019173128564.csv", sep = ",", header = TRUE)

str(bankprofit)
'data.frame':   12609 obs. of  11 variables:
$ ï..ITEM   : Factor w/ 49 levels "BA14TE","BA15TE",..: 24 24 24 24 24 24 24 24 24 24 ...
$ Item      : Factor w/ 49 levels "1. Interest income",..: 1 1 1 1 1 1 1 1 1 1 ...
$ BANK      : Factor w/ 1 level "ALL": 1 1 1 1 1 1 1 1 1 1 ...
$ Bank      : Factor w/ 1 level "All banks": 1 1 1 1 1 1 1 1 1 1 ...
$ COU       : Factor w/ 28 levels "AUT","BEL","CAN",..: 1 1 1 1 1 1 1 1 1 1 ...
$ Country   : Factor w/ 28 levels "Austria","Belgium",..: 1 1 1 1 1 1 1 1 1 1 ...
$ YEA       : int  1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 ...
$ Year      : int  1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 ...
$ Value     : num  22333 27413 26867 23522 20846 ...
$ Flag.Codes: logi  NA NA NA NA NA NA ...
$ Flags     : logi  NA NA NA NA NA NA ...

bankprofit[bankprofit$Country == "Chile",]
```

Examinemos la instruccion anterior:

- **bankprofit** es el nombre de la tabla que recibirá la información leída por read.table; de no hacerse la asignación, R se limitará a imprimirlos en la consola.
- " BPF1_12022019173128564.csv " (¡entrecomillado!) es la ruta o nombre del fichero de interés. Se trata de una ruta relativa al directorio de trabajo, que se supone que contiene el subdirectorio data_dir y, dentro de él, el fichero BPF1_12022019173128564.csv.
- sep = "," indica que los campos del fichero están separados por comas(,) si fueran tabuladores (sí, el tabulador es \t). Muchos ficheros tienen campos separados por,

además del tabulador, caracteres tales como ,, ;, | u otros. Es necesario indicárselo a R y la manera de averiguar qué separador usa un fichero, si es que no se sabe, es abriéndolo previamente con un editor de texto decente.

- **header = TRUE** indica que la primera fila del fichero contiene los nombres de las columnas. Si olvidas especificarlo y la primera fila del fichero contiene efectivamente el nombre de las columnas, R interpretará estas, erróneamente, como datos.

Con una expresión similar a esa, tal vez cambiando el separador, se leen la mayoría de los ficheros de texto habituales. Otras opciones de las muchas que tiene read.table que pueden ser útiles en determinadas ocasiones son:

- dec, para indicar el separador de decimales. Por defecto es ., pero en ocasiones hay que cambiarlo a dec = "," para que interprete correctamente, p.e., el antiguo estándar español (p.e., 67,56 en lugar de 67.56).
- quote, que indica qué caracter se usa para acotar campos de texto. En algunas ocasiones aparecen campos de texto que contienen apóstrofes (p.e., calle O'Donnell) y la carga de datos puede fracasar de no indicarse quote = "". Esta expresión desactiva el papel especial de acotación de campos de texto que por defecto tienen las comillas.

26. Lee el fichero paro.csv usando la función read.table. Comprueba que está correctamente importado usando head, tail, nrow, summary, etc. Para leer la tabla necesitarás leer con cierto detenimiento ?read.table.

27. Repite el ejercicio anterior eliminando la opción header = TRUE. Examina el resultado y comprueba que, efectivamente, los datos no se han cargado correctamente.

28. En read.table y sus derivados puedes indicar, además de ficheros disponibles en el disco duro, la URL de uno disponible en internet. Prueba a leer directamente el fichero disponible en http://<url servidor amazon>/R/data/datos_treemap.txt. Nota: es un fichero de texto separado por tabuladores y con nombres de columna.