



## Datos en R – parte I

### 1) Primeros pasos en R: tipos de datos (Recordatorio)

Realice una serie de experimentos para familiarizarse o recordar los tipos de datos más usuales en R.

- Construya un vector, muéstrelo, nombre cada componente con un nombre o etiqueta, muestre un valor en concreto, realice la suma de todos sus valores en caso de ser un vector numérico, etc.
- Construya una matriz 3x3 con los números del 1 al 9. Realícelo de varias formas posibles. Muestre un valor en concreto de la matriz.
- Construya un *DATA FRAME* a partir de tres vectores, uno de cadenas y otros dos numéricos. Estudie su estructura, muestre sus primeros valores y estudie las formas de acceder tanto a sus valores concretos como a sus filas y columnas.
- Construya una lista a partir de un vector, una matriz y una data frame que serán su contenido. Estudie como se accede al contenido de dicha lista y las distintas maneras de hacerlo.

### 2) Primeros pasos en R: utilidades (Recordatorio)

- ¿Cómo se sabe desde una instrucción de código cuál es el directorio de trabajo?
- ¿Cómo se puede cambiar de directorio de trabajo desde una línea de código?
- ¿Cómo se pueden averiguar los objetos que están cargados en memoria en la sesión en la que estamos trabajando?
- ¿Sabe cómo borrarlos?
- ¿Cómo se instala un paquete concreto de una librería?
- ¿Cómo se carga para que esté disponible?
- ¿Sabía que la función `search()` le suministra la lista de paquetes que tiene cargados o que la función `ls()` le indica los objetos disponibles en memoria?

### 3) Primeros pasos en R: cargando información

Las funciones más importantes para importar un fichero con un data set a un objeto de R, por ejemplo un data frame, son:

- `read.table(---)`
- `read.delim(----)`
- `read.csv(----)`

estudie en la ayuda de R estas tres funciones.

Se le proporciona como material un fichero .csv “de juguete” de nombre “datos1\_prueba.csv”. Impórtelo de las dos siguientes maneras y estudie la diferencia:

```
df<-read.csv("datos1_prueba.csv")  
df2<-read.csv("datos1_prueba.csv",stringsAsFactors = FALSE)
```

### 4) Funciones gather y spread de la librería tidyr

La función `gather` y `spread` de la librería `tidyr` son muy útiles para preprocesar datos. La primera función reúne una serie variables de tan sólo una pareja de pares y valores, de decir, una dataset que tenga muchas columnas lo convierte en uno con menos columnas. La

función `spread` hace justo lo contrario. Usando el dataframe `df`, usado en el apartado anterior, ejecute el siguiente código y estudie el funcionamiento de estas dos funciones.

```
df_aux<-gather(df,time,val,t1:t3)
df_aux

#Se expanden
#una dataset "largo" lo convierto en "ancho"
df_aux2<-spread(df_aux,time,val)
df_aux2
```

Así mismo, estudie mediante la función `is.na(---)`, que no es de este paquete, si hay algún missing value en este dataframe.

A modo de curiosidad, ejecute en la consola: `edit(data.frame())`

5) Algunas utilidades:

- En la librería `stringr` se encuentran las funciones `str_trim(---)` para eliminar espacios de una cadena de caracteres y la función `str_replace(--)`, entre otras. Cree varios cadenas a modo de ejemplo y estudie el comportamiento de estas funciones.
- En la librería `lubridate` se encuentran una serie de utilidades para trabajar con fechas. Estudie el funcionamiento de las funciones `ymd()` y `ymd_hms()`

6) **Limpieza de datos: un caso práctico**

Se le proporciona un fichero de nombre “weather.csv” con datos de observaciones meteorológicas del área de Boston (EEUU). Dicho fichero no está listo para su análisis y debe ser previamente preparado. Es decir, se trata de aplicar la fase de pre procesado, o de limpieza de datos, antes de dejarlo listo para analizar o modelizar. Así mismo, se le proporciona un script de R en que se le guía dicho proceso. Descargue el fichero de los datos y el script, incompleto, y lleve a cabo la tarea de limpieza.

7) Sintaxis de escritura de funciones en R

Escriba una función que:

- dado dos valores devuelva el mayor de ellos.
- imprima diez veces un nombre por consola: utilice un `**_for_**`
- imprima diez veces un nombre por consola: utilice un `**_while_**`
- sume dos números y tome por defecto el segundo valor a diez.
- dados tres números devuelva el mayor.
- dada una lista de valores numéricos devuelva otra lista con aquellos valores que sean pares.
- dada una lista con valores numéricos devuelva la suma de los valores pares.
- dada una lista de valores devuelva el mayor de los valores pares.
- dada una lista y un valor indique si se encuentra o no en dicha lista.