

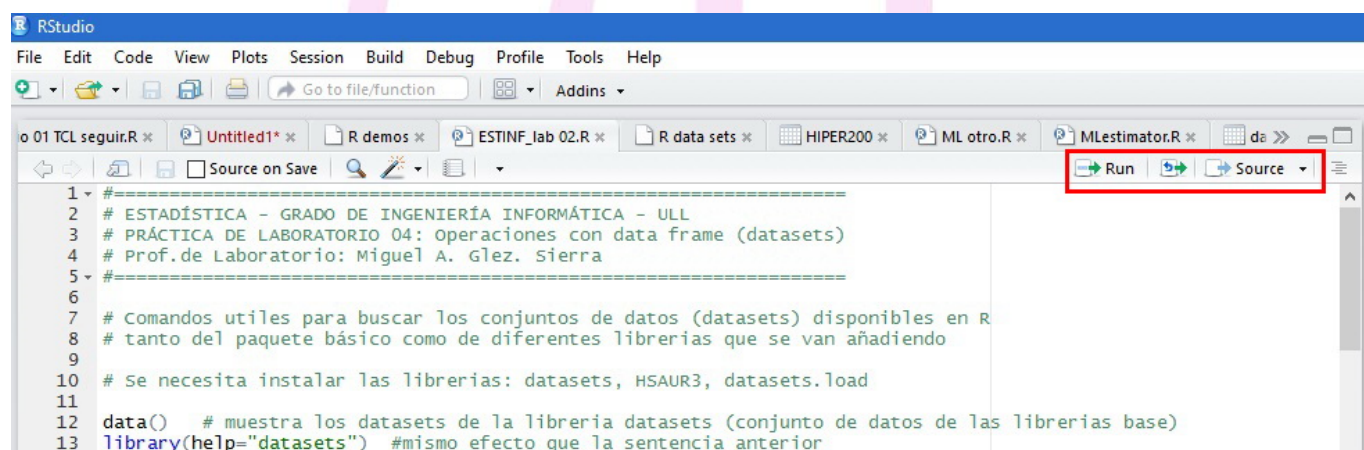
## PRÁCTICA 01: Instrucciones básicas de R

### DESARROLLO DE LA PRÁCTICA

En esta primera práctica de laboratorio, de introducción al software estadístico R, nuestro objetivo primordial es hacernos lo más familiar posible con el entorno de trabajo de RStudio y del R Commander. Nos centraremos en los aspectos necesarios e indispensables para poder realizar al final del curso un análisis estadístico elemental a un conjunto de datos. Como en cualquier lenguaje de programación existen unos preliminares que trataremos de ver en esta práctica.

### PREVIO A LA PRÁCTICA

Cuando se carga un fichero script, como es esta ocasión, las sentencias se pueden ejecutar de varias formas



Pulsando **Run** se ejecuta la línea en la que se encuentra el cursor, o si hemos seleccionado varias líneas se ejecutarán todas. Con el siguiente icono desde donde se está hasta el inicio. Con **Source** se ejecuta todo el contenido del script. Si seleccionamos parte de una sentencia con **Run** solo se ejecuta lo sombreado. Además, con Run, no tenemos que seguir el orden secuencial del fichero, se puede ejecutar cualquier línea solo basta desplazar el cursor a dicha línea.

### NOCIONES BÁSICAS DEL LENGUAJE DE R

#### Introducción

Como en cualquier lenguaje de programación, con R podemos hablar de constantes, variables, vectores, matrices, arrays, caracteres, lógicos y operaciones entre ellos de acuerdo con su tipo. Prácticamente, cualquier cosa (vector, matriz, data frame,...) es un **objeto** en R. Por defecto R trabaja en lenguaje de alto nivel dirigido hacia objetos, donde se presupone ciertas cosas y valores por defecto. Así cuando ponemos un comando no se compila y luego se ejecuta, sino que tenemos un “interprete” de instrucciones de acuerdo a unas normas generales

#### Generalidades básicas:

-En principio, cada instrucción (sentencia) **acaba con un signo de punto y coma (;)**.

-Si la sentencia es muy larga **su continuación** en la línea siguiente **se indica por el signo +**.

- **Sentencia de asignación:**  $a \leftarrow -3 + 2$  (preferentemente  $\leftarrow$ ). También puede usarse  $\rightarrow$  y últimamente el signo igual  $=$ . Se puede utilizar espacios en blanco para una mejor visión de los comandos

-Comentarios: se inician con el símbolo #.

#### #Ejemplos de asignación numérica

```
a <- -2
```

```
3 -> b
```

```
c = 5
```

```
c = 7
```

Comenzando por los elementos más simples de cualquier lenguaje, podemos hablar de una **constante**, preferentemente se entiende que es **numérica**, sin entrar si es entero, real o complejo de simple o doble precisión, o **no numérica**, en cuyo caso trabajamos con un factor, sinónimo de una variable que no asume valores numéricos, a lo máximo que sean ordenados. También podemos hablar de character,

#### #Ejemplos de asignación no numérica

```
a <- "2"
```

```
b <- "casado"
```

Podemos pasar de hablar de constantes a mencionar los **vectores**, números o caracteres, formado por dos o más elementos del mismo tipo.

- $c()$  es una función genérica que combina (asigna) valores en un vector o una lista:

```
a <- c(1,5,3,2,9,-1)
```

```
b <- c("d","g","m","z","13")
```

```
d <- c(3:15)
```

Si hablamos de **matrices** (numéricas o no numéricas, pero también puede ser lógica, de fechas), todos los elementos de la matriz deben ser del mismo tipo

#### # Ejemplos de instrucciones básicas para trabajar en R con matrices:

```
d <- matrix( 1:10, nrow=2)      # matriz
```

```
d <- matrix( 1:10, ncol=2 )      # matriz
```

```
d <- matrix( 1:10, nrow=5, ncol=2 )  # matriz
```

```
d <- matrix( 1:1000, ncol=5 )      # matriz
```

```
d <- matrix(c(18,15,12,27,5, 4,9,7,-7,-5,-4,-2), nrow=3)
```

```
d <- matrix(c(18,15,12,27,5, 4,9,7,-7,-5,-4,-2), ncol=3)
```

```
d <- matrix(c(18,15,12,27,5, 4,9,7,-7,-5,-4,-2), nrow=4, ncol=3)
```

```
d <- matrix(c(18,15,12,27,5, 4,9,7,-7,-5,-4,-2), nrow=3, ncol=4)
```

Ver la ayuda sobre "matrix"

---

-Si disponemos de una matriz A de orden nxp, el acceso a sus elementos puede hacerse según:

A[i,j] nos referimos a un elemento determinado

A[i,] nos referimos a la fila i

A[,j] nos referimos a la columna j

### # Ejemplos de acceso a los elementos de la última matriz d creada anteriormente:

d[3,2] # elemento que ocupa el lugar (3,2) en la matriz d

d[2,] # elementos que forman la fila 2 en la matriz d

d[,3] # elementos que forman la columnas 3 en la matriz d

De igual manera se accede a los elementos de un **vector**.

Si seguimos ascendiendo en la complejidad de las estructuras de los datos, tenemos el **data frame**. R usa el termino **data.frame** para contener datos estructurados en forma de variables (columnas) para los diferentes individuos (filas). Las variables de un data frame pueden ser de distinto tipo, pero los vectores deben tener la misma longitud. Es la denominación común cuando recogemos una serie de variables (p) sobre un conjunto de n individuos. Se forma una matriz de nxp, donde los datos de los individuos van dispuestos en las filas y los valores de las variables en columnas.

Casi finalizando, tenemos las **listas** como una colección de elementos que forman un vector o una matriz, pero cuyos elementos no tienen que ser del mismo tipo ni de la misma dimensión.

Finalmente tenemos los **objetos**. Prácticamente, cualquier cosa (vector, matriz, data frame,...) es un **objeto** en R. Por defecto R trabaja en lenguaje de alto nivel dirigido hacia objetos, donde se presupone ciertas cosas y valores por defecto.

### Utilidades varias

-El nombre de un objeto de R, puede ser cualquier cadena alfanumérica formada por letras, números del 0 al 9 y el símbolo "." (punto). Debe empezarse por una letra y recordar que R distingue entre minúsculas y mayúsculas. Por ejemplo "**Estoesunnombre.345.inicial**".

De paso, indiquemos que el filetype de un fichero script es R, por ejemplo "**prueba01a.R**" y el de un dataframe es RData, por ejemplo "**datos123.RData**".

-Para visualizar el contenido de un objeto en la consola sólo es necesario escribir su nombre.

-**workspace**: todas las variables u objetos creados en R son almacenados en el espacio de trabajo (workspace).

-**ls()** permite ver todas las variables u objetos activos en R

-Como en cualquier otro lenguaje de programación, en R existen instrucciones para controlar el flujo de un programa: **for**, **if else**, **while**, **repeat**, etc.

-Una de las ventajas de R es la posibilidad de poder crear funciones que se adapten a la necesidad del usuario.

-R usa el objeto **data frame** para contener datos estructurados en forma de variables (columnas) para los diferentes individuos (filas). Las variables de un data frame pueden ser de distinto tipo, pero los vectores deben tener la misma longitud.

-La función **paste()** permite concatenar elementos de diferentes tipos.

-R está en constante desarrollo: existen 11.110 paquetes, en julio de 2017, complementarios al paquete base, todos ellos de libre utilización.

### Operaciones aritméticas, lógicas y de comparación

ARITMÉTICOS	COMPARACIÓN	LÓGICOS
+ suma	= = igualdad	& y lógico
- resta	!= diferente de	! no lógico
* multiplicación	< menor que	o lógico
/ división	> mayor que	
^ potencia	<= menor o igual	
%/% división entera	>= mayor o igual	

### Algunas funciones elementales:

Raíz cuadrada de x	sqrt(x)
Exponencial de x	exp(x)
Logaritmo neperiano	log(x)
	log10, log(x,base)
Seno de x	sin(x)
Coseno de x	cos(x)
Tangente de x	tan(x)
Valor absoluto de x	abs(x)

### Valores faltantes (missing values)

Cuando no se recoge el valor de una variable sobre un individuo se le asigna un valor especial NA (acrónimo de Not Available). En general, cualquier operación donde intervenga un valor NA da por resultado NA. Además, hay una clase adicional de valores omitidos producidos por el cálculo NAN (acrónimo de Not A Number) para designar valores como infinito.

### Funciones usuales de vectores,....

<b>round(x, n)</b>	redondea los elementos de x a ncifras decimales
<b>sort(x)</b>	ordena los elementos de x en orden ascendente
<b>rev(x)</b>	invierte el orden de los elementos en x
<b>rank(x)</b>	asigna rangos a los elementos de x
<b>log(x, base)</b>	calcula el logaritmo de x en base "base"
<b>scale(x)</b>	si x es una matriz, centra y reduce los datos; si se desea centrar solamente utilizar scale=FALSE, para reducir solamente usar center=FALSE
<b>length(x)</b>	número de elementos en x
<b>sum(x)</b>	suma de los elementos de x
<b>prod(x)</b>	producto de los elementos de x
<b>max(x)</b>	valor máximo en el objeto x

<b>min(x)</b>	valor mínimo en el objeto x
<b>range(x)</b>	rango de x o c(min(x), max(x))
<b>mean(x)</b>	promedio de los elementos de x
<b>median(x)</b>	mediana de los elementos de x
<b>var(x) o cov(x)</b>	varianza de los elementos de x(calculada con $n - 1$ ); si x es una matriz o un conjunto de datos (data frame), se calcula la matriz de varianza-covarianza
<b>cor(x)</b>	matriz de correlación de x si es una matriz o un conjunto de datos (1 si x es un vector)
<b>var(x, y) o cov(x, y)</b>	covarianza entre x e y, o entre las columnas de x e y si son matrices o conjunto de datos
<b>cor(x, y)</b>	correlación lineal entre x e y, o la matriz de correlación si x e y son matrices o conjunto de datos

### Algunos comandos básicos

<b>data()</b>	conjuntos de datos disponibles. Se listan los ficheros de datos que suministra o incorpora el programa.
<b>attach(fichero)</b>	carga en memoria las variables del data frame
<b>detach(fichero)</b>	descarga de la memoria las variables del data frame
<b>ls()</b>	lista de los objetos que están en memoria
<b>rm(objeto)</b>	elimina el objeto en memoria
<b>attributes(data.frame)</b>	lista de los objetos en memoria
<b>getwd()</b>	directorio de trabajo
<b>setwd()</b>	establece el directorio de trabajo
<b>na.omit(x)</b>	elimina las observaciones con datos ausentes (NA)
<b>is.na(x)</b>	devuelve TRUE cuando encuentra valor omitido (NA)
<b>which(is.nan(x))</b>	devuelve los índices de los valores con TRUE para (NA)
<b>unique(x)</b>	si x es un vector o un conjunto de datos, devuelve un objeto similar pero suprimiendo elementos duplicados
<b>subset(x, ...)</b>	devuelve una selección de x con respecto al criterio (...típicamente comparaciones: $x\$V1 < 10$ ); si x es un conjunto de datos, la opción selectproporciona las variables que se mantienen (o se ignoran con -)

---

EJERCICIOS

1- Proponer cinco vectores de dimensión 5, llamémosles a, b, c, d y e, respectivamente.

- a) Sumar los dos primeros vectores  $a+b$ , Realizar el producto escalar entre a y b. Calcular  $2a-5b+3d$
- b) Crear una matriz de  $5 \times 5$  con dichos vectores como columnas. Calcular el determinante de A. Calcular sus raíces y vectores característicos. Su inversa, si existe.
- c) Combinar los 25 datos en un vector x. Ordenarlos de menor a mayor. Determinar su valor mínimo. Determinar su valor máximo. La suma de todos los datos. La media aritmética de todos los datos.

2.- Resuelve el siguiente sistema de ecuaciones:

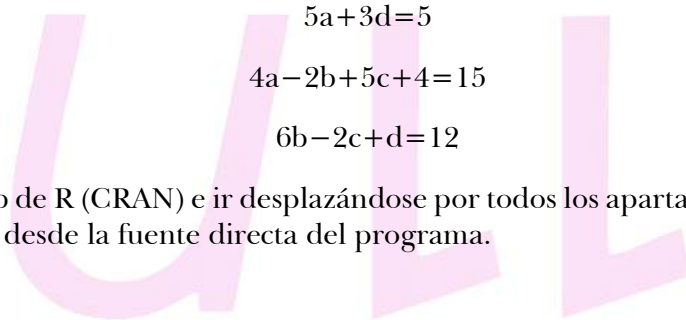
$$7a+6b+3c+2d=10$$

$$5a+3d=5$$

$$4a-2b+5c+4=15$$

$$6b-2c+d=12$$

3- Acudir a la página web de R (CRAN) e ir desplazándose por todos los apartados, para ver la información que tenemos disponible desde la fuente directa del programa.



Universidad  
de La Laguna

Departamento de Matemáticas,  
Estadística e Investigación Operativa