Apuntes didácticos R: Práctica 1

¿Qué es R?

En la actualidad R es un lenguaje interpretado que emplea variables, vectores, matrices, gráficos con los que se realizan operaciones aritméticas. Es muy utilizado en el campo de la estadística y con múltiples aplicaciones en los campos de la investigación científica, el aprendizaje automático (machine learning), minería de datos, econometría, investigación biomédica, bioinformática y en el campo económico-financiero.

Existen dos variedades programa: R y Rstudio. Cada una cuenta con algunas ventajas, en el caso de R la consola y scripts se mantienen mejor organizados, mientras que en Rstudio se marcan las funciones predefinidas asegurando que se han escrito bien.

Operaciones aritméticas

Los símbolos para operar son:

- + : Suma
- - : Resta
- * : Producto
- / :> Cociente
- ** :> Potencia
- "<" : Menor
- ">" :Mayor
- "<=" :Menor o igual
- ">=" :Mayor o igual
- "!=" : Distinto
- "==" : Igualdad lógica

Algunas funciones matemáticas:

- Funciones logarítmicas
  - log(x): logaritmo neperiano.
  - log10(x): logaritmo en base 10.
  - log2(x): logaritmo en base 2.
  - logb(x,base): logaritmo en cualquier base.
  - exp(x): función exponencial.
- Funciones trigonométricas
  - sin(x): seno.
  - cos(x): coseno.
  - tan(x): tangente trigonométrica.
  - asin(x): arco seno.
  - acos(x): arco coseno.
  - atan(x): arco tangente.

Otras funciones:

- abs(x): valor absoluto.
- sqrt(x): ra´ız cuadrada.
- factorial(x): factorial.
- choose(n,x): binomio de Newton

Las operaciones se ejecutan siguiendo el orden tradicional de:

- Operaciones entre paréntesis
- Potencias
- Productos y divisiones
- Sumas y restas

Ej:

```
>2**4+6*5-5*(4/2-2)

[1] 46
```

1º Se ejecuta la operación el cociente de 4 entre 2 menos 2

2º Se realiza la potencia de 2 elevado a la 4

3º Se realiza el producto de 6 por 5 y 5 por 0

4º Se realiza la suma de 16 más 30 y menos 0

Nota

Se debe tener en cuenta que las simplificaciones cuando hay variables del estilo 2x, dan lugar a errores. Se debe tener una sintaxis adecuada, por lo que la operación sería 2*x

Variable

En R se pueden almacenar en expresiones, utilizando <= o =, tanto valores numéricos:

```
> a=4
> a
[1] 4
```

Permite ver el valor almacenado en la variable, también se utiliza print => print(a)

Como palabras y otras expresiones:

```
> b="Alberto"
> b
[1] "Alberto"
```

(Para añadir texto utilizamos comillas "…")

Estas variables pueden contener mayúsculas y secuencias alfa numéricas. Además, al asignar un segundo valor a una variable, el primer valor previamente almacenado se pierde.

Ej:

```
> a=2
> a
[1] 2
> a=3
```

> a

[1] 3

Consola de operaciones

La consola cumple varias funciones:

- Realizar operaciones de manera similar a una calculadora

  Ej: escribimos la suma 5+3 y 4-9 en la consola y pulsamos intro.

  > 5+3

  [1] 8

  > 4-9

  [1] -5

- Ejecutar los programas

  La consola será el espacio en el que ejecutaremos los scripts o programas que realicemos. Es conveniente realizar ejecuciones periódicas para asegurar que la parte ya programada realice la función que queremos. La consola recordará el valor de las variables hayamos utilizado, para evitar que se tomen valores antiguos deberemos añadir al principio del programa la secuencia: > rm(list=ls(all=TRUE))

Funciones predefinidas

Muestreo y toma de valores aleatorios

- rnorm(x)

  Nos permite tomar un valor aleatorio muestreado de una distribución normal de varianza 1 y media 0. Siendo “x” el número de valores aleatorios que se quieren obtener.

  Ej:

  > rnorm(1)

  [1] -1.682738

  > rnorm(2)

  [1] -0.2113294 -1.3060296

- > runif(x)

  Nos permite tomar un valor aleatorio en el intervalo [0,1] , siendo “x” de nuevo el número de valores aleatorios que queremos

  Ej:

> runif(1)

[1] 0.8099334

> runif(2)

[1] 0.3179668 0.2735054

Se puede utilizar el producto de esos valores para obtener valores en el rango que queramos. Sin embargo, se suele utilizar la función “sample”.

- sample(x:y,z,replace=TRUE/FALSE)

  La función sample nos permite tomar valores aleatorios en un intervalo [x,y], siendo “z” el número de valores aleatorios que queremos obtener. Si no queremos obtener valores aleatorios repetidos, añadimos la secuencia replace=FALSE, de lo contrario sustituímos FALSE por TRUE.

  Ej:

  > sample(1:5,2,replace=TRUE)

  [1] 4 4

  > sample(1:5,2,replace=FALSE)

  [1] 5 1

Listado de variables

- ls()

  Esta función nos permite listar todas las variables registradas en la consola.

  Ej:

  > a=4

  > b=120

  > c="Antonio"

  > ls()

  [1] "a" "b" "c"

  Si añadimos .str(), es decir ls.str(), podremos ver los valores almacenados en las variables.

  Ej:

```
> ls.str()
a :  num 4
b :  num 120
c :  chr "Antonio"
```

Se nos indica además si dentro de la variable se almacena un número (num) o texto (chr)

Si en el interior del paréntesis añadimos la función pat=x, es decir, ls(pat=x) siendo x cualquier letra o valor numérico, se mostrarán aquellas variables cuyo nombre contenga dicha letra.

Ej:

```
> mapa=2
> pato="Bueno"
> Alberto=7
> ls(pat="p")
[1] "mapa" "pato"
```

Nota

Si quisiéramos que aparezcan las variables que empiezan por una determinada letra bastaría con utilizar ^, tal que "^a".

Eliminar variables

- rm()

Esta función nos permite eliminar las variables registradas en la consola. Para eliminar todas las variables se utiliza la función rm(list=ls(all=TRUE)), pero si se quiere eliminar una en concreto, se añade el nombre de la variable en el paréntesis rm(x), siendo x el nombre de la variable

Ej:

```
> Antonio=2
> Paco=3
> Juan=4
> rm(Antonio)
> ls()
[1] "Juan" "Paco"
> rm(list=ls(all=TRUE))
> ls()
character(0)
```

No se han encontrado variables

Ejercicio

- Crear los objetos pais1, pais2, poblacion1, poblacion2 de manera que pais1 contenga el valor Italia, pais2 contenga el valor Alemania, poblacion1 contenga el valor 50, poblacion2 contenga el valor 80.
- Listar todo lo que hay en memoria. listar objetos que contengan el carácter p.
- Listar objetos que contengan el carácter b.
- Listar objetos que comiencen con el carácter b.
- Listar objetos que comiencen con el carácter p.
- Listar todos los objetos que hay en memoria de manera que se obtengan sus características.
- Borrar los objetos pais1, poblacion1.
- Volver a listar el contenido de la memoria.

Solución

Para resolver este ejercicio haremos uso de las funciones predefinidas anteriormente abordadas.

Programa:

```
rm(list=ls(all=TRUE))

pais1="Italia"

pais2="Alemania"

poblacion1=50

poblacion2=80

ls()

ls(pat="p")

ls(pat="b")

ls(pat="^b")

ls(pat="^p")

ls.str()

rm(pais1)

rm(poblacion1)

ls()
```

Que tendrá como resultado al introducirlo en la consola:

```
> rm(list=ls(all=TRUE))

> pais1="Italia"

> pais2="Alemania"

> poblacion1=50

> poblacion2=80
```

```
> ls()
[1] "pais1"     "pais2"     "poblacion1" "poblacion2"
> ls(pat="p")
[1] "pais1"     "pais2"     "poblacion1" "poblacion2"
> ls(pat="b")
[1] "poblacion1" "poblacion2"
> ls(pat="^b")
character(0)
> ls(pat="^p")
[1] "pais1"     "pais2"     "poblacion1" "poblacion2"
> ls.str()
pais1 :  chr "Italia"
pais2 :  chr "Alemania"
poblacion1 :  num 50
poblacion2 :  num 80
> rm(pais1)
> rm(poblacion1)
> ls()
[1] "pais2"     "poblacion2"
```