ACTIVIDAD 1

Les proponemos realizar las siguientes actividades para que se familiaricen con el entorno de RStudio y hagan su primer contacto con el programa R. Los ejercicios propuestos son simples y hasta puede ser que los encuentre sin-sentido pero son la excusa perfecta para comprender su funcionamiento del entorno R.

Ejercicio 1- Usando R como calculadora

En la consola de RStudio realiza los siguientes cálculos y observa los resultados

4+5

27-11

3*9

23/4

23%/%4

23%%4

2/3+4/7

sqrt(36)

sqrt(79)

sin(pi)

sin(3.14)

Ejercicio 2- Más cálculos

Te proponemos que calcules la raíz cuadrada de 144, dividido la raíz cubica de 27.

(Ayudita: la raíz cúbica hay que crearla con exponenciación)

Calcular el valor absoluto del resultado de: 356 menos 366, multiplicado por el valor pi. Redondear el resultado con dos decimales.

Ejercicio 3- Mi primer Script

Antes de comenzar abre en el entorno de RStudio el archivo denominado: "Tips_y_Generalidades.R" Luego te pedimos que especifiques tu espacio de trabajo siguiendo las indicaciones del Script y crea tu primer Script al que llamaremos "Actividad_bloque1.R"



Escribir las siguientes líneas en el editor de código y ejecutarlas.

No olvides guardar el contenido del script.

a <- 5

b <- 4

c <- a + b

a <- b * c

 $b <- (c - a)^2$

c <- a * b

(Ayuda: puedes copiar y pegar las líneas.)

Finalmente: Cuánto valen los objetos a, b y c?

Ejercicio 4- Trabajando con vectores

i) Crea los objetos **x** e **y** ejecutando las siguientes líneas de código.

x <- c(1,3,5,7,9)

y < -c(2,3,5,7,11,13)

ii) Suma 1 a cada elemento del vector x

iii) Suma 1 en cada elemento del vector y

iv) Al vector y multiplícalo por 3

v) Suma la cantidad de elementos del vector x e y

vi) Sumar x+y

Ejercicio 5- Operaciones con vectores

- i) Escribe un vector con los números consecutivos del 1 al 10
- ii) Escribe un vector de números consecutivos decrecientes a partir del 10
- **iii)** Otra forma de crear vectores es utilizando función "seq(a,b,c)", que genera secuencias de números reales, donde el primer elemento indicara el principio de la secuencia, el segundo el final y el tercero el incremento que se debe usar para generar la secuencia. También podemos poner la función de estas formas: "seq(length = d; from = a; to = b)" o "seq(by = c; from = a; to = b)" siendo "d" la longitud del vector.

Te proponemos que usando esas funciones crees un vector con la secuencia del 1 al 10 de números impares, y otro vector con una secuencia del 1 al 10 que contenga 6 elementos (usa el argumento *length*)



iv) Con la función "rep(a; b)" podemos crear un vector con "b" elementos idénticos al valor "a". Crea un vector que repita 10 veces el elemento 5.

Repetir cada elemento de un vector 3 veces de en 5.

- v) Cree el vector llamado meses que contenga los meses del año. Luego llamelo para verlo en la consola
- vi) Obtenga la estructura del vector meses con str(). ¿Qué nos indica?
- vii) Obtenga el modo y la longitud del vector meses

Ejercicio 6- Bloque Environment

Guarde el entorno (bloque Environment) para almacenar los objetos creados en el workspace, bajo el nombre **Actividad_1** (.RData)

A continuación elimine desde la consola el objeto "meses", con la función adecuada.

Pida en la misma consola que R muestre los objetos activos en memoria que quedaron.

Cierre y vuelva a abrir RStudio. (si aparece una ventana con la pregunta Save workspace image to .RData? pulse el botón Don't Save)

Abra el archivo Actividad_1.RData (puede hacerlo directamente desde el bloque Environment

haciendo click en ícono de la carpeta 🥣



Usando la función *ls()* observe qué objetos aparecen en el entorno. ¿Está el objeto "meses"?



RESPUESTAS

Ejercicio 1- Usando R como calculadora

```
> 4+5
[1] 9
> 27-11
[1] 16
> 3*9
[1] 27
> 23/4
[1] 5.75
> 23%/%4
[1] 5
> 23%%4
[1] 3
> 2/3+4/7
[1] 1.238095
> sqrt(36)
[1] 6
> sqrt(79)
[1] 8.888194
> sin(pi)
[1] 1.224606e-16
> sin(3.14)
[1] 0.001592653
```

Ejercicio 2- Más cálculos

```
> sqrt(144) / (27^(1/3))
[1] 4
> round(abs((356-366)*pi),2)
[1] 31.42
```

Ejercicio 3 - Mi primer Script

```
> setwd("C:/CURSO_R_SADIO") ## elije el alumno su ruta
> getwd() ## verifico la ruta
[1] "C:/CURSO_R_SADIO"
```



```
> a <- 5

> b <- 4

> c <- a + b

> a <- b * c

> b <- (c - a)^2

> c <- a * b
```

Para saber cuánto vales a, b, y c

```
> a
[1] 36
> b
[1] 729
> C
[1] 26244
```

Ejercicio 4- Trabajando con vectores

i) Crea los objetos **x** e **y** ejecutando las siguientes líneas de código.

```
> x <- c(1,3,5,7,9)
> y <- c(2,3,5,7,11,13)
```

ii) Suma 1 a cada elemento del vector x

iii) Suma 1 en cada elemento del vector y

```
> y+1
[1] 3 4 6 8 12 14
```

iv) Al vector y multiplícalo por 3

```
> y*3
[1] 6 9 15 21 33 39
```

v) Suma la cantidad de elementos del vector x e y

```
> length(x) + length(y)
[1] 11
```

vi) Sumando los vectores

```
> x+y
[1] 3 6 10 14 20 14
```



```
Warning message:
In x + y: longer object length is not a multiple of shorter object length
```

Nota: Como la longitud de los vectores es diferente, nos devuelve un aviso de error. De todas maneras realiza igualmente la suma, pero al llegar al último elemento del objeto más largo (y) hace la suma con el primer elemento de x, es decir vuelve a iniciar el vector.

Ejercicio 5- Operaciones con vectores

```
    i) > 1:10
        [1] 1 2 3 4 5 6 7 8 9 10
    ii) > 10:1
        [1] 10 9 8 7 6 5 4 3 2 1
```

Nota: el vector comienza en el primer número suministrado y finaliza en el segundo o en un número anterior sin sobrepasarlo, tanto en orden ascendente como descendente)

```
iii) > seq(1,10,2)
[1] 1 3 5 7 9
> seq(from = 1, to = 10, length =6)
[1] 1.0 2.8 4.6 6.4 8.2 10.0
iv) > rep(5,10)
 [1] 5 5 5 5 5 5 5 5 5 5
> rep(1:3,rep(5,3))
 [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
v)
> meses<- c ( "Enero","Febrero","Marzo","Abril","mayo","Junio","Julio","Agos
to","Septiembre","Octubre","Noviembre","Diciembre")</pre>
> meses
 [1] "Enero"
[6] "Junio"
                     "Febrero"
                                     "Marzo"
                                                    "Abril"
                                                                    "mavo"
                     "Julio"
                                                    "Septiembre" "Octubre"
                                     "Agosto"
[11] "Noviembre"
                     "Diciembre"
```



```
vi) > str(meses)
chr [1:12] "Enero" "Febrero" "Marzo" "Abril" "mayo" "Junio" "Julio" ...

vii) > mode(meses)
[1] "character"
> length(meses)
[1] 12

Ejercicio 6
> save.image("C:/CURSO_R_SADIO/INICIACION EN EL LENGUAJE R/Actividad_1.RData ")
> rm(meses)
> ls()
[1] "a" "b" "c" "meses" "x" "y"
```

Sí, "meses" aparece en el listado porque quedó almacenado en el archivo RData antes de su eliminación.

Nota: Para borrar objetos almacenados en la memoria, utilizamos la función "rm()", por ejemplo rm(meses) elimina el objeto meses; rm(x; y) elimina ambos objetos x e y, con "rm(list = ls())" podemos eliminar todos los objetos que están en la memoria. Tenga en cuenta que las opciones mencionadas para la función ls() pueden aplicarse para borrar selectivamente algunos objetos.

