

Lab 01 - Introducción a R

Contents

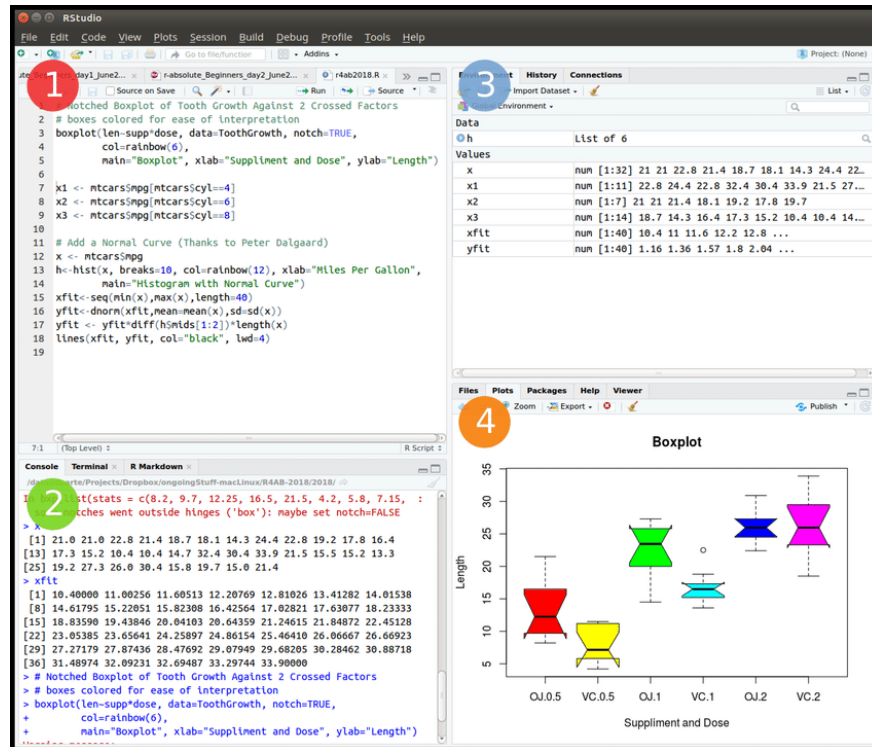
El lenguaje de programación R	1
Ejecutar comandos en R	2
R como calculadora	3
Tu turno	4
Asignando valores	4
Tipos de Variables	5
Tu turno	6
Funciones de redondeo	6
Ejercicios	6

El lenguaje de programación R

R es un **software libre** de programación especializado en estadística. RStudio es una interfaz que facilita la programación en R.

- R: lenguaje de comunicación
- RStudio: forma de comunicarse con máquina





En el siguiente enlace puede encontrarse información acerca de la instalación de R y RStudio en diferentes sistemas operativos.

<https://rstudio-education.github.io/hopr/starting.html>.

Ejecutar comandos en R

Nos comunicamos con R a través de la consola de comandos. Por ejemplo, con el siguiente código damos a R la orden de multiplicar 123 por 65.

```
123 * 65
```

```
## [1] 7995
```

La respuesta en consola aparece como

```
> 123 * 65
[1] 7995
```

Suele ser buena práctica explicar el código que se crea. Esto se hace a través de comentarios. Los comentarios se inician con el caracter `#`. De esta manera, R reconoce que lo que sucede a este caracter es un comentario y no lo ejecuta. Por ejemplo:

```
# Esto es un comentario
# Este código multiplica 100*12
100*12
```

```
## [1] 1200
```

R como calculadora

Podemos usar R como calculadora. Algunas operaciones básicas son:

```
# Suma  
2+2
```

```
## [1] 4
```

```
# Resta  
2-2
```

```
## [1] 0
```

```
# Multiplicación  
2*3
```

```
## [1] 6
```

```
# División  
2/3
```

```
## [1] 0.6666667
```

```
# Potencias  
2^3
```

```
## [1] 8
```

```
# Potencias (otra forma)  
2**3
```

```
## [1] 8
```

```
# Raíz cuadrada  
sqrt(9)
```

```
## [1] 3
```

```
# Logaritmo Natural  
log(100)
```

```
## [1] 4.60517
```

```
# Logaritmo en base 10  
log(100,10)
```

```
## [1] 2
```

```
# Exponencial  
exp(1)
```

```
## [1] 2.718282
```

```
# Valor Absoluto  
abs(-4)
```

```
## [1] 4
```

```
# Factorial  
factorial(3)
```

```
## [1] 6
```

```
# Combinaciones de 4 en 2  
choose(4,2)
```

```
## [1] 6
```

Ante operaciones de resultado indeterminado, R devuelve un NaN (not a number).

```
0/0
```

```
## [1] NaN
```

Tu turno

Escribe y ejecuta las siguientes operaciones aritméticas en R:

- Suma dos más 3
- Calcula 2 elevado a 100, ¿qué observas?
- Calcula el logaritmo en base dos de 1024

```
## Tu código aquí
```

Asignando valores

Si queremos un valor (o conjunto de valores) de forma recurrente hay que **guardarlos en memoria**, dándole un nombre.

```
# Esto crea un objeto de R con valor pi  
a <- pi  
# Sumo 3 al valor guardado  
a + 3
```

```
## [1] 6.141593
```

Puedes nombrar un objeto en R casi cualquier cosa que quieras, pero hay algunas reglas. Primero, un nombre no puede comenzar con un número. Segundo, un nombre no puede usar algunos símbolos especiales, como `^`, `!`, `$`, `@`, `+`, `-`, `/` o `*`. Tercero, evita espacios en los nombres: en lugar de `mi nombre` usa `mi_nombre`.

Podemos guardar múltiples valores

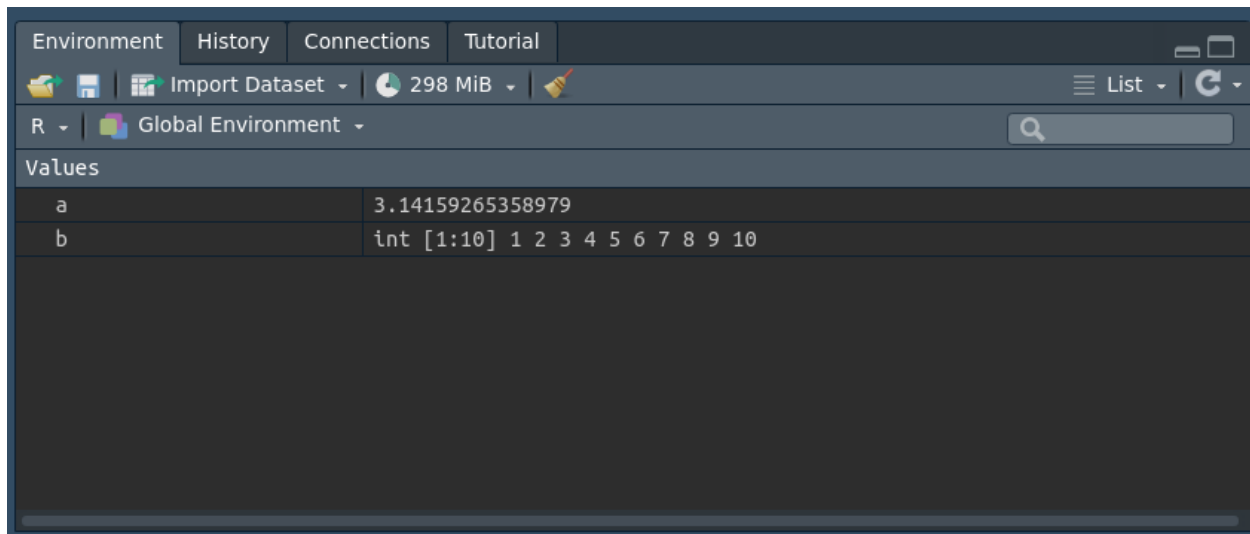
```
# Esto guarda los números del 1 al 5  
b <- 1:5  
b
```

```
## [1] 1 2 3 4 5
```

Podemos concatenar valores

```
# Esto hace lo mismo que el trozo anterior, pero de forma manual  
nums <- c(1,2,3,4,5)
```

Los objetos guardados aparecen en la ventana de *Environment* de RStudio



El nombre de un objeto no puede empezar con un número, ni tampoco incluir ciertos caracteres especiales como ^, \$, @, etc.

Importante: R sobre-escribe. Observa lo que sucede en este código.

```
c <- 5  
c
```

```
## [1] 5
```

```
c <- 20  
c
```

```
## [1] 20
```

Tipos de Variables

Los tipos más importantes son:

- Numéricas
- Caracteres. ¡Con comillas!
- Lógicas (toma valores TRUE o FALSE. También T o F)

Puedes comprobar el tipo de una variable con el comando `class(x)` o `typeof(x)`:

```
num <- 3  
class(num)
```

```
## [1] "numeric"
```

```
color <- "negro"  
class(color)
```

```
## [1] "character"
```

```
logica <- F  
class(logica)
```

```
## [1] "logical"
```

```
logica <- TRUE  
class(logica)
```

```
## [1] "logical"
```

Es posible cambiar el tipo de variables usando:

- `as.numeric(x)`
- `as.character(x)`
- `as.logical(x)`

Tu turno

- Crea una variable con el nombre que desees que almacene los valores “Negro”, “Blanco” y “Verde”. Determina el tipo de esta variable.
- Crea una variable con el nombre que desees que almacene los valores 3, 4 y “Verde”. Determina el tipo de esta variable. ¿Qué sucede?
- Crea una variable con el nombre que desees que almacene los valores TRUE, FALSE y “False”. Determina el tipo de esta variable. ¿Qué sucede?

```
## Tu código aquí
```

Funciones de redondeo

```
x <- 2.17
# Redondea con 1 dígito de precisión al número más cercano
round(x, digits=1)
```

```
## [1] 2.2
```

```
# Toma el entero inmediatamente anterior
floor(x)
```

```
## [1] 2
```

```
# Toma el entero inmediatamente superior
ceiling(x)
```

```
## [1] 3
```

Ejercicios

1. Ejecuta una expresión en R, por ejemplo `5 -`. ¿Qué sucede?
2. Divide 6 entre el resultado de restar 4 menos 1.
3. Realiza los siguientes pasos:
 - Escoge cualquier número y guárdalo en una variable llamada `x`
 - Suma dos a `x`
 - Multiplica el resultado anterior por 3
 - Resta 6 del resultado anterior
 - Divide lo que hayas obtenido en el paso anterior por 3

¿Qué obtienes?

4. Crea un script de R llamado `mi_script.R` donde guardes todos los pasos anteriores. Ejecuta el script.

5. Crea una variable llamada Numero que contenga el número 3 y otra llamada numero que también contenga el valor 6. Suma 7 a la variable Numero, ¿qué obtienes?
6. Utiliza la ayuda para determinar la utilidad de la función `ls()`. ¿Qué obtienes al ejecutarla?
7. Obtén el valor del número e.