

# Practica\_1\_R

Cristian Zaldaña

2022-07-28

## Índice

<b>CONSOLA DE R (R-CONSOLE)</b>	<b>2</b>
<b>TRABAJANDO CON SCRIPT</b>	<b>4</b>
<b>VECTORES DE DATOS</b>	<b>5</b>
VECTORES NUMÉRICOS . . . . .	5

## CONSOLA DE R (R-CONSOLE)

Ejemplo 1. Encontrar el resultado de operar: 2 más 10 por 3 entre 5 Escriba en la Consola de R:  $2+10*3/5$  y oprima la tecla ENTER

```
2+10*3/5
```

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

Note que en R se respecta el mismo orden de preferencia de la mayoría de los lenguajes de programación, la multiplicación y la división tienen prioridad a la suma y resta

Ejemplo 2. Encontrar el resultado de operar: 3 elevado a la potencia 100  $3^{100}$  o también `format(3^100, sci = FALSE)`

```
format(3^100, sci = FALSE)
```

```
## [1] "515377520732011324202202224420402268886864624842"
```

Sci=FALSE le indica a R que muestre todos los dígitos del resultado, de lo contrario (Sci=TRUE) solamente se mostrará la representación científica.

Ejemplo 3. Encontrar el resultado anterior con 15 cifras decimales y guardarlo en la variable y `y <- format(3^100, digits = 15);y` o `y = format(3^100, digits = 15)`

```
y=format(3^100, digits = 15)
y
```

```
## [1] "5.15377520732011e+47"
```

Note que en R, la asignación de valores a una variable puede hacerse con “=” o con “<-”.

Ejemplo 4. Redondear el valor de  $\pi$  a 4 dígitos decimales `round(pi, 4)` Aplique las funciones: `trunc(pi)`, `floor(pi)` y `ceiling(pi)`

```
round(pi, 4)
```

```
## [1] 3.1416
```

```
trunc(pi)
```

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

```
floor(pi)
```

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

```
ceiling(pi)
```

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

Ejemplo 5. Guardar en la variable `n` el valor 150 y luego calcular el valor de `n` `n = 150 factorial(n)`

```
n=150
```

```
factorial(n)
```

```
## [1] 5.713384e+262
```

Ejemplo 6. Operar el complejo  $(2+3i)$  elevado a la potencia 10  $(2+3i)^{10}$  o también `format((2+3i)^10, sci = TRUE)`

```
(2+3i)^10
```

```
## [1] -341525-145668i
```

Ejemplo 7. Calcular la integral entre 0 y  $\pi$  de la función Seno(x) `f <- function(x) {sin(x)}` `integrate(f, lower = 0, upper = pi)`

```
f=function(x) {sin(x)}
```

```
integrate(f,lower = 0,upper = pi)
```

```
## 2 with absolute error < 2.2e-14
```

## TRABAJANDO CON SCRIPT

A medida que estemos realizando un trabajo de pequeña, mediana o de gran complejidad, será muy útil manejar todas las entradas que solicitemos a R en un entorno donde podamos corregirlas, retocarlas, repetirlas, guardarlas para continuar el trabajo en otro momento, con otros datos, etc. Esta es la función del editor de R, a los archivos creados en este editor se les conoce como Script. Es posible incluir comentarios que R no leerá si utilizamos líneas que comiencen con el carácter `#` (o en cualquier parte de la línea). Por el contrario, si escribimos cualquier orden no antecedida de `#` R (sin importar en que parte se encuentre) lo reconocerá como instrucciones que deben ejecutarse.

Este archivo se enviara por a parte...

## VECTORES DE DATOS

Este tipo de objetos se denominan estructuras atómicas ya que todos sus elementos son del mismo tipo o modo: character (carácter) o numeric (numérico) que puede ser integer (entero), double (real), complex (complejo), logical (lógico).

### VECTORES NUMÉRICOS

FORMA 1-Crear un vector numérico vacío y añadirle luego sus elementos.

Ejemplo 1: `v <- numeric(3);v`

```
v <- numeric(3)
v
```

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

El vector tiene longitud 3 y sus componentes serán NA (Not Available/“Missing” Values) que es la forma como R maneja los datos omitidos o faltantes.

Ejemplo 2: `v[3] <- 17; v`

```
v[3] <- 17
v
```

```
## [1] 0 0 17
```

asigna el valor de 17 en la tercera posición del vector v.

FORMA 2-Crear un vector numérico asignándole todos sus elementos o valores.

Ejemplo 1: `x <- c(2, 4, 3.1, 8, 6)` , revise el modo con `is.integer(x)` y `is.double(x)`; encuentre la longitud con: `length(x)`

```
x<- c(2,4,3.1,8,6)
is.integer(x)
```

```
## [1] FALSE
```

```
is.double(x)
```

```
## [1] TRUE
```

```
length(x)
```

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

Ejemplo 2: Modifique el vector agregándole el valor 9 en la posición 3, use la siguiente la función de edición: `x <- edit(x)`

```
edit(x)
```

```
## [1] 2.0 4.0 3.1 8.0 6.0
```

FORMA 3-Crear un vector numérico dando un rango de valores.

Ejemplo 1:  $y = 1:4$ ; `y`

```
y = 1:4  
y
```

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

Crea un vector de valores enteros en que su primer elemento es 1 su último es 4

Ejemplo 2: Modificación de los elementos de un vector: `y[2] <- 5` (para modificar un elemento de un vector se escribe su nombre (del vector) y entre corchetes el índice del elemento que se quiera modificar).

```
y[2] <- 5  
y
```

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

Ejemplo 3: Crear un vector con elementos de otro; `u <- 1:12; u1=u[2 * 1:5]` (vector de tamaño 5 con elementos de las posiciones pares de `u`)

```
u <- 1:12  
u1=u[2*1:5]  
u1
```

```
## [1] 2 4 6 8 10
```