



# Introducción a la sintaxis de R

Christian Salas Eljatib, Ph.D.

**E-mail:** [cseljatib@gmail.com](mailto:cseljatib@gmail.com)

**Web:** <https://eljatib.com>

4 de enero de 2023  
Santiago, Chile

# Contenidos

- 1 Calculadora
- 2 Funciones
  - Estructura de una función
  - Funciones de distribución de probabilidad
  - Funciones matemáticas
  - Anidado de funciones
  - Otras funciones útiles
- 3 Llevando un registro
  - Historial de comandos
  - Espacio de trabajo
  - *Script*
- 4 Buscando ayuda sobre R

# R como una calculadora

## R como una calculadora

```
> 1+1
```

```
[1] 2
```

```
> 100/10
```

```
[1] 10
```

```
> 1*4
```

```
[1] 4
```

```
> 1-3
```

```
[1] -2
```

# Agregando un comentario

```
> 110+45      #suma
[1] 155
> 35/3        #division
[1] 11.667
> 0.4*7       #multiplicacion
[1] 2.8
> .4*7        #da el mismo resultado que lo anterior
[1] 2.8
```

¿Qué pasa cuando cometo un error? Compare el resultado de lo siguiente

```
> 0.002*23 #esta sintaxis esta bien
> 0,002*23 #esta sintaxis no esta bien
```

# Estructura de una función computacional

## Estructura de una función en R

```
nombreFx(objeto, opciones)
```

Donde:

- `nombreFx` es el nombre de la función.
- `objeto` es el nombre del objeto al cual se le aplicará la función.
- `opciones` son especificaciones que se necesitan para aplicar la función.

# Funciones de distribución de probabilidad

## La función rnorm

```
> rnorm(1)
[1] -0.71967
> rnorm(2)
[1] -0.21620 -0.27613
> rnorm(1, mean=10, sd=20)
[1] 17.001
```

## La función runif

```
> runif(1)
[1] 0.72564
> runif(1, min=-10, max=10)
[1] -8.766
> runif(3, min=-10, max=10)
[1] -5.8180  9.5592  2.6909
```

# Funciones matemáticas

Expresión	Función en R
$\ln(y)$	<code>log()</code>
$\log(y)$	<code>log10()</code>
$y^2$	
$e^y$	<code>exp()</code>
$\sqrt{y}$	<code>sqrt()</code>
$ y $	<code>abs()</code>
$\pi$	<code>pi</code>

## Ejemplos

```
> log(3)
[1] 1.0986
> log10(4)
[1] 0.60206
> 7^2
[1] 49
> exp(-0.2)
[1] 0.81873
> sqrt(64)
[1] 8
> abs(-10)
[1] 10
> pi/40000
[1] 7.854e-05
```



# R permite anidar funciones

Por ejemplo, la expresión matemática

$$\sqrt{\ln 5} \quad (1)$$

se expresa como

```
> sqrt(log(5))  
[1] 1.2686
```

Otra característica importante de R, es que no es sensible a los espacios (esto es una ventaja), ya que la expresión anterior **también se puede escribir como**

```
> sqrt(   log   (   5   )   )  
[1] 1.2686
```

**o incluso como**

```
> sqrt(  
      log(5)  
      )  
[1] 1.2686
```

# Otro ejemplo I

La expresión matemática

$$\frac{1}{e^{\sqrt{\ln|-10,4^{1,2}|}}} \quad (2)$$

¿Cómo la escribiría Ud en R?

## Otro ejemplo II

una alternativa, es escribirla como

```
> 1/ ( exp( sqrt( log( abs(-10.4^1.2) ) ) ) ) )  
[1] 0.18705
```

## Otro ejemplo III

y otra alternativa podría ser

```
> 1/( exp(
      sqrt(
        log(
          abs(-10.4^1.2)
        )
      )
    )
  )
[1] 0.18705
```

# Funciones trigonométricas

- Las funciones trigonométricas en R se basan en radianes, no en ángulos
- Se debe transformar desde ángulos a radianes, e.g., un ángulo  $\alpha$  medido en grados, se transforma a radianes mediante la siguiente fórmula

$$\alpha \times \frac{\pi}{180} \quad (3)$$

Expresión	Función en R
$\cos(\alpha)$	<code>cos()</code>
$\sin(\alpha)$	<code>sin()</code>
$\tan(\alpha)$	<code>tan()</code>
$\cos^{-1}(\alpha)$	<code>acos()</code>
$\sin^{-1}(\alpha)$	<code>asin()</code>
$\tan^{-1}(\alpha)$	<code>atan()</code>

## Ejemplos

```
> cos(90*(pi/180))  
[1] 6.1232e-17  
> sin(0*(pi/180))  
[1] 0  
> tan(45*(pi/180))  
[1] 1
```

# Función seq |

## La función seq()

```
> seq(0, 3) #serie de numeros, 0-1
```

```
[1] 0 1 2 3
```

```
> seq(10, 11, by=0.2) #serie de numeros, 10-11, cada 0.1
```

```
[1] 10.0 10.2 10.4 10.6 10.8 11.0
```

# Función seq II

Otra forma de obtener el mismo resultado que produce la sintaxis `seq(0,3)`  
compare con

```
> 0:3
```

```
[1] 0 1 2 3
```

# Función `sample()`

## La función `sample()`

```
> sample(seq(1:1000),3) #selecciona 3 de 1000
[1] 869 384 223
> sample(1:6,5,replace=TRUE)#tiremos los dados 5 veces
[1] 3 6 1 1 1
```



# Historial de comandos

## Guardando el historial de comandos

```
> savehistory(file = "misComandos.Rhistory")
```

```
> loadhistory(file = "misComandos.Rhistory")
```

- Ojo: guarda todo, incluso sintaxis errónea

# Espacio de trabajo

## Guardando el espacio de trabajo

```
> save.image("miEspTbjo.Rdata")
```

```
> load("miEspTbjo.Rdata")
```

- Graba comandos y objetos
- Ideal para procesos largos y que involucran varios análisis. (e.g., ajuste de modelos, simulaciones, etc.)

# Archivo con sintaxis de análisis (*Script*)

- Documentar toda la sintaxis empleada para un análisis desarrollado.
- Archivo de texto simple con los comandos, sintaxis y comentarios empleados, el cual posteriormente se abre y corre dentro de R.
- El script debiera tener la extensión **.R**
- Tiene la gran virtud de poder reproducir íntegramente los resultados obtenidos (y/o publicados).

## Recomendaciones

- Evite ocupar tildes y mayúsculas.
- No emplear espacios para nombrar archivos.

# dentro de R y journals

## Estando en una sesión de R

```
> ?rnorm  
> help(runif)
```

## Journals

- *The R Journal*

<http://journal.r-project.org/>

- *Journal of Computational and Graphical Statistics*

<https://amstat.tandfonline.com/loi/ucgs20>

# en Internet

## Internet: buscador

[R] random number generator

[R] regression

## Internet: *Mailing list* y similares

Por ejemplo

<https://stat.ethz.ch/mailman/listinfo/r-help>

<https://stackoverflow.com/questions/tagged/r>

<https://rseek.org>