

COMANDOS Y HERRAMIENTAS ÚTILES DE R

NIVEL FÁCIL



<https://www.google.com/url?sa=i&source=images&cd=&ved=2ahUKewiE4fbRl:jmAhVHzYUKHY77BpwQjRx6BAg8FEAQ&url=https%3A%2F%2Fblog.dinahosting.com%2Flos-10-lenguajes-de-programacion-mas-usados%2F&psig=AOvVaw2v5c2bUPAXjmUvKZY5oiSL&ust=1576517247162169>

Este documento ha sido realizado y revisado por alumnos de primero del grado de Biotecnología de la Universidad Politécnica de Madrid. Se basa en los contenidos de la asignatura de Fundamentos de Programación, impartida por Arturo Hidalgo y Ángel Fidalgo.

El lenguaje de programación utilizado es R, enfocado en el análisis estadístico y muy utilizado en investigación científica y biomédica, bioinformática, aprendizaje automático (machine learning), minería de datos, y matemáticas financieras.

El documento está formado por una explicación de los comandos y herramientas más útiles de R, con imágenes que los ejemplifican. Por lo que su finalidad es la de servir como referencia a próximos alumnos y a personas con interés con los campos citados, facilitando el aprendizaje de este lenguaje.

¿Cómo almacenamos un valor en una variable?

Elegimos el nombre que le queremos dar a la variable y añadiendo o un igual o una flecha dirigida a la variable, escribimos el valor deseado.

```
> a=2      > a<-2
> a        > a
[1] 2      [1] 2
```

Si deseamos observar el valor de la variable, utilizamos el comando **print(n)** siendo n el nombre de la variable.

```
> print(a)
[1] 2
```

¿Cómo generamos datos al azar?

Utilizamos el comando **rnorm(n)** siendo n el número de valores que queremos.

```
> rnorm(3)
[1] -0.5347761  0.9441476 -0.5449957
> rnorm(5)
[1]  1.7734831 -2.0085725 -1.6738059  0.5851164  1.4962717
```

Si lo queremos generar en un intervalo de [0,1], utilizamos **runif(n)** siendo n el número de valores que queremos.

```
> runif(2)
[1] 0.6481829 0.3126400
> runif(4)
[1] 0.8822806 0.8496863 0.5965254 0.4482463
```

¿Cómo saber que variables tenemos almacenadas?

Utilizando el comando **ls()** y con **ls.str()** nos especifica que valores hay dentro.

```
> ls()
[1] "a"
> ls.str()
a : num 2
```

Si queremos buscar un elemento que contenga una letra en particular utilizamos **ls(pat="x")** siendo x la letra que buscamos., mientras que si queremos buscar un elemento que empiece por una letra en particular utilizamos **ls(pat="^x")** siendo x la letra por la que empieza el elemento. (**NOTA:** en el caso de no haber ningún elemento, aparecerá `character(0)`)

```
> altura=7
> poblacion=9
> ls(pat="a")
[1] "a"          "altura"     "poblacion"
> ls(pat="^a")
[1] "a"          "altura"
```

¿Cómo borramos elementos de la memoria?

Utilizamos el comando **rm(n)** siendo n el elemento que se desea borrar.

```
> rm(a)
> a
Error: objeto 'a' no encontrado
```

OPERACIONES ARITMÉTICAS

En R podemos realizar operaciones como: suma, resta, multiplicación, división y elevar una potencia mediante operadores.

¿Cómo sumar?

Asignamos dos valores a dos variables y luego utilizando el operador **suma(+)** , sumamos ambos valores y se imprimirá el resultado.

```
> a=5
> b=7
> a+b
[1] 12
```

¿Cómo restar?

Asignamos dos valores a dos variables y luego utilizando el operador **resta(-)** , restamos ambos valores y se imprimirá el resultado.

```
> a=8
> b=2
> a-b
[1] 6
```

¿Cómo multiplicar?

Asignamos dos números a dos variables y utilizando el operador **multiplicación(*)**, multiplicamos ambos valores y se imprimirá el resultado en la consola.

```
> a=5
> b=2
> a*b
[1] 10
```

¿Cómo dividir?

Asignamos dos números a dos variables y utilizando el operador **división(/)**, dividimos ambos valores y se imprimirá el resultado en la consola.

```
> a=20
> b=10
> a/b
[1] 2
```

¿Cómo elevar a una potencia?

Asignamos dos números a dos variables y utilizando el operador potencia (**), se elevarán los valores y se imprimirá el resultado en la consola.

```
> a=2
> b=8
> a**b
[1] 256
```

Cuando hay varias operaciones aritméticas el orden en el que el ordenador las ejecuta es:

1. Potencias
2. Productos y divisiones
3. Sumas y restas

Pero, el orden en que se ejecutan las operaciones se puede modificar empleando paréntesis.