

# Funciones

Arturo Pérez

## 1. Características generales de una función.

Las funciones representan uno de los aspectos más poderosos en programación, se trata de la transición entre usuario y programador.

Las funciones se crea utilizando el comando `function()` y se compone de las siguientes partes:

1. Nombre de la función
2. Argumentos
3. Cuerpo de la función

```
x <- function(<argumentos>){  
  ## Cuerpo de la función  
}
```

```
soyunafuncion <- function(x,y){  
  x+y  
}  
  
soyunafuncion <- function(x,y=10){  
  result <- x+y  
  print(result)  
}
```

Las funciones son objetos de primera clase y pertenecen a la clase de objetos “function”, nosotros podemos diseñar las funciones para que puedan hacer lo que nosotros queramos.

Pueden ser pasadas como argumentos a otras funciones

Pueden anidarse, es decir, meter una función dentro de otra función.

El valor regresado de una función es la última expresión en el cuerpo del código de la misma.

### Argumentos

**Argumentos formales** Son los argumentos incluidos cuando definimos una función.

Las funciones tienen varios argumentos, dentro de los cuales podemos encontrar argumentos nombrados con valores por default. No todas las funciones hacen uso de todos sus argumentos formales. Por ejemplo, una función con 10 argumentos puede tener valores especificados o nosotros debemos especificarlos (forzosamente, si no la función no hace nada). Para ver los argumentos de una función podemos hacer uso de los comandos `str()` o `formals()`

```
formals(rnorm)
```

```
## $n
##
##
## $mean
## [1] 0
##
## $sd
## [1] 1
```

```
str(rnorm)
```

```
## function (n, mean = 0, sd = 1)
```

**argument matching** Los argumentos de las funciones pueden unirse por su posición o por el nombre, por ejemplo.

```
## por posición
set.seed(1)
rnorm(5, 3, 2)
```

```
## [1] 1.747092 3.367287 1.328743 6.190562 3.659016
```

```
##por nombre
set.seed(1)
rnorm(n=5, mean = 3, sd = 2)
```

```
## [1] 1.747092 3.367287 1.328743 6.190562 3.659016
```

```
##No recomendado invertir el orden de los args.
set.seed(1)
rnorm(mean = 3, 5, 2)
```

```
## [1] 1.747092 3.367287 1.328743 6.190562 3.659016
```

Cuando tenemos muchos argumentos, podemos mezclar el posicionamiento por nombre y por posición.

```
##equivalentes
lm(data = mydata, x~y, model = FALSE, 1:100)
lm(x~y, mydata, 1:100, model = FALSE) ##personalmente me gusta más este.
```

No es necesario especificar todo el tiempo, a menos que sea necesario o nos sea más cómodo. Por lo general, queremos nombrar los argumentos cuando tenemos una gran lista de estos y queremos modificar su valor por default a otro, además es útil escribirlos si sabes el nombre de los argumentos, pero no su posición en la lista.

**Matching parcial** Cuando especificamos una parte del argumento, R une y le asigna el valor correcto

```
set.seed(1)
rnorm(10, m =4, 1)
```

```
## [1] 3.373546 4.183643 3.164371 5.595281 4.329508 3.179532 4.487429 4.738325
## [9] 4.575781 3.694612
```

Entonces tenemos 3 tipos de match para los argumentos

- 1) Match exacto (por nombre)
- 2) Match parcial (por nombre, pero sin mencionar  
    Todo el argumento)
- 3) Match por posición