

Creación de funciones en R

La estructura general de una función en R es la siguiente:

nombre = function(argumento1 , argumento2,) comandos

Por ejemplo, podemos definir una función que calcule la desviación típica:

```
> desv = function(x){sqrt(var(x))} # Definimos la función
> x<-1:10                          # Generamos datos
> desv(x)                          # Utilizamos la función
[1] 3.027650
> sd(x)                            # La definida en R coincide con la nuestra
[1] 3.027650
```

Una vez definida una función, se la puede llamar y utilizar como a cualquiera otra función predefinida en el sistema. Por ejemplo, vamos a utilizar la función apply combinada con desv para calcular las desviaciones típicas de las columnas de una matriz:

```
> x = matrix(rnorm(15),nrow=3)
> x
      [,1] [,2] [,3] [,4] [,5]
[1,] 0.1578703 1.6712974 -0.5419452 0.03345786 -0.6675674
[2,] 0.3215741 -0.6352143 -1.0222260 0.39006069 0.3609624
[3,] 0.4770036 -0.3508383 -0.5147970 1.36219826 -1.6669992
> apply(x,2,desv)
[1] 0.1595845 1.2576365 0.2854502 0.6877219 1.0140156
```

(<http://ocw.uc3m.es/estadistica/aprendizaje-del-software-estadistico-r-un-entorno-para-simulacion-y-computacion-estadistica/algunas-estructuras-de-programacion-creacion-de-funciones-en-r>)

Como usar funciones en r

Funciones y expresiones.

La gran utilidad de las expresiones es que nos permiten ejecutar varios comandos de una única vez. Pero donde gana mayor utilidad esta forma de trabajar es a la hora de crear nuevos objetos que ejecuten diversas expresiones utilizando como entrada unos objetos (argumentos) y devolviendo otros objetos. Estos objetos (cuyo modo es function) constituyen las nuevas funciones de R, que se pueden utilizar a su vez en expresiones posteriores. En este proceso, el lenguaje gana en potencia, comodidad y elegancia. Muchas funciones del lenguaje R están escritas en código interno, otras utilizan conexiones a C, Fortran, etc. Pero otras muchas, como mean o var, están de hecho escritas en R y, por tanto, no dieran materialmente de las funciones que nosotros podamos escribir. Aprender a escribir funciones que nos puedan ser de futura utilidad es una de las mejores formas de conseguir que el uso de R nos sea cómodo y productivo.

2.- Ejecución condicional: if y else.

El lenguaje R tiene la posibilidad de ejecutar expresiones condicionalmente:

```
if (expre1) expre2 else expre3
```

Si expre1=TRUE calcula expre2

Si expre1=FALSE calcula expre3

Ejemplo

```
if (10>3) cat("SI 10>3 \n") else cat("NO 10>3 \n")
```

En la primera expresión podemos incluir varios requerimientos utilizando los operadores lógicos presentados en el tema 2.

Ejemplo

```
x<-0
if (is.numeric(x)&min(x)>0) rax<-sqrt(x) else stop("x debe
ser numérico y positivo \n")
```

Ejecución repetitiva: for y while.

```
for (name in values) expre
```

La expre es evaluada asignado a name sucesivamente cada uno de los elementos de values.

Recordar que si la expre tiene más de un comando va entre llaves.

Ejemplo

```
for (i in 1:5) cat("caso ",i,"\n")
```

```
while (condi) expre
```

La expre es evaluada mientras la condi sea cierta.

Recordar que si la expre tiene más de un comando va entre llaves.

Ejemplo

```
i<-5
while (i >0) {cat("caso ",i,"\n"); i<-i-1}
```