

Estructuras de programación

R permite crear estructuras repetitivas (loops) y la ejecución condicional de sentencias. A este fin, los comandos pueden agruparse entre llaves, utilizando la siguiente sintaxis:

{comando1 ; comando2; comando3 ;}

○ *El bucle for*

Para crear un bucle repetitivo (un bucle for), la sintaxis es la siguiente:

for (i in listadevalores) { secuencia de comandos }

Por ejemplo:

```
> for(i in 1:10) { print(i)}
```

```
[1] 1  
[1] 2  
[1] 3  
[1] 4  
[1] 5  
[1] 6  
[1] 7  
[1] 8  
[1] 9  
[1] 10
```

○ *El bucle while*

La sintaxis es como sigue:

while (condicion logica) { expresiones a ejecutar }

Por ejemplo, si queremos calcular qué número es el mayor cuyo cuadrado no excede de 1000, podemos hacer:

```

> cuadrado = 0
> while(cuadrado<=1000)
+ {
+ n<-n+1
+ cuadrado<-n^2
+ }
> cuadrado
[1] 1024
> n
[1] 32
> 32^2
[1] 1024

```

¿Qué ha sucedido? El cuadrado de 32 excede 1000. En realidad, cuando n valía 31, su cuadrado (961) no excedía 1000, y el while () permitió entrar en el bucle, lo que hizo n=32. El número correcto sería en este caso $n-1 = 31$.

○ *Ejecución condicional: if*

La sintaxis general es:

if (condicion) comando1 else comando2

Por ejemplo, vamos a crear dos listas; una para guardar los números pares de 1 a 10, y otra para los impares:

```

> n = 10      # Se inicializa n
> pares = c() # Se crea un vector vacío
> impares = c() # Idem
> for(i in 1:n){ # Se van a procesar los números de 1 a n
+ if(i%%2==0) pares<-c(pares,i) # Si al dividir por 2 sale 0
+ else impares<-c(impares,i)} # el número es par, impar en otro caso
> pares
[1] 2 4 6 8 10
> impares
[1] 1 3 5 7 9

```

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>