Bucles y estructuras condicionales

Los bucles permiten repetir un mismo bloque de código todas las veces necesarias sin tener que escribirlo más de una vez y las estructuras condicionales permiten ejecutar distintos bloques de código dependiendo del cumplimiento de una condición. Su uso hace que el código ya no siga una estructura puramente lineal.

Bucles *for*

Los bucles *for* repiten un bloque de código un determinado número de veces en los que una variable toma distintos valores (números enteros). Los valores de la variable se indican entre paréntesis y el código que se ejecuta cada vuelta entre llaves. La estructura de estos bucles es:

```
for (i in n:m){
   operaciones
}
```

n:m→ la i toma valores de n a m con intervalos de 1

La i se puede usar como variable dentro del bucle y su valor es el que se le asigna cada vuelta.
Ejemplo (suma de todas las componentes de un vector):

bucle *for*

```
> v=c(1,2,3,4,5,6,7,8,9,10)
> S=0
> for (i in 1:length(v)){
+    S=S+v[i]
+ }
> S
[1] 55
```

-la función length() mide el número de componentes de un vector, en este caso 10
-hay que inicializar el sumatorio a 0
-en este caso la i se utiliza para indicar las distintas posiciones del vector

Bucles *while*

Los bucles *while* repiten un bloque de código mientras se cumpla una determinada condición. Esta condición se indica entre paréntesis y el código que se ejecuta cada vuelta entre llaves. La estructura de este tipo de bucles es pues:

```
while (condición){
   operaciones
}
```

La condición es la comparación de una determinada variable con un valor concreto, puede ser de igualdad, desigualdad, etc.
El código del bucle debe hacer que el valor de esta variable cambie de forma que llegado un punto deje de cumplir la condición ya que si no es así el bucle sería infinito y R dejaría de funcionar.

Ejemplo (suma de números naturales hasta superar el 100):

bucle *while*

```
> S=0
> i=0
> while (S<=100){
+   i=i+1
+   S=S+i
+ }
> S
[1] 105
> i
[1] 14
```

-la i es la que va tomando el valor de los distintos números naturales cada vuelta del bucle
-al superar la S el valor de 100 el bucle se detiene

símbolos

< → menor
> → mayor
<= → menor o igual
>= → mayor o igual
!= → distinto
== → igual

Estructuras condicionales

Las estructuras condicionales permiten ejecutar un bloque de código únicamente si se cumpla una determinada condición. Hay distintos tipos de estructuras condicionales pero en todas ellas la condición se escribe entre paréntesis y el código entre llaves. La estructura más sencilla es:

```
if (condición){
   operaciones
}
```

También se puede añadir código para que se ejecute si (y solo si) no se cumple la condición usando else:

```
if (condición){
   operaciones
} else {
   operaciones
}
```

Se pueden incluir distintas condiciones que ejecutan distinto código en una misma estructura usando else if:

```
if (condición 1){
   operaciones
} else if (condición 2){
   operaciones
} else if (condición 3){
   operaciones
}
.
.
.
else {
   operaciones
}
```

Ejemplo (asignación de categoría a un valor de IMC):

estructura condicional

```
> M=23
> if (M<=15){
+   print('Delgadez muy severa')
+ } else if (M<=15.9){
+   print('Delgadez severa')
+ } else if (M<=18.4){
+   print('Delgadez')
+ } else if (M<=24.9){
+   print('Peso saludable')
+ } else if (M<=29.9){
+   print('Sobrepeso')
+ } else if (M<=34.9){
+   print('Obesidad moderada')
+ } else if (M<=39.9){
+   print('Obesidad severa')
+ } else {
+   print('Obesidad muy severa (obesidad morbida)')
+ }
[1] "Peso saludable"
```

-la estructura condicional sitúa al valor en un intervalo e indica a que categoría pertenece
-el comando print() devuelve el texto que se escriba dentro o, en caso de que sea una variable, su valor