Guía de Estructuras de Control en R

Larios Ponce Hector Manuel

22-07-2023

Introducción

En R, las estructuras de control son fundamentales para dirigir el flujo de un programa. Estas incluyen condicionales (`if`, `else if`, `else`), bucles (`for`, `while`), y declaraciones de control (`break`, `next`). Esta guía te mostrará cómo utilizarlas con ejemplos prácticos.

Condicionales

`if` y `else`

La estructura `if` se utiliza para ejecutar código basado en una condición. El `else` se utiliza para ejecutar código si la condición del `if` es falsa.

```
x <- 10

if (x > 5) {
  print("x es mayor que 5")
} else {
  print("x no es mayor que 5")
}
```

```
## [1] "x es mayor que 5"
```

`else if`

La estructura `else if` se utiliza para evaluar múltiples condiciones.

```
x <- 10

if (x > 15) {
  print("x es mayor que 15")
} else if (x > 5) {
  print("x es mayor que 5 pero menor o igual a 15")
} else {
  print("x es menor o igual a 5")
}
```

```
## [1] "x es mayor que 5 pero menor o igual a 15"
```

Bucles

Los bucles se utilizan para repetir una tarea varias veces. Dependiendo de si sabemos de antemano cuántas veces queremos repetir la tarea, podemos utilizar un bucle `for` o un bucle `while`.

Si sabemos cuántas veces queremos repetir la tarea, utilizamos un bucle `for`. Por ejemplo, si queremos imprimir los números del 1 al 5, podemos hacerlo de la siguiente manera:

```
for (i in 1:5) {
  # 1:5 indica el rango de valores que tomara i en cada repetición
  print(i)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

Si no sabemos cuántas veces queremos repetir la tarea, utilizamos un bucle `while`. Por ejemplo, si queremos imprimir x * y mientras que x * y sea menor o igual que 100, podemos hacerlo de la siguiente manera:

```
x <- 3
y <- 8

while (x * y <= 100) {
  print(x * y)
  if (x * y > 100) {
    break
  }
  x <- x + 1
}
```

```
## [1] 24
## [1] 32
## [1] 40
## [1] 48
## [1] 56
## [1] 64
## [1] 72
## [1] 80
## [1] 88
## [1] 96
```

Iterar sobre un vector

Ya sabemos como repetir una tarea varias veces, pero ¿qué pasa si queremos repetir una tarea para cada elemento de un vector? En este caso, podemos utilizar un bucle `for` para iterar sobre los elementos del vector.

Existen al menos dos formas de hacer esto:

1. Iterar sobre los índices del vector:

```r
vec <- c(1, 2, 3, 4, 5)

for (i in seq_along(vec)) {
  print(vec[i])
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

2. Iterar sobre los elementos del vector:

```r
for (elemento in vec) {
  print(elemento)
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
```

Iterar sobre una matríz de vectores

Si queremos iterar sobre una matriz de vectores, podemos utilizar un bucle `for` anidado. Por ejemplo, si tenemos la siguiente matriz de vectores:

```r
matriz <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3, byrow = TRUE)
```

Podemos iterar sobre cada vector de la matriz de la siguiente manera:

```r
for (i in seq_len(nrow(matriz))) {
  for (j in seq_len(ncol(matriz))) {
    cat(matriz[i, j], " ")
  }
  cat("\n")
}
```

```
## 1  2  3
## 4  5  6
```

Alternativamente, podemos utilizar la función `apply` para aplicar una función a cada fila o columna de la matriz:

```r
apply(matriz, 2, print) # Aplica la función print a cada fila
```

```
## [1] 1 4
## [1] 2 5
## [1] 3 6
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6
```