

Fundamentos de ciencia de datos con R - Módulo 1

Clase 5: Programación básica en R: if, while, for

CEPAL - Unidad de Estadísticas Sociales

2025-10-29

Introducción

La programación es la base de la automatización en ciencia de datos.

En R, permite **tomar decisiones, repetir tareas y controlar el flujo lógico del análisis**, haciendo los procesos más eficientes y reproducibles.

A través de estructuras como `if`, `while` y `for`, podemos enseñar al programa **qué hacer, cuándo hacerlo y cuántas veces hacerlo**.

Estas herramientas son esenciales para pasar de ejecutar comandos sueltos a escribir scripts dinámicos y escalables.

¿Qué es la programación estructurada?

La programación estructurada organiza el código en bloques lógicos que **siguen un flujo claro de ejecución**.

Este enfoque se basa en tres principios:

- ▶ **Secuencia:** ejecutar instrucciones en orden.
- ▶ **Selección:** tomar decisiones con `if`.
- ▶ **Iteración:** repetir acciones con `while` o `for`.

¿Qué es la programación estructurada?

i Ejemplo de flujo lógico

Si la base de datos tiene valores faltantes → mostrar advertencia.

Mientras existan valores faltantes → imputarlos.

Para cada variable → calcular promedio y desviación estándar.

Estructura condicional – if y else

Permite **tomar decisiones** en función de una condición lógica.

Sintaxis básica

```
if (condición) {  
    # código si la condición es verdadera  
} else {  
    # código si la condición es falsa
```

Estructura condicional – if y else

Ejemplo

```
x <- 10

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

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

Estructura condicional – if y else

Nota

El bloque if evalúa una condición lógica (TRUE/FALSE). Si se cumple (TRUE), ejecuta el código; si no, pasa al bloque else.

Estructura repetitiva – while

El bucle while ejecuta un bloque de código mientras una condición sea verdadera.

```
while (condición) {  
    # instrucciones a repetir  
}
```


Estructura repetitiva – while

Ejemplo

```
contador <- 1

while (contador <= 5) {
  print(paste("Iteración", contador))
  contador <- contador + 1
}
```

```
[1] "Iteración 1"
[1] "Iteración 2"
[1] "Iteración 3"
[1] "Iteración 4"
[1] "Iteración 5"
```

Estructura repetitiva – while



Advertencia

Si la condición nunca cambia a FALSE, el bucle se vuelve infinito. Siempre asegúrate de actualizar la variable dentro del while.

Estructura repetitiva – for

El bucle for se usa cuando sabemos cuántas veces queremos repetir algo.

Sintaxis

```
for (variable in secuencia) {  
    # código a ejecutar  
}
```

Estructura repetitiva – for

Ejemplo

```
for (i in 1:5) {  
  print(paste("Iteración número", i))  
}
```

```
[1] "Iteración número 1"  
[1] "Iteración número 2"  
[1] "Iteración número 3"  
[1] "Iteración número 4"  
[1] "Iteración número 5"
```

Estructura repetitiva – for

Nota

La variable `i` cambia automáticamente en cada ciclo. Podemos recorrer vectores, listas o secuencias numéricas.

Combinando estructuras

Podemos mezclar bucles y condiciones para generar rutinas más inteligentes.

```
for (i in 1:6) {  
  if (i %% 2 == 0) {  
    print(paste(i, "es par"))  
  } else {  
    print(paste(i, "es impar"))  
  }  
}
```

```
[1] "1 es impar"  
[1] "2 es par"  
[1] "3 es impar"  
[1] "4 es par"  
[1] "5 es impar"  
[1] "6 es par"
```

Este ejemplo recorre los números del 1 al 6 e imprime si cada uno es par o impar.

Control adicional en bucles

R permite controlar los bucles con dos comandos:

Comando	Descripción	Ejemplo
<code>break</code>	Detiene el bucle inmediatamente	<code>if (i == 5) break</code>
<code>next</code>	Salta a la siguiente iteración	<code>if (i == 3) next</code>

Control adicional en bucles

Ejemplo

```
for (i in 1:5) {  
  if (i == 3) next  
  print(i)  
}
```

[1] 1

[1] 2

[1] 4

[1] 5

Resultado: 1, 2, 4, 5 (salta el 3).

Nota

Usa next para omitir casos específicos y break para detener el bucle por completo.

Ejemplo aplicado

Clasificar una lista de valores según sean positivos, negativos o cero:

```
valores <- c(-3, 0, 5, -1, 2)

for (v in valores) {
  if (v > 0) {
    print(paste(v, "es positivo"))
  } else if (v < 0) {
    print(paste(v, "es negativo"))
  } else {
    print(paste(v, "es cero"))
  }
}
```

```
[1] "-3 es negativo"
```

```
[1] "0 es cero"
```

```
[1] "5 es positivo"
```

```
[1] "-1 es negativo"
```

```
[1] "2 es positivo"
```