

Introducción a la Programación en R

Mihai M. Craiu

7/9/2020

Contents

Condicionales	1
Ejercicios	4
Bucles	6

Condicionales

Las **estructuras selectivas** o **condicionales** son aquellas que nos permiten hacer una selección entre dos o varias rutas de ejecución posible. La selección se llevará a cabo según el valor de una condición. En lo sucesivo entenderemos por expresión a cualquier agrupación de comandos entre llaves, bien separadas por punto y coma, o bien por salto de línea. La primera estructura condicional que vamos a conocer es **if**, y no es más que un **si condicional**:

if (condición1) expresión1

Si la condición de **condición1** se verifica (TRUE), entonces la sentencia a ejecutar recogida en *expresión1* se lleva a cabo. Si el resultado de la evaluación es falso, la sentencia a ejecutar simplemente se ignora.

Ejemplo: Número positivo

```
x <- 20
if (x>0) {paste(x,"es positivo")}
```

```
## [1] "20 es positivo"
```

Ejemplo: Número positivo, negativo o cero

```
y <- -15
if (y>0) {paste(y, "es positivo")}
if (y==0) {paste(y,"es cero")}
if (y<0) {paste(y, "es negativo")}
```

```
## [1] "-15 es negativo"
```

Ejemplo: Número par o impar (supone un uso interactivo de R)

```
n <- readline(prompt = "Introduzca un entero: ")
n <- as.numeric(n)
if (n%%2 == 0) {paste(n, "es par")}
if (n%%2 == 1) {paste(n, "es impar")}
```

En los dos últimos ejemplos hemos visto que a veces necesitamos que una sentencia se ejecute cuando se cumple la condición, y que otra sentencia se ejecute cuando dicha condición sea falsa. Si nos encontramos en este caso, la forma de resolverlo únicamente a través de sentencias **if** es engorrosa y poco práctica, por lo que existe otra forma más sencilla para llevar a cabo dicha tarea: la estructura **if-else**:

if (condición1) expresión1 else expresión2

Si **condición1** es evaluada como verdadera, entonces se ejecuta la **expresión1**, en cambio, cuando la condición se evalúa como falsa, se ejecuta la **expresión2**

Las estructuras if-else se pueden **anidar** y el lenguaje R dispone además de una versión vectorizada de if, *ifelse()*, sobre la que volveremos posteriormente.

Ejemplo: Número par o impar (supone un uso interactivo de R)

```
n <- readline(prompt= "Introduzca un entero: ") n <- as.numeric(n) if (n%%2 == 0) {paste(n,
"es par")} else {n, "es impar"}
```

```
y <- -15
if (y>0){
  paste(y, "es positivo")
}else if (y==0){
  paste(y, "es cero")
}else {paste(y, "es negativo")}
```

```
## [1] "-15 es negativo"
```

Ejemplo: Tipo de Bono

```
Sector_Emisor <- "Industria_Textil"
Vencimiento <- 5

if ( (Sector_Emisor=="Gobierno") | (Sector_Emisor=="GOBIERNO")){
  if(Vencimiento<=1){
    Clasif_Bono <- "GOB_CORTO"
  }else if (Vencimiento<=3){
    Clasif_Bono <- "GOB_MEDIO"
  }else{Clasif_Bono <- "GOB_LARGO"}
}else if ((Sector_Emisor=="Financiero") | (Sector_Emisor=="FINANCIERO")){
  if(Vencimiento<=1){
    Clasif_Bono <- "FIN_CORTO"
  }else if(Vencimiento>=3){
    Clasif_Bono <- "FIN_MEDIO"
  }else{Clasif_Bono <- "FIN_LARGO"}
}else {
  if(Vencimiento<=1){
    Clasif_Bono <- "NOFIN_CORTO"
  }else if (Vencimiento<=3){
```

```

        Clasif_Bono <- "NOFIN_MEDIO"
      }else{Clasif_Bono <- "NOFIN_LARGO"}
    }
  Clasif_Bono

```

```
## [1] "NOFIN_LARGO"
```

Ejemplo: Múltiplos de 4

```

m <- 15
cociente <- floor(m/4)
if(m%%4 ==0){
  paste(m,"es de la forma 4k, con k=", cociente)
}else if(m%%4==1){
  paste(m,"es de la fomra 4k+1, con k=", cociente)
}else if(m%%4==2){
  paste(m,"es de la forma 4k+2, con k=", cociente)
}else {paste(m,"es de la forma 4k+3,con k=", cociente)}

```

```
## [1] "15 es de la forma 4k+3,con k= 3"
```

A menudo sucede, como en el ejemplo anterior, que tenemos difernetes opciones dependiendo del valor de una variable. Para evitar la concatenación de excesivos condicionales if, tenemos a nuestra disposición la estructura **switch**:

switch(variable, expresión1, expresión2, ...,expresiónN)

Si **variable** es un entero positivo i, entonces se ejecuta la expresión i-ésima.

switch(variable, string1=expresión1, ..., stringN=expresiónN)

Si **variable** es del tipo cadena de caracteres, entonces se ejecuta la expresión cuyo identificador coincida con el contenido de la variable. A diferencia del caso numérico, en caso de no coincidencia de la variable con los indetificadores, se ejectura la expresión por defecto **default**.

Ejemplos:

```
switch(1,"Madrid","Barcelona","Valencia")
```

```
## [1] "Madrid"
```

```
switch(3,"Madrid","Barcelona","Valencia")
```

```
## [1] "Valencia"
```

```

switch(0,"Madrid","Barcelona","Valencia")
switch("tres",uno="1",dos="2",tres="3")

```

```
## [1] "3"
```

```
switch("siete",uno="1",dos="2",tres="3","Otro")
```

```
## [1] "Otro"
```

Ejemplo: Tipo de seguro

```
Identif_seguro <- 3
switch(Identif_seguro,
      {TIPO_SEGURO <- "Hogar"},
      {TIPO_SEGURO <- "Autos"},
      {TIPO_SEGURO <- "Vida"})
TIPO_SEGURO
```

```
## [1] "Vida"
```

Ejemplo: Múltiplos de 4... ¿es correcto?

```
m <- 9
cociente <- floor(m/4)
resto <- m%4
switch(resto,
      {paste(m,"es de la forma 4k+1, con k =", cociente)},
      {paste(m,"es de la forma 4k+2, con k =", cociente)},
      {paste(m,"es de la forma 4k+3, con k =", cociente)},
      paste(m,"es de la forma 4k, con k =", cociente) )
```

```
## [1] "9 es de la forma 4k+1, con k = 2"
```

Ejemplo: Múltiplos de 4

```
m <- 12
cociente <- floor(m/4)
resto <- as.character(m%4)
switch(resto,
      "1" = {paste(m,"es de la forma 4k+1, con k =", cociente)},
      "2" = {paste(m,"es de la forma 4k+2, con k =", cociente)},
      "3" = {paste(m,"es de la forma 4k+3, con k =", cociente)},
      paste(m,"es de la forma 4k, con k =", cociente) )
```

```
## [1] "12 es de la forma 4k, con k = 3"
```

Ejercicios

Ejercicio propuesto: Clasificación por edad

Escribe un código que permita clasificar la variable RIESGO_ASEGURADO en “Alto”, “Bajo” o “Moderado”, en función de la variable EDAD_ASEGURADO, según que ésta sea menor de 25 años o mayor que 70.

```

EDAD_ASEGURADO <- 35

if (EDAD_ASEGURADO < 25){
  RIESGO_ASEGURADO <- "Alto"; RIESGO_ASEGURADO
} else if (EDAD_ASEGURADO > 70) {
  RIESGO_ASEGURADO <- "Medio"; RIESGO_ASEGURADO
} else {RIESGO_ASEGURADO <- "Bajo"; RIESGO_ASEGURADO}

## [1] "Bajo"

```

Ejercicio propuesto: Clasificación por edad y sexo

Modifica el código anterior para que, en caso de que la variable SEXO ASEGURADO sea “M”(mujer, frente a “H”, hombre), el correspondiente perfil de riesgo disminuya un nivel.

```

EDAD_ASEGURADO <- 18
SEXO_ASEGURADO <- "M"

if (EDAD_ASEGURADO < 25){
  RIESGO_ASEGURADO <- "Alto"
  if (SEXO_ASEGURADO == "M") {RIESGO_ASEGURADO <- "Medio"}
  RIESGO_ASEGURADO
} else if (EDAD_ASEGURADO > 70) {
  RIESGO_ASEGURADO <- "Medio"
  if (SEXO_ASEGURADO == "M") {RIESGO_ASEGURADO <- "Bajo"}
  RIESGO_ASEGURADO
} else {RIESGO_ASEGURADO <- "Bajo"; RIESGO_ASEGURADO}

## [1] "Medio"

```

Ejercicio propuesto: Opción call

Escribe un código que permita obtener el beneficio/pérdida para un inversor que compra por 1e el día 3 de septiembre de 2018 una opción de compra (call) sobre unas acciones de Telefónica con vencimiento el 28 de septiembre de 2018, a un precio de ejercicio de 7.19e.

```

Strike <- 7.19
Prima <- 1
PyG <- 0

Cotiz_Telef_28_09_2018 <- 8.19

if(Cotiz_Telef_28_09_2018 <= Strike){
  PyG <- -Prima; PyG
} else {PyG <- Cotiz_Telef_28_09_2018 - (Strike + Prima); PyG}

## [1] -1.776357e-15

```

```
#Observación: Para R, debido a la aritmética de punto flotante que emplea para representar  
#números reales, sucede que  
sqrt(2)^2 == 2 #Arroja FALSE
```

```
## [1] FALSE
```

```
#En tal caso es preferible hacer:  
all.equal(sqrt(2)^2, 2) #Devuelve TRUE
```

```
## [1] TRUE
```

```
#En consecuencia,  
all.equal(0, PyG)
```

```
## [1] TRUE
```

Bucles

Los **bucles** o **ciclos** son estructuras que nos permiten repetir la misma instrucción un