

Sumatorios y productorios en RStudio

En este recurso se utilizarán dos recursos creados previamente: “[Sumatorios.pdf](#)” y “[Explicación del ejercicio M y B.pdf](#)” para poder ver de manera más clara el paso de algoritmia a R.

A continuación, se adjuntan imágenes con las explicaciones correspondientes:

- Sumatorio en R:

```
2 #Explicación general, utilizando la fórmula del primer ejemplo
3 cat("\n014")
4 rm(list=ls())
5 #Comenzamos dando la variable de entrada (por ejemplo el vector A):
6 A= c(3,4,5)
7 #Después definimos la variable n como la longitud del vector, pues es el "tramo" en el que queremos que se aplique el sumatorio:
8 n=length(A)
9 #Una vez tenemos esto, podemos comenzar con la fórmula del sumatorio en sí, recordaremos los pasos:
10 #1. Inicializamos a 0 la variable que resultará de aplicar el sumatorio al vector A (S):
11 S= 0
12 #2. Introducimos nuestro bucle for, el cual afectará a la variable "i" del vector:
13 for (i in 1:n) { #Esto significa que i varía desde 1 hasta n, y como no indicamos otra cosa, lo hará de 1 en 1
14   S= S + (A[i] + i)
15 }
16 #El símbolo "}" indica que finaliza el bucle
17
18 #3. Pedimos que el programa nos devuelva el resultado por pantalla:
19 print(S)
```

- Productorio en R:

```
24 #Ahora utilizaremos la misma fórmula pero en productorios, es decir S = S * (A[i]+i):
25 #Comenzamos dando la variable de entrada (por ejemplo el vector A):
26 A= c(3,4,5)
27 #Después definimos la variable n como la longitud del vector, pues es el "tramo" en el que queremos que se aplique el productorio:
28 n=length(A)
29 #1. Esta vez, inicializaremos la variable P a 1, ya que al tratarse de un productorio, si inicializamos a 0 multiplicaríamos por 0 todo el rato.
30 P=1
31 #2. Una vez tenemos esto, el proceso es el mismo, de manera que declaramos el bucle for:
32 for (i in 1:n) {
33   P= P * (A[i] + i) #Aquí cambiamos "+" por "*" ya que se trata de un productorio
34 }
35
36 #3. Pedimos que imprima por pantalla la variable:
37 print(P)
```

Por otra parte, una vez ha quedado clara la teoría de una manera más general, pasamos a **resolver el ejercicio** mencionado anteriormente, esta vez en R:

```

1 #Explicación del ejercicio M y B
2 cat("\n014")
3 rm(list=ls())
4 #Comenzamos dando las variable de entrada (en este caso deben ser el vector S y el vector Y):
5 S= c(5,7,3)
6 Y= c(3,2,6)
7 #Más adelante, declaramos la variable n que mide la longitud del vector S:
8 n=length(S)
9
10 #1. Una vez tenemos todo esto, comenzamos con la traducción de fórmulas. Empezamos inicializando nuestras variables M y B:
11 M=matrix(0, nrow=n, ncol=n) #De esta manera definimos la matriz M inicializada a 0, con el mismo número de filas y columnas que la longitud de
12 #nuestro vector S
13 B= numeric(n) #Con este comando conseguimos que la variable B sea un vector, también inicializado a 0 (condición necesaria para sumatorios)
14 #y con n "huecos" disponibles
15
16 #2. Comenzamos con los bucles, que esta vez afectarán a las variables "i" y "j":
17 for (i in 1:n) {
18   for (j in 1:n) {
19     M[i,j]=S[i]^(j-1) #Ponemos la fórmula con la que nos indican que rellenaremos la matriz M
20   }
21 }
22 #3. Después del cierre del bucle "for" de la variable j, introducimos la fórmula que nos definirá el vector B:
23 B[i]=Y[i] #No ponemos B[1,i] como indica el enunciado ya que R lo trataría como matriz y daría error al haber definido B como vector.
24 #Si pusieramos dicha fórmula, habría que haber declarado B como una matriz de la siguiente forma: B=matrix(0, nrow=1, ncol=n),
25 #para que nos diera una matriz de una sola fila
26 }
27
28 #Con todo esto, ya podemos pedir que nos imprima por pantalla los valores de M y B:
29 print(M)
30 print(B)

```

Para poder tener el código guardado, se adjuntan los scripts utilizados para la realización de este recurso:

- [Sumatorios y productorios](#)
- [Explicación ejercicio](#)

Por último, se plantea el siguiente ejercicio a resolver, para comprobar que se ha entendido lo que se ha explicado en este recurso:

Disponemos de un vector R formado por n puntos en el eje de abscisas, y un vector Z que contiene los valores resultantes de aplicar una función $g(x)$ a cada elemento de T.

Es decir, T y Z representan las coordenadas de n puntos (T_i, Z_i) .

Se pide (debes escribir el código en R):

1. Calcular la matriz A, siendo:

$$A(i, j) = (R_i)^j \cdot \prod_{k=1}^j R_k$$

donde:

- La matriz A es cuadrada, de tamaño $n \times n$.
- Para cada posición (i, j) , se eleva el valor R_i a j y se multiplica por el productorio de los primeros j elementos del vector T.

2. Calcular el vector C, siendo:

$$C(i) = \sum_{k=1}^i Z_k$$

donde:

- **C** es un vector con n elementos.
- Cada componente $C(i)$ es la suma acumulada de los primeros i elementos del vector **Z**.
-

Aquí tenemos la solución (se aconseja no mirarla hasta haberlo intentado individualmente):

```
1 #Explicación del ejercicio R y Z
2 cat("\014")
3 rm(list=ls())
4 #Comenzamos dando las variable de entrada (en este caso deben ser el vector R y el vector Z), aunque podríamos haberlas pedido por consola:
5 R= c(5,7,3)
6 Z= c(3,2,6)
7 #Más adelante, declaramos la variable n que mide la longitud del vector R:
8 n=length(R)
9
10 #1. Una vez lo tenemos, comenzamos con la traducción de fórmulas. Empezamos inicializando nuestras variables A y C:
11 A=matrix(0, nrow=n, ncol=n)
12 C= numeric(n)
13
14 #2. Comenzamos con los bucles, que otra vez afectarán a las variables "i" y "j":
15 for (i in 1:n) {
16   for (j in 1:n) {
17     P = 1 # Declaramos la variable P para el productorio e inicializamos a 1
18     for (k in 1:j) { #Ponemos el bucle for que afectará al productorio
19       P= P*R[k] #Calculamos el productorio según nos indica la fórmula
20       A[i, j] <- (R[i]^j) * P #Rellenamos la matriz A según nos indica la fórmula
21     }
22   }
23   #3. Ahora pasamos a rellenar el vector C
24   S=0 #Declaramos la variable S para el sumatorio e inicializamos a 0
25   for (k in 1:i) { #Ponemos el bucle for que afectará al sumatorio
26     S= S+Z[k] #Calculamos el sumatorio según nos indica la fórmula
27     C[i]= S #Rellenamos el vector C según nos indica la fórmula
28   }
29 }
30
31
32 #Con todo esto, ya podemos pedir que nos imprima por pantalla los valores de A y C:
33 print(A)
34 print(C)
```

Y al igual que en los anteriores ejemplos, aquí tenemos el script:

- [Ejercicio para practicar](#)