

PRÁCTICA DE PROGRAMACIÓN Nº3

Durante esta tercera práctica se trabajaron conceptos relacionados con los bucles, las matrices y las gráficas.

Bucles "For"

- Este comando programado de R nos permite realizar una o varias operaciones repetidamente a medida que variamos el valor de nuestra variable. Es decir, creamos un "loop" que se va repitiendo una y otra vez en base a unas condiciones que habremos preestablecido anteriormente.

VECTORES: EJERCICIO EXPLICATIVO 1

```
v=c(12,-3,5,18.7)
```

```
w<-c(12,0.25,77,exp(2))
```

```
n=length(v)
```

```
s=c(0,0,0,0)
```

DATOS DEL PROBLEMA

En este paso inicializamos el vector "s" en 0

```
for (i in 1:n){
```

```
  s[i]=v[i]+w[i]
```

```
}
```

Este comando nos está indicando que la operación que vamos a llevar a cabo a continuación va a realizarse en bucle para los valores de la variable "i" desde el 1 hasta "n" siendo este último un valor que hemos definido anteriormente.

!!!NO TE OLVIDES DE CERRAR LOS CORCHETES QUE HAYAS ABIERTO!!!!

Esta es la operación que vamos a querer que se haga siguiendo el bucle for que hemos impuesto. En este ejemplo se trata de una suma de vectores

```
> v=c(12,-3,5,18.7)
> w<-c(12,0.25,77,exp(2))
> n=length(v)
> s=c(0,0,0,0)
> for(i in 1:n){
+ s[i]= v[i]+w[i]
+ }
> s
```

```
[1] 24.00000 -2.75000
82.00000 26.08906
```

```
> v=c(12,-3,5,18.7)
> w<-c(12,0.25,77,exp(2))
> n=length(v)
> s=c(0,0,0,0)
> for(i in 2:n){
+ s[i]= v[i]+w[i]
+ }
> s
```

```
[1] 0.00000 -2.75000
82.00000 26.08906
```

```
> v=c(12,-3,5,18.7)
> w<-c(12,0.25,77,exp(2))
> n=length(v)
> s=c(0,0,0,0)
> for(i in 3:n){
+ s[i]= v[i]+w[i]
+ }
> s
```

```
[1] 0.00000 0.00000
82.00000 26.08906
```

```
> v=c(12,-3,5,18.7)
> w<-c(12,0.25,77,exp(2))
> n=length(v)
> s=c(0,2,0,0)
> for(i in 3:n){
+ s[i]= v[i]+w[i]
+ }
> s
```

```
[1] 0.00000 2.00000
82.00000 26.08906
```

La explicación de la operación sería que vamos a almacenar en el hueco "i" (que en este caso va a variar de 1 a 4 ya que son vectores de 4 componentes) de nuestro vector "s" la suma del término que ocupa la posición "i" dentro de los vectores "v" y "w". Los ejemplos que hemos puesto arriba en los recuadros están pensados para que el alumno visualice como variaría el resultado en función de cómo o qué se modifica (principalmente en el bucle for). De esta manera se hace más visual y es más fácil relacionarlo con otros ejercicios similares.

#VECTORES: EJERCICIO EXPLICATIVO 2

```
v= c(12,-3,5,18.7)
w<-c(12,0.25,77,exp(2))
n=length(v)
SUMA= c(0,0,0,0)
for(i in 1:n){
    SUMA=SUMA+v[i]
}
SUMA
32.7 32.7 32.7 32.7
```

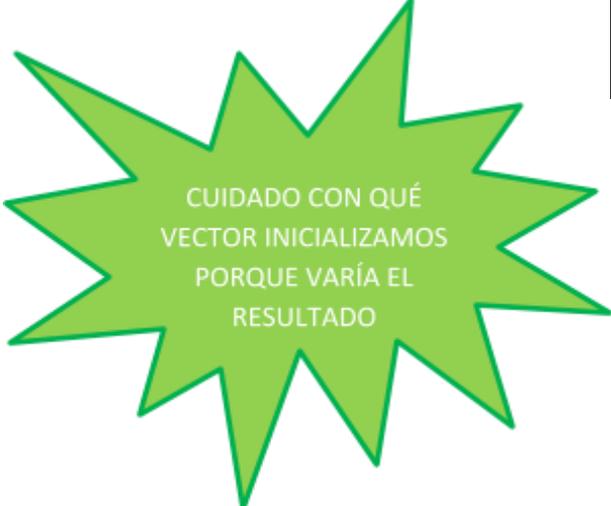
Este otro caso es muy similar al anterior pero en este ejemplo en lugar de sumar componentes de dos vectores, vamos a sumar la componente "i" del vector v al vector SUMA y la vamos a almacenar en ese mismo vector SUMA. Este proceso va a repetirse en bucle desde 1 hasta n.

En este otro caso he modificado el bucle "for" para que vaya desde 2 hasta n para poder ver el cambio que se produce. En este caso se van a ir sumando las componentes del vector V y acumulándose en la variable SUMA. La diferencia es que ahora solo se han sumado tres de las componentes (desde 2 hasta n que en este caso es 4).

```
> v= c(12,-3,5,18.7)
> w<-c(12,0.25,77,exp(2))
> n=length(v)
> SUMA= c(0,0,0,0)
> for(i in 2:n){
+ SUMA=SUMA+v[i]
+ }
> SUMA
[1] 20.7 20.7 20.7 20.7
```

EL RESULTADO FINAL CORRESPONDE CON LA SUMA DE TODOS LOS VALORES DEL VECTOR "V"

EL RESULTADO FINAL SERÁ LA SUMA DE TRES DE LAS COMPONENTES DEL VECTOR "V". ES DECIR, NO TOMA EN CUENTA EL PRIMER VALOR



#VECTORES: EJERCICIO EXPLICATIVO 3

```
> v=c(12,-3,5,18.7)
> w<-c(12,0.25,77,exp(2))
> pesc=c(0)
> for (ko in 1:n){
+ pesc=pesc+v[ko]*w[ko]
+ }
> pesc
[1] 666.4253
```



RECORDAD QUE EL PRODUCTO ESCALAR PUEDE REALIZARSE O POR BUCLES O CON LA FÓRMULA: $v \% * \% w$

En este otro caso de bucles tenemos dentro del propio bucle la siguiente operación:
 $pesc=pesc+v[ko]*w[ko]$

El vector "pesc" representa el producto escalar y por tanto lo que va a hacer el programa es multiplicar $v[ko]*w[ko]$ y posteriormente sumárselo a "pesc" el cual hemos inicializado anteriormente. Una vez realizado este proceso y, al igual que en los otros casos propuestos, el valor de la "i" varía para la segunda repetición del bucle y así sucesivamente hasta usar todos los valores de [ko].

BUCLES "FOR" ANIDADOS

LOS BUCLES ANIDADOS, COMO SU PROPIO NOMBRE DA A ENTENDER, SON UNA SUCESIÓN DE BUCLES "FOR" INCLUIDOS DENTRO DE OTROS BUCLES. ES DECIR, GRACIAS A ESTOS VAMOS A PODER ASOCIAR UNA SERIE DE CONDICIONES A UNA O VARIAS OPERACIONES QUE QUERAMOS REALIZAR. ESTOS BUCLES VAN A SER MUY ÚTILES PARA GESTIONES CON VECTORES QUE SIGUEN UNA FORMULA REPETITIVA. SU APLICACIÓN BIOTECNOLÓGICA MÁS DIRECTA SERÍA LA DE LA RAMA COMPUTACIONAL QUE SE CENTRA EN EL DESARROLLO DE NUEVOS INSTRUMENTOS DE UTILIDAD EN EL CAMPO.

ESTRUCTURA DE LOS BUCLES ANIDADOS

```
for (i in 1:2){
  for (j in 1:2){
    for (k in 1:2){
      E[i,j]=C[i,j]+D[i,j]
    }
  }
}
```

TEN EN CUENTA EL ORDEN EN EL QUE SE CIERRAN LOS CORCHETES QUE HEMOS IDO ABIRIENDO.

CONSEJO: DALE AL TABULADOR PARA IR DEJANDO ESPACIOS DE MANERA QUE VAYA QUEDANDO DE FORMA ESCALONADA. ASÍ SERÁ MÁS FÁCIL DIFERENCIARLOS Y EVITARÁS FALLOS

BUCLES ANIDADOS

```
for (i in v1:vf1) {  
  for (j in v2:vf2){  
    for (k in v3:vf3){  
      Proceso de cálculo  
    }  
  }  
}
```

EJERCICIO EXPLICATIVO: BUCLES ANIDADOS

```
> C=matrix(c(2,4,5.5,1),nrow=2,ncol=2)  
> D=matrix(c((1/2),sqrt(7),1/3,log(12)),nrow=2,ncol=2)  
> D  
      [,1] [,2]  
[1,] 0.500000 0.3333333  
[2,] 2.645751 2.4849066  
> E=matrix(c(0),nrow=n,ncol=m)  
> for (i in 1:2){  
+ for (j in 1:2){  
+ E[i,j]=C[i,j]+D[i,j]  
+ }  
+ }  
> E  
      [,1] [,2]  
[1,] 2.500000 5.833333  
[2,] 6.645751 3.484907
```

OBJETIVO

Almacenar en una matriz "E" la suma de las matrices "C" y "D"

BUCLE EN "i"

Es el primer bucle de nuestro "nido" y tal y como se indica va desde 1 hasta 2.

NOTA: Podríamos haber puesto desde 1 hasta nrow ("n"). Ponemos nrow porque el bucle "i" lo hemos relacionado con las filas de nuestra matriz.

BUCLE EN "j"

Es el segundo bucle de nuestro "nido" y tal y como se indica va desde 1 hasta 2.

NOTA: Podríamos haber puesto desde 1 hasta ncol ("m") (que es 2). Ponemos ncol porque el bucle "j" lo hemos relacionado con las columnas de nuestra matriz.

*** Puedes hacer todos los bucles que necesites.**