**Segundo parcial de Fundamentos de Programación de 2015: Ejercicio 2º**

**Enunciado:** Los valores de la concentración del colesterol LDL en muestras de sangre de tres pacientes se almacenan en sendos vectores u, v, w, respectivamente. El número de muestras de cada paciente son: p, q, r, respectivamente. Siendo N (hay que deducirlo dentro del programa) el mayor valor entre p, q y r.

Se pide un programa para construir los vectores u, v y w, para calcular N y para crear una matriz L de 3 columnas y N filas de manera que la primera columna esté formada por los elementos del vector u, la segunda por los elementos del vector v y la tercera por los elementos del vector w.

**Nota:** La matriz L se inicia a 0. Los números p, q y r tomarán valores naturales aleatorios entre uno y diez, los valores dentro de los vectores u, v y w serán números naturales aleatorios entre 125 y 200.

**Resolución:**

p<-sample(1:10,1); u<-0

Asignación de valores aleatorios a p, q y r la longitud de u, v, y w, que se crean para que el bucle pueda trabajar con ellos.

q<-sample(1:10,1); v<-0

r<-sample(1:10,1); w<-0

Bucles para dar valor a los vectores u, v, y w. Se generará un valor aleatorio hasta que i sea más grande que la longitud establecida en el anterior paso.

Así, p está formado por u valores aleatorios.

Hay que igualar cada vez i a 0 porque si no tendría el valor que acabó teniendo en el último bucle.

i<-0

while(i<=p){

u[i]=sample(120:200,1)

i=i+1

}

i<-0

while(i<=q){

v[i]=sample(120:200,1)

i=i+1

}

i<-0

while(i<=r){

w[i]=sample(120:200,1)

i=i+1

}

if(p >= q && p >= r){

Valor máximo entre p, q y r. “&&” es un operador lógico y significa “y”. Para N= r no hace falta el operador lógico porque llegados a ese punto r tiene que ser el valor máximo de los tres. Dado que el operador es “>=” y no solamente “>”, esto funciona aunque los p, q y r sean iguales.

N= p

} else if(q >= p && q >= r){

N= q

} else{

N= r

}

Crear la matriz L. Se crea una matriz de ceros de tres columnas y N filas.

L=matrix(c(0),ncol=3, nrow=N)

Bucles para dar valores a la matriz. Hay varias opciones.

Primera opción: bucle de 1 a p para que la primera columna de L adopte todos los valores de u (dado que u mide p). Los demás valores siguen siendo los ceros asignados antes.

Segunda opción: bucle para que en las filas inferiores que o iguales a q se tomen los valores de v mientras que las superiores tomen el valor 0. Es útil si la matriz no ha sido iniciada a 0. Esta es la opción más recomendable por ser la más completa.

Tercera opción: bucle de 1 a N en que para que la tercera columna de L adopte todos los valores de w y hasta que se quede sin valores. Luego no asignará ningún valor, sino “NA” (“not assigned”) en cada fila para la que el vector w no tenga un valor. Esta es la opción menos recomendable porque imposibilita algunos cálculos con la matriz al convertirla en no cuadrada.

En estos bucles no hace falta igualar i a 1 porque en “for(i in 1:N)” ya definimos que la i va de 1 a N.

for(i in 1: p){

L[i,1]=u[i]

}

for(i in 1:N){

if(i<= q){

L[i,2]= v [i]

} else {

L[i,2]=0

}

}

for(i in 1:N){

L[i,3]= w [i]

}

L

[,1] [,2] [,3]

Este es un resultado posible. “NA” no es igual a 0, por lo que ésta no es una matriz cuadrada.

[1,] 131 194 141

[2,] 181 147 124

[3,] 0 161 133

[4,] 0 120 156

[5,] 0 129 162

[6,] 0 187 NA

[7,] 0 135 NA

[8,] 0 174 NA