

# R Programming Structures

*Juan Ramón Gómez Berzosa*

*24/10/2018*

## R Programming Structures

1. Crea una función creciente que indique si los elementos de un vector dado son estrictamente crecientes. No se permite ordenar el vector.

```
creciente = function(vector){  
  
  creciente = TRUE  
  i = 1  
  while (creciente & (i < length(vector))) {  
    creciente = ifelse(vector[i] < vector[i+1], TRUE, FALSE)  
    i = i+1  
  }  
  creciente  
}  
  
vector = c(1:15)  
creciente(vector)
```

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

```
vector = c(15:1)  
creciente(vector)
```

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

2. Crea una función montecarlo que calcule la estimación de la siguiente integral.

```
montecarlo.bucles= function(N){  
  
  hits = 0  
  i = 1  
  for(i in 1:N){  
    random = runif(2,0,1)  
    if(random[2] < (random[1]**2))  
      hits = hits + 1  
  }  
  hits/N  
}  
  
montecarlo.optimizado = function(N){  
  
  random = runif(2*N,0,1)  
  random  
  r1 = random[seq(1,2*N,by=2)]
```

```

r2 = random[seq(2,2*N,by=2)]

sol = r2 < (r1**2)
hits = length( sol[sol==TRUE] )
hits/N
}
system.time(montecarlo.bucles(10000000))

```

```

##      user  system elapsed
## 28.140   1.758   30.098

```

```

system.time(montecarlo.optimizado(10000000))

```

```

##      user  system elapsed
##  1.370    0.200    1.575

```

3. Crea una lista de 5 vectores numéricos y ordena todos los vectores de la lista.

```

rand = c(1:1000)
lista = list(v1=sample(rand, size = 10), v2=sample(rand, size = 10),
            v3=sample(rand, size = 10), v4=sample(rand, size = 10),
            v5=sample(rand, size = 10))

listaOrdenada = lapply(lista,function(lis){lis[order(lis)]})
listaOrdenada

```

```

## $v1
## [1] 25 119 174 251 315 321 678 815 942 973
##
## $v2
## [1] 229 272 364 391 400 688 752 906 991 993
##
## $v3
## [1] 22 205 220 355 389 402 447 516 524 758
##
## $v4
## [1] 73 76 102 104 138 332 416 796 840 944
##
## $v5
## [1] 63 93 158 369 610 651 716 787 818 837

```

4. Calcula el valor mínimo de cada columna de una matriz, pero toma los valores impares como números negativos y los pares como positivos.

```

rand = c(1:20)

matriz = matrix(sample(rand, size = 16, replace = TRUE),nrow=4,ncol = 4)
matrizTransformada = ifelse(matriz%%2 !=0, matriz*-1, matriz)

v = apply(matrizTransformada, 2, min )
v

```

```

## [1] -11 -5 2 -15

```

5. Dada una matriz devuelva una lista con los valores mayores a 7 de cada fila.

```
rand = c(1:20)

matriz = matrix(sample(rand, size = 16,replace = TRUE),nrow=4,ncol = 4)

#Imprimimos la matriz
matriz

##      [,1] [,2] [,3] [,4]
## [1,]    2    7    7    7
## [2,]    4   11   11   16
## [3,]    9   15    8    2
## [4,]    4    8    3    9

v = apply(matriz, 1,function(matriz){ matriz[matriz >7 ] })

#Imprimimos los valores mayores que 7 por filas
v

## [[1]]
## integer(0)
##
## [[2]]
## [1] 11 11 16
##
## [[3]]
## [1]  9 15  8
##
## [[4]]
## [1]  8  9
```