

# EjerciciosREstructuras

*Alberto Armijo Ruiz*

*24 de octubre de 2018*

## Ejercicios Estructuras en R

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(x){  
  creciente = TRUE  
  i = 1  
  max_l = length(x)-1  
  while (i <= max_l && creciente) {  
  
    if(x[i] > x[i+1]){  
      creciente = FALSE  
    }  
    i = i+1  
  }  
  
  creciente  
}  
  
p = c(1,2,3,4,5)  
creciente(p)
```

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

```
p2 = c(3,1,2,3,4)  
creciente(p2)
```

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

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

```
montecarlo = function(x){  
  hits = 0  
  for( i in 1:x ){  
    r1r2 = runif(2,0,1)  
    if(r1r2[2] < r1r2[1]**2 )  
      hits = hits +1  
  }  
  
  hits/x  
}  
  
montecarlo(4)
```

```
## [1] 0.75
```

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

```
v1 = c(1,2,3,4,5)
v2 = c(3,4,5,1,2)
v3 = c(2,5,4,2,1)
v4 = c(4,5,6,8,2)
v5 = c(6,2,3,9,10)

m_list = list(v1,v2,v3,v4,v5);

for (v in 1:length(m_list)) {
  m_list[[v]] = sort(m_list[[v]])
}

str(m_list)
```

```
## List of 5
## $ : num [1:5] 1 2 3 4 5
## $ : num [1:5] 1 2 3 4 5
## $ : num [1:5] 1 2 2 4 5
## $ : num [1:5] 2 4 5 6 8
## $ : num [1:5] 2 3 6 9 10

# Otra forma
m_list2 = list(v1,v2,v3,v4,v5)

m_list2 = lapply(m_list2,sort)
str(m_list2)
```

```
## List of 5
## $ : num [1:5] 1 2 3 4 5
## $ : num [1:5] 1 2 3 4 5
## $ : num [1:5] 1 2 2 4 5
## $ : num [1:5] 2 4 5 6 8
## $ : num [1:5] 2 3 6 9 10
```

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

```
paroiimpar = function(x){
  ifelse(x%%2!=0,-1*x,x)
}

m_matrix = matrix(c(1,4,3,4,5,5,6,3,4,5,1,23,4,7,8,8,1,3),ncol=3)
m_matrix
```

```
##      [,1] [,2] [,3]
## [1,]    1    6    4
## [2,]    4    3    7
## [3,]    3    4    8
## [4,]    4    5    8
## [5,]    5    1    1
## [6,]    5   23    3
```

```
m_transformada = apply(m_matrix,c(1,2),paroimpar); m_transformada
```

```
##      [,1] [,2] [,3]
## [1,]  -1   6   4
## [2,]   4  -3  -7
## [3,]  -3   4   8
## [4,]   4  -5   8
## [5,]  -5  -1  -1
## [6,]  -5 -23  -3
```

```
col_menores = apply(m_transformada,1,which.min)
pos_menores = cbind(1:nrow(m_transformada),col_menores)
valores_menores = m_matrix[pos_menores]; valores_menores
```

```
## [1]  1  7  3  5  5 23
```

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

```
m = matrix(c(1,8,12,3,4,10,11,1,2),ncol = 3)
m
```

```
##      [,1] [,2] [,3]
## [1,]   1   3  11
## [2,]   8   4   1
## [3,]  12  10   2
```

```
mayores = list(m[which(m > 7)])
mayores
```

```
## [[1]]
## [1]  8 12 10 11
```