

Ejercicios

Victor Lopez

2023-01-12

1. Crea un vector llamado Harry formado por la sucesión de números consecutivos entre el -10 y 27. Pídele a R que devuelva el elemento de índice 7. Escribe el resultado.

```
Harry = -10:27
Harry[7]
```

```
## [1] -4
```

2. Da el maximo de la sucesion $100 \cdot 2^n - 7 \cdot 3^n$ con $n = 0, \dots, 200$

```
n = 0:200
sucesion = 100 * 2^n - 7 * 3^n
max(sucesion)
```

```
## [1] 1499
```

3. Crea la sucesión de números consecutivos entre 0 y 40. A continuación, crea el vector $3 \cdot 5^n - 1$ con $n = 0, \dots, 40$. Ponle como nombre x. Ahora, da el subvector de los elementos que son estrictamente mayores a 3.5.

```
n = 0:40
x = 3 * 5^n - 1
x[x > 3.5]
```

```
## [1] 1.400000e+01 7.400000e+01 3.740000e+02 1.874000e+03 9.374000e+03
## [6] 4.687400e+04 2.343740e+05 1.171874e+06 5.859374e+06 2.929687e+07
## [11] 1.464844e+08 7.324219e+08 3.662109e+09 1.831055e+10 9.155273e+10
## [16] 4.577637e+11 2.288818e+12 1.144409e+13 5.722046e+13 2.861023e+14
## [21] 1.430511e+15 7.152557e+15 3.576279e+16 1.788139e+17 8.940697e+17
## [26] 4.470348e+18 2.235174e+19 1.117587e+20 5.587935e+20 2.793968e+21
## [31] 1.396984e+22 6.984919e+22 3.492460e+23 1.746230e+24 8.731149e+24
## [36] 4.365575e+25 2.182787e+26 1.091394e+27 5.456968e+27 2.728484e+28
```

4. Crea una función que devuelva la parte real, la imaginaria, el módulo, el argumento y el conjugado de un número, mostrando solo 2 cifras significativas.

```
numImag = function(num){
  print(c(Re(num), Im(num), Mod(num), Arg(num), Conj(num)), 2)
}
numImag(9 + 13i)
```

```
## [1] 9.00+ 0i 13.00+ 0i 15.81+ 0i 0.97+ 0i 9.00-13i
```

5. Crea una función que resuelva ecuaciones de segundo grado (de la forma $Ax^2 + Bx + C = 0$). No importa, por ahora, que tengas en cuenta las ecuaciones de segundo grado que no tienen solución real.

```
tercer = function(a, b, c) {  
  x1 = (-b + sqrt(b^2 - 4 * a * c))/(2* a)  
  x2 = (-b - sqrt(b^2 - 4 * a * c))/(2* a)  
  
  print(paste("x1 = ", x1))  
  print(paste("x2 = ", x2))  
}  
tercer(-1, -2, 3)
```

```
## [1] "x1 = -3"  
## [1] "x2 = 1"
```

6. Tomando el vector `vec = c(0,9,98,2,6,7,5,19,88,20,16,0)`, da 3 opciones diferentes para calcular el sub-vector `c(9,19,20,16)`

Tomando el vector `vec` definido en el apartado anterior, busca

- qué entradas son pares
- qué entradas no son pares y mayores que 20
- dónde toma `vec` su valor máximo
- dónde toma `vec` sus valores mínimos

```
vec = c(0,9,98,2,6,7,5,19,88,20,16,0)  
sort(vec)[7:10]
```

```
## [1] 9 16 19 20
```

```
vec[vec >= 9 & vec <= 20]
```

```
## [1] 9 19 20 16
```

```
vec[which(vec >= 9 & vec <= 20)]
```

```
## [1] 9 19 20 16
```

```
vec[c(2, 8, 10, 11)]
```

```
## [1] 9 19 20 16
```

7. Da la entrada (2, 2) de $A(A + A) \cdot A$ con:

$$\begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$$

```
A = rbind(c(1, 3), c(2, 4))
res = A %*% (A + A) %*% A
A[2, 2]
```

```
## [1] 4
```

8. Da los valores propios de la matriz

$$\begin{pmatrix} 2 & 4 & -6 \\ 0 & 0 & 3 \\ 0 & -2 & 5 \end{pmatrix}$$

$$B = \begin{bmatrix} 2 & 4 & -6 \\ 0 & 0 & 3 \\ 0 & -2 & 5 \end{bmatrix}$$

```
B = rbind(c(2, 4, -6), c(0, 0, 3), c(0, -2, 5))
eigen(B)$values
```

```
## [1] 3 2 2
```

9. Da, redondeando a 3 cifras decimales, los vectores propios de la matriz $C = \begin{bmatrix} -48 & 35 & -12 \\ -134 & 95 & -32 \\ -194 & 133 & -44 \end{bmatrix}$

$$\begin{vmatrix} -48 & 35 & -12 \\ -134 & 95 & -32 \\ -194 & 133 & -44 \end{vmatrix}$$

```
C = rbind(c(-48, 35, -12), c(-134, 95, -32), c(-194, 133, -44))
round(eigen(C)$vectors, 3)
```

```
##      [,1] [,2] [,3]
## [1,] 0.371 0.169 0.098
## [2,] 0.743 0.507 -0.195
## [3,] 0.557 0.845 -0.976
```

10. Da el rango de la matriz $D = \begin{bmatrix} -2 & -8 & -2 & 3 \\ -3 & -6 & -1 & 2 \\ -9 & -22 & -3 & 7 \\ -18 & -44 & -8 & 15 \end{bmatrix}$

$$\begin{Bmatrix} -2 & -8 & -2 & 3 \\ -3 & -6 & -1 & 2 \\ -9 & -22 & -3 & 7 \\ -18 & -44 & -8 & 15 \end{Bmatrix}$$

```
D = rbind(c(-2, -8, -2, 3), c(-3, -6, -1, 2), c(-9, -22, -3, 7), c(-18, -44, -8, 15))
qr(D)$rank
```

```
## [1] 3
```

1. Realiza los siguientes productos de matrices siguiente en R:

```
A = rbind( c(1, 2, 3, 4), c(4, 3, 2, 1), c(0, 1, 0, 2), c(3, 0, 4, 0) )
B = rbind( c(4, 3, 2, 1), c(0, 3, 0, 4), c(1, 2, 3, 4), c(0, 1, 0, 2) )
```

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \\ 0 & 1 & 0 & 2 \\ 3 & 0 & 4 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 0 & 3 & 0 & 4 \\ 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 2 \end{pmatrix}$$

$$A \cdot B$$

```
A %*% B
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    7   19   11   29
## [2,]   18   26   14   26
## [3,]    0    5    0    8
## [4,]   16   17   18   19
```

$$B \cdot A$$

```
B %*% A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   19   19   22   23
## [2,]   24    9   22    3
## [3,]   21   11   23   12
## [4,]   10    3   10    1
```

$$(A \cdot B)^t$$

```
t(A %*% B)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    7   18    0   16
## [2,]   19   26    5   17
## [3,]   11   14    0   18
## [4,]   29   26    8   19
```

$$B^t \cdot A$$

```
t(B) %*% A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    4    9   12   18
## [2,]   18   17   19   19
## [3,]    2    7    6   14
## [4,]   23   18   19   16
```

$$(A \cdot B)^{-1}$$

```
solve(A %*% B)
```

```
##      [,1] [,2] [,3] [,4]
## [1,] -1.66 -0.65  4.52  1.52
## [2,]  1.60  0.80 -4.60 -1.60
## [3,]  1.02  0.35 -2.84 -0.84
## [4,] -1.00 -0.50  3.00  1.00
```

$$A^{-1} \cdot B^{-1}$$

```
solve(A) %*% solve(B)
```

```
##      [,1] [,2] [,3] [,4]
## [1,] -0.40  0.70  0.80 -2.50
## [2,]  0.42 -0.25 -0.88  1.65
## [3,]  0.30 -0.65 -0.60  2.25
## [4,] -0.26  0.25  0.64 -1.45
```

2. Considera en un vector los números de tu DNI (puedes inventártelos) y llámalo dni. Por ejemplo, si tu DNI es 54201567K, tu vector será

```
dni = (5, 4, 2, 0, 1, 5, 6, 7)
```

Define el vector en R. Calcula con R el vector dni al cuadrado, la raíz cuadrada del vector dni y, por último, la suma de todas las cifras del vector dni.

Redacta todos tus resultados y utiliza LaTeX cuando pertoque.

```
dni = c(5, 4, 2, 0, 1, 5, 6, 7)
```

```
# Potencia
dni^2
```

```
## [1] 25 16  4  0  1 25 36 49
```

```
# Raiz
sqrt(dni)
```

```
## [1] 2.236068 2.000000 1.414214 0.000000 1.000000 2.236068 2.449490 2.645751
```

```
# Suma
sum(dni)
```

```
## [1] 30
```

3. Considera el vector de las letras de tu nombre y apellido. Llámalo name. Por ejemplo,

```
name = (M, A, R, I, A, S, A, N, T, O, S)
```

Define dicho vector en R. Calcula el subvector que solo contenga tu nombre. Calcula también el subvector que contenga solo tu apellido. Ordénalo alfabéticamente. Crea una matriz con este vector.

Redacta todos tus resultados y utiliza LaTeX cuando pertoque.

```
name = c("V", "I", "C", "T", "O", "R", "L", "O", "P", "E", "Z")
name1 = name[1:6]
surname = name[7:length(name)]
ordenado = sort(name)
matrix = matrix(name, nrow = 4, ncol = 3, byrow = TRUE)
```

```
## Warning in matrix(name, nrow = 4, ncol = 3, byrow = TRUE): la longitud de los
## datos [11] no es un submúltiplo o múltiplo del número de filas [4] en la matriz
```

```
name1
```

```
## [1] "V" "I" "C" "T" "O" "R"
```

```
surname
```

```
## [1] "L" "O" "P" "E" "Z"
```

```
ordenado
```

```
## [1] "C" "E" "I" "L" "O" "O" "P" "R" "T" "V" "Z"
```

```
matrix
```

```
##      [,1] [,2] [,3]
## [1,] "V"  "I"  "C"
## [2,] "T"  "O"  "R"
## [3,] "L"  "O"  "P"
## [4,] "E"  "Z"  "V"
```