

Tarea 3, 1-6

Téllez Gerardo Rubén

6/4/2021

1. Cread un vector llamado “Harry” formado por la sucesión de números consecutivos entre el -10 y 27. Pedidle a R que os devuelva el elemento de índice 7. Escribid el resultado.

```
Harry <- c(-10:27)
Harry[7]
```

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

2. Dad el máximo de la sucesión $(100 \cdot 2^n - 7 \cdot 3^n)$ con $n=0, \dots, 200$

```
sucesion1 <- function(n) {100*2^n-7*3^n}
A <- sucesion1(0:200)
max(A)
```

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

3. Cread la sucesión de números consecutivos entre 0 y 40. A continuación, cread el vector $(3 \cdot 5^n - 1)$ con $n=0, \dots, 40$. Ponedle como nombre x. Ahora, dad el subvector de los elementos que son estrictamente mayores que 3.5

```
s = 0:40
sucesion2 <- function(n) {3*5^n - 1}
x <- sucesion2(s)
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. Cread una función que os devuelva la parte real, la imaginaria, el módulo, el argumento y el conjugado de un número, mostrando solo 2 cifras significativas

RECOMENDACIÓN: En algún momento os hará falta utilizar vectores

```
funcion1 <- function(n) {print(list(Parte_real = Re(n),
                                   Parte_imaginaria = Im(n),
                                   Modulo = Mod(n),
                                   Argumento = Arg(n),
                                   Conjugado = Conj(n)), 2)}

funcion1(8.91+3.21i)
```

```
## $Parte_real
## [1] 8.9
##
## $Parte_imaginaria
## [1] 3.2
##
## $Modulo
## [1] 9.5
##
## $Argumento
## [1] 0.35
##
## $Conjugado
## [1] 8.9-3.2i
```

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

```
segundo_grado <- function(vect) {c((( -vect[2] + sqrt(vect[2]^2 - 4*vect[1]*vect[3]))/2*vect[1]), ((-vect[2] - sqrt(vect[2]^2 - 4*vect[1]*vect[3]))/2*vect[1]))}

segundo_grado(c(1, -5, 6))
```

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

RECOMENDACIÓN: En algún momento os hará falta utilizar vectores

6. Tomando el vector $vec = c(0,9,98,2,6,7,5,19,88,20,16,0)$, dad 3 opciones diferentes para calcular el subvector $c(9,19,20,16)$

```
vec <- c(0,9,98,2,6,7,5,19,88,20,16,0)

# Calcular c(9,19,20,16)
#1:
vec[c(2, 8, 10, 11)]
```

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

```
#2:
vec[vec == 9 | vec == 19 | vec == 20 | vec == 16]
```

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

```
#3  
vec[vec >= 9 & vec <=20]
```

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

```
#4  
vec[!vec > 20 & !vec < 9 ]
```

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

Tomando el vector vec definido en el apartado anterior, buscad

- 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

```
#Entradas pares  
vec[(vec %% 2) == 0]
```

```
## [1] 0 98 2 6 88 20 16 0
```

```
#Entradas no pares y mayores que 20  
vec[(vec %% 2) != 0 & vec > 20]
```

```
## numeric(0)
```

```
#Dónde vec toma su valor máximo  
which.max(vec)
```

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

```
#Dónde toma vec sus valores mínimos  
which(vec == min(vec))
```

```
## [1] 1 12
```