

Laboratorio 1

Alex Avila

2 de junio de 2018

Practica 1

Nombre : Alex Avila Santos 20160332F

Ejercicio7:

(a) Almacena el siguiente vector:

```
f1<- c(13563,-14156,-14319,16981,12921,11979,9568,8833,-12968, 8133,2)
```

Entonces realiza lo siguiente:

#####. Muestra la salida todos los elementos de f1 que, cuando se eleva a una potencia de 75, no son infinitos. #####. Devuelve los elementos de f1, excluyendo aquellos que resultan en infinito negativo cuando se eleva a una potencia de 75

```
#Nuestro vector f1
f1 <- c(13563,-14156,-14319,16981,12921,11979,9568,8833,-12968, 8133)

#Creamos un vector para almacenar las potencias
f2 <- c(1,1,1,1,1,1,1,1,1,1)
#hacemos un for para elevar cada numero de f1 a una potencia de 75
for ( i in 1:75)
  for( j in 1:10){
    f2[j] <- f2[j]*f1[j]
    # print(f2[j])
  }

#nuestro for recorrera todo el vector para ver cuales son no infinitos y los imprimirá
for(k in 1:10 )
  if(!is.infinite(f2[k])){
    print(f2[k])
  }
```

```
## [1] 7.612764e+305
## [1] 3.644077e+298
## [1] 9.080657e+295
## [1] 1.856983e+293
```

```
#Ahora mostramos los elementos de f1, excluyendo aquellos que resultan en infinito negativo cuando se eleva a una potencia de 75
for(k in 1:10 )
  if(f2[k]>=0){
    print(f2[k])
  }
```

```
## [1] Inf
## [1] Inf
## [1] Inf
## [1] 7.612764e+305
```

```
## [1] 3.644077e+298
## [1] 9.080657e+295
## [1] 1.856983e+293
```

(b) Almacenamos la siguiente matriz 3 x 4 dada en el laboratorio, como el objeto varMatriz

```
varMatriz <- matrix(ncol = 4,nrow = 3, c(77875.40,-35466.25,-39803.81,27551.45,-73333.85, 55976.34, 23764.30,-70585.69,76694.82,47032.00,-36478.88,36599.69))
varMatriz

##           [,1]      [,2]      [,3]      [,4]
## [1,]  77875.40  27551.45  23764.30 -36478.88
## [2,] -35466.25 -73333.85  36599.69 -70585.69
## [3,] -39803.81  55976.34  76694.82  47032.00
```

Entonces realizamos lo siguiente:

#####. Identifica los índices específicos de las entradas de varMatriz que son NaN cuando se eleva varMatriz a una potencia de 65 y se divide por infinito.

```
##Creamos la matriz para almacenar la potencia 65
potMatriz <- matrix(ncol = 4,nrow = 3,data = 1)

##calculamos la potencia 65 de la matriz varMatriz
for ( i in 1:65)
  for( j in 1:12){
    potMatriz[j] <- potMatriz[j]*varMatriz[j]
    #print(potMatriz[j])
  }
##mostramos la matriz
potMatriz

##           [,1]      [,2]      [,3]      [,4]
## [1,]      Inf  4.068084e+288  2.723496e+284 -3.409551e+296
## [2,] -5.470067e+295      -Inf  4.226992e+296      -Inf
## [3,] -9.887851e+298      Inf      Inf  5.076613e+303

## Identificando los índices específicos de las entradas de potMatriz que son NaN cuando se divide por
for(k in 1:12 )
  if(is.nan(potMatriz[k]/Inf)){
    print(k)
  }

## [1] 1
## [1] 5
## [1] 6
## [1] 9
## [1] 11
```

#####. Devuelve los valores en varMatriz que NO son NaN cuando se eleva varMatriz a una potencia de 67 y se añade infinito al resultado. Confirma que esto es idéntico a identificar aquellos valores en varMatriz que, cuando aumentan a una potencia de 67, no son iguales al infinito negativo.

```
##Creamos la matriz para almacenar la potencia 67
potMatriz67 <- matrix(ncol = 4,nrow = 3,data = 1)

##calculamos la potencia 65 de la matriz varMatriz
for ( i in 1:67)
```

```

for( j in 1:12){
  potMatriz67[j] <- potMatriz67[j]*varMatriz[j]
  #print(potMatriz[j])
}
##Identificamos los valores en varMatriz que no son NaN cuando se eleva varMatriz a una potencia
for(k in 1:12 )
  if(!is.nan(potMatriz67[k]+Inf)){
    print(varMatriz[k])
  }

```

```

## [1] 77875.4
## [1] -35466.25
## [1] -39803.81
## [1] 27551.45
## [1] 55976.34
## [1] 23764.3
## [1] 36599.69
## [1] 76694.82
## [1] -36478.88
## [1] 47032

```

```

##Confirmamos que esto es idéntico a identificar aquellos valores en varMatriz que, cuando aumentamos la potencia
for(k in 1:12 )
  if(!(is.infinite(potMatriz67[k]) && potMatriz67[k]<0)){
    print(varMatriz[k])
  }

```

```

## [1] 77875.4
## [1] -35466.25
## [1] -39803.81
## [1] 27551.45
## [1] 55976.34
## [1] 23764.3
## [1] 36599.69
## [1] 76694.82
## [1] -36478.88
## [1] 47032

```

```

print("-----")

```

```

## [1] "-----"

```

```

print("Comprobado")

```

```

## [1] "Comprobado"

```

#####. Identifique los valores en varMatriz que sean infinito negativo o finito cuando eleva varMatriz a una potencia de 67.

```

for(k in 1:12 )
  ##imprimos los valores de la matriz que cuando elevamos a la potencia 67 sean infinitos negativos
  if((is.infinite(potMatriz67[k]) && potMatriz67[k]<0) || is.finite(potMatriz67[k])){
    print(varMatriz[k])
  }

```

```

## [1] -35466.25
## [1] -39803.81

```

```
## [1] 27551.45
## [1] -73333.85
## [1] 23764.3
## [1] 36599.69
## [1] -36478.88
## [1] -70585.69
```

(c) Considere la siguiente línea de código:

```
f2 <- c(4.3,2.2,NULL,2.4,NaN,3.3,3.1,NULL,3.4,NA)
```

Decide cuál de las siguientes afirmaciones son verdaderas y cuáles son falsas y luego usa R para confirmar:

#####. La longitud de f2 es 8. #####. Llamando a `which(x=is.na(x=f2))`, no resultará en 4 y 8.
#####. Verificando `is.null(x=f2)`, proporciona la localización de dos valores NULL, presentes.

Respuesta VERDADERO la longitud de f2 es 8.

```
length(f2)
```

```
## [1] 8
```

Respuesta FALSO el llamado a `which(x=is.na(x=f2))`, sí resultará en 4 y 8.

```
which(x=is.na(x=f2))
```

```
## [1] 4 8
```

Respuesta FALSO Verificando `is.null(x=f2)`, no proporciona la localización de dos valores NULL, presentes.

```
is.null(x=f2)
```

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