Parcial Segundo Corte

Santiago Lozano

24 de abril de 2020

1)

Escriba un loop que tome un vector de números `all_pos` y que imprima `TRUE` si todos los elementos de la lista son positivos y `FALSE` en otro caso, cree el código y pruébelo con un ejemplo

```
## [1]   1  -5   8 -76   3  -5   8  -6
```

y despliegue

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

2)

Escriba un código que a partir de un vector de números, cree otro vector que sólo contenga los valores positivos del vector inicial

```
## [1]   1  -5   8 -76   3  -5   8  -6
```

y despliegue

```
## [1] 1 8 3 8
```

3)

Escriba un código que a partir de un vector de números cree un nuevo vector `nl` en la que el i-ésimo elemento de `nl` tiene el valor `TRUE` si el i-ésimo elemento del vector inicial tiene un valor positivo y `FALSE` en otro caso.

```
## [1]   1  -5   8 -76   3  -5   8  -6
```

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

4)

Escriba una función que determine si un dato pertenece a un vector

```
nombre.de.funcion(vector,elemento.del.vector)
```

despliegue

```
ubicacion.del.elemento.en.el.vector
```

5)

Cierto profesor califica de 0 a 100 un examen y da notas cualitativas a este de la siguiente manera entre 90-100:A,80-89:B, 70-79:C,60-69:D,menos de 60:F. Escriba un programa que le pida al usuario la nota del 0 a 100 del examen y este le devuelva la nota cualitativa

`nombre.de.funcion(95)`

despliegue

`A`

6)*

Hace una clase vimos lo que hacía la función `match` donde

```
primero <- c(5,8,3,5,3,6,4,4,2,8,8,8,4,4,6)
segundo <- c(8,6,4,2)
match(primero,segundo)
```

```
##  [1] NA  1 NA NA NA  2  3  3  4  1  1  1  3  3  2
```

vemos que aquí la forma en cómo trabaja la función, así, ¿Dónde el 5 del primer vector aparece en el segundo vector?, no aparece, entonces R despiega NA en el output,¿dónde el 8 del primer vector aparece en el segundo? aparece en el primero, y el programa despliega 1, y así sucesivamente.

Cree un programa a pedal que haga esta mismo `match` y pruébelo con lo qvectores anteriormente mencionados

7)*

Haga una función que reciba una matriz cuadrada y retorne su misma matriz pero triangular superior

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    6   11   16   21
## [2,]    2    7   12   17   22
## [3,]    3    8   13   18   23
## [4,]    4    9   14   19   24
## [5,]    5   10   15   20   25
```

despliegue

```
M1 <- matrix(c(1,6,11,16,21,0,7,12,17,22,0,0,13,18,23,0,0,0,19,24,0,0,0,0,25),byrow=T,nrow = 5)
M1
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    6   11   16   21
## [2,]    0    7   12   17   22
## [3,]    0    0   13   18   23
## [4,]    0    0    0   19   24
## [5,]    0    0    0    0   25
```

8)

Escriba un progrma que use un `while` loop para investigar el número de términos antes que el producto

1x2x3x4x5x6x. . . ..

llegue a 10 millones

9)*

Implemente un juego de multiplicación donde use un `while` loop donde R le dé al usuario dos números entre 9 y 16 y le pregunte al usuario si los multiplica. El juego sólo debe acabar si el usuario tiene 5 respuestas correctas

10)*

Cree el siguiente dataframe

```
student.df = data.frame( nombre = c("Sue", "Eva", "Henry", "Jean"),
                         sexo = c("f", "f", "m", "m"),
                         anos = c(21,31,29,19))
student.df
```

```
##   nombre sexo anos
## 1    Sue    f   21
## 2    Eva    f   31
## 3  Henry    m   29
## 4   Jean    m   19
```

Use loops y condicionales para crear una nueva variable `hombre.joven` que sea de tipo `logical` y sea `TRUE` si el individuo es hombre menos de 20 años, de lo contrario sea `FALSE` y agregue esa variable al data frame de manera que obtenga

```
##   nombre sexo anos hombre.joven
## 1    Sue    f   21        FALSE
## 2    Eva    f   31        FALSE
## 3  Henry    m   29        FALSE
## 4   Jean    m   19         TRUE
```

11)*

Si usted tiene una urna de pelotas enumeradas del 1 al 100, el experimento consiste es sacar pelotas de la urna y ver el número y si este no es 55, devolver la pelota a la urna, lo que queremos saber es cuantas veces debo hacer el experimento antes de encontrar el número 55. Este experimento se debe hacer máximo 1000 veces, si pasan las 1000 repeticiones, despliegue un mensaje que diga que no acaó el juego (sugerencias: el comando `sample(1:100,1,replace=TRUE)` es el que recrea la situación de la urna)

12)

El archivo `rivers` el cual puede traer con la función `data(rivers)` da la distancia de ciertos ríos

```
data("rivers")
View(rivers)
```

Cree un loops que imprima

“rio corto” si el río es más pequeño de 500

“río largo” si el río es más grande de 2000

y el valor original si no está en ninguna de las opciones anteriores

13)

Importe el archivo `base22.txt` con la función `scan()` y convierta el archivo en un dataframe

14)

Importe el archivo `scan1.txt` con la función `scan()` y convierta el archivo en un data frame

15)

Importe el archivo `murders.txt` con la función `readLines()` y convierta el archivo en un dataframe

16)

Importe el archivo `scan2.txt` con la función `readLines()` y convierta el archivo en un dataframe