

De Luna Ocampo Yanina

Programación para Ciencia de Datos

Galindo Durán Cristal Karina

3AM1

Práctica3

### PROBLEMA1

"Escribe una función que dado un valor y un vector devuelva cuántas veces se encuentra el valor en el vector"

Creo una función para pedir la cantidad de valores que tendrá mi función y cuáles serán esos valores, creamos un vector vacío para almacenarlos ahí.

```
my_func <- function() {  
  vec <- c()  
  cat("Digita el numero de valores que tendra la funcion")  
  n <- readLines(n = 1)  
  n <- as.integer(n)  
  
  cat("Digita los valores")  
  for(i in 1:n){  
    valor <- as.integer(readLines(n=1))  
    vec <- append(vec, valor)  
  }  
  return(vec)  
}
```

Creo otra función llamada repite para hacer el proceso de encontrar el número que se repite con la función dada. Mando llamar toda la función de arriba para poder solamente llamar a esta en la consola y se ejecute todo de una vez.

```
repite <- function(vec) {  
  my_func()  
  cat("Ingresa el numero el cual buscas que se repite")  
  r <- readLines(n = 1)  
  r <- as.numeric(r)  
  vec %in% r  
}
```

## PROBLEMA2

"Una función recibe como parámetro una matriz. La matriz representa ciclista en varias etapas. Cada fila representa una etapa. La primera columna de la matriz almacena el número de horas, la segunda columna el número de minutos y la tercera columna el número de segundos que tardó en completar 1 ejemplo, si se recibe:

```
2 30 50
1 55 20
```

El ciclista ha completado dos etapas. En la primera etapa ha tardado 2 horas, 30 minutos y 50 segundos. La función debe devolver una lista con el número total de horas, minutos y segundos empleados por el ciclista en cubrir el total de etapas. Para los datos de ejemplo se devolvería 4 horas, 26 minutos y 10 segundos."

Creamos nuestra función principal para mandarla llamar y solo llamar una función, esta leerá el número de tiempo que deseemos meter

```
mi_prin <- function(){
  c <- readline("Digite el numero de tiempos que registrara: \n")
  c <- as.integer(c)
  mi_mat(c)
}
```

Creo la función de la matriz, le asigno de valores para que genere números de 1 a la cantidad al cuadrado del número que metí arriba. Mis columnas solo serán 3 porque solo son horas, minutos y segundos. Digo el tiempo, lo asigno y utilizo apply para la función a través de nuestra matriz.

```
mi_mat <- function(c){
  mtz1 <- matrix(seq(1:c*c), nrow = c, ncol = 3, byrow = FALSE)
  for (i in c) {
    cat("Digite el tiempo: ")
    t1 <- readLines(n=3)
    t1 <- as.integer(t1)

    mtz1[i, ] <- t1
    t1 <- NULL
  }
  mtz1
  apply(mtz1, 2, sum)
}
```

### PROBLEMA3

"Una matriz es simétrica si es cuadrada y es igual a su transpuesta. Escribe una función que devuelva un valor lógico indicando si una matriz es simétrica."

Asignamos a la parte inferior de una matriz triangular superior, los valores de la parte superior. Con `lower.tri` obtenemos el triángulo inferior. Le asignamos la parte superior que **transponemos previamente** la matriz `t(df)`, por eso volvemos a usar `lower.tri`. Pues si usáramos `upper.tri` el resultado no sería una matriz simétrica.

```
mtzSimetrica <- function() {  
  df <- structure(list(v1 = c(0L, 0L, 0L, 0L, 0L),  
                      v2 = c(2L, 0L, 0L, 0L, 0L),  
                      v3 = c(3L, 6L, 0L, 0L, 0L),  
                      v4 = c(4L, 8L, 12L, 0L, 0L),  
                      v5 = c(5L, 10L, 15L, 20L, 0L)),  
                  .Names = c("v1", "v2", "v3", "v4", "v5"), class = "data.frame",  
  
  df + t(df)  
  df[lower.tri(df)] <- t(df)[lower.tri(df)]  
  df[lower.tri(df)] <- df[upper.tri(df)]  
  print(df)  
  print(t(df))  
  if(df==t(df)){  
    cat("La matriz es simetrica. \n")  
  }  
}
```

```
"data.frame", row.names = c("1", "2", "3", "4", "5"))
```

Bibliografía de apoyo:

<https://nubededatos.blogspot.com/2015/09/como-crear-una-matriz-simetrica-en-r.html>

### CONCLUSIONES

Con esta práctica reforzamos los conocimientos que hemos estado viendo en clase, como funciones, vectores, funciones de matrices, matrices, for.

También que hay que explorar más las funciones para poder entenderlas e ir mejorando poco a poco en el uso al momento de que nos pidan hacer algo diferente a lo visto en clase.

Se pueden crear varias cosas con cada una de las cosas vistas en clase, sin embargo hay que tener claros los conceptos.