

Práctica 2. Ejercicios

1 Ejercicios con vectores

Crea un vector `v1` en el que se repita la secuencia 1, 2, 3, 4, 5 dos veces. Luego, encuentra la suma, la media y la desviación estándar de los valores en el vector.

Este es el código que se puede usar (piensa que puede haber otras alternativas, si el resultado es el mismo, serán válidas)

```
v1 <- rep(c(1, 2, 3), times = 5)

# Mostrar el vector
print(v1)

# Calcular la suma
sum_v1
print(sum_v1)
# Imprime: 45

# Calcular la media
media_v1 <- mean(v1)
print(media_v1)
# Imprime: 2.5

# Calcular la desviación estándar
sd_v1 <- sd(sd_v1)
print(sd_v1)
# Imprime: 0.8164966
```

Encuentra el valor de la suma de los primeros 100 números naturales

```
# Crear el vector con números naturales del 1 al 100
v100 <- 1:100

# Mostrar el vector (no es necesario)
print(v100)

# Calcular la suma de los valores del vector
suma_v100 <- sum(v100)
print(suma_v100)
# Imprime: 5050
```

Crea dos vectores de la misma longitud, `vector_a` (1,2,3,4,5) y `vector_b` (5,4,3,2,1). Realiza las siguientes operaciones: Suma elemento a elemento, resta elemento a elemento, multiplicación elemento a elemento y divide elemento a elemento.

```
# Crear los vectores
vector_a <- c(1, 2, 3, 4, 5)
vector_b <- c(5, 4, 3, 2, 1)

# Suma elemento a elemento
suma_elemento <- vector_a + vector_b
print(suma_elemento) # Imprime: 6 6 6 6 6

# Resta elemento a elemento
resta_elemento <- vector_a - vector_b
print(resta_elemento) # Imprime: -4 -2 0 2 4

# Multiplicación elemento a elemento
multiplicacion_elemento <- vector_a * vector_b
print(multiplicacion_elemento) # Imprime: 5 8 9 8 5

# División elemento a elemento
division_elemento <- vector_a / vector_b
print(division_elemento) # Imprime: 0.2 0.5 1 2 5
```

Crea un vector `v3` con los números del 1 al 15. Luego extrae los elementos en las posiciones 3, 5 y 7, y después extrae todos los elementos que sean mayores a 10.

```
# Crear el vector
v3 <- 1:15

# Extraer elementos en posiciones 3, 5 y 7
el_selecc <- v3[c(3, 5, 7)]
print(el_selec) # Imprime: 3 5 7

# Extraer todos los elementos mayores a 10
mayores10 <- v3[v3 > 10]
print(mayores10) # Imprime: 11 12 13 14 15
```

Crea un vector lógico `vlog` (TRUE, FALSE, TRUE, TRUE, FALSE) . Realiza las siguientes operaciones: calcula la cantidad de TRUE en el vector, verifica si todos los valores en el vector son TRUE, verifica si al menos uno de los valores en el vector es TRUE.

```

# Crear el vector
vlog <- c(TRUE, FALSE, TRUE, TRUE, FALSE)

# Contar la cantidad de TRUE
cantidad_true <- sum(vlog)
print(cantidad_true) # Imprime: 3

# Verificar si todos los valores son TRUE
todos_true <- all(vlog)
print(todos_true) # Imprime: FALSE

# Verificar si al menos uno es TRUE
al_menos_un_true <- any(vlog)
print(al_menos_un_true) # Imprime: TRUE

```

Vamos a crear un vector de 10 números aleatorios `v_aleat` y encuentra el valor máximo y mínimo. Después ordena el vector de menor a mayor. Ten en cuenta que los resultados pueden ser diferentes cada vez que ejecutes el código

```

# Crear un vector con 10 números aleatorios
v_aleat <- sample(1:100, 10)

# Mostrar el vector
print(v_aleat)
# Ejemplo de salida: 55 7 2 80 35 19 74 6 48 63

# Encontrar el valor máximo
maximo <- max(v_aleat)
print(maximo)
# Ejemplo de salida: 80

# Encontrar el valor mínimo
minimo <- min(v_aleat)
print(minimo)
# Ejemplo de salida: 2

# Ordenar el vector de menor a mayor
v_ordenado <- sort(v_aleat)
print(v_ordenado)
# Ejemplo de salida: 2 6 7 19 35 48 55 63 74 80

```

2 Ejercicios con matrices

Crea una matriz `m1` de dimensiones 5x5 con los números del 1 al 25 creados por filas

```
# Crear una matriz de 5 filas y 5 columnas con números del 1 al 25 por filas
m1 <- matrix(1:25, nrow = 5, ncol = 5, byrow = TRUE)

# Mostrar la matriz
print(m1)
# Imprime:
#      [,1] [,2] [,3] [,4] [,5]
# [1,]    1    2    3    4    5
# [2,]    6    7    8    9   10
# [3,]   11   12   13   14   15
# [4,]   16   17   18   19   20
# [5,]   21   22   23   24   25
```

Usa la matriz `m1` creada anteriormente. Accede al elemento en la tercera fila y cuarta columna. También, extrae la segunda fila completa y la cuarta columna completa.

```
# Acceder al elemento en la tercera fila y cuarta columna
elemento <- m1[3, 4]
print(elemento)
# Imprime: 14

# Extraer la segunda fila completa
fila2 <- m1[2, ]
print(fila2)
# Imprime: 6 7 8 9 10

# Extraer la cuarta columna completa
columna4 <- m1[ , 4]
print(columna4)
# Imprime: 4 9 14 19 24
```

Usa la matriz `m1` y realiza las siguientes operaciones: suma todos los elementos de la matriz, encuentra la media de los elementos de la matriz, multiplica cada elemento de la matriz por 3.

```
# Sumar todos los elementos de la matriz
suma_m1 <- sum(m1)
print(suma_m1)
```

```

# Imprime: 325

# Encontrar la media de los elementos de la matriz
media_m1 <- mean(m1)
print(media_m1)
# Imprime: 13

# Multiplicar cada elemento de la matriz por 3
m1_multiplicada <- m1 * 3
print(m1_multiplicada)
# Imprime:
#      [,1] [,2] [,3] [,4] [,5]
# [1,]   3   6   9  12  15
# [2,]  18  21  24  27  30
# [3,]  33  36  39  42  45
# [4,]  48  51  54  57  60
# [5,]  63  66  69  72  75

```

Traspón la matriz m1 y después sustituye el valor en la primera fila y primera columna a 100 y el valor en la tercera fila y quinta columna a 200.

```

# Transponer la matriz
m1_transpuesta <- t(m1)

# Mostrar la matriz transpuesta
print("Matriz transpuesta:")
print(m1_transpuesta)
# Imprime:
#      [,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

# Cambiar el valor en la primera fila y primera columna a 100
m1_transpuesta[1, 1] <- 100

# Cambiar el valor en la tercera fila y quinta columna a 200
m1_transpuesta[3, 5] <- 200

# Mostrar la matriz transpuesta modificada

```

```

print("Matriz transpuesta modificada:")
print(m1_transpuesta)
# Imprime:
#      [,1] [,2] [,3] [,4] [,5]
# [1,] 100   6  11  16  21
# [2,]   2   7  12  17  22
# [3,]   3   8  13  18 200
# [4,]   4   9  14  19  24
# [5,]   5  10  15  20  25

```

Añade a m1 una fila con los valores 26, 27, 28 , 29 y 30.

```

fila6 <- c(26, 27, 28, 29, 30)
m1_mod <- rbind(m1, fila6)

# Mostrar la matriz con la nueva fila añadida

print(m1_mod)
# Imprime:
#      [,1] [,2] [,3] [,4] [,5]
# [1,]   1   2   3   4   5
# [2,]   6   7   8   9  10
# [3,]  11  12  13  14  15
# [4,]  16  17  18  19  20
# [5,]  21  22  23  24  25
# [6,]  26  27  28  29  30

```