

```
#Manuel Tristán Martín Ruiz
```

```
#Practica 1
```

```
# Ej.1
```

```
x <- c(1,2,3,4,5)
```

```
y <- c(9.1,2.4,7.5,1.3,3.4)
```

```
# Hablamos de un vector como de una estructura que almacena entre parentesis datos del mismo tipo (numéricos, cadena de caracteres y/o lógicos.), en este caso numéricos. Se trata de la estructura de datos más sencilla en R.
```

```
# La diferencia entre vector y lista, es que si bien son parecidas el vector puede almacenar solamente datos del mismo tipo mientras que la lista permite almacenar datos de diferente tipo.
```

```
# la longitud de X es:5 y la de y es 5 también
```

```
length(x)
```

```
length(y)
```

```
# La diferencia es que integer solamente puede almacenar números enteros, sin decimales, mientras que double puede almacenar numeros reales con decimales.
```

```
# Un vector que almacene valores enteros será computado más facilmente que uno que almacene datos decimales, y ocupará menos memoria. Pero un tipo de datos double puede ser más preciso.
```

```
# Ej.2
```

```
valor_suma <- x+y
```

```
print(valor_suma)
```

```
valor_resta <- x-y
```

```
print(valor_resta)
```

```
valor_multiplicación <- x*y
```

```
print(valor_multiplicación)
```

```
valor_división <- x/y
```

```
print(valor_división)
```

```
valor_elevación <- x^y
```

```
print(valor_elevación)
```

```
# Ej.3
```

```
valor_elevación_cuadrado <- y^2
```

```
print(valor_elevación_cuadrado)
```

```
#Ej.4
```

```
valor_multiplicación_10 <- y*10
```

```
print(valor_multiplicación_10)
```

```
#Ej.5
```

```
valor_suma_25 <- y+25
```

```
print(valor_suma_25)
```

```
#Ej.6
```

```
maximo_y <- max(y)
```

```
print(maximo_y)
```

```
#Ej.7
```

```
minimo_y <- min(y)
```

```
print(minimo_y)
```

```
#Ej.8
```

```
absoluto_y <- abs(y[1])
```

```
print(absoluto_y)
```

```
#Ej.9
```

```
raiz_y <- sqrt(y)
```

```
print(raiz_y)
```

```
#Ej.10
```

```
suma_y <- sum(y)
```

```
print(suma_y)
```

```
#Ej.11
```

```
lista <- c(x,y)
```

```
print(lista)
```

```
length(lista)
```

```
# la longitud de este objeto "lista" es 10
```

```
class(lista)
```

```
# Utiliza el tipo de dato double, porque al crearse un nuevo vector numérico con números  
decimales le asigna por defecto ese tipo de dato. Aunque los números de x sean  
entremos, se transforman al tipo double para que la lista pueda contenerlos a todos  
juntos.
```

```
#Ej.12
```

```
is.na(lista)
```

```
#No existen valores "NA" en el objeto "lista"
```

```
#Ej.13
```

```
z <- c(29,NA,12,46,73)
```

```
#Ej.14
```

```
media_z <- mean(z)
```

```
print(media_z)
```

```
#Da como resultado NA, porque hay un valor no asignado
```

```
is.na(media_z)
```

```
media_z <- mean(z, na.rm = TRUE)
```

```
print(media_z)
```

```
#Ej.15
```

```
matriz <- matrix(c(24,69,3,90,23,56,1,63,87,21,77,19),
```

```
          nrow = 4, ncol = 3)
```

```
print(matriz)
```

```
#Los valores se rellenan según las columnas [1,1], [2,1], [3,1], [4,1] y así con las columnas  
2 y 3.
```

#Ej.16

```
matriz_filas <- matrix(c(24,69,3,90,23,56,1,63,87,21,77,19),  
  nrow = 4, ncol = 3,  
  byrow = T)  
print(matriz_filas)
```

#Ej.17

```
a <- matrix(c(1:9),  
  nrow = 3, ncol = 3)  
print(a)  
b <- matrix(c(10:18),  
  nrow = 3, ncol = 3)  
print(b)
```

#Ej.18

```
sumar_matriz <- a+b  
print(sumar_matriz)
```

```
restar_matriz <- a-b  
print(restar_matriz)
```

#Resta los valores dentro de la matriz posición por posición/ elemento a elemento, eso es lo que provoca el resultado.

#Ej.19

```
matriz_t <- t(matriz)  
print(matriz_t)
```

#Ej.20

```
matriz_mult_constante <- 3*matriz  
print(matriz_mult_constante)
```

#Ej.21

```
matriz_mult <- a*b
```

```
print(matriz_mult)
```

#Ej.22

```
print(matriz_mult[[2,3]])
```

```
print(matriz_mult[2,])
```

```
print(matriz_mult[,3])
```

#Practica 2

#Ej.1

```
set.seed(123)
```

```
num_cuentas <- round(rnorm(100, mean = 50, sd = 10))
```

```
print(num_cuentas)
```

#Ej.2

```
mean(num_cuentas)
```

#Ej.3

```
median(num_cuentas)
```

#Ej.4

```
moda <- as.numeric(names(sort(table(num_cuentas), decreasing = TRUE)[1]))
```

```
print(paste("la moda de las cuentas en el yacimiento es:", moda))
```

#Ej.5

```
rango <- max(num_cuentas)-min(num_cuentas)
```

```
print(rango)
```

#Ej.6

```
primer_cuartil <- quantile(num_cuentas, probs = 0.25)
print(primer_cuartil)
```

#Ej.7

```
percentil_75 <- quantile(num_cuentas, probs = 0.75)
print(percentil_75)
```

#Ej.8

```
varianza <- var(num_cuentas)
print(varianza)
```

#Ej.9

```
desviación_estandar <- sd(num_cuentas)
print(desviación_estandar)
```

#Ej.10

```
library(ggplot2)
```

```
hist(num_cuentas,
      main = "Histograma de frecuencia",
      xlab = "Número de cuentas",
      ylab = "Frecuencia",
      col = "blue",
      border = "black",
      breaks = 10)
```

#Ej.11

```
boxplot(num_cuentas,
        main = "Diagrama de caja",
        ylab = "Número de cuentas",
```

```

    col = "lightblue",
    border = "black")

#Ej.12
densidad <- density(num_cuentas)
plot(densidad,
     main = "Gráfico de densidad de cuentas",
     xlab = "Número de cuentas",
     ylab = "Densidad")

#Ej.13
barplot(num_cuentas,
     main = "Gráfico de Barras",
     xlab = "Intervalos",
     ylab = "Frecuencia",
     col = "lightblue",
     border = "black")

#Ej.14
tipo_artefacto <- sample(c("Vajilla", "Fibula", "Cuentas", "Monedas", "Tinajas"), 10, replace
= TRUE)

material <- sample(c("Cerámica", "Metal", "Vidrio", "Piedra"), 10, replace = TRUE)

periodo_cultural <- sample(c("Romano Republicano", "Romano Imperial",
"Tardoantigüo","Emiral","Calífal"), 10, replace = TRUE)

estado_conservacion <- sample(c("Muy Bueno","Bueno", "Regular", "Malo"), 10, replace =
TRUE)

ubicacion <- sample(c("Granada", "Mertola", "Córdoba","Málaga"), 10, replace = TRUE)

Datos_ej_14 <- data.frame(
  tipo_artefacto = tipo_artefacto,
  material = material,
  periodo_cultural = periodo_cultural,
  estado_conservacion = estado_conservacion,
  ubicacion = ubicacion

```

```
)
```

```
print(Datos_ej_14)
```

```
View(Datos_ej_14)
```

```
tabla_tipo_artefacto <- table(tipo_artefacto)
```

```
View(tabla_tipo_artefacto)
```

```
tabla_material <- table(material)
```

```
View(tabla_material)
```

```
tabla_periodo_cultural <- table(periodo_cultural)
```

```
View(tabla_periodo_cultural)
```

```
tabla_estado_conservacion <- table(estado_conservacion)
```

```
View(tabla_estado_conservacion)
```

```
tabla_ubicacion <- table(ubicacion)
```

```
View(tabla_ubicacion)
```