#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 trasnforman 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","Califal"), 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)