#Name: ALEENA JOBY

#Register No: 22122303

#Date:- 15/05/2023

SECTION A

#1 - Create a vector ‘x’ with values 1, 2, 3, 4, 5 and a vector ‘y’ with values 2, 4, 6, 8, 10. By using family functions, find out the mean, median, sum, maximum and minimum of both vectors.

#Answer:

```
# create vectors x and y

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

y <- c(2, 4, 6, 8, 10)


# find mean of vectors x and y

mean_x <- mean(x)

mean_y <- mean(y)


# find median of vectors x and y

median_x <- median(x)

median_y <- median(y)


# find sum of vectors x and y

sum_x <- sum(x)
```

```
sum_y <- sum(y)


# find maximum of vectors x and y
max_x <- max(x)
max_y <- max(y)


# find minimum of vectors x and y
min_x <- min(x)
min_y <- min(y)


# print the results
cat("Vector x: ", x, "\n")
cat("Vector y: ", y, "\n")
cat("Mean of x: ", mean_x, "\n")
cat("Mean of y: ", mean_y, "\n")
cat("Median of x: ", median_x, "\n")
cat("Median of y: ", median_y, "\n")
cat("Sum of x: ", sum_x, "\n")
cat("Sum of y: ", sum_y, "\n")
cat("Maximum of x: ", max_x, "\n")
cat("Maximum of y: ", max_y, "\n")
```

```
cat("Minimum of x: ", min_x, "\n")

cat("Minimum of y: ", min_y, "\n")
```

OUTPUT

```
> cat("Vector x: ", x, "\n")

Vector x:  1 2 3 4 5

> cat("Vector y: ", y, "\n")

Vector y:  2 4 6 8 10

> cat("Mean of x: ", mean_x, "\n")

Mean of x:  3

> cat("Mean of y: ", mean_y, "\n")

Mean of y:  6

> cat("Median of x: ", median_x, "\n")

Median of x:  3

> cat("Median of y: ", median_y, "\n")

Median of y:  6

> cat("Sum of x: ", sum_x, "\n")

Sum of x:  15

> cat("Sum of y: ", sum_y, "\n")

Sum of y:  30

> cat("Maximum of x: ", max_x, "\n")
```

Maximum of x: 5

> cat("Maximum of y: ", max_y, "\n")

Maximum of y: 10

> cat("Minimum of x: ", min_x, "\n")

Minimum of x: 1

> cat("Minimum of y: ", min_y, "\n")

Minimum of y: 2

#2 - Write a R program to create a matrix taking a given vector of numbers as input and define the column and row names. Display the matrix.

#Answer:

Define input vector

input_vector <- c(1, 2, 3, 4, 5, 6)

Define column and row names

column_names <- c("A", "B")

row_names <- c("C", "D", "E")

Create matrix with given input vector and names

matrix_output <- matrix(input_vector, nrow = length(row_names), ncol = length(column_names), byrow = TRUE, dimnames = list(row_names, column_names))

Display matrix

print(matrix_output)

OUTPUT

A B

C 1 2

D 3 4

E 5 6

#3 - Write a R program to print the numbers from 1 to 100 and print "multiple of 3" for multiples of 3, print "multiple of 5" for multiples of 5, and print "multiple of both 3 & 5" for multiples of both.

```
for (i in 1:100) {

 if (i %% 3 == 0 && i %% 5 == 0) {

  print("multiple of both 3 & 5")

 } else if (i %% 3 == 0) {

  print("multiple of 3")

 } else if (i %% 5 == 0) {

  print("multiple of 5")

 } else {

  print(i)

 }
```

}

OUTPUT

```
[1] 1
[1] 2
[1] "multiple of 3"
[1] 4
[1] "multiple of 5"
[1] "multiple of 3"
[1] 7
[1] 8
[1] "multiple of 3"
[1] "multiple of 5"
[1] 11
[1] "multiple of 3"
[1] 13
[1] 14
[1] "multiple of both 3 & 5"
[1] 16
[1] 17
[1] "multiple of 3"
[1] 19
[1] "multiple of 5"
[1] "multiple of 3"
[1] 22
[1] 23
[1] "multiple of 3"
[1] "multiple of 5"
[1] 26
[1] "multiple of 3"
[1] 28
[1] 29
[1] "multiple of both 3 & 5"
[1] 31
[1] 32
[1] "multiple of 3"
[1] 34
[1] "multiple of 5"
[1] "multiple of 3"
[1] 37
```

```
[1] 38
[1] "multiple of 3"
[1] "multiple of 5"
[1] 41
[1] "multiple of 3"
[1] 43
[1] 44
[1] "multiple of both 3 & 5"
[1] 46
[1] 47
[1] "multiple of 3"
[1] 49
[1] "multiple of 5"
[1] "multiple of 3"
[1] 52
[1] 53
[1] "multiple of 3"
[1] "multiple of 5"
[1] 56
[1] "multiple of 3"
[1] 58
[1] 59
[1] "multiple of both 3 & 5"
[1] 61
[1] 62
[1] "multiple of 3"
[1] 64
[1] "multiple of 5"
[1] "multiple of 3"
[1] 67
[1] 68
[1] "multiple of 3"
[1] "multiple of 5"
[1] 71
[1] "multiple of 3"
[1] 73
[1] 74
[1] "multiple of both 3 & 5"
[1] 76
[1] 77
```

[1] "multiple of 3"
[1] 79
[1] "multiple of 5"
[1] "multiple of 3"
[1] 82
[1] 83
[1] "multiple of 3"
[1] "multiple of 5"
[1] 86
[1] "multiple of 3"
[1] 88
[1] 89
[1] "multiple of both 3 & 5"
[1] 91
[1] 92
[1] "multiple of 3"
[1] 94
[1] "multiple of 5"
[1] "multiple of 3"
[1] 97
[1] 98
[1] "multiple of 3"
[1] "multiple of 5"

#4 - Create a data frame which have minimum three columns and ten rows

#a) Extract the 3rd and 5th rows with 1st and 3rd columns from the data frame.

#b) Write a R program to add a new column into the given data frame.

#c) Write a R program to sort the given data frame by a particular variable.

#Answer:

create the data frame

my_df <- data.frame(A = 1:10, B = letters[1:10], C = rnorm(10))

print the data frame

my_df

```
   A B           C
1   1 a  0.80628211
2   2 b  0.69806982
3   3 c  0.97433653
4   4 d -0.05276501
5   5 e -0.86037631
6   6 f -1.11880229
7   7 g -0.02334576
8   8 h -0.85615983
9   9 i  1.43085898
10 10 j  0.88281552
```

#a) extract the rows and columns

new_df <- my_df[c(3, 5), c(1, 3)]

print the new data frame

new_df

```
  A          C
3 3  0.9743365
5 5 -0.8603763
```

#b) To add a new column to the data frame my_df:

create the new column

my_df$D <- rpois(10, 2)

my_df

```
   A B           C D
1   1 a  0.80628211 4
2   2 b  0.69806982 4
3   3 c  0.97433653 2
4   4 d -0.05276501 1
5   5 e -0.86037631 2
```

6   6 f -1.11880229   0
7   7 g -0.02334576 0
8   8 h -0.85615983 2
9   9 i 1.43085898   4
10 10 j   0.88281552 1

#c) To sort the data frame by a particular variable (e.g. column C):

sort the data frame by column C

sorted_df <- my_df[order(my_df$C),]

print the sorted data frame

sorted_df

```
   A B          C       D
6   6 f -1.11880229   0
5   5 e -0.86037631  2
8   8 h -0.85615983  2
4   4 d -0.05276501  1
7   7 g -0.02334576  0
2   2 b  0.69806982  4
1   1 a  0.80628211  4
10 10 j   0.88281552 1
3   3 c  0.97433653  2
9   9 i  1.43085898  4
```

#5 - Create a function "Rectangle" which takes "length" and "width" of the rectangle and returns area and perimeter of that rectangle.

#Answer :

define the function "Rectangle"

Rectangle <- function(length, width) {

calculate the area of the rectangle

area <- length * width

calculate the perimeter of the rectangle

perimeter <- 2*(length + width)

return the area and perimeter as a list

  return(list(area = area, perimeter = perimeter))

}

test the function with example inputs

result <- Rectangle(5, 7)

print(result)

OUTPUT

$area
[1] 35
$perimeter
[1] 24

SECTION B

#6 -

 # a) Give a small description about the dataset.

#The Iris dataset is a collection of measurements of 50 flowers from three different species of irises.

#The measurements include the sepal length, sepal width, petal length, and petal width.

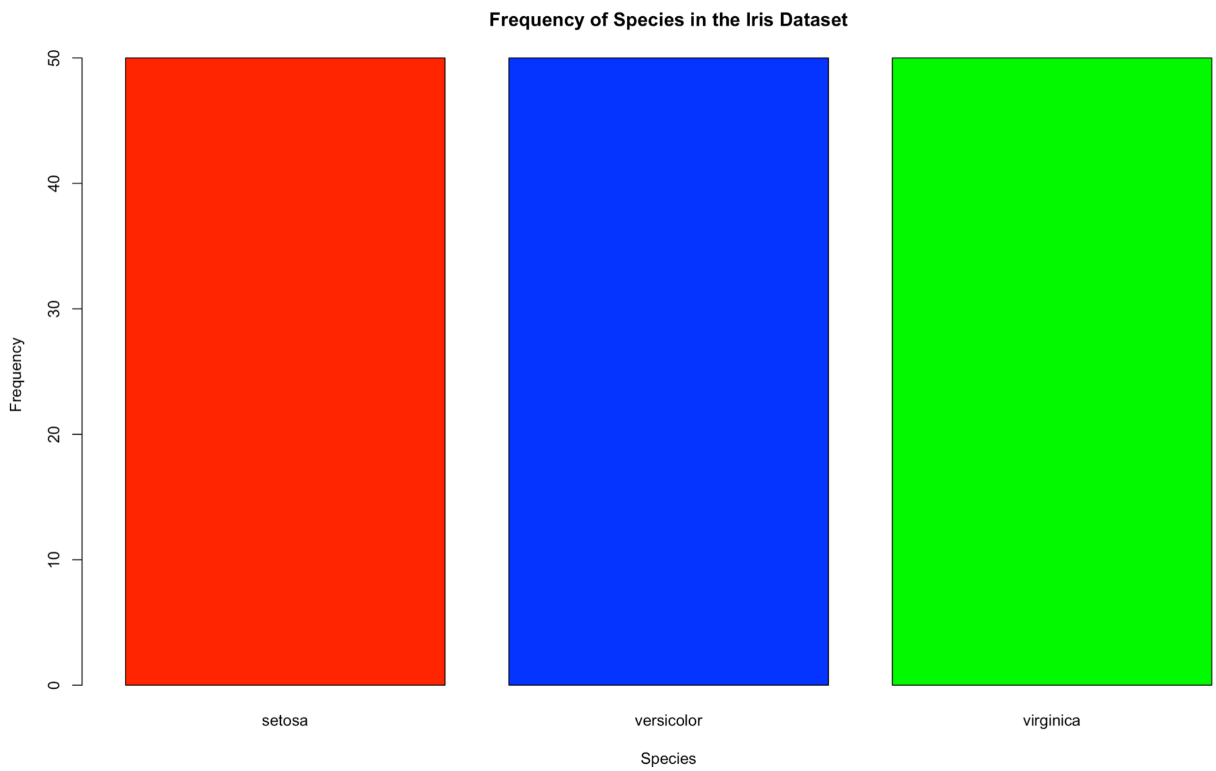
#The three species of irises are Iris setosa, Iris versicolor, and Iris virginica.

 # b) Create a new variable "sepal_ratio" that is the ratio of sepal length to sepal width.

sepal_ratio <- iris$Sepal.Length / iris$Sepal.Width

 # c) Then, create a boxplot of sepal_ratio by Species with proper explanation of the output.

boxplot(sepal_ratio ~ Species, data = iris)



#The boxplot shows that the sepal ratio is different for the three species of irises.

#The sepal ratio for Iris setosa is the smallest, followed by Iris versicolor, and then Iris virginica.

#This suggests that the sepals of Iris setosa are narrower relative to their length than the sepals of the other two species.

 # d) Find out the correlation between all the variable and plot the correlation.

cor(iris[, 1:4])

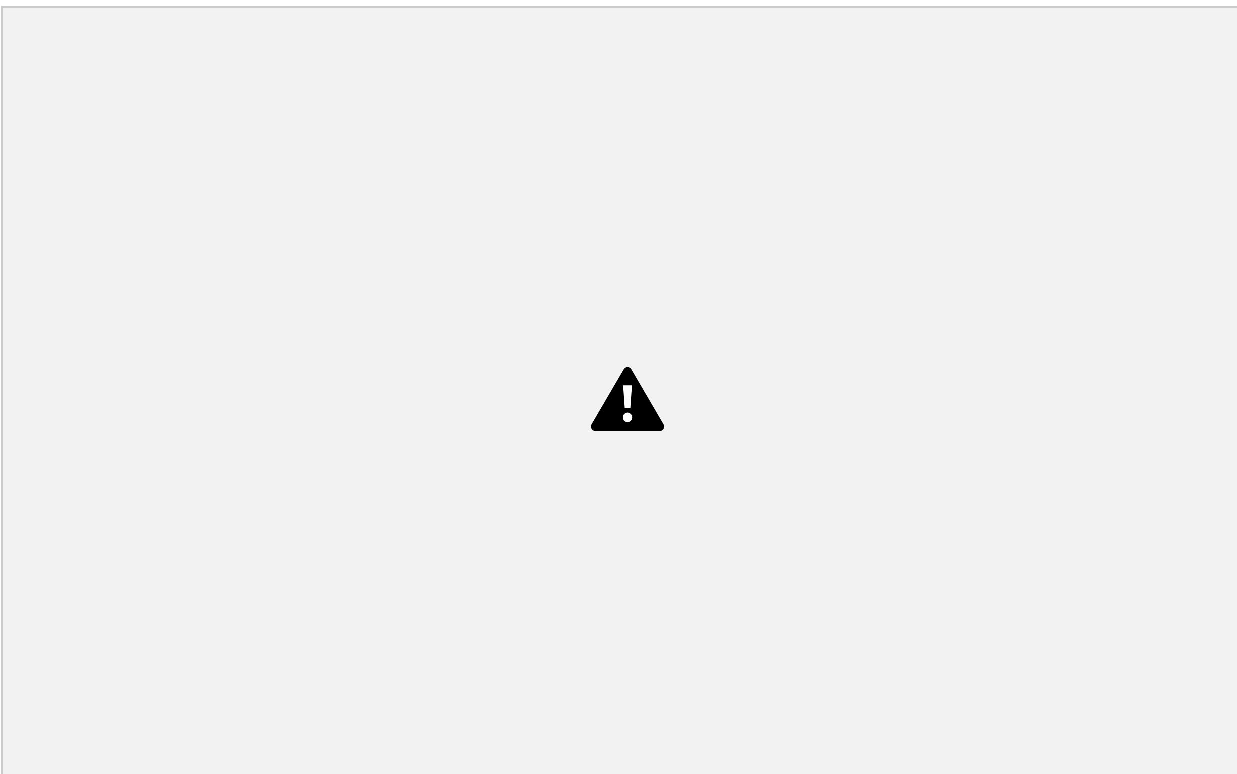
#The correlation matrix shows that the sepal length and sepal width are strongly correlated,

#as are the petal length and petal width.

#The sepal length and petal length are also correlated, but to a lesser extent.

#The sepal width and petal width are not correlated.

plot(iris[, 1:4])



#The scatterplot matrix shows the relationships between the four variables.

#The sepal length and sepal width are positively correlated, as are the petal length and petal width.

#The sepal length and petal length are also positively correlated, but to a lesser extent.

#The sepal width and petal width are not correlated.

e) Write the conclusion based on the correlation.

#The correlation matrix and scatterplot matrix show that the sepal length and sepal width are strongly correlated,

#as are the petal length and petal width. The sepal length and petal length are also correlated, but to a lesser extent.

#The sepal width and petal width are not correlated.

#These results suggest that the four variables are not independent of each other.

#The sepal length and sepal width are likely to be correlated because they are both measurements of the same flower.

#The petal length and petal width are also likely to be correlated because they are both measurements of the same flower(correlation coefficient = 0.96).

#The sepal length and petal length are correlated because they are both measurements of the size of the flower.

#The sepal width and petal width are not correlated because they are measurements of different parts of the flower.

#7 - Come up with the appropriate visualization for below mentioned variables and write the inferences also

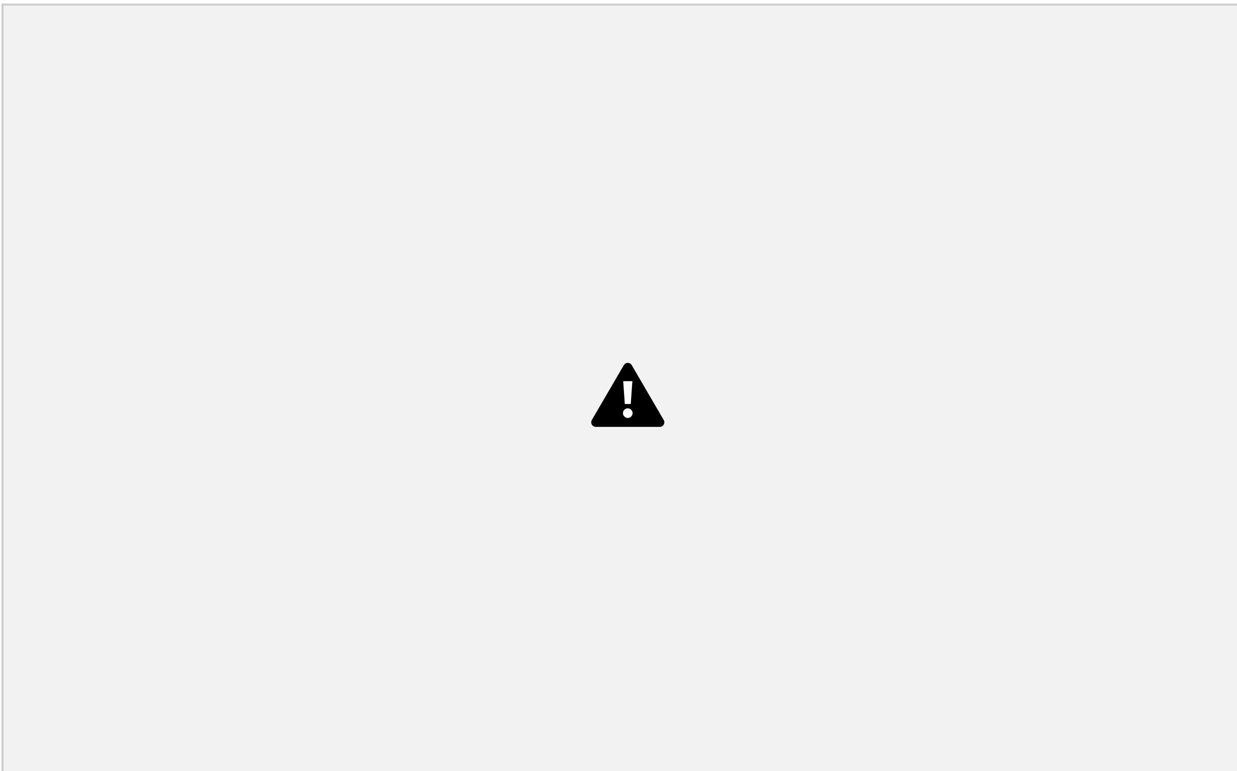
#a) Species

#The most appropriate visualization for the Species variable is a bar chart.

#A bar chart is a good way to compare the frequencies of different categories.

#In this case, the categories are the three species of irises.

#The bar chart will show how many flowers are in each species.



```
# Load the iris dataset

data(iris)


# Create a bar chart of the Species variable

barplot(table(iris$Species), col = c("red", "blue", "green"), main = "Frequency of Species in the Iris Dataset", xlab = "Species", ylab = "Frequency")
```

#Inference: From the bar plot, we can see that the three species in the dataset (Setosa, Versicolor, and Virginica) are equally represented,

#with 50 samples for each species.

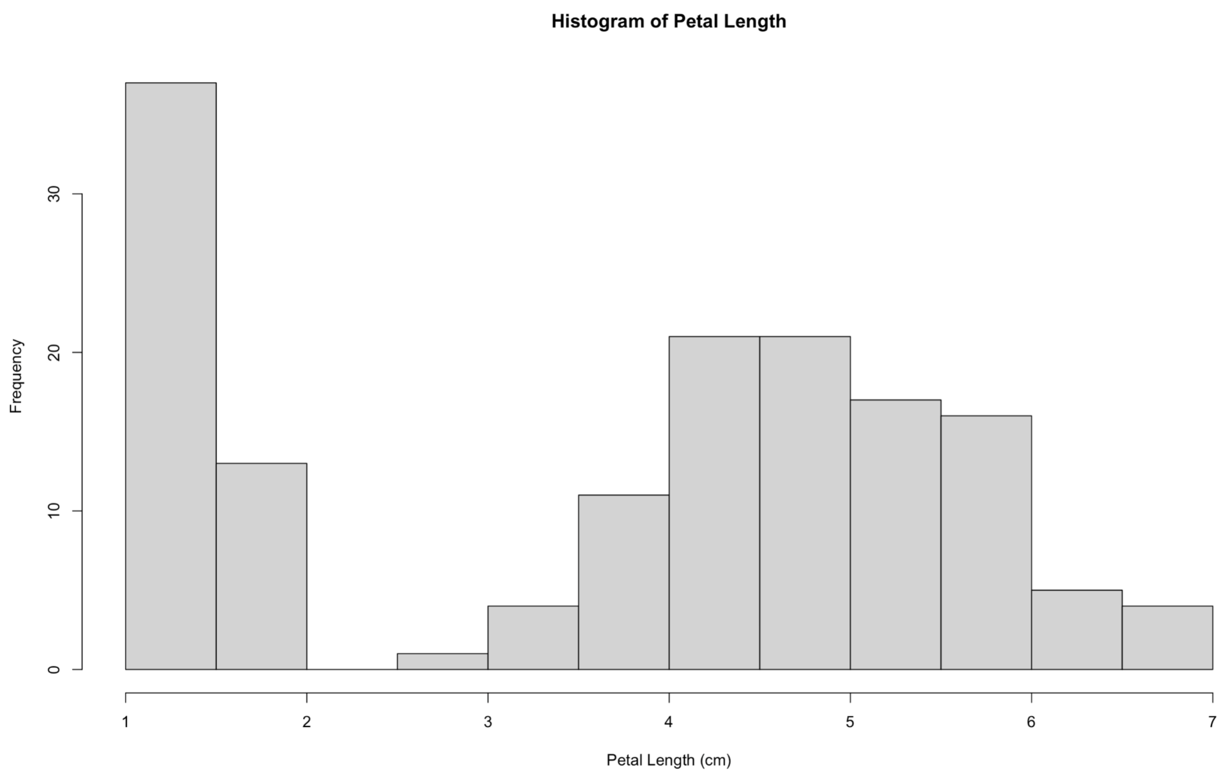
#b) PetalLengthCm

#The most appropriate visualization for the PetalLengthCm variable is a histogram.

#A histogram is a good way to show the distribution of data.

#In this case, the data is the petal lengths of the irises.

#The histogram will show how the petal lengths are distributed.



hist(iris$Petal.Length, main = "Histogram of Petal Length", xlab = "Petal Length (cm)", ylab = "Frequency")

#Inference: From the histogram, we can see that the Petal Lengths in Iris dataset

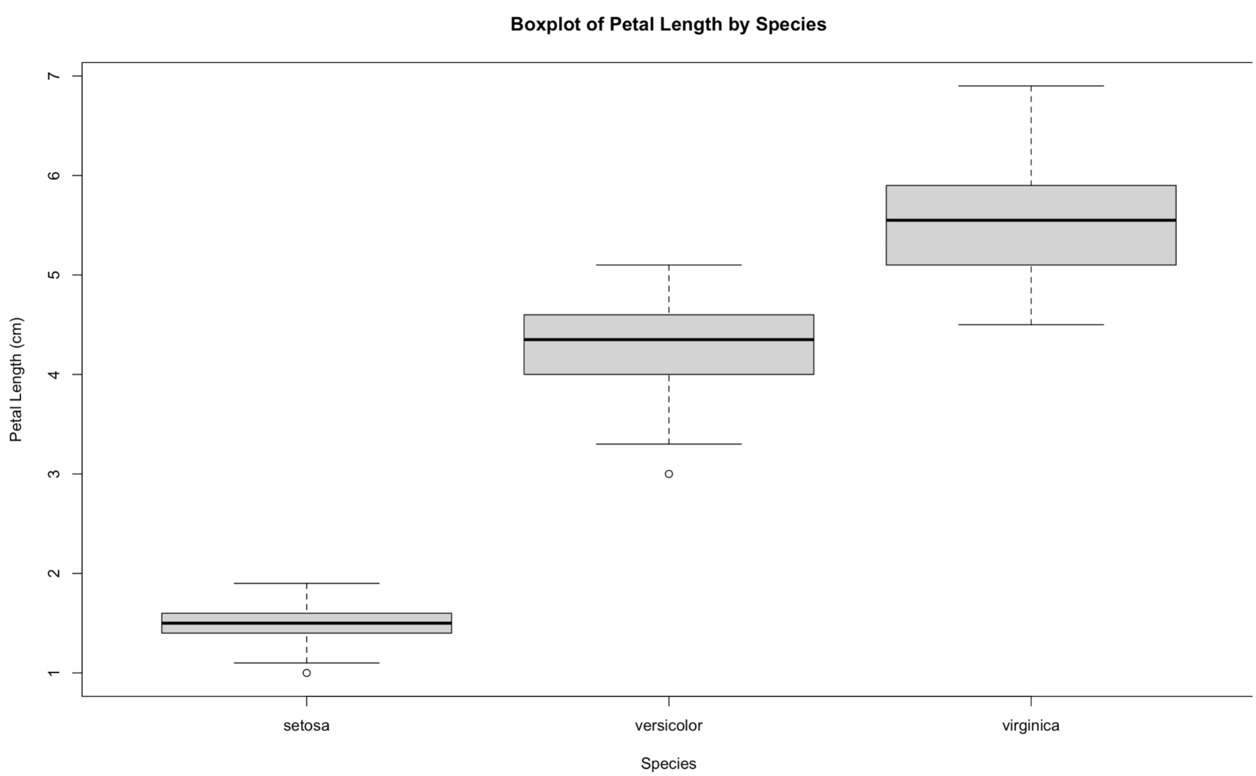
#are normally distributed and centred around 4 to 5cm.

#c) Species and PetalLengthCm

#The appropriate visualization for both Species and PetalLengthCm is a boxplot,

#which can be created using the boxplot() function. This boxplot will display the PetalLengthCm values

#for each species as separate boxes.



boxplot(Petal.Length ~ Species, data = iris, xlab = "Species", ylab = "Petal Length (cm)", main = "Boxplot of Petal Length by Species")

#Inference: From the boxplot, we can see that the average Petal Length for the Setosa species

#is much smaller than those of the other two species. Also, Versicolor Petal lengths are varying widely,

#while Virginica Petal Lengths are longer and have less variation.

#d) SepalLengthCm and SepalWidthCm

```
# Create a scatterplot of the SepalLengthCm and SepalWidthCm variables

plot(iris$SepalLengthCm, iris$SepalWidthCm, col = "lightblue", pch = 19, cex = 1.5)
```

#The most appropriate visualization for the SepalLengthCm and SepalWidthCm variables is a scatterplot.

#A scatterplot is a good way to show the relationship between two variables.

#In this case, the two variables are the sepal length and sepal width.

#The scatterplot will show how the sepal length and sepal width are related.

Inferences:

#The bar chart shows that there are more Iris setosa flowers than Iris versicolor flowers,

#and more Iris versicolor flowers than Iris virginica flowers.

#The histogram shows that the petal lengths of the irises are distributed normally.

#The scatterplot shows that there is a positive correlation between the species of iris and the petal length.

#The scatterplot also shows that there is a positive correlation between the sepal length and sepal width.