Worksheet_#4b

Carl

2023-11-05

```
#1 Using the for loop, create an R script that will display a 5x5 matrix as shown in Figure 1. It must

vecZero <- c(0,0,0,0,0)
matZero <- matrix(vecZero, 5, 5)

matZero
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
```

```
vectorA <- c(1,2,3,4,5)
matVecA <- matrix(vectorA, 5, 5)

for (i in 1:length(vectorA)) {
  matZero[i, ] <- abs(vectorA - vectorA[i]   )
}


print(matZero)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    1    2    3    4
## [2,]    1    0    1    2    3
## [3,]    2    1    0    1    2
## [4,]    3    2    1    0    1
## [5,]    4    3    2    1    0
```

```
#2 Print the string "*" using for() function. The output should be the same as shown in Figure

star <- "*"

for (i in 1:5) {
  starnew <- rep(star, i)
  print(starnew)
}
```

```
## [1] "*"
## [1] "*" "*"
## [1] "*" "*" "*"
## [1] "*" "*" "*" "*"
## [1] "*" "*" "*" "*" "*"
```

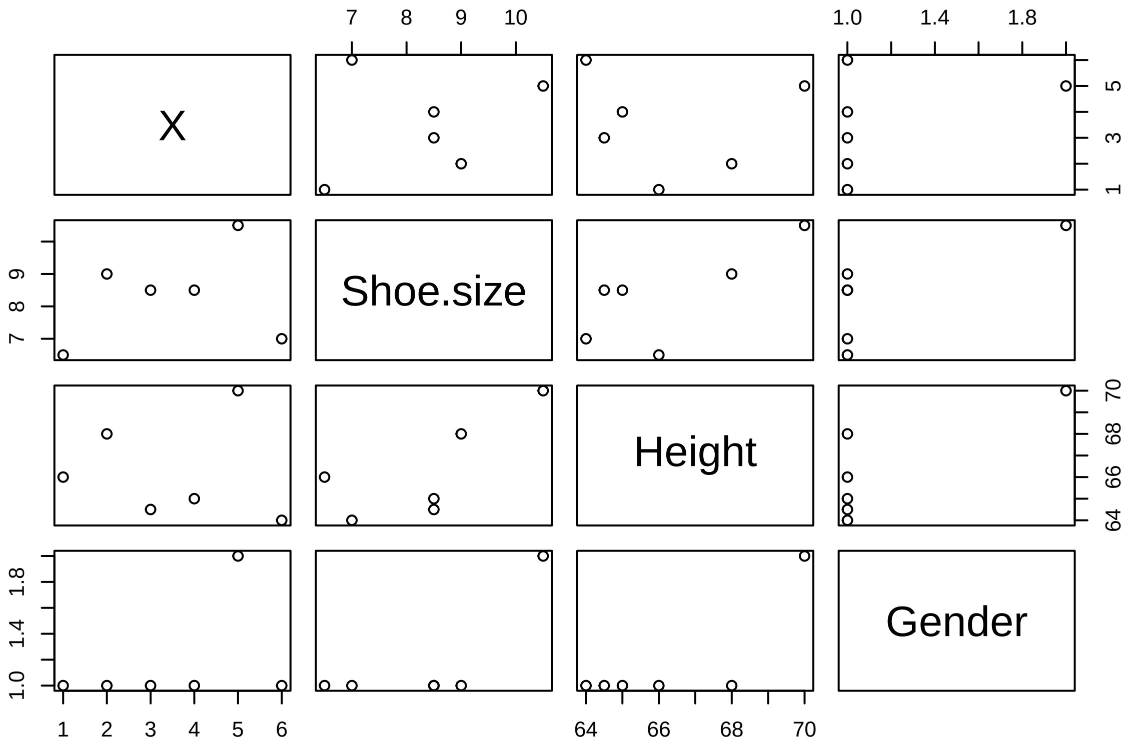
```r
#3 Get an input from the user to print the Fibonacci sequence starting from the 1st input up to 500. Us

readlineInput <- as.integer(readline("Enter the starting Fibonacci sequence number: "))
```

```
## Enter the starting Fibonacci sequence number:
```

```r
if(is.na(readlineInput || readlineInput < 0)) {
cat("Error: Enter a number!")
} else {

userinput <- readlineInput
a <- userinput
b <- 0


cat("Fibonacci sequence starting from", userinput, ":\n")

repeat {
  next_num <- a + b
  if (next_num > 500){
    cat("STOPPED!!! next sequence will be over 500")
    break
  }
  cat(next_num, " ")
  a <- b
  b <- next_num
}

cat("\n")
}
```

```
## Error: Enter a number!
```

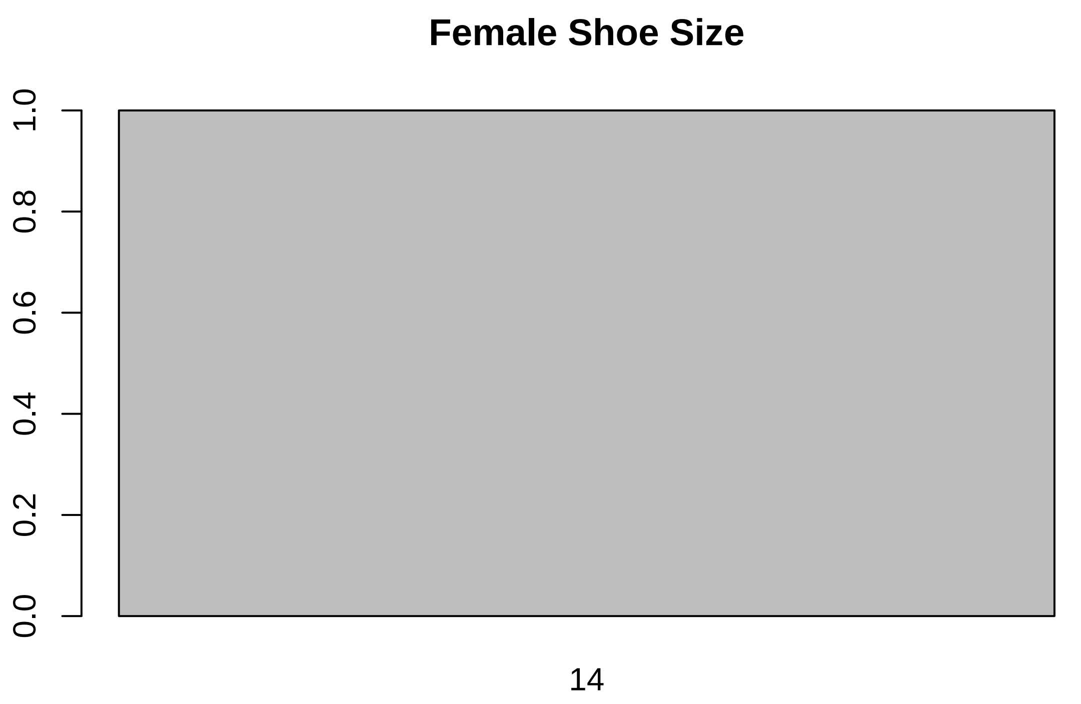
```r
#4 Import the dataset as shown in Figure 1 you have created previously.


#4a What is the R script for importing an excel or a csv file? Display the first 6 rows of the dataset?

imp <- read.csv("prevdata")

plot(head(imp,6))
```



```
#4b Create a subset for gender(female and male). How many observations are there in Male? How about in

numofFem <- subset(imp, Gender == "F")
numofMale <- subset(imp, Gender == "M")

numofFem <- nrow(numofFem)
numofMale <- nrow(numofMale)

cat("Number of observations in Female subset: ", numofFem, "\n")
```

```
## Number of observations in Female subset:  14
```

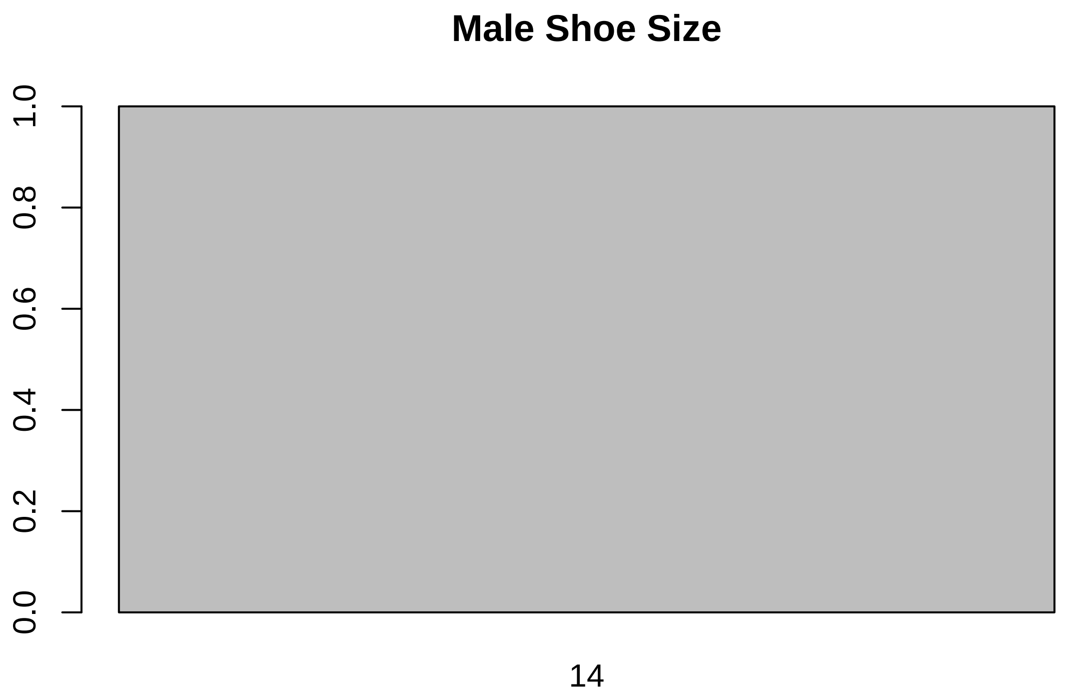
```
barplot(table(subset(numofFem, imp$Gender == "F" )), main = "Female Shoe Size")
```



```
cat("Number of observations in Male subset: ", numofMale, "\n")
```

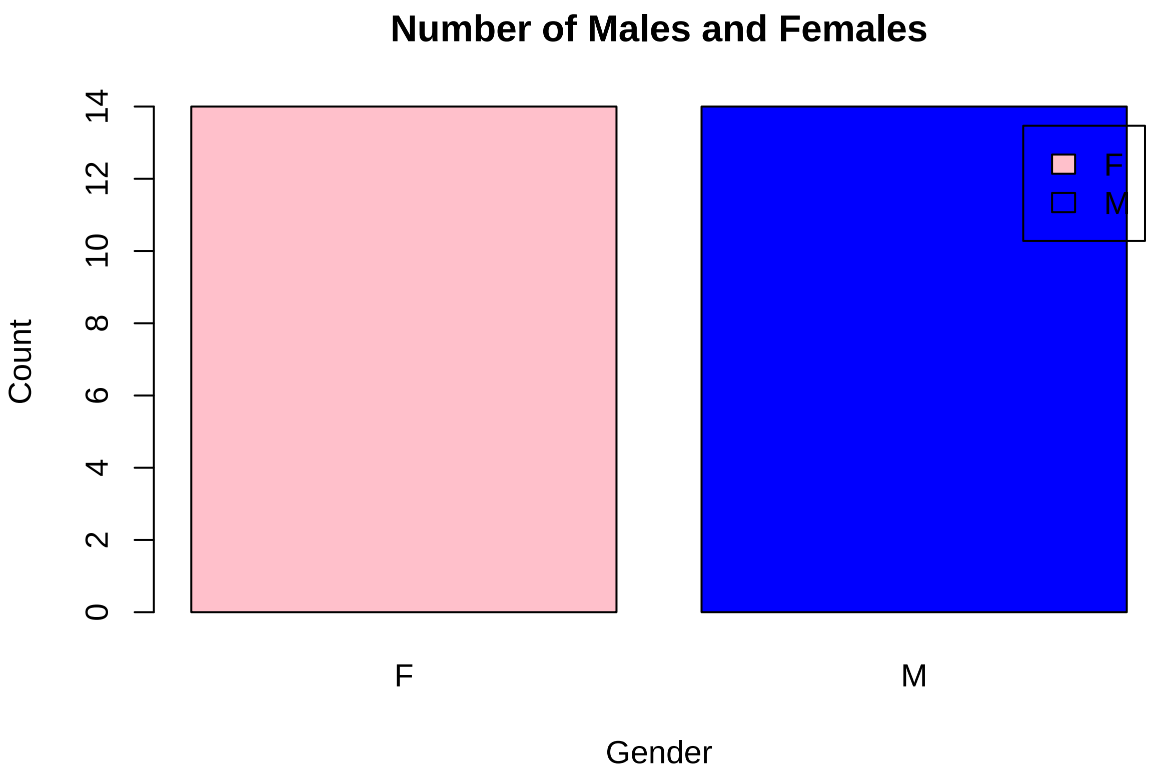
```
## Number of observations in Male subset:  14
```

```
barplot(table(subset(numofMale, imp$Gender == "F")), main = "Male Shoe Size")
```



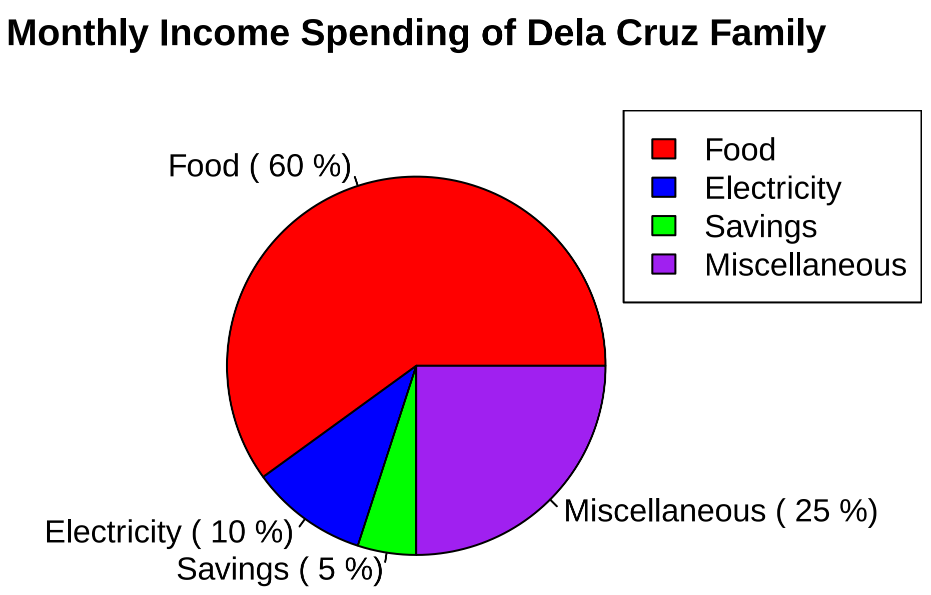
```
#4c Create a graph for the number of males and females for Household Data. Use plot(), chart type = bar

totalofMF <- table(imp$Gender)
```

```r
barplot(totalofMF,
        main = "Number of Males and Females",
        xlab = "Gender",
        ylab = "Count",
        col = c("pink", "blue"),
        legend.text = rownames(totalofMF),
        beside = TRUE)
```



```r
#5 The monthly income of Dela Cruz family was spent on the following:'

#Food Electricity Savings Miscellaneous
# 60     10             5       25
```

```r
#5a Create a piechart that will include labels in percentage.Add some colors and title of the chart. Wr

spending_data <- data.frame(
  Category = c("Food", "Electricity", "Savings", "Miscellaneous"),
  Value = c(60, 10, 5, 25)
)

spending_data$Percentage <- spending_data$Value / sum(spending_data$Value) * 100

pie(spending_data$Value,
    labels = paste(spending_data$Category, "(", spending_data$Percentage, "%)"),
    col = c("red", "blue", "green", "purple"),
    main = "Monthly Income Spending of Dela Cruz Family")

legend("topright", spending_data$Category, fill = c("red", "blue", "green", "purple"))
```



```r
#6Use the iris dataset.

data("iris")

#6a Check for the structure of the dataset using the str() function. Describe what you have seen in the

str(iris)
```

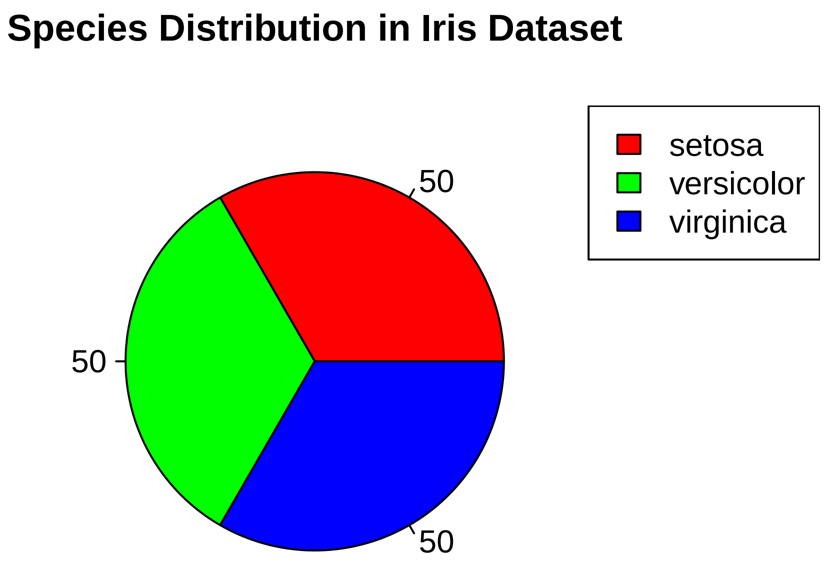
```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```r
#The iris dataset is a collection of information about iris flowers. It includes data on the length and
```

```r
#6b Create an R object that will contain the mean of the sepal.length, sepal.width,petal.length,and pet

meanofIris <- colMeans(iris[,1:4])

meanofIris
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

```r
#6c Create a pie chart for the Species distribution. Add title, legends, and colors. Write the R script

speciesofIris <- table(iris$Species)
nameofSpecies <- c("Setosa", "Versicolor", "Virginica")

pie(speciesofIris,
    labels = speciesofIris,
    col = c("red", "green", "blue"),
    main = "Species Distribution in Iris Dataset")

legend("topright", legend = levels(iris$Species), fill = c("red", "green", "blue"),)
```



```
#6d Subset the species into setosa, versicolor, and virginica. Write the R scripts and show the last si


subSetosa <- subset(iris, Species == "setosa")
subVersi <- subset(iris, Species == "versicolor")
subVirginica <- subset(iris, Species == "virginica")

tail(subSetosa, 6)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 45          5.1         3.8          1.9         0.4  setosa
## 46          4.8         3.0          1.4         0.3  setosa
## 47          5.1         3.8          1.6         0.2  setosa
## 48          4.6         3.2          1.4         0.2  setosa
## 49          5.3         3.7          1.5         0.2  setosa
## 50          5.0         3.3          1.4         0.2  setosa
```

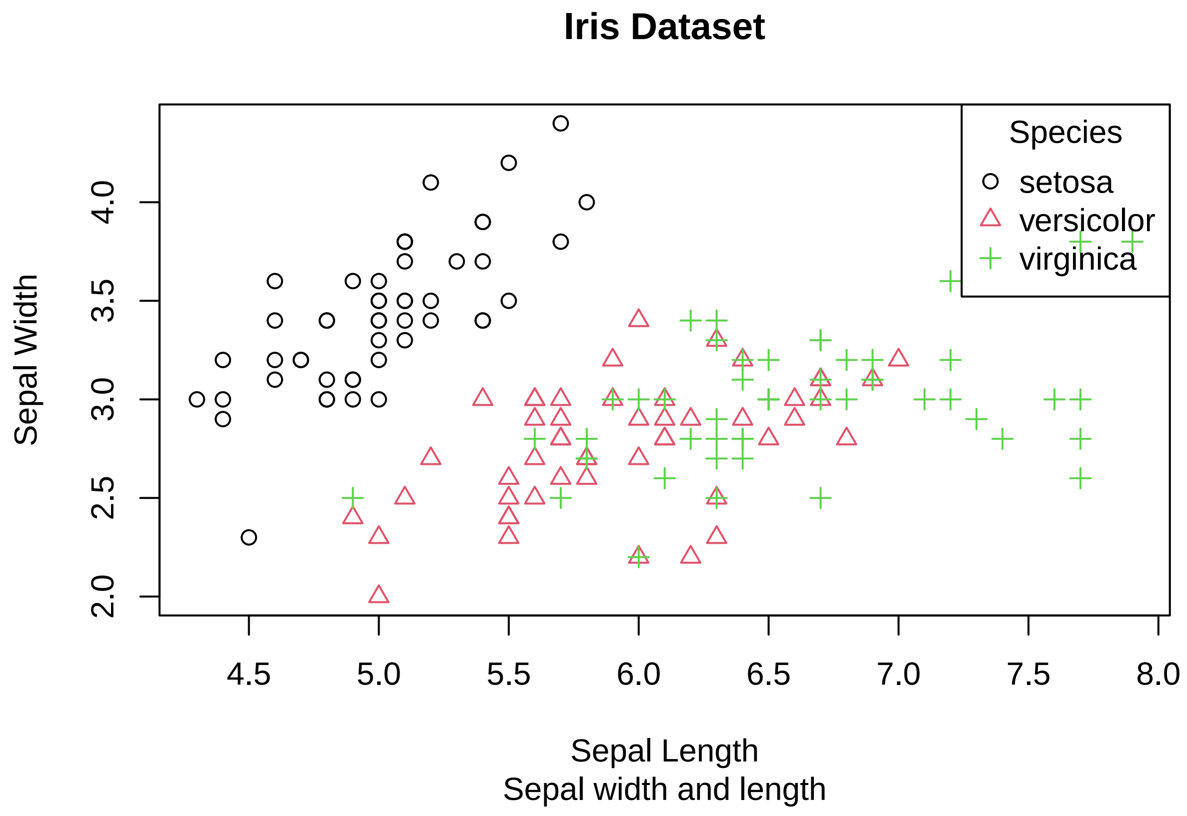
```
tail(subVersi, 6)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 95           5.6         2.7          4.2         1.3 versicolor
## 96           5.7         3.0          4.2         1.2 versicolor
## 97           5.7         2.9          4.2         1.3 versicolor
## 98           6.2         2.9          4.3         1.3 versicolor
## 99           5.1         2.5          3.0         1.1 versicolor
## 100          5.7         2.8          4.1         1.3 versicolor
```

```
tail(subVirginica, 6)
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 145          6.7         3.3          5.7         2.5 virginica
## 146          6.7         3.0          5.2         2.3 virginica
## 147          6.3         2.5          5.0         1.9 virginica
## 148          6.5         3.0          5.2         2.0 virginica
## 149          6.2         3.4          5.4         2.3 virginica
## 150          5.9         3.0          5.1         1.8 virginica
```

```r
#6e Create a scatterplot of the sepal.length and sepal.width using the different species(setosa,versico

plot(iris$Sepal.Length, iris$Sepal.Width,
     xlab = "Sepal Length", ylab = "Sepal Width",
     main = "Iris Dataset", sub = "Sepal width and length",
     pch = as.numeric(iris$Species), col = as.numeric(iris$Species))

legend("topright", legend = levels(iris$Species),
       pch = 1:3, col = 1:3,
       title = "Species")
```



```r
#6f Interpret the result

#In This Iris dataset helps us see how iris flowers of different species are different in terms of sepa

# -Setosa: These flowers typically have short sepal length and wide sepal width. They are grouped in th

# -Versicolor: Versicolor flowers have average sepal length and width. They are in the middle part.

# - Virginica: Virginica flowers are usually long in sepal length and have narrower sepal width. They f

#This plot makes it easy to see the differences between the three iris species based on sepal length an
```

7.Import the alexa-file.xlsx. Check on the variations. Notice that there are ex-tra whitespaces among black variants (Black Dot, Black Plus, Black Show, Black Spot). Also on the white variants (White Dot, White Plus, White Show, White Spot).

```r
library(readxl)
alexa_file <- read_excel("alexa_file.xlsx")
```

7 a.Rename the white and black variants by using gsub() function.

```
alexa_file$variation <- gsub("Black  Dot" , "BlackDot", alexa_file$variation)
alexa_file$variation <- gsub("Black  Plus" , "BlackPlus", alexa_file$variation )
alexa_file$variation <- gsub("Black  Show" , "BlackShow", alexa_file$variation )
alexa_file$variation <- gsub("Black  Spot" , "BlackSpot", alexa_file$variation )


alexa_file$variation <- gsub("White  Dot" , "WhiteDot", alexa_file$variation )
alexa_file$variation <- gsub("White  Plus" , "WhitePlus", alexa_file$variation )
alexa_file$variation <- gsub("White  Show" , "WhiteShow", alexa_file$variation )
alexa_file$variation <- gsub("White  Spot" , "WhiteSpot", alexa_file$variation )
```

7b. Get the total number of each variations and save it into another object. Save the object as variations.RData. Write the R scripts. What is its result? Hint: Use the dplyr package. Make sure to install it before loading the package.

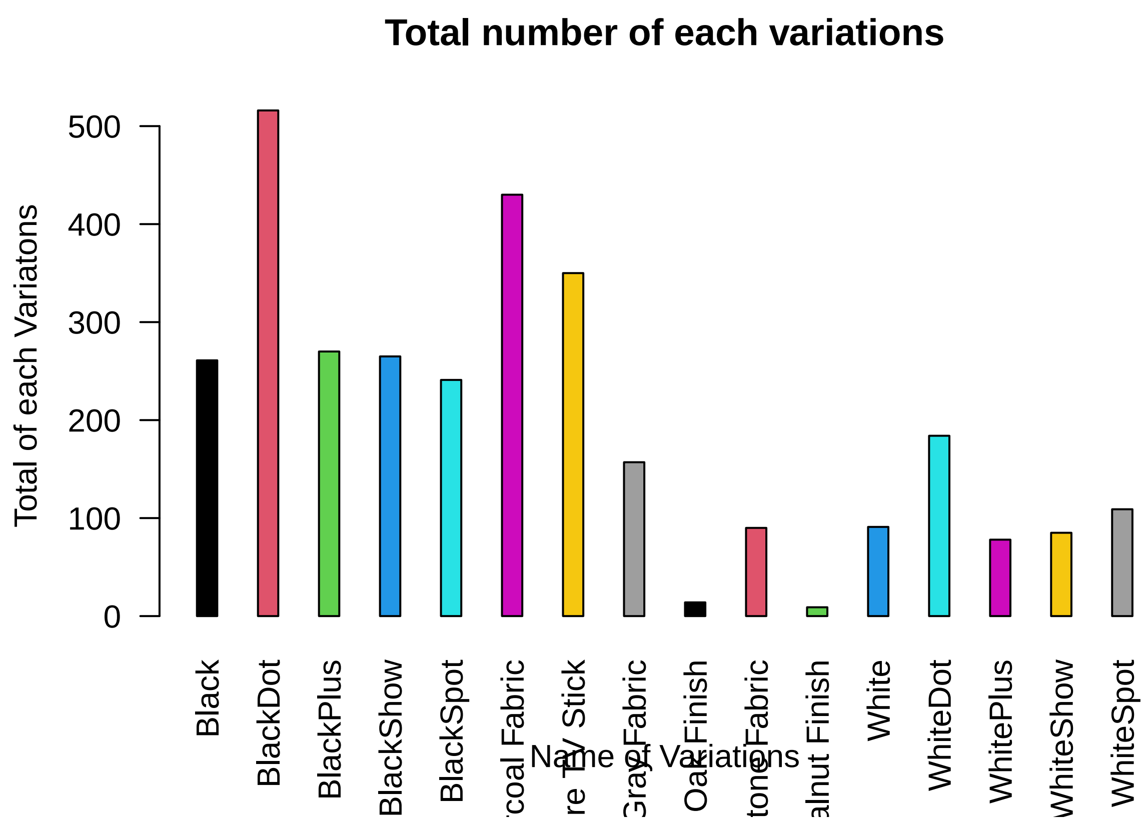
```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
  Variationsofalexa <- sort(alexa_file$variation)
  Vardata <- alexa_file$variation

   Variations.RDara <-  alexa_file %>%
    count(alexa_file$variation)
```

7.c From the variations.RData, create a barplot(). Complete the details of the chart which include the title, color, labels of each bar.

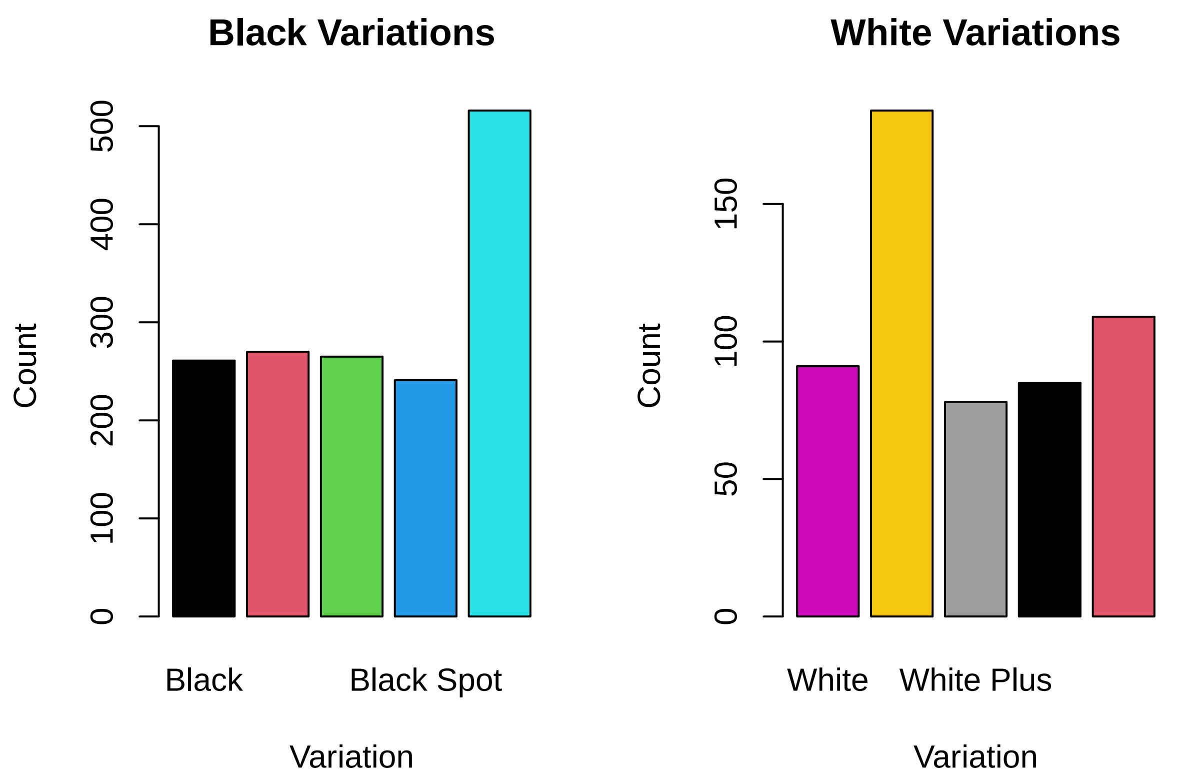
```
  totalofeachVar <- Variations.RDara$n
  nameOfVar <- Variations.RDara$`alexa_file$variation`

varPlot <- barplot(totalofeachVar,
        names.arg = nameOfVar,
        main = "Total number of each variations",
        xlab = "Name of Variations",
        ylab = "Total of each Variatons",
        col = 1:16,
        las = 2,
        space = 2)
```



```r
png("varPlot.png")
```

7.d. Create a barplot() for the black and white variations. Plot it in 1 frame, side by side. Complete the details of the chart.

```r
library(RColorBrewer)

par(mfrow = c(1,2))
blackPlot <- barplot(height = c(261,270,265,241,516),
                     names.arg = c("Black","Black Plus","Black Show","Black Spot","Black Dot"), main =
                     col = 1:5,
                     xlab = "Variation",
                     ylab = "Count")





whitePlot <- barplot(height = c(91,184,78,85,109),
        names.arg = c("White", "White Dot", "White Plus", "White Show", "White Spot"),
        main = "White Variations",
        col = 6:10,
        xlab = "Variation",
        ylab = "Count",)
```



```
png("whitePlot.png")
png("blackPlot.png")

par(mfrow = c(1,1))
```