RWorksheet_Huervana#4B

2023-10-04

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

It must contain vectorA = [1,2,3,4,5] and a 5 x 5 zero matrix.

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

matrx <- matrix(0,nrow = 5, ncol =5)

for (j in 1:5)
  for (k in 1:5)
  {
    matrx[j,k] <- abs (vectorA[j] - vectorA[k])
  }

matrx
```

```
##      [,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 2

```
for(i in 1:5) {
  stars <- rep("*", i)
  print(stars)
}
```

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

3. Get an input from the user to print the Fibonacci sequence starting from the 1st input up to 500. Use repeat and break statements. Write the R Scripts and its output.

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

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

```
if(is.na(userInput || userInput < 0)) {
  cat("Please enter something")
} else {
  x <- userInput
  y <- 0
cat("Fibonacci sequence starting from", userInput, ":\n")

repeat {

  next_num <- x + y

  if (next_num > 500){
    break
  }
  cat(next_num, " ")
  x <- y
  y <- next_num

}
}
```

```
## Please enter something
```

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

A. What is the R script for importing an excel or a csv file? Display the first 6 rows of the dataset? Show your codes and its result.

```
importData <- read.csv("HouseholdData.csv")

head(importData)
```

```
##   X Shoe.Size Height Gender
## 1 1       6.5   66.0      F
## 2 2       9.0   68.0      F
## 3 3       8.5   64.5      F
## 4 4       8.5   65.0      F
## 5 5      10.5   70.0      M
## 6 6       7.0   64.0      F
```

B. Create a subset for gender(female and male). How many observations are there in Male? How about in Female? Write the R scripts and its output.

```
males <- importData[importData$Gender == "M",]
males
```

```
##     X Shoe.Size Height Gender
## 5   5      10.5   70.0      M
## 9   9      13.0   72.0      M
## 11 11      10.5   74.5      M
## 13 13      12.0   71.0      M
## 14 14      10.5   71.0      M
## 15 15      13.0   77.0      M
## 16 16      11.5   72.0      M
## 19 19      10.0   72.0      M
## 22 22       8.5   67.0      M
## 23 23      10.5   73.0      M
## 25 25      10.5   72.0      M
## 26 26      11.0   70.0      M
## 27 27       9.0   69.0      M
## 28 28      13.0   70.0      M
```

```
females <- importData[importData$Gender == "F",]
females
```

```
##     X Shoe.Size Height Gender
## 1   1       6.5   66.0      F
## 2   2       9.0   68.0      F
## 3   3       8.5   64.5      F
## 4   4       8.5   65.0      F
## 6   6       7.0   64.0      F
## 7   7       9.5   70.0      F
## 8   8       9.0   71.0      F
## 10 10       7.5   64.0      F
## 12 12       8.5   67.0      F
## 17 17       8.5   59.0      F
## 18 18       5.0   62.0      F
## 20 20       6.5   66.0      F
## 21 21       7.5   64.0      F
## 24 24       8.5   69.0      F
```

```
numofMale <- nrow(males)
numofMale
```

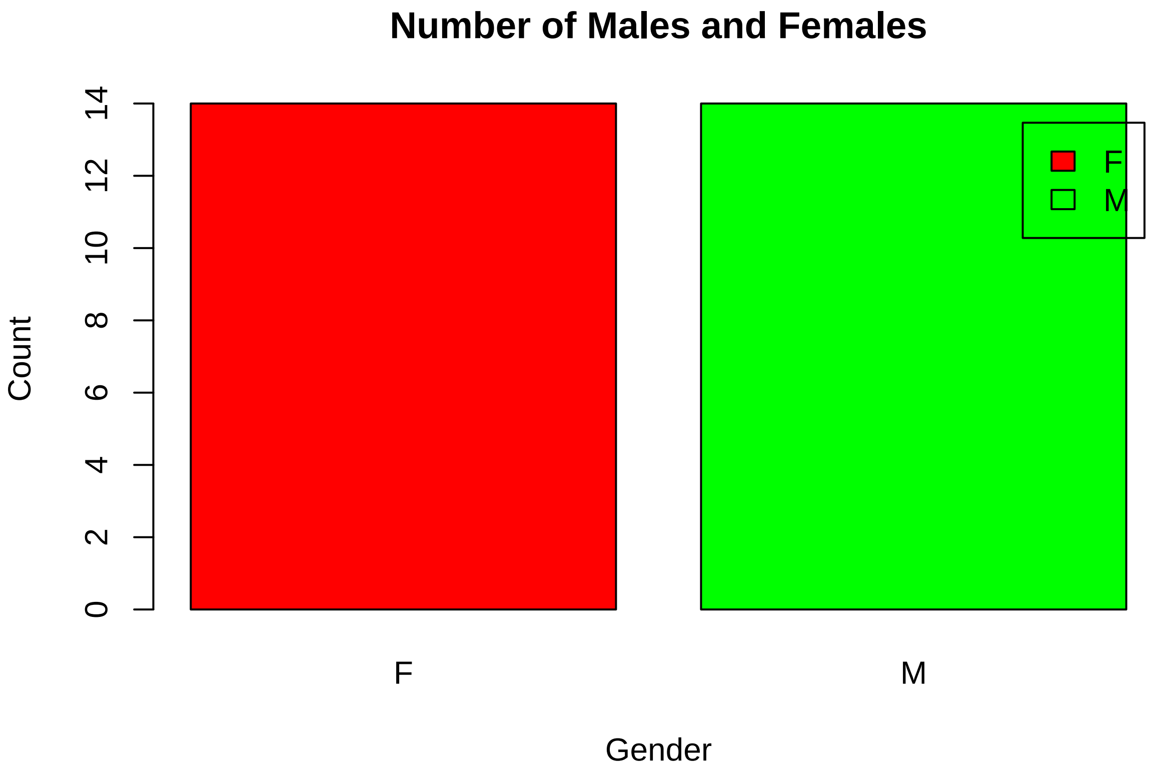
```
## [1] 14
```

```
numofFem <- nrow(females)
numofFem
```

```
## [1] 14
```

C. Create a graph for the number of males and females for Household Data.

```
MaleFemale <- table(importData$Gender)
barplot(MaleFemale,

        main = "Number of Males and Females",

        xlab = "Gender",

        ylab = "Count",

        col = c("red", "green"),

        legend.text = rownames(MaleFemale),

        beside = TRUE)
```

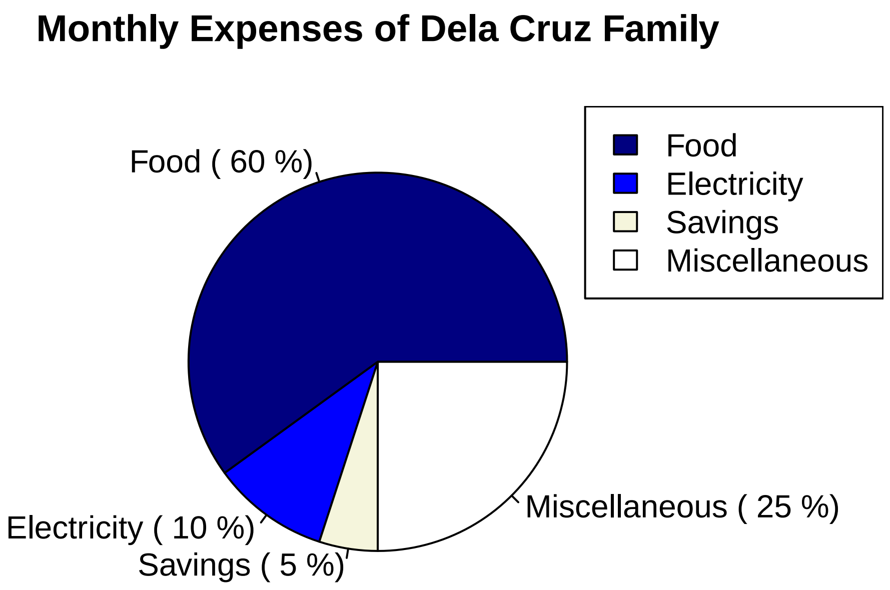


5. The monthly income of Dela Cruz family was spent on the following:”

Food Electricity Savings Miscellaneous 60 10 5 25

A. Create a piechart that will include labels in percentage.Add some colors and title of the chart.

```
expenses <- data.frame(
  expenseCat = c("Food", "Electricity", "Savings", "Miscellaneous"),
  cost = c(60, 10, 5, 25)
)

expenses$Percentage <- expenses$cost / sum(expenses$cost) * 100

colors <- c("navy", "blue", "beige", "white")

pie(expenses$cost,

    labels = paste(expenses$expenseCat, "(",expenses$Percentage,"%)"),
    col = colors,
    main = "Monthly Expenses of Dela Cruz Family")

legend("topright", expenses$expenseCat, fill = colors)
```



6.Use the iris dataset. data(iris)

A. Check for the structure of the dataset using the str() function. Describe what you have seen in the output.

```
data(iris)
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 ...
```

```
#This is a function in R used to load datasets that come pre-installed with R or from packages.
#It provides a concise summary of the structure of the iris dataset. It shows information about the dat
```

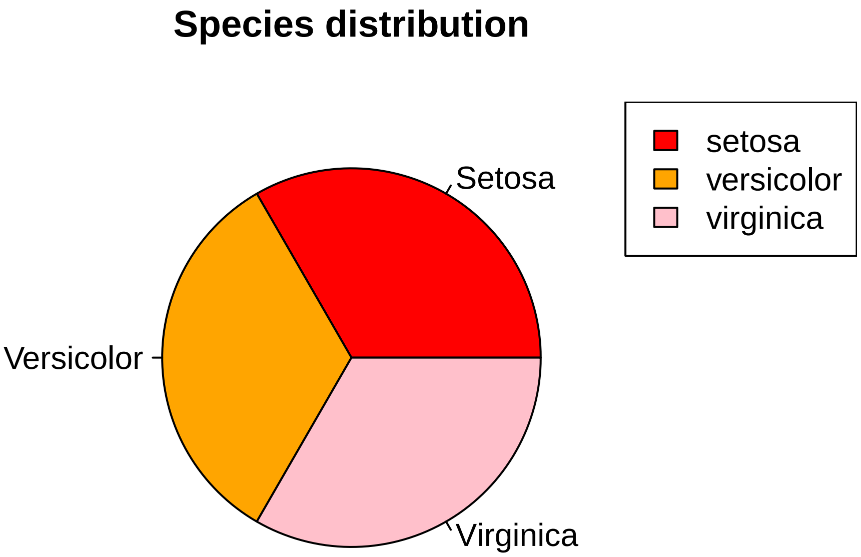
B. Create an R object that will contain the mean of the sepal.length, sepal.width,petal.length,and petal.width.

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

mean
```

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

C. Create a pie chart for the Species distribution. Add title, legends, and colors.

```
species_dis <- table(iris$Species)
Species <- c("Setosa", "Versicolor", "Virginica")
pie(species_dis,
    labels = Species,

    col = c("red", "orange", "pink"),

    main = "Species distribution")

legend("topright", legend = levels(iris$Species), fill = c("red", "orange", "pink"),)
```



D. Subset the species into setosa, versicolor, and virginica.

```
setosa <- subset(iris, Species == "setosa")
versicolor <- subset(iris, Species == "versicolor")
virginica <- subset(iris, Species == "virginica")



tail(setosa, 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(versicolor, 6)
```

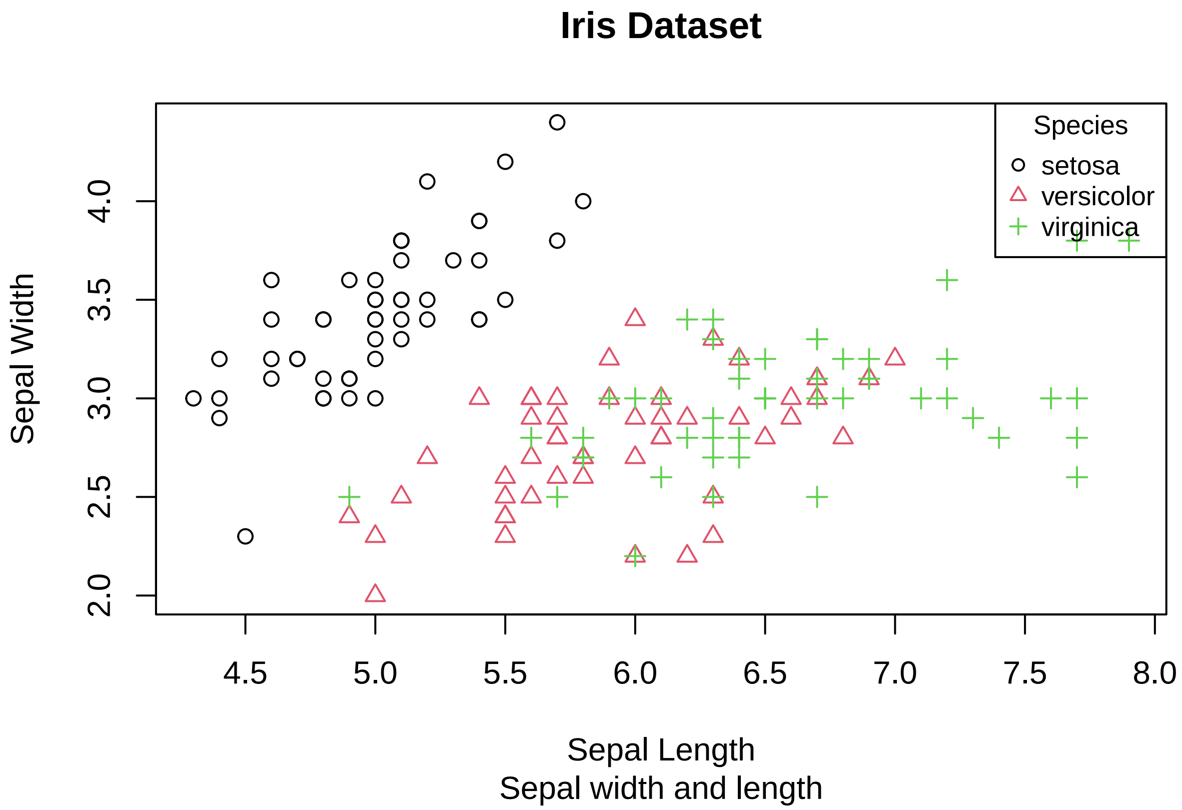
```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 95           5.6         2.7          4.2         1.3 versicolor
## 96           5.7         3.0          4.2         1.2 versicolor
## 97           5.7         2.9          4.2         1.3 versicolor
## 98           6.2         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(virginica, 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
```

E. Create a scatterplot of the sepal.length and sepal.width using the different species(setosa,versicolor,virginica).

```r
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), col = 1:3, pch = 1:3, cex = 0.8, title = "Species")
```



F.

The scatter plot visually represents the relationship between Sepal Length and Sepal Width for each species in the Iris dataset. Each point corresponds to an individual observation, and the points are differentiated by species using different colors and plotting characters. The legend provides a key to interpret the colors and symbols associated with each species.

7. Import the alexa-file.xlsx. Check on the variations. Notice that there are extra 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 <- read_excel("alexa_file.xlsx")
alexa
```

```
## # A tibble: 3,150 x 5
##    rating date                variation        verified_reviews      feedback
##     <dbl> <dttm>              <chr>            <chr>                    <dbl>
##  1      5 2018-07-31 00:00:00 Charcoal Fabric  Love my Echo!                1
##  2      5 2018-07-31 00:00:00 Charcoal Fabric  Loved it!                    1
##  3      4 2018-07-31 00:00:00 Walnut Finish    Sometimes while play~        1
##  4      5 2018-07-31 00:00:00 Charcoal Fabric  I have had a lot of ~        1
##  5      5 2018-07-31 00:00:00 Charcoal Fabric  Music                        1
##  6      5 2018-07-31 00:00:00 Heather Gray Fabric I received the echo ~        1
##  7      3 2018-07-31 00:00:00 Sandstone Fabric    Without having a cel~        1
##  8      5 2018-07-31 00:00:00 Charcoal Fabric     I think this is the ~        1
##  9      5 2018-07-30 00:00:00 Heather Gray Fabric looks great                  1
## 10      5 2018-07-30 00:00:00 Heather Gray Fabric Love it! I've listen~        1
## # i 3,140 more rows
```

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

```r
alexa$variation <- gsub("Black  Dot", "BlackDot",alexa$variation)

alexa$variation <- gsub("Black  Plus","BlackPlus", alexa$variation)

alexa$variation <- gsub("Black  Show", "BlackShow", alexa$variation)

alexa$variation <- gsub("Black  Spot", "BlackSpot", alexa$variation)

alexa$variation <- gsub("White  Dot","WhiteDot",alexa$variation)

alexa$variation <- gsub("White  Plus","WhitePlus", alexa$variation)

alexa$variation <- gsub("White  Show","WhiteShow", alexa$variation)

alexa$variation <- gsub("White  Spot","WhiteSpot",alexa$variation)

alexa
```

```
## # A tibble: 3,150 x 5
##    rating date                variation           verified_reviews      feedback
##     <dbl> <dttm>              <chr>               <chr>                    <dbl>
##  1      5 2018-07-31 00:00:00 Charcoal Fabric     Love my Echo!                1
##  2      5 2018-07-31 00:00:00 Charcoal Fabric     Loved it!                    1
##  3      4 2018-07-31 00:00:00 Walnut Finish       Sometimes while play~        1
##  4      5 2018-07-31 00:00:00 Charcoal Fabric     I have had a lot of ~        1
##  5      5 2018-07-31 00:00:00 Charcoal Fabric     Music                        1
##  6      5 2018-07-31 00:00:00 Heather Gray Fabric I received the echo ~        1
##  7      3 2018-07-31 00:00:00 Sandstone Fabric    Without having a cel~        1
##  8      5 2018-07-31 00:00:00 Charcoal Fabric     I think this is the ~        1
##  9      5 2018-07-30 00:00:00 Heather Gray Fabric looks great                  1
## 10      5 2018-07-30 00:00:00 Heather Gray Fabric Love it! I've listen~        1
## # i 3,140 more rows
```

B. Get the total number of each variations and save it into another object. Save the object as variations.RData.

```r
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
```

```
vartotal <- alexa %>%
  count(alexa$variation)

vartotal
```

```
## # A tibble: 16 x 2
##    `alexa$variation`                n
##    <chr>                        <int>
##  1 Black                          261
##  2 BlackDot                       516
##  3 BlackPlus                      270
##  4 BlackShow                      265
##  5 BlackSpot                      241
##  6 Charcoal Fabric                430
##  7 Configuration: Fire TV Stick   350
##  8 Heather Gray Fabric            157
##  9 Oak Finish                      14
## 10 Sandstone Fabric                90
## 11 Walnut Finish                    9
## 12 White                           91
## 13 WhiteDot                       184
## 14 WhitePlus                       78
## 15 WhiteShow                       85
## 16 WhiteSpot                      109
```

```
save(vartotal, file = "variations.RData")
```

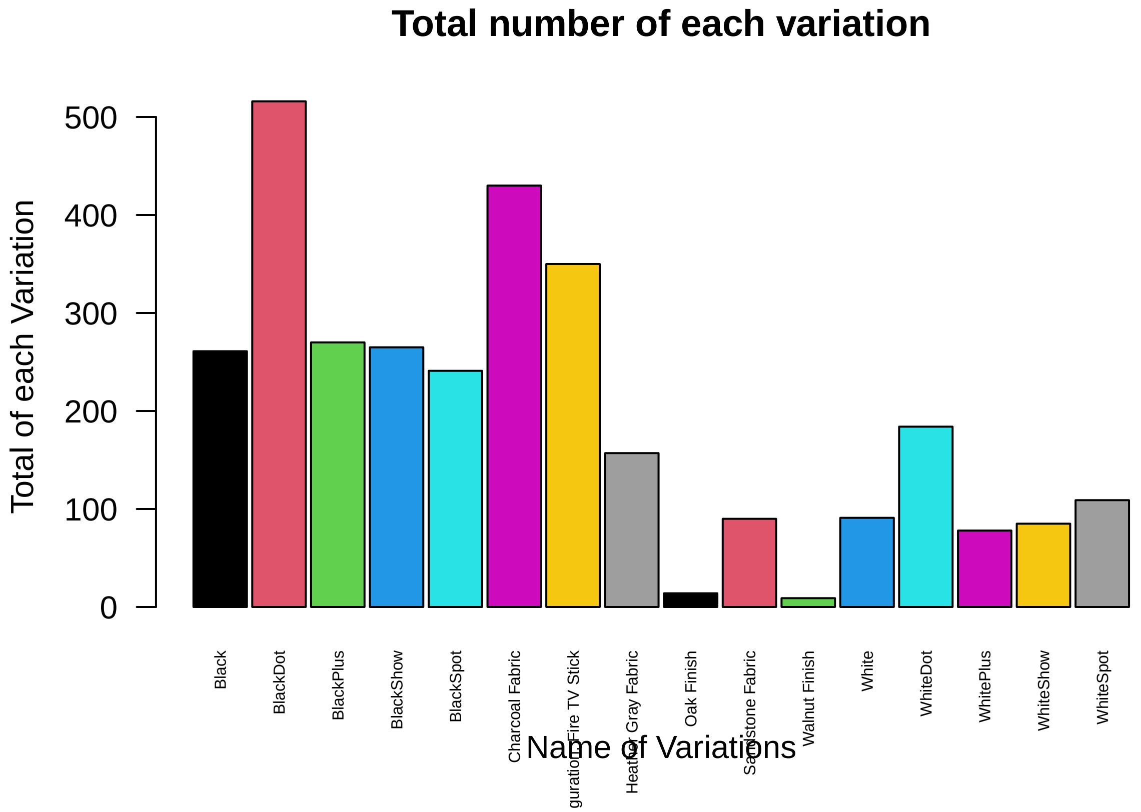
C.From the variations.RData, create a barplot(). Complete the details of the chart which include the title, color, labels of each bar.

```
load("variations.RData")

vartotal
```

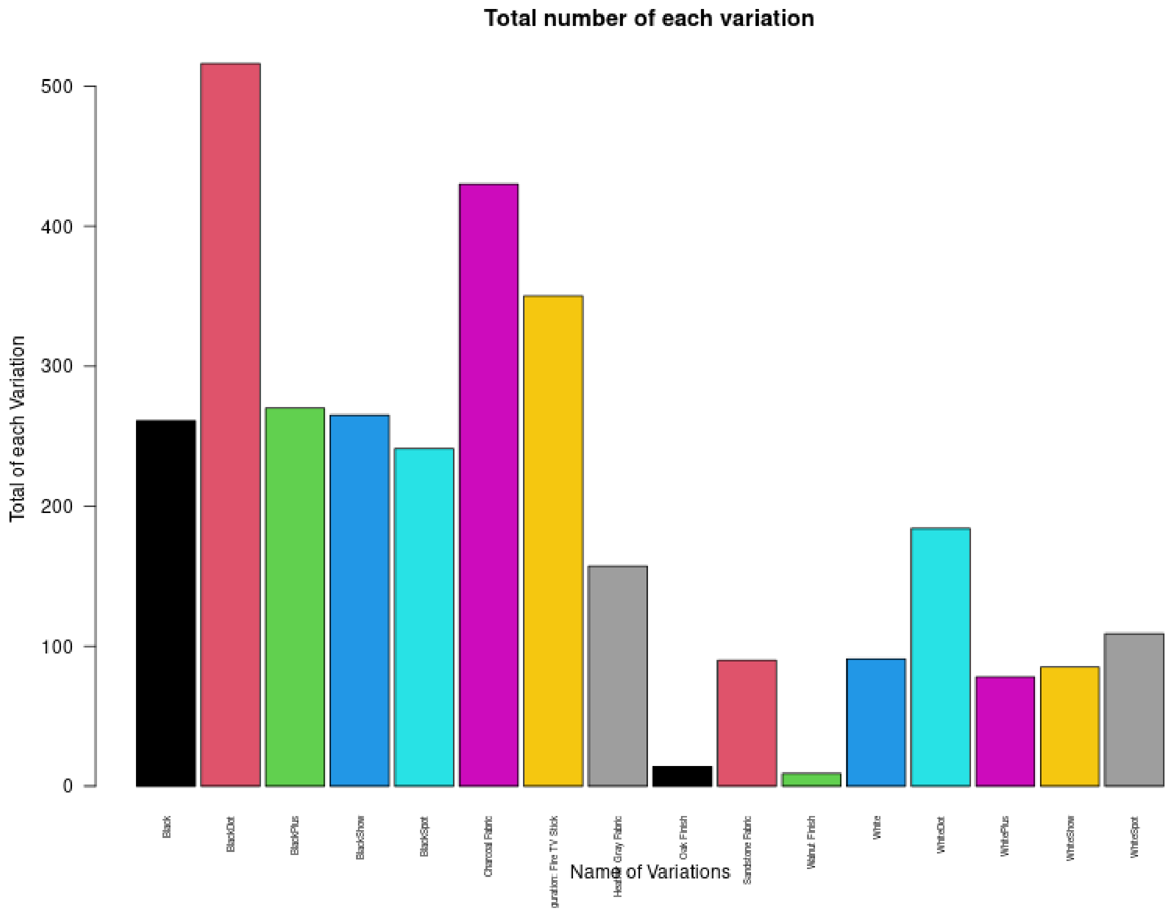
```
## # A tibble: 16 x 2
##    `alexa$variation`                n
##    <chr>                        <int>
##  1 Black                          261
##  2 BlackDot                       516
##  3 BlackPlus                      270
##  4 BlackShow                      265
##  5 BlackSpot                      241
##  6 Charcoal Fabric                430
##  7 Configuration: Fire TV Stick   350
##  8 Heather Gray Fabric            157
##  9 Oak Finish                      14
## 10 Sandstone Fabric                90
## 11 Walnut Finish                    9
## 12 White                           91
## 13 WhiteDot                       184
## 14 WhitePlus                       78
## 15 WhiteShow                       85
## 16 WhiteSpot                      109
```

```
varNames <- vartotal$`alexa$variation`
```

```r
totalPlot <- barplot(vartotal$n,
        names.arg = varNames,
        main = "Total number of each variation",
        xlab = "Name of Variations",
        ylab = "Total of each Variation",
        col = 1:16,
        space = 0.1,
        cex.names = 0.5,
        las = 2)
```



```r
png("/cloud/project/RWorksheet#4/vartotal.png", width = 800, height = 600, units = "px", pointsize = 12
knitr::include_graphics("/cloud/project/RWorksheet#4/vartotal.png")
```

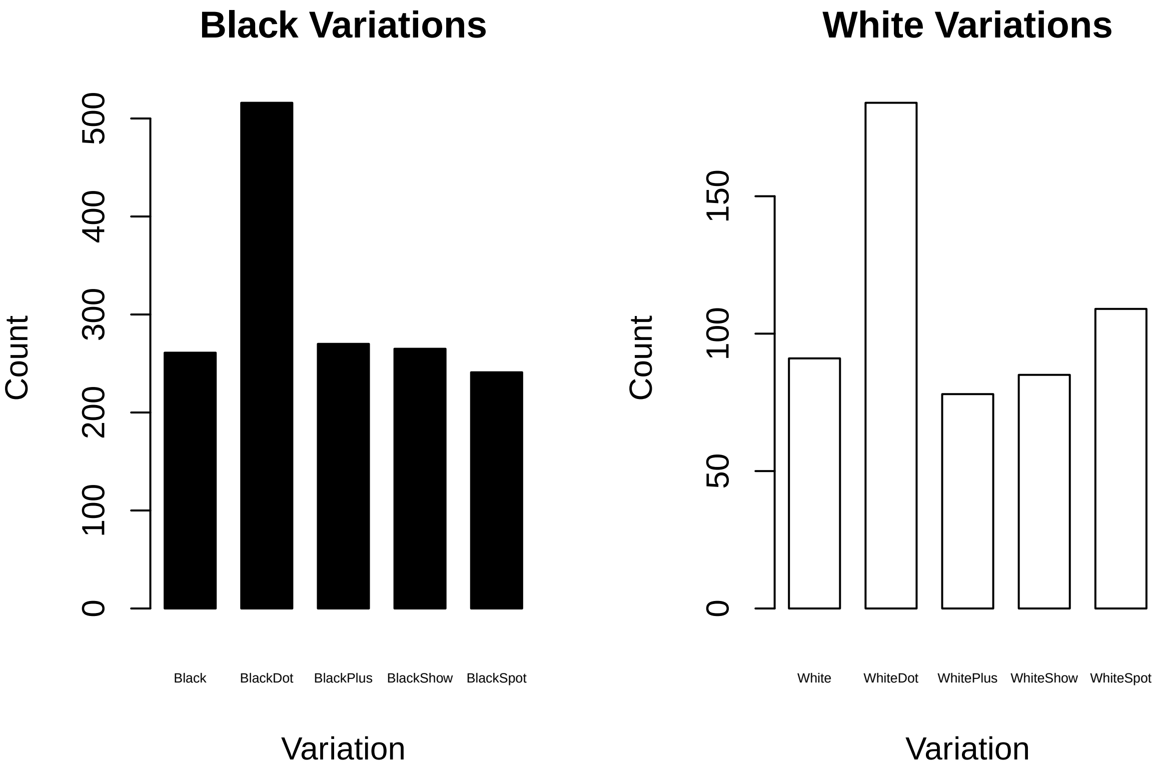


D. Create a barplot() for the black and white variations. Plot it in 1 frame, side by side. Complete the details of the chart.

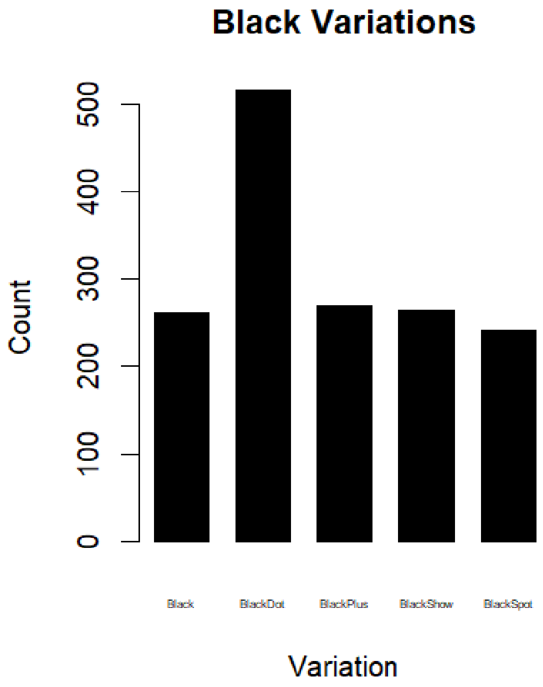
```
blackVars <- vartotal[vartotal$`alexa$variation` %in% c("Black", "BlackPlus" , "BlackShow" ,"BlackSpot"

whiteVars <- vartotal[vartotal$`alexa$variation` %in% c("White", "WhiteDot", "WhitePlus", "WhiteShow",

par(mfrow = c(1,2))
blackVars
```

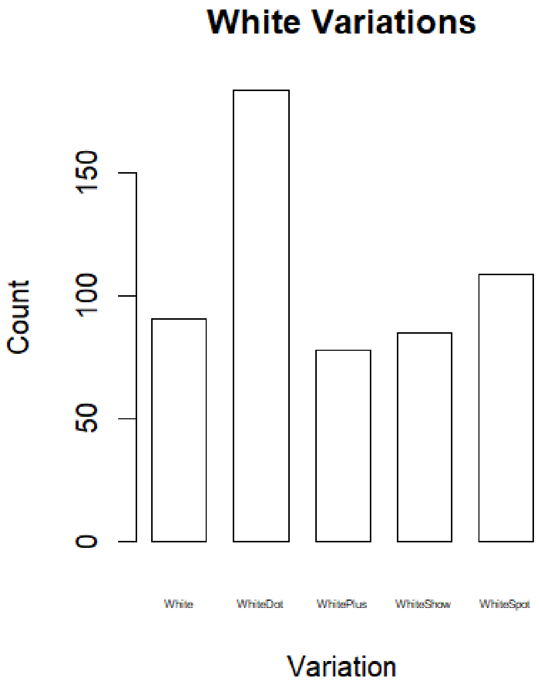
```
## # A tibble: 5 x 2
##   `alexa$variation`     n
##   <chr>             <int>
## 1 Black               261
## 2 BlackDot            516
## 3 BlackPlus           270
## 4 BlackShow           265
## 5 BlackSpot           241
```

```
blackPlot <- barplot(height = blackVars$n,
        names.arg = blackVars$`alexa$variation`,
        col = c("black"),
        main = "Black Variations",
        xlab = "Variation",
        ylab = "Count",
        border = "black",
        space = 0.5,
        cex.names = 0.4)


whitePlot <- barplot(height = whiteVars$n,
        names.arg = whiteVars$`alexa$variation`,
        col = c("white"),
        main = "White Variations",
        xlab = "Variation",
        ylab = "Count",
        border = "black",
        space = 0.5,
        cex.names = 0.4)
```



```
knitr::include_graphics("/cloud/project/RWorksheet#4/blackNwhiteVars.png")
```


Black Variations
Count
0
100
200
300
400
500
Black
BlackDot
BlackPlus
BlackShow
BlackSpot
Variation


White Variations
Count
0
50
100
150
White
WhiteDot
WhitePlus
WhiteShow
WhiteSpot
Variation