Worksheet#4

Marivic Amuan

2023-11-11

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

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

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

for (i in 1:5) {
  for (p in 1:5) {
    if (i == p) {
      x [i, p] <- vectorA[i]
    }

  }
}


abs(x)
```

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

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

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. U
start_num <- as.numeric(readline("Enter the starting number for the Fibonacci sequence: "))
```

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

```r
if (is.na(start_num)) {
  cat("Enter a valid numeric starting number.\n")
} else {

  num1 <- 0
  num2 <- 1

  while (num2 <= 500) {
    if (!is.na(start_num) && num2 >= start_num) {
      cat(num2, " ")
    }

    fib_sum <- num1 + num2
    num1 <- num2
    num2 <- fib_sum
  }
}
```

```
## Enter a valid numeric starting number.
```

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

#4A.

prevDATA <- read.csv("householdData.csv")
head(prevDATA)
```

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

```r
#4B.

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

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

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

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

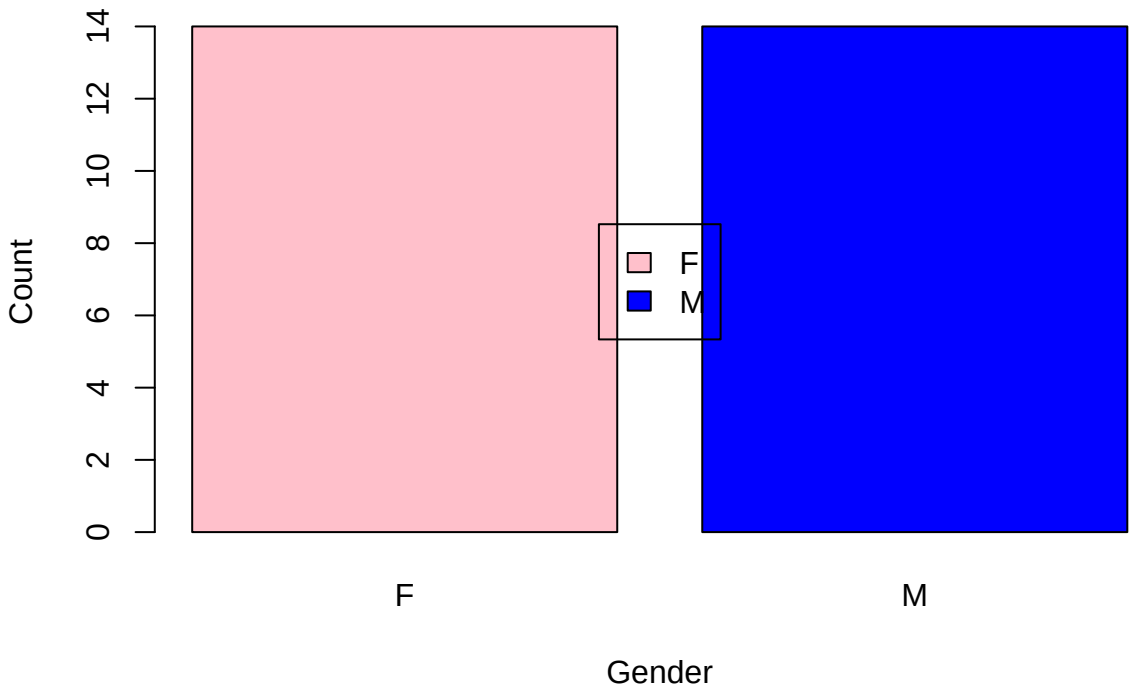
```r
numOfMales <- nrow(males)
numOfMales
```

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

```r
numOfFemales <- nrow(females)
numOfFemales
```

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

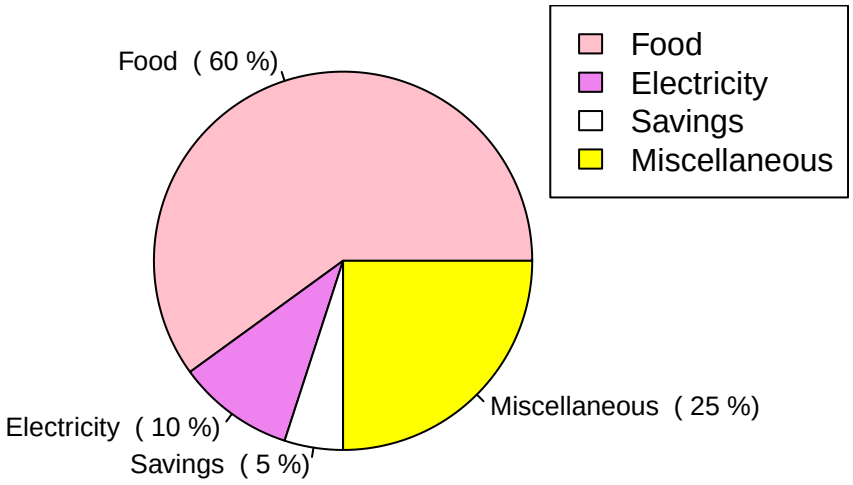
```r
#4C.

totalMaleFemale <- table(prevDATA$Gender)
barplot(totalMaleFemale, main = "Number of Females and Males", xlab = "Gender", ylab = "Count", col = c
        legend("center", legend = rownames(totalMaleFemale), fill = c("Pink", "Blue"))
```




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

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

spendingData$Percentage <- spendingData$Value / sum (spendingData$Value) * 100
colors <- c("Blue", "Yellow", "White", "Green")
```

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

spendingData$Percentage <- spendingData$Value / sum (spendingData$Value) * 100
colors <- c("Pink", "Violet", "White", "Yellow")

pie(spendingData$Value,
    labels = paste(spendingData$Category, " (", spendingData$Percentage, "%)"),
    col = colors,
    main = "The Monthly Income Spending of Dela Cruz Family",cex = 0.8)

legend("topright", spendingData$Category, fill = colors)
```

The Monthly Income Spending of Dela Cruz Family




```
#6. Use the iris dataset.
#6.A. Check for the structure of the dataset using the str() function. Describe what you have seen in t

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 ...
```

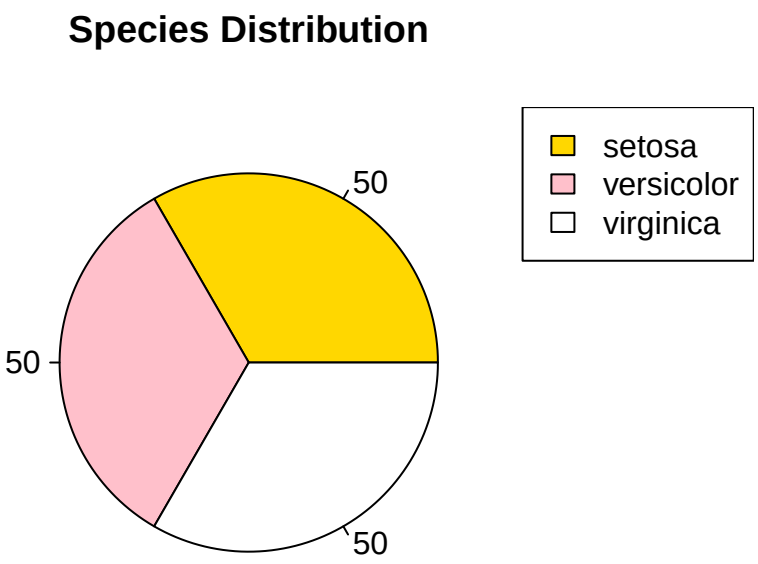
```
#OUTPUT:
#ANSWER: The dataset, structured as a data frame with 150 observations and 5 variables, includes numeri

#'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 ...
```

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

valueMeans <- c(
    Lsepal <- mean(iris$Sepal.Length),
    Wsepal <- mean(iris$Sepal.Width) ,
    Lpetal <- mean(iris$Petal.Length),
    Wpetal <- mean(iris$Petal.Width)
  )
valueMeans
```

```
## [1] 5.843333 3.057333 3.758000 1.199333
```

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

species <- table(iris$Species)
colors <- c("Gold", "Pink", "White")
```

```
pie(species, col = colors, labels = species )
legend("topright", legend = levels(iris$Species), fill = colors)
title("Species Distribution")
```




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

setosaSubset <- iris[iris$Species == "Setosa", ]
setosaSubset
```

```
## [1] Sepal.Length Sepal.Width  Petal.Length Petal.Width  Species
## <0 rows> (or 0-length row.names)
```

```
versicolorSubset <- iris[iris$Species == "VersiColor", ]
versicolorSubset
```

```
## [1] Sepal.Length Sepal.Width  Petal.Length Petal.Width  Species
## <0 rows> (or 0-length row.names)
```

```
virginicaSubset <- iris[iris$Species == "Virginica", ]
virginicaSubset
```

```
## [1] Sepal.Length Sepal.Width  Petal.Length Petal.Width  Species
## <0 rows> (or 0-length row.names)
```

```
tail(setosaSubset, 6)
```

```
## [1] Sepal.Length Sepal.Width  Petal.Length Petal.Width  Species
## <0 rows> (or 0-length row.names)
```

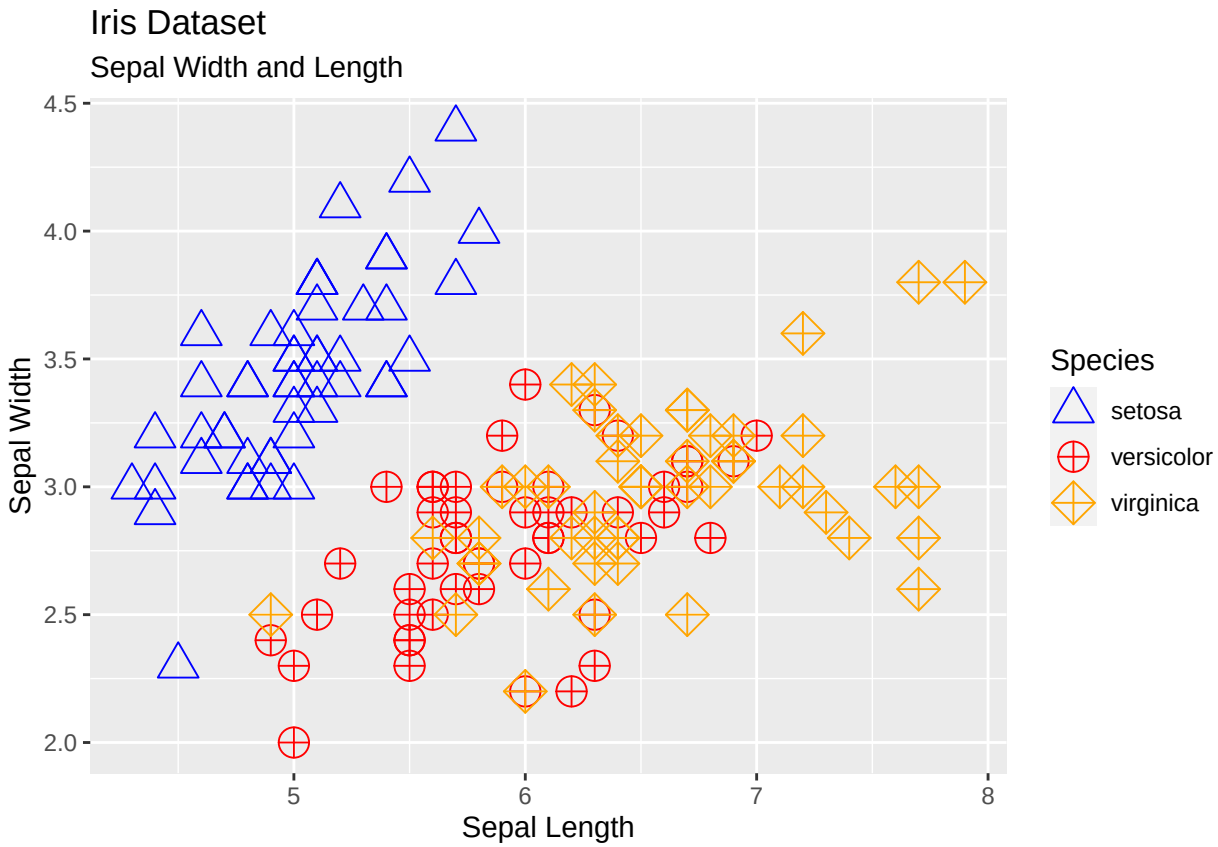
```
tail(versicolorSubset, 6)
```

```
## [1] Sepal.Length Sepal.Width  Petal.Length Petal.Width  Species
## <0 rows> (or 0-length row.names)
```

```
tail(virginicaSubset, 6)
```

```
## [1] Sepal.Length Sepal.Width  Petal.Length Petal.Width  Species
## <0 rows> (or 0-length row.names)
```

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

library(ggplot2)

iris$Species <- as.factor(iris$Species)
scatterplot <- ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species, shape = Species)) +
  geom_point(size = 5) +
  labs(
    title = "Iris Dataset",
    subtitle = "Sepal Width and Length",
    x = "Sepal Length",
    y = "Sepal Width"
  ) +
  scale_color_manual(values = c("setosa" = "Blue", "versicolor" = "Red", "virginica" = "Orange")) +
  scale_shape_manual(values = c("setosa" = 2, "versicolor" = 10, "virginica" = 9))

print(scatterplot)
```




```r
#ANSWER: The scatterplot visually demonstrates distinct clusters for each Iris species (setosa, versico

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

library(readxl)

alexa_file <- read_excel("alexa_file.xlsx")
alexa_file
```

```
## # 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
```

```
#7A. Rename the white and black variants by using gsub() function.

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

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

alexa_file
```

```
## # 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
```

```
#7B. Get the total number of each variations and save it into another object. Save the object as variat

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
```

```
var <-alexa_file %>%
  count(alexa_file$variation)
var
```

```
## # A tibble: 16 x 2
##    `alexa_file$variation`           n
##    <chr>                        <int>
##  1 Black                          261
##  2 Black  Dot                    516
##  3 Black  Plus                   270
##  4 Black  Show                   265
##  5 Black  Spot                   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 White  Dot                    184
## 14 White  Plus                    78
## 15 White  Show                    85
## 16 White  Spot                   109
```

```
var
```

```
## # A tibble: 16 x 2
##    `alexa_file$variation`           n
##    <chr>                        <int>
##  1 Black                          261
##  2 Black  Dot                    516
##  3 Black  Plus                   270
##  4 Black  Show                   265
##  5 Black  Spot                   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 White  Dot                    184
## 14 White  Plus                    78
## 15 White  Show                    85
## 16 White  Spot                   109
```

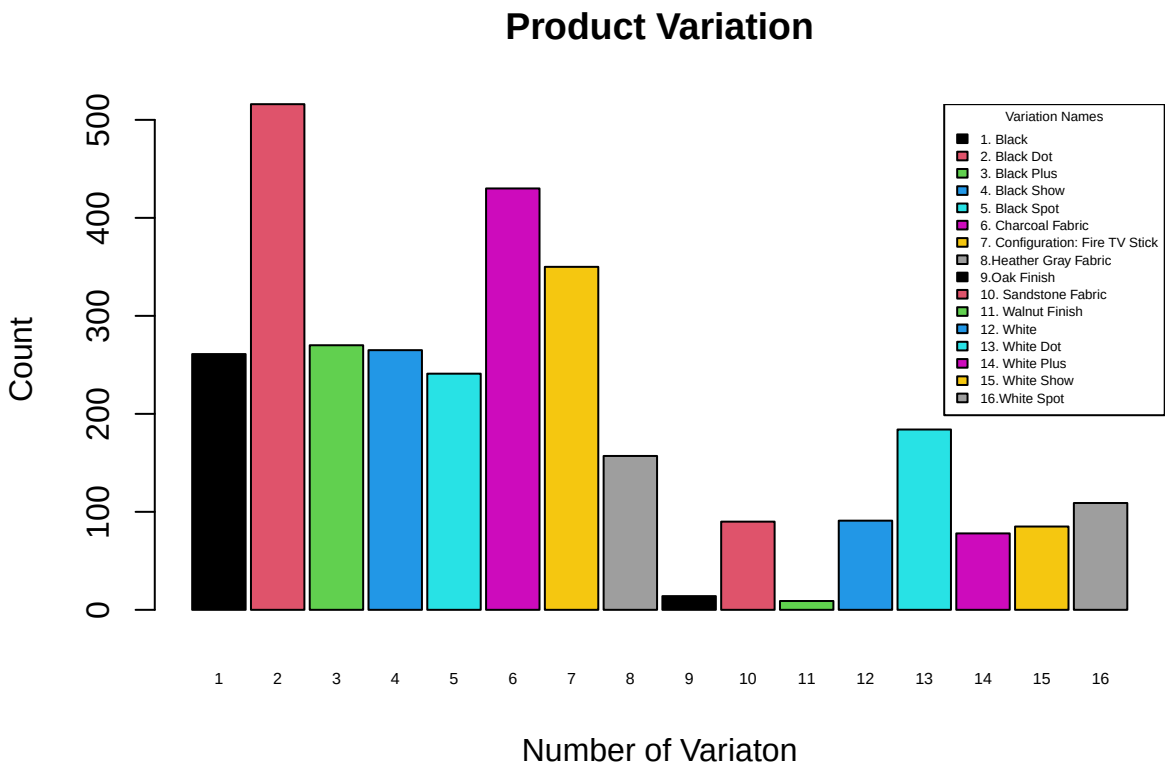
```
save(var, file = "variations.RData")
```

```
#7C. From the variations.RData, create a barplot(). Complete the details of the chart which include the

load("variations.RData")

varNames <- c(
  "1. Black", "2. Black Dot", "3. Black Plus", "4. Black Show",
```

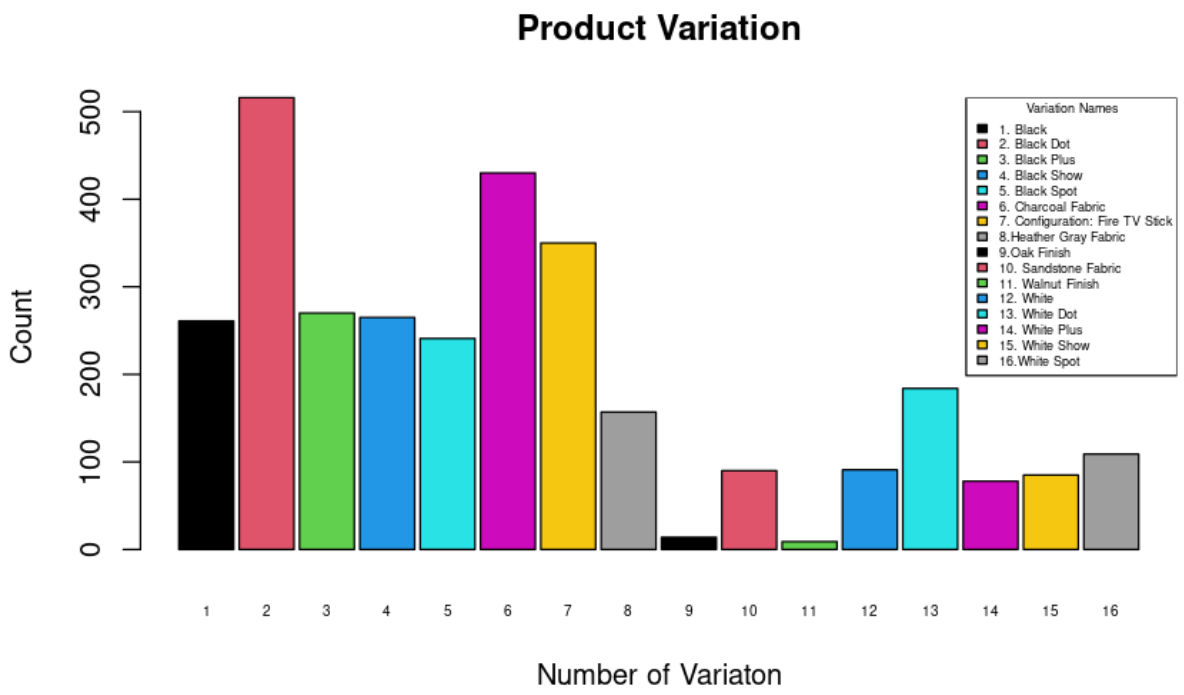
```
  "5. Black Spot", "6. Charcoal Fabric", "7. Configuration: Fire TV Stick",
  "8.Heather Gray Fabric", "9.Oak Finish", "10. Sandstone Fabric",
  "11. Walnut Finish", "12. White", "13. White Dot", "14. White Plus", "15. White Show", "16.White Spot"
)
varNames
```

```
##  [1] "1. Black"                        "2. Black Dot"
##  [3] "3. Black Plus"                   "4. Black Show"
##  [5] "5. Black Spot"                   "6. Charcoal Fabric"
##  [7] "7. Configuration: Fire TV Stick" "8.Heather Gray Fabric"
##  [9] "9.Oak Finish"                    "10. Sandstone Fabric"
## [11] "11. Walnut Finish"               "12. White"
## [13] "13. White Dot"                   "14. White Plus"
## [15] "15. White Show"                  "16.White Spot"
```

```
compPlot <- barplot(var$n,
            names.arg = 1:16,
            col = 1:16,
            main = "Product Variation",
            xlab = "Number of Variaton",
            ylab = "Count",
            las = 0.10,
            cex.names = 0.50,
            space = 0.10)
legend("topright", legend = varNames, fill = 1:16, title = "Variation Names", cex = 0.45)
```

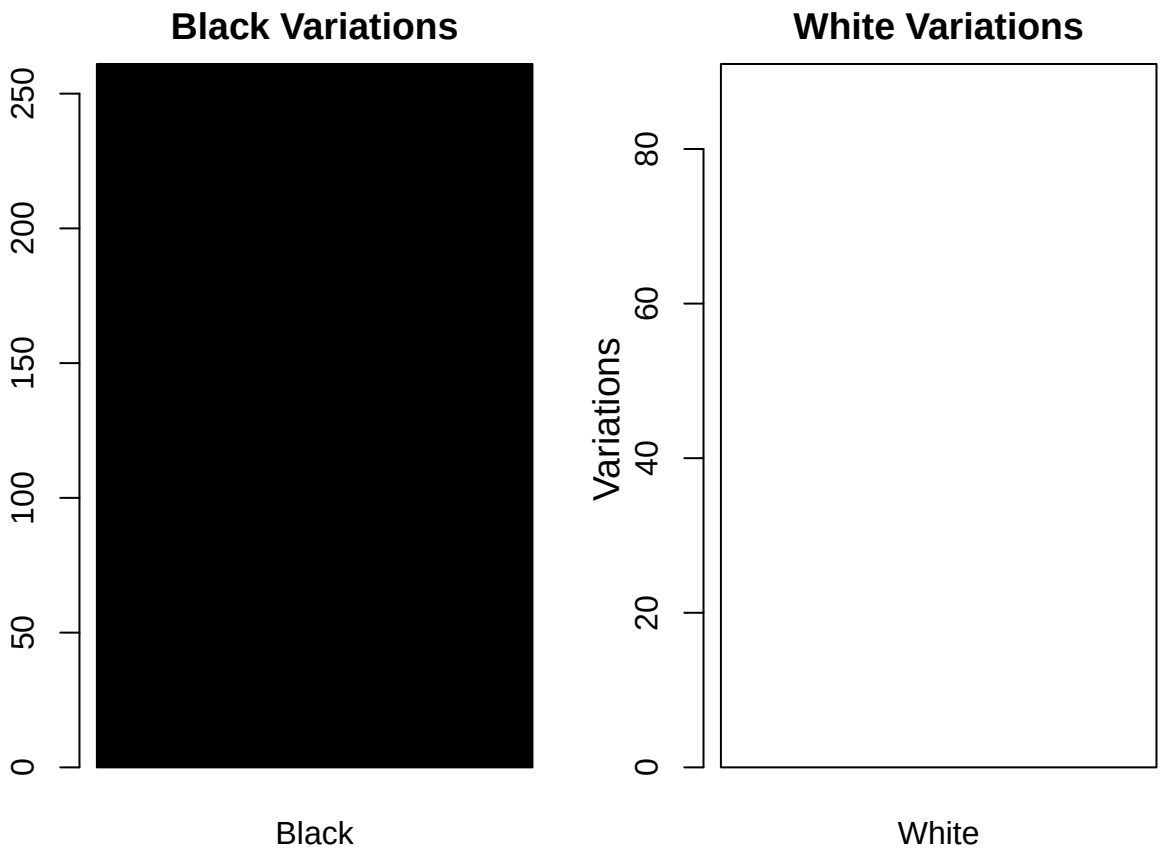



```
knitr::include_graphics("/cloud/project/Worksheet#4/Product Variation.png")
```

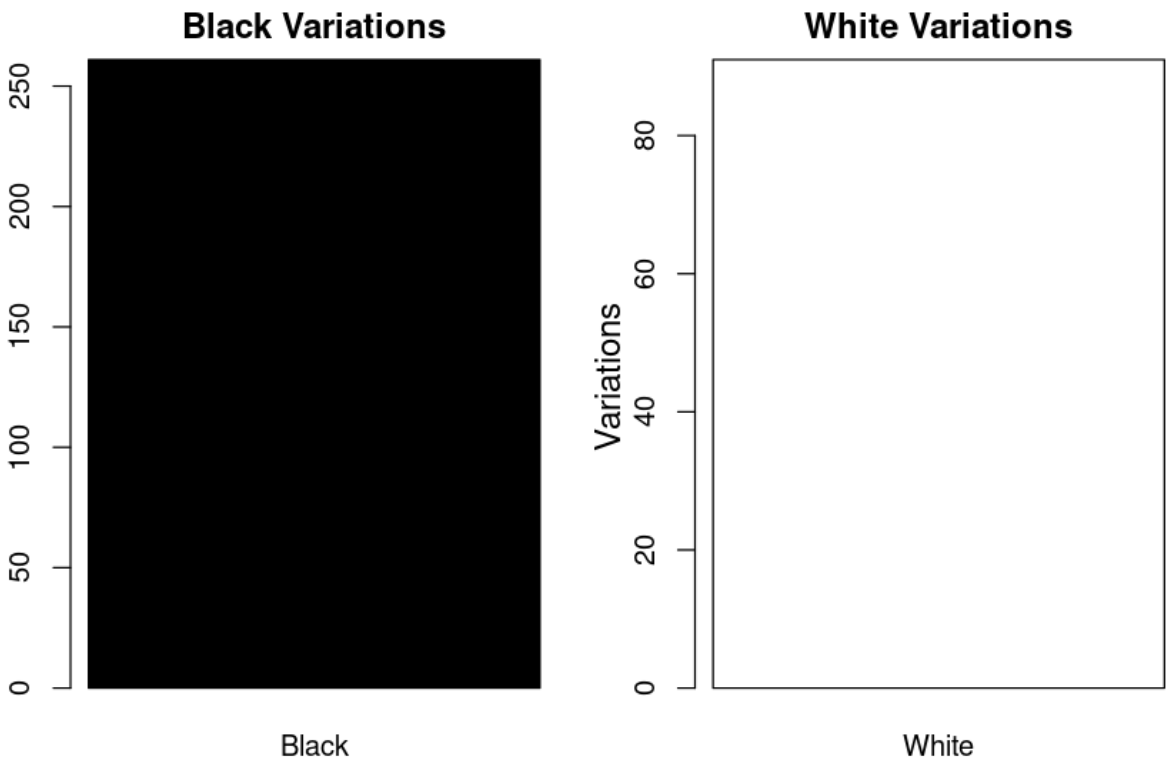



```
#7D. Create a barplot() for the black and white variations. Plot it in 1 frame, side by side. Complete

library(ggplot2)
library(magrittr)

blackPlot <- var [var$`alexa_file$variation` %in% c("Black", "BlackDot", "BlackShow", "BlackPlus", "Bla
whitePlot <- var [var$`alexa_file$variation` %in% c("White", "WhiteDot", "WhiteShow", "WhitePlus", "Whi

par(mfrow = c(1,2), mar = c(2,2,2,2))

black <- barplot(height = blackPlot$n,
                 names.arg = blackPlot$`alexa_file$variation`,
                 col = "Black",
                 main = "Black Variations",
                 xlab = "Number of Variation",
                 ylab = "Count",
                 las = 0.0,
                 cex.names = 1.0,
                 space = 0.1)

white <- barplot(height = whitePlot$n,
                 names.arg = whitePlot$`alexa_file$variation`,
                 col = "White",
                 main = "White Variations",
                 xlab = "Number of Variation",
                 ylab = "Count",
                 las = 0.0,
                 cex.names = 1.0,
                 space = 0.1
```

```
)
mtext("Variations", side = 2, line = 2, cex = 1.2)
```




```
knitr::include_graphics("/cloud/project/Worksheet#4/BlackWhite Variation.png")
```


Black Variations
0
50
100
150
200
250
Black
White Variations
Variations
0
20
40
60
80
White