Rworksheet_Barrientos#4B

Barrientos, Milfrance D.

10/30/2024

Using For Loop Function

```
#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.
# Hint Use abs() function to get the absolute value

vectorA <- c(1,2,3,4,5)
matrixA <- matrix(0,5,5)

for (i in 1:5) {
  for (j in 1:5) {
    matrixA[i,j] <- abs(vectorA[i] - vectorA[j])
  }
}
print(matrixA)
```

```
##      [,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.

rows <- 5
for (i in 1:rows) {
cat(paste(rep("*", i), collapse=" "), "\n")
}
```

```
## *
## * *
## * * *
## * * * *
## * * * * *
```

```
#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.

n <- 600
n1 <- 0
n2 <- 1
repeat {

  if (n1 >= n) {
    cat(n1, "\n")
  }


  if (n1 > 500) {
    break
  }


  fib <- n1 + n2
  n1 <- n2
  n2 <- fib
}
```

```
## 610
```

#Using Basic Graphics (plot(),barplot(),pie(),hist())

```
household_data <- data.frame(
  Shoe_size = c(6.5, 9.0, 8.5, 8.5, 10.5, 7.0, 9.5, 9.0, 13.0, 7.5, 10.5, 8.5, 12.0, 10.5,
13.0, 11.5, 8.5, 5.0, 10.0, 6.5, 7.5,
8.5, 10.5, 8.5, 10.5, 11.0, 9.0, 13.0),
  Height = c(66.0, 68.0, 64.5, 65.0, 70.0, 64.0, 70.0, 71.0, 72.0, 64.0, 74.5, 67.0, 71.0, 71.0,
77.0, 72.0, 59.0, 62.0, 72.0, 66.0, 64.0,
67.0, 73.0, 69.0, 72.0, 70.0, 69.0, 70.0),
  Gender = c("F","F","F","F","M","F","F","F","M",
"F","M","F","M","M","M","M","F","F","M",
"F","F","M","M","F","M","M","M","M")
)
household_data
```

```
##    Shoe_size 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
## 7        9.5   70.0      F
## 8        9.0   71.0      F
## 9       13.0   72.0      M
## 10       7.5   64.0      F
## 11      10.5   74.5      M
## 12       8.5   67.0      F
```

```
## 13      12.0   71.0      M
## 14      10.5   71.0      M
## 15      13.0   77.0      M
## 16      11.5   72.0      M
## 17       8.5   59.0      F
## 18       5.0   62.0      F
## 19      10.0   72.0      M
## 20       6.5   66.0      F
## 21       7.5   64.0      F
## 22       8.5   67.0      M
## 23      10.5   73.0      M
## 24       8.5   69.0      F
## 25      10.5   72.0      M
## 26      11.0   70.0      M
## 27       9.0   69.0      M
## 28      13.0   70.0      M
```

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

#install.packages("readxl") for importing an excel file
#install.packages("readr") for importing a csv file
library(readxl)
library(readr)
household_data <- read_excel("C:/PROJ/household_data.xlsx")

head(household_data)
```

```
## # A tibble: 6 x 3
##   Shoe_size Height Gender
##       <dbl>  <dbl> <chr>
## 1       6.5   66   F
## 2       9     68   F
## 3       8.5   64.5 F
## 4       8.5   65   F
## 5      10.5   70   M
## 6       7     64   F
```

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

fem_data <- subset(household_data, Gender == "F")
male_data <- subset(household_data, Gender == "M")

num_fem <- nrow(fem_data)
num_male <- nrow(male_data)

cat("Number of female observations: ", num_fem, "\n")
```

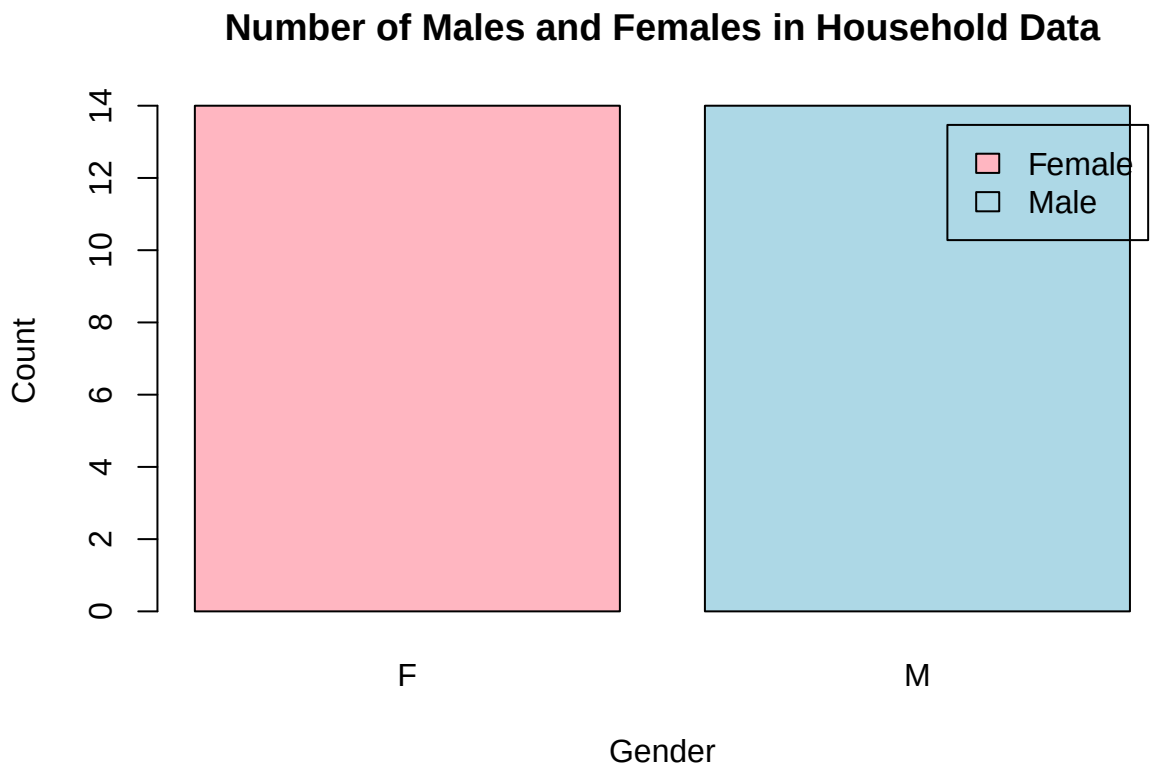
```
## Number of female observations:  14
```

```
cat("Number of male obserbations: ",  num_male, "\n")
```

```
## Number of male obserbations:  14
```

```r
#c. Creating a Bar Plot for the Number of Males and Females
gender_counts <- table(household_data$Gender)

barplot(gender_counts,
        main = "Number of Males and Females in Household Data",
        xlab = "Gender",
        ylab = "Count",
        col = c("lightpink", "lightblue"),
        legend = c("Female", "Male"))
```

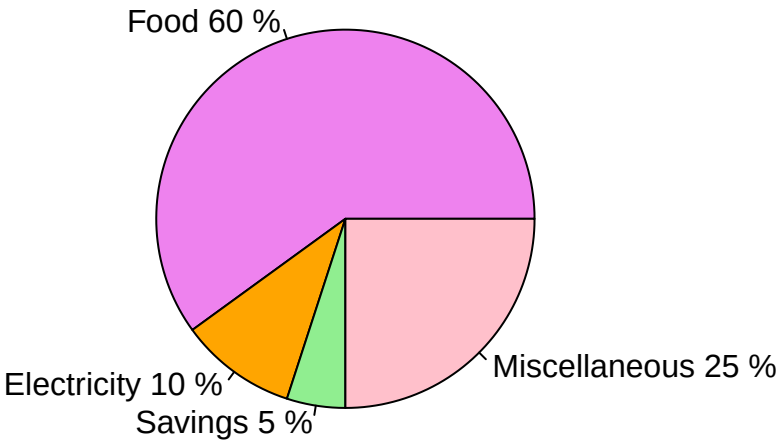



```r
# 5.
# a. Create a piechart that will include labels in percentage.Add some colors and title of the chart. W

expenses <- c(Food = 60, Electricity = 10, Savings = 5, Miscellaneous = 25)

percent_expenses <- round(100 * expenses / sum(expenses))
labels <- paste(names(percent_expenses), percent_expenses, "%")
colors <- c("violet", "orange", "lightgreen", "pink")

pie(expenses, labels = labels, col = colors, main = "Dela Cruz Family Monthly Expenses")
```



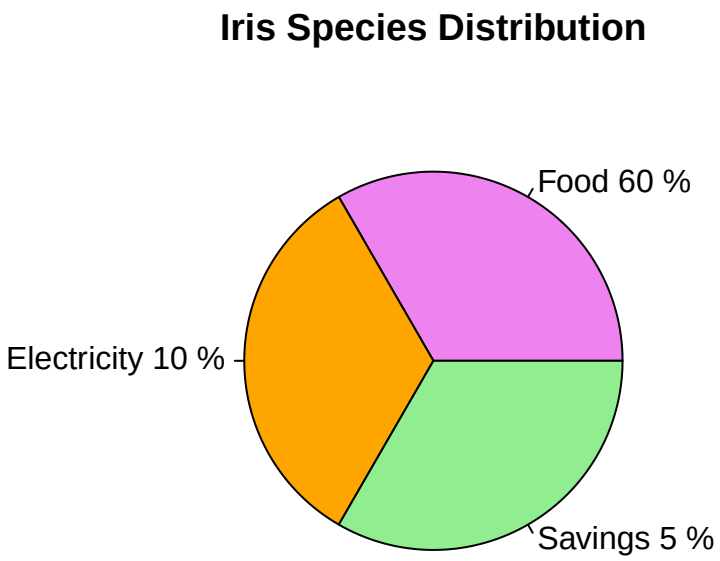

```r
# 6. Use the iris dataset.
#a. Check for the structure of the dataset using the str() function. Describe what you have seen in the
str(iris)
```

```
## '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
#b. Create an R object that will contain the mean of the sepal.length, sepal.width,petal.length,and pet
mean_values <- c(mean(iris$Sepal.Length), mean(iris$Sepal.Width), mean(iris$Petal.Length), mean(iris$Pe
mean_values
```

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

```r
#c. Create a pie chart for the Species distribution. Add title, legends, and colors. Write the R script
species_counts <- table(iris$Species)
colors <- c("violet", "orange", "lightgreen")
pie(species_counts, labels = labels, col = colors, main = "Iris Species Distribution")
```




```
#d. Subset the species into setosa, versicolor, and virginica. Write the R scripts and show the last si
setosa <- subset(iris, Species == "setosa")
versicolor <- subset(iris, Species == "versicolor")
virginica <- subset(iris, Species == "virginica")

tail(setosa)
```

```
##    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
```

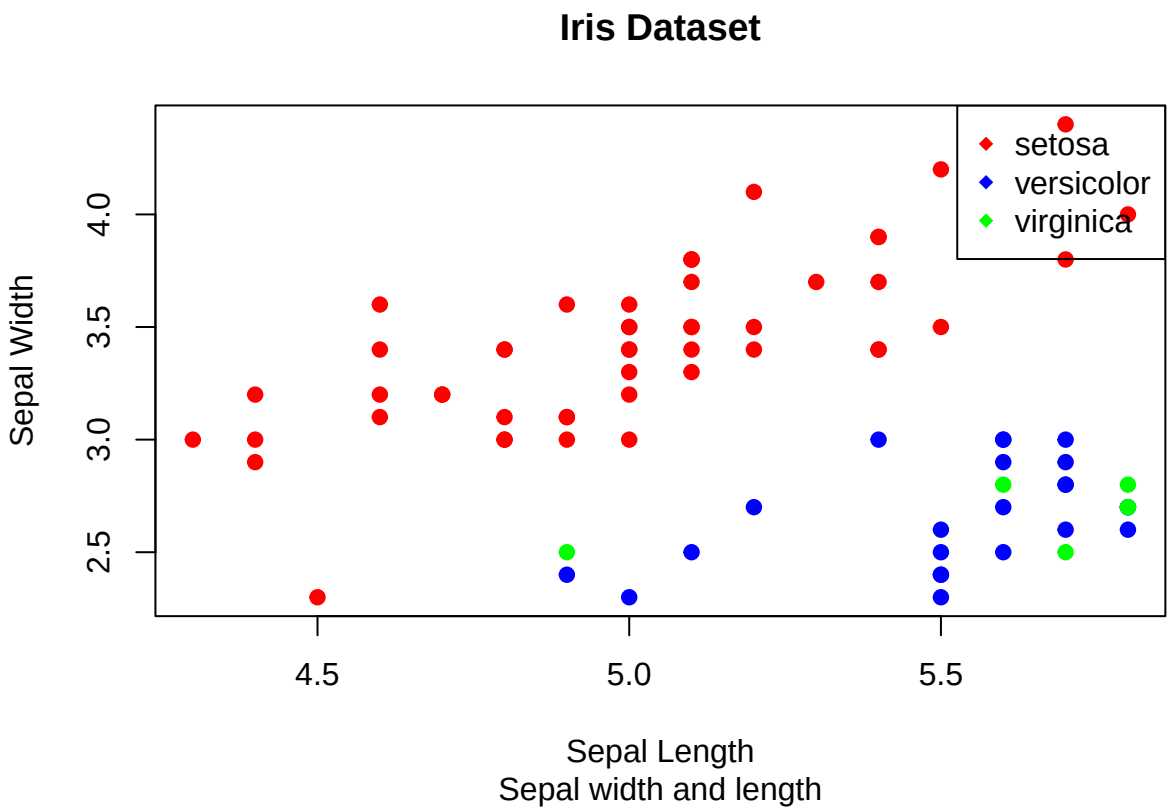
```
#e. Create a scatterplot of the sepal.length and sepal.width using the different species(setosa,versico
#Add a title = "Iris Dataset", subtitle = "Sepal
#width and length, labels for the x and y axis, the pch symbol and colors should be based on the specie

#Hint: Need to convert to factors the species to store categorical variables.
as.factor(iris$Species)
```

```
##   [1] setosa     setosa     setosa     setosa     setosa     setosa
##   [7] setosa     setosa     setosa     setosa     setosa     setosa
##  [13] setosa     setosa     setosa     setosa     setosa     setosa
```

```
##  [19] setosa     setosa     setosa     setosa     setosa     setosa
##  [25] setosa     setosa     setosa     setosa     setosa     setosa
##  [31] setosa     setosa     setosa     setosa     setosa     setosa
##  [37] setosa     setosa     setosa     setosa     setosa     setosa
##  [43] setosa     setosa     setosa     setosa     setosa     setosa
##  [49] setosa     setosa     versicolor versicolor versicolor versicolor
##  [55] versicolor versicolor versicolor versicolor versicolor versicolor
##  [61] versicolor versicolor versicolor versicolor versicolor versicolor
##  [67] versicolor versicolor versicolor versicolor versicolor versicolor
##  [73] versicolor versicolor versicolor versicolor versicolor versicolor
##  [79] versicolor versicolor versicolor versicolor versicolor versicolor
##  [85] versicolor versicolor versicolor versicolor versicolor versicolor
##  [91] versicolor versicolor versicolor versicolor versicolor versicolor
##  [97] versicolor versicolor versicolor versicolor virginica  virginica
## [103] virginica  virginica  virginica  virginica  virginica  virginica
## [109] virginica  virginica  virginica  virginica  virginica  virginica
## [115] virginica  virginica  virginica  virginica  virginica  virginica
## [121] virginica  virginica  virginica  virginica  virginica  virginica
## [127] virginica  virginica  virginica  virginica  virginica  virginica
## [133] virginica  virginica  virginica  virginica  virginica  virginica
## [139] virginica  virginica  virginica  virginica  virginica  virginica
## [145] virginica  virginica  virginica  virginica  virginica  virginica
## Levels: setosa versicolor virginica
```

```r
plot(setosa$Sepal.Length, setosa$Sepal.Width, pch = 19, col = "red", xlab = "Sepal Length", ylab = "Sepa
points(versicolor$Sepal.Length, versicolor$Sepal.Width, pch = 19, col = "blue")
points(virginica$Sepal.Length, virginica$Sepal.Width, pch = 19, col = "green")
legend("topright", legend = levels(iris$Species), col = c("red", "blue", "green"), pch = 18)
```




```
#f. Interpret the result.
#The scatterplot shows the relationship between the sepal length and sepal width of the iris dataset.
```

Basic Cleaning and Transformation of Objects

```
# Load necessary libraries
library(readxl)
library(dplyr) # for data manipulation
```

```
##
## 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
```

```r
library(knitr) # for kable


alexa_data <- read_excel("C:/PROJ/alexa_file.xlsx")

str(alexa_data)
```

```
## tibble [3,150 x 5] (S3: tbl_df/tbl/data.frame)
##  $ rating          : num [1:3150] 5 5 4 5 5 5 3 5 5 5 ...
##  $ date            : POSIXct[1:3150], format: "2018-07-31" "2018-07-31" ...
##  $ variation       : chr [1:3150] "Charcoal Fabric" "Charcoal Fabric" "Walnut Finish" "Charcoal Fabri
##  $ verified_reviews: chr [1:3150] "Love my Echo!" "Loved it!" "Sometimes while playing a game, you ca
##  $ feedback        : num [1:3150] 1 1 1 1 1 1 1 1 1 1 ...
```

```r
alexa_data$variation <- gsub("Black Dot", "BlackDot", alexa_data$variation)
alexa_data$variation <- gsub("Black Plus", "BlackPlus", alexa_data$variation)
alexa_data$variation <- gsub("Black Show", "BlackShow", alexa_data$variation)
alexa_data$variation <- gsub("Black Spot", "BlackSpot", alexa_data$variation)
alexa_data$variation <- gsub("White Dot", "WhiteDot", alexa_data$variation)
alexa_data$variation <- gsub("White Plus", "WhitePlus", alexa_data$variation)
alexa_data$variation <- gsub("White Show", "WhiteShow", alexa_data$variation)
alexa_data$variation <- gsub("White Spot", "WhiteSpot", alexa_data$variation)


head(alexa_data)
```

```
## # A tibble: 6 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 playi~        1
## 4      5 2018-07-31 00:00:00 Charcoal Fabric     I have had a lot of f~        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 a~        1
```

```r
knitr::kable(head(alexa_data))
```

rating	date	variation	verified_reviews	feedback
5	2018-07-31	Charcoal Fabric	Love my Echo!	1
5	2018-07-31	Charcoal Fabric	Loved it!	1
4	2018-07-31	Walnut Finish	Sometimes while playing a game, you can answer a question correctly but Alexa says you got it wrong and answers the same as you. I like being able to turn lights on and off while away from home.	1

rating	date	variation	verified_reviews	feedback
5	2018-07-31	Charcoal Fabric	I have had a lot of fun with this thing. My 4 yr old learns about dinosaurs, i control the lights and play games like categories. Has nice sound when playing music as well.	1
5	2018-07-31	Charcoal Fabric	Music	1
5	2018-07-31	Heather Gray Fabric	I received the echo as a gift. I needed another Bluetooth or something to play music easily accessible, and found this smart speaker. Can't wait to see what else it can do.	1

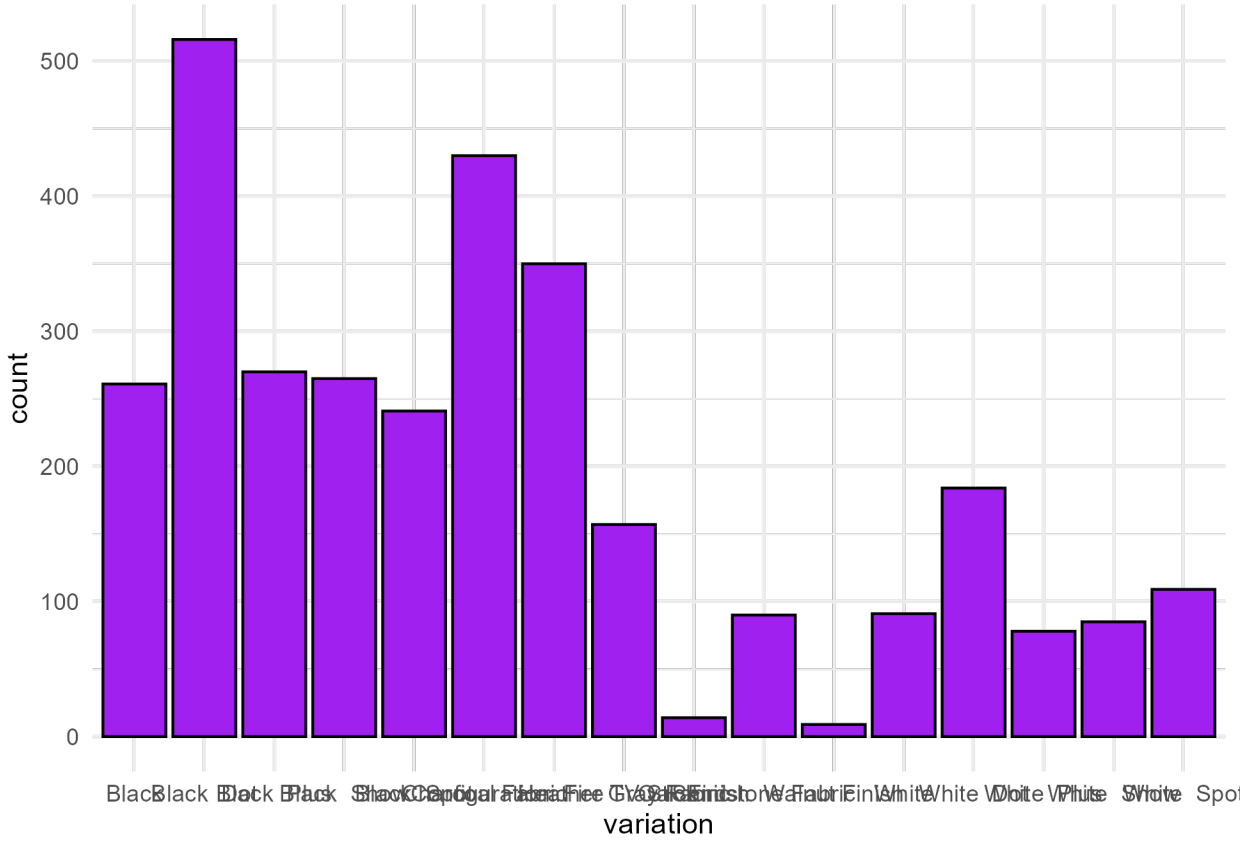
```
library(ggplot2)


plot <- ggplot(alexa_data, aes(x = variation)) +
        geom_bar(fill = "purple", color = "black") +
        theme_minimal()


ggsave("C:/PROJ/alexa_plot.png", plot)
```

```
## Saving 6.5 x 4.5 in image
```

```
knitr::include_graphics("C:/PROJ/alexa_plot.png")
```



```r
#b. 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.
#Syntax for dplyr RObject %>% count(RObject$columnName)

library(dplyr)
variation <- alexa_data %>%
  count(variation)

save(variation, file = "C:/PROJ/variations.RData")
print(variation)
```

```
## # A tibble: 16 x 2
##    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
```

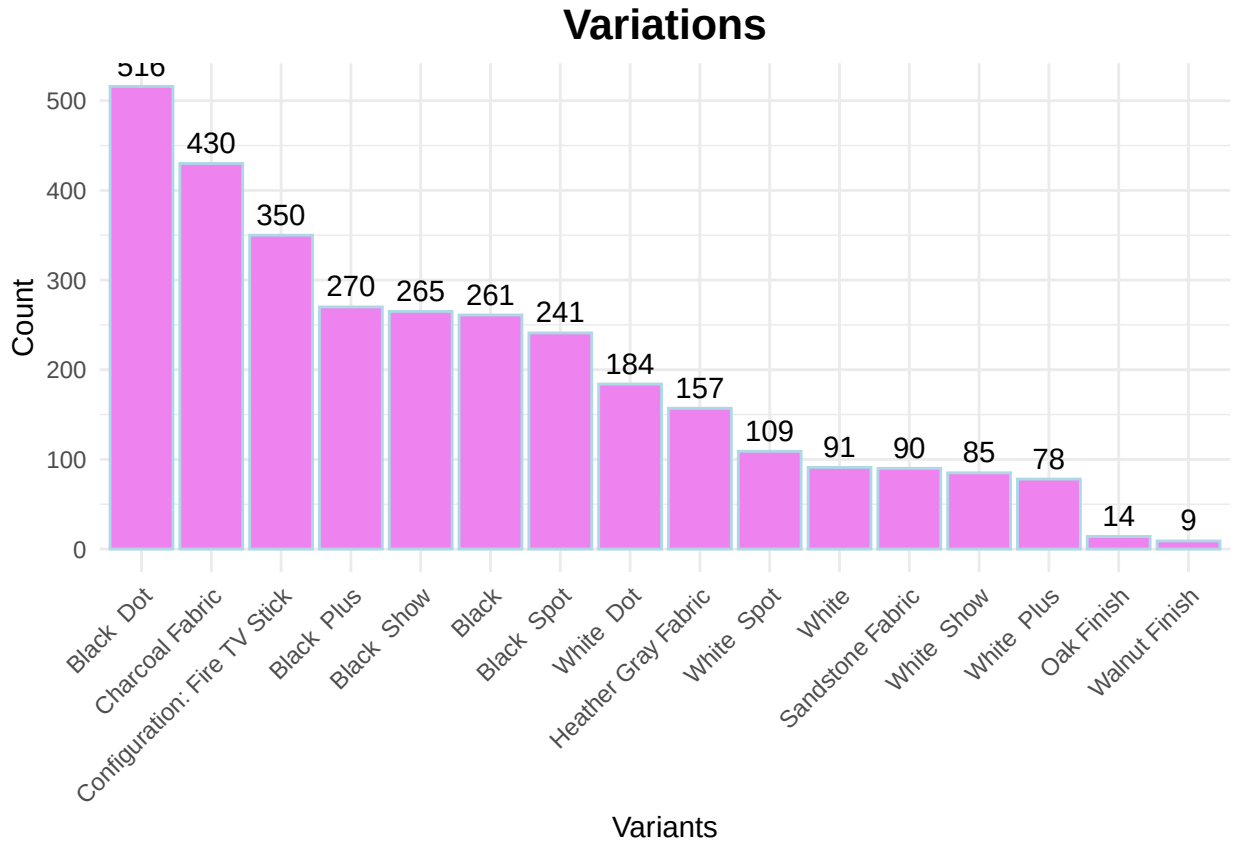
```r
#c. From the variations.RData, create a barplot(). Complete the details of the chart which include the

library(dplyr)
library(ggplot2)

load("C:/PROJ/variations.RData")

plot <- ggplot(variation, aes(x = reorder(variation, -n), y = n)) +
  geom_bar(stat = "identity", fill = "violet", color = "lightblue") +
  labs(title = "Variations",
       x = "Variants",
       y = "Count") +
  geom_text(aes(label = n), vjust = -0.5) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )

print(plot)
```



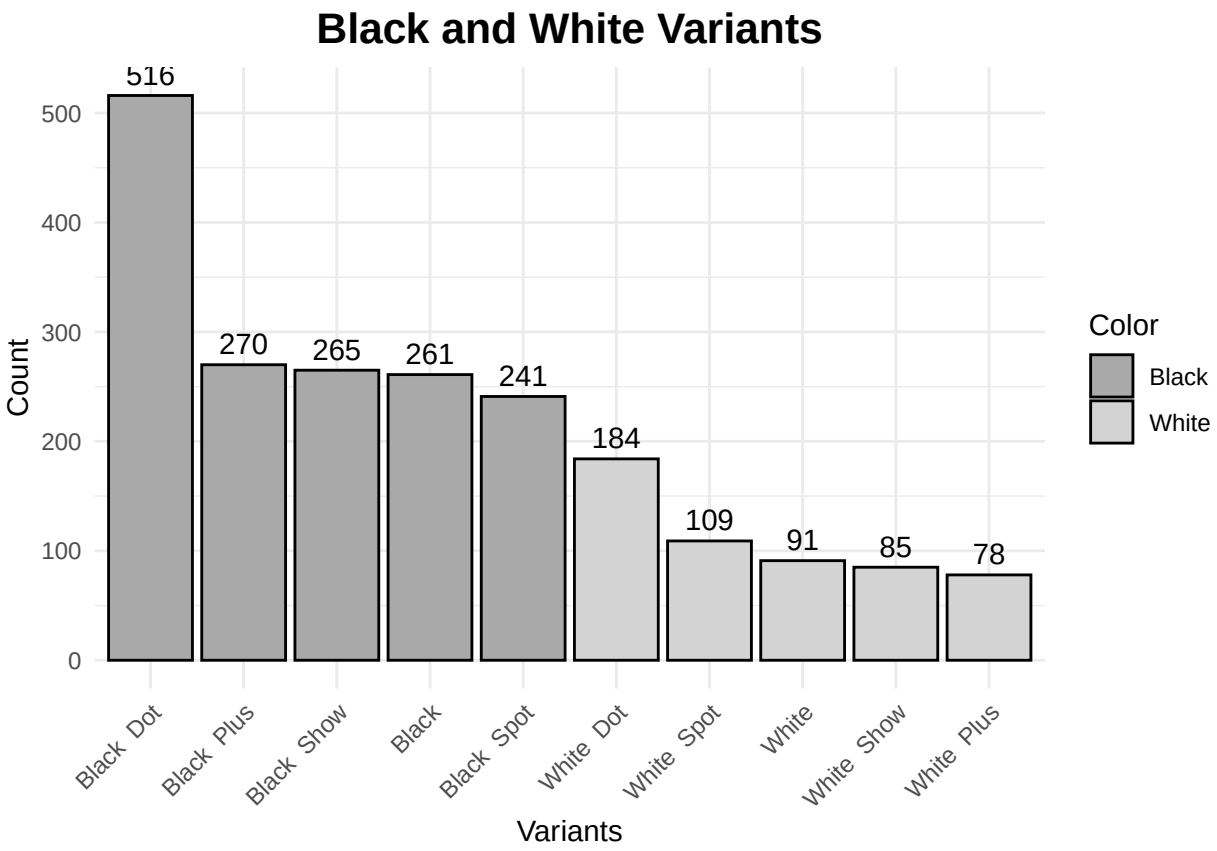

```r
#d. Create a barplot() for the black and white variations. Plot it in 1 frame, side by side. Complete t
library(dplyr)
library(ggplot2)

load("C:/PROJ/variations.RData")

black_white_variations <- variation %>%
  filter(grepl("Black|White", variation))

plot <- ggplot(black_white_variations, aes(x = reorder(variation, -n), y = n, fill = ifelse(grepl("Blac
  geom_bar(stat = "identity", color = "black", position = position_dodge(width = 0.9)) +
  labs(title = "Black and White Variants",
       x = "Variants",
       y = "Count",
       fill = "Color") +
  geom_text(aes(label = n), vjust = -0.5, position = position_dodge(width = 0.9)) +
  scale_fill_manual(values = c("Black" = "darkgray", "White" = "lightgray")) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )

print(plot)
```


Black and White Variants
516
270
265
261
241
184
109
91
85
78
500
400
300
200
100
0
Count
Black Dot
Black Plus
Black Show
Black
Black Spot
White Dot
White Spot
White
White Show
White Plus
Variants
Color
Black
White