RWorksheet_Gerona#4b

Mariel M. Gerona

#1. Create a 5x5 Matrix with for Loop

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

matrixA <- matrix(0, nrow = 5, ncol = 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 star pattern using for Loop

```
# Prints a triangle with 5 rows of stars
for(i in 1:5) {
  for(j in 1:i) {
    cat("*")
  }
  cat("\n")
}
```

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

#3. Fibonacci Sequence up to 500

```
fibonacci <- function(n) {
  a <- 0
  b <- 1
  repeat {
    cat(a, " ")
    temp <- a + b
    a <- b
    b <- temp
    if (a > 500) break
```

```
  }
}
fibonacci()
```

```
## 0  1  1  2  3  5  8  13  21  34  55  89  144  233  377
```

#4. Importing and Analyzing Dataset #a. Import CSV file and display the first 6 rows

```
library(readr)

shoe_sizes <- read_csv("Shoe Sizes.csv")
```

```
## Rows: 28 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Gender
## dbl (2): Shoe size, Height
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(shoe_sizes)
```

```
## # 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 and count observations:

```
male <- subset(shoe_sizes, Gender == "M")
female <- subset(shoe_sizes, Gender == "F")
num_males <- nrow(male)
num_females <- nrow(female)

cat("Number of Male observations:", num_males, "\n")
```

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

```
cat("Number of Female observations:", num_females, "\n")
```

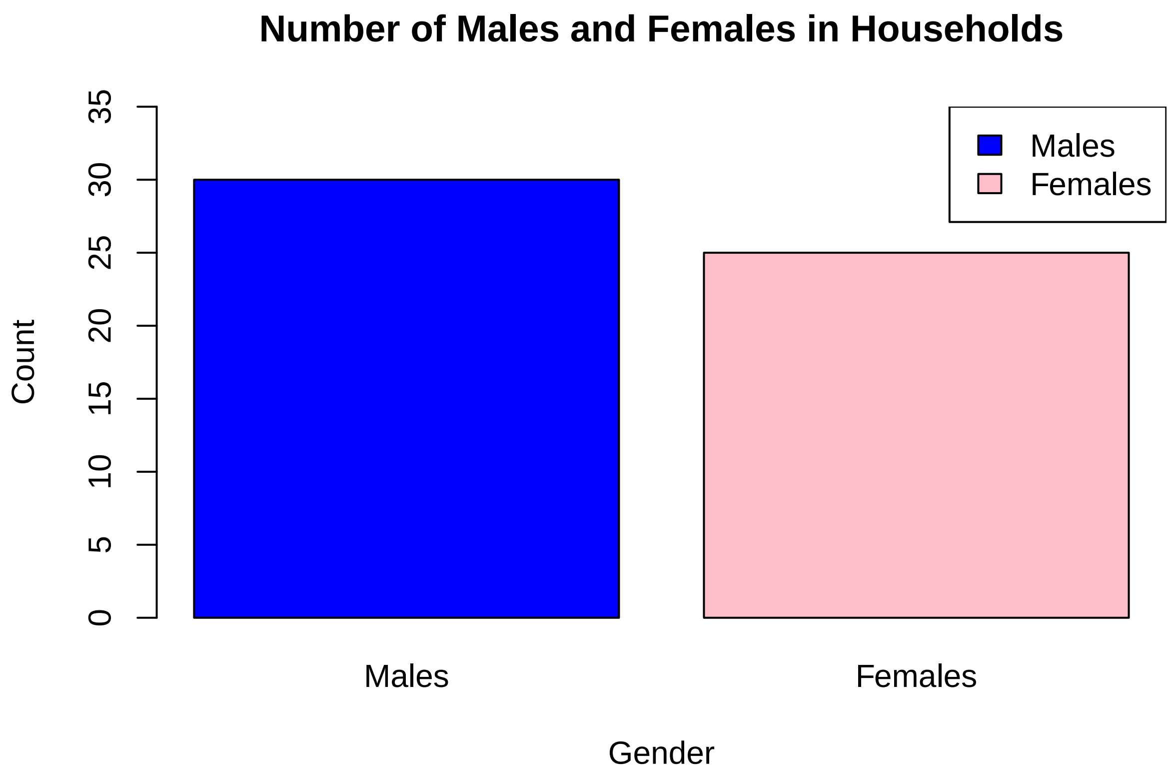
```
## Number of Female observations: 14
```

#c. Barplot for Male and Female Counts

```
gender_counts <- c(Males = 30, Females = 25)

barplot(gender_counts,
        main = "Number of Males and Females in Households",
        xlab = "Gender",
        ylab = "Count",
        col = c("blue", "pink"),
        ylim = c(0, max(gender_counts) + 5),
        beside = TRUE)
```

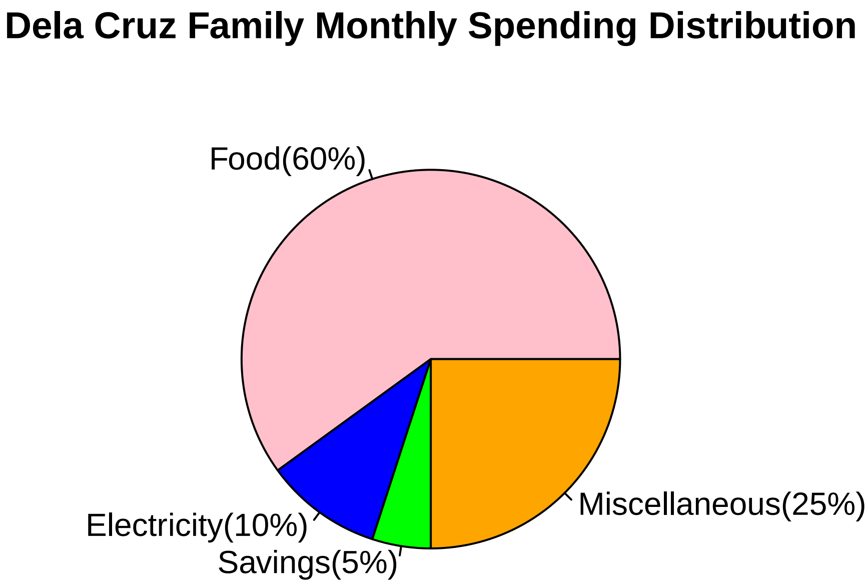
```r
legend("topright",
       legend = names(gender_counts),
       fill = c("blue", "pink"))
```



#5. Dela Cruz Family Monthly Income Pie Chart

```r
# Data for Dela Cruz family spending
categories <- c("Food", "Electricity", "Savings", "Miscellaneous")
spending <- c(60, 10, 5, 25)

# Create a pie chart
pie(spending,
    labels = paste(categories, "(", spending, "%)", sep=""),
    col = c("pink", "blue", "green", "orange"),
    main = "Dela Cruz Family Monthly Spending Distribution")
```



#6. Iris Dataset Analysis #a. Check structure of the iris dataset:

```
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 ...
```

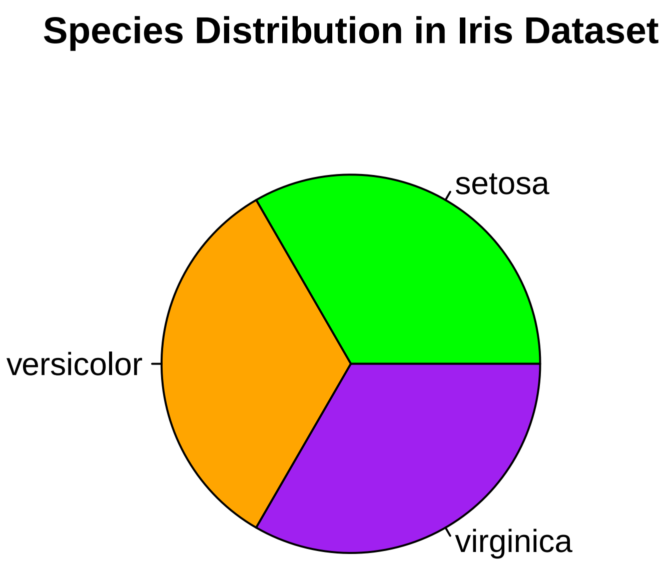
#b. Mean of Sepal and Petal dimensions:

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

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

#c. Species Distribution Pie Chart

```
pie(table(iris$Species), main = "Species Distribution in Iris Dataset", col = c("green", "orange", "pur
```



#d. Subset by Species and Display Last 6 Rows: # Subset by species

```
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
```

```
tail(versicolor)
```

```
##     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
```

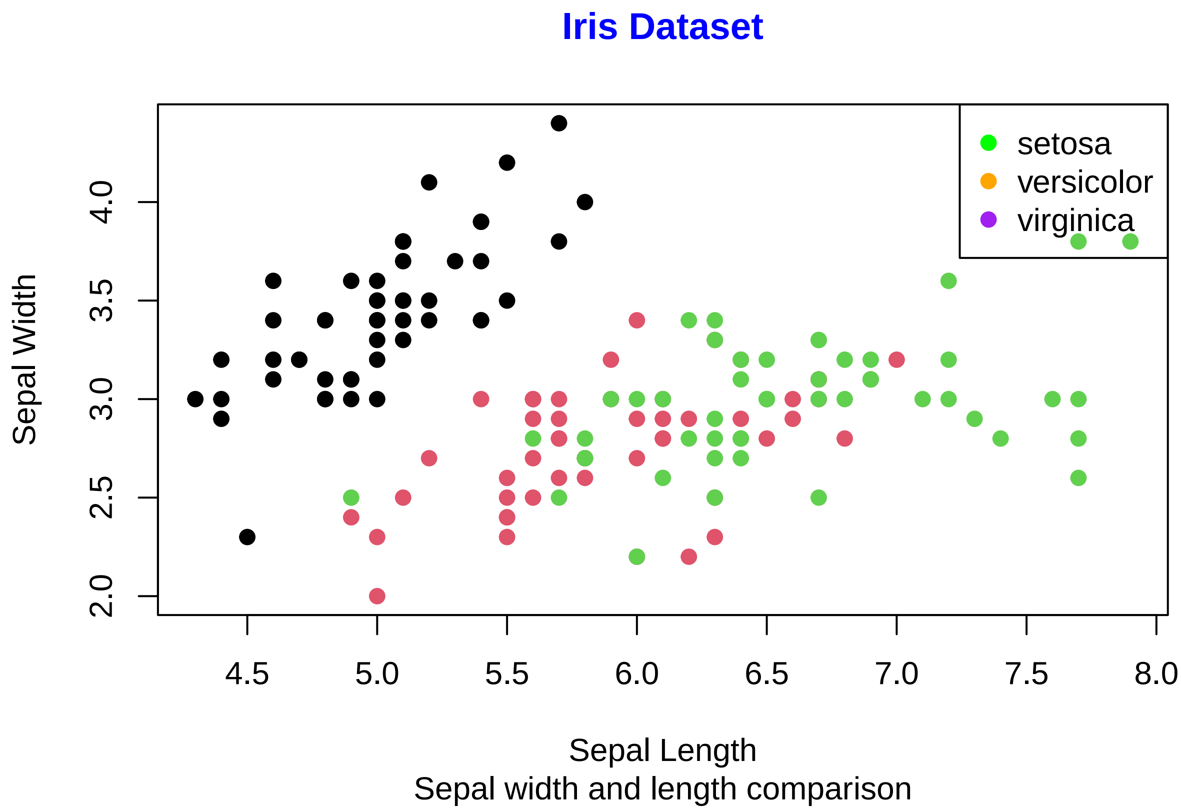
```
tail(virginica)
```

```
##     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. Scatter Plot for Sepal Dimensions by Species: # Scatter plot for Sepal.Length and Sepal.Width

```
plot(iris$Sepal.Length, iris$Sepal.Width, col = iris$Species, pch = 19,
     xlab = "Sepal Length", ylab = "Sepal Width", main = "Iris Dataset",
```

```
     sub = "Sepal width and length comparison", col.main = "blue")
legend("topright", legend = levels(iris$Species), col = c("green", "orange", "purple"), pch = 19)
```



#6f. Interpretation of the result. The scatterplot illustrates the relationship between Sepal.Length and Sepal.Width across three species of flowers. Setosa points, shown in orange, typically have shorter sepal lengths and widths, clearly distinguishing them from the other species. Versicolor points, depicted in pink, are more widely distributed and exhibit slight overlap with Virginica points in yellow, indicating some similarities between these two species. In contrast, Virginica points tend to have larger sepal lengths and widths, further distinguishing them from Setosa while overlapping with some Versicolor points. Overall, the scatterplot effectively highlights the distinct separation of Setosa from the other species, with Versicolor and Virginica demonstrating some degree of overlap.

#7.Import the alexa-file.xlsx

```
library(readxl)
data <- read_excel("/cloud/project/Worksheet#4/Worksheet#4b/alexa-file.xlsx")

data$variation <- gsub("Black Dot", "Black Dot", data$variation)
data$variation <- gsub("White Plus", "White Plus", data$variation)

head(data$variation)
```

```
## [1] "Black Dot" "Black Dot" "Black Dot" "Black Dot" "Black Dot"
```

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

```
library(readxl) # Load the readxl package

# Import the data and assign it to the variable 'alexa_file'
alexa_file <- read_excel("alexa-file.xlsx")
```

```r
# Rename the variations using gsub()
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)

print(head(alexa_file))
```

```
## # A tibble: 5 x 5
##   rating date       variation verified_reviews                          feedback
##    <dbl> <chr>      <chr>     <chr>                                        <dbl>
## 1      5 2018-07-30 Black Dot It works great!!                                 1
## 2      5 2018-07-30 Black Dot PHENOMENAL                                       1
## 3      4 2018-07-30 Black Dot I used it to control my smart home devic~        1
## 4      5 2018-07-30 Black Dot Very convenient                                  1
## 5      3 2018-07-30 Black Dot Amazing product!                                 1
```

#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?

```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
```

```r
variations_count <- alexa_file %>%
  count(variation)

save(variations_count, file = "variations.RData")
print(variations_count)
```

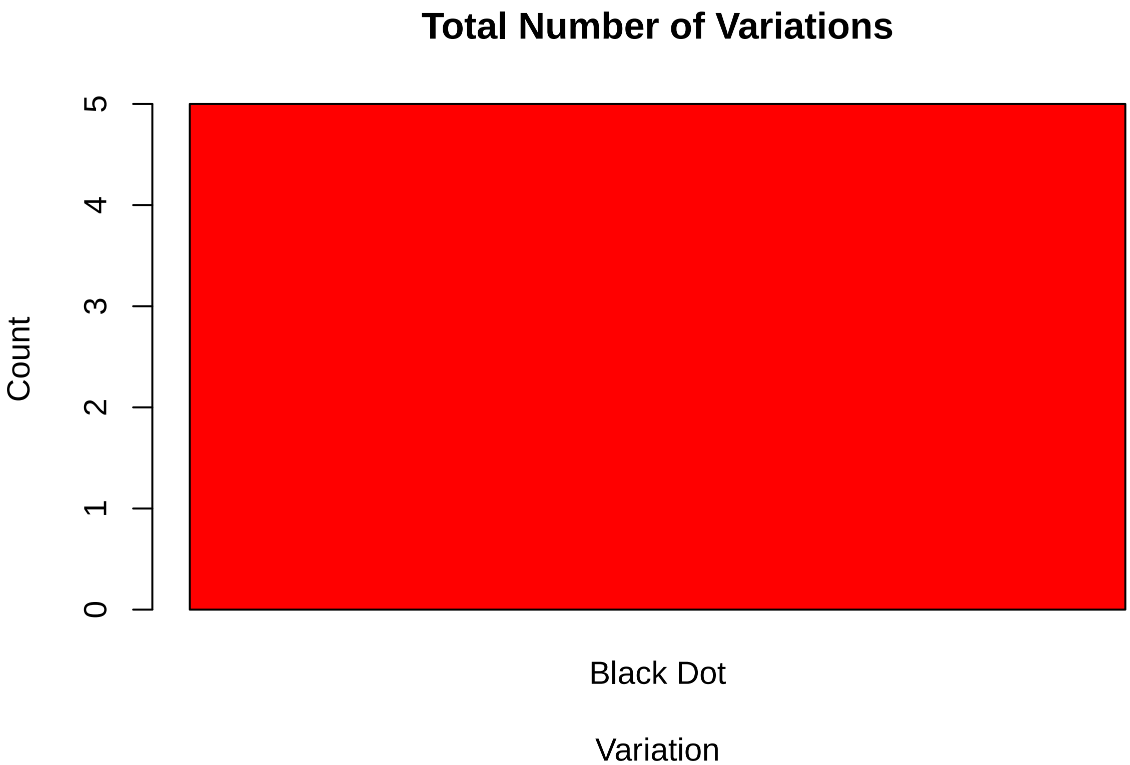
```
## # A tibble: 1 x 2
##   variation     n
##   <chr>     <int>
## 1 Black Dot     5
```

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

```r
load("variations.RData")

barplot(variations_count$n, names.arg = variations_count$variation,
        col = rainbow(nrow(variations_count)),
        main = "Total Number of Variations",
        xlab = "Variation",
```

```r
        ylab = "Count")
```



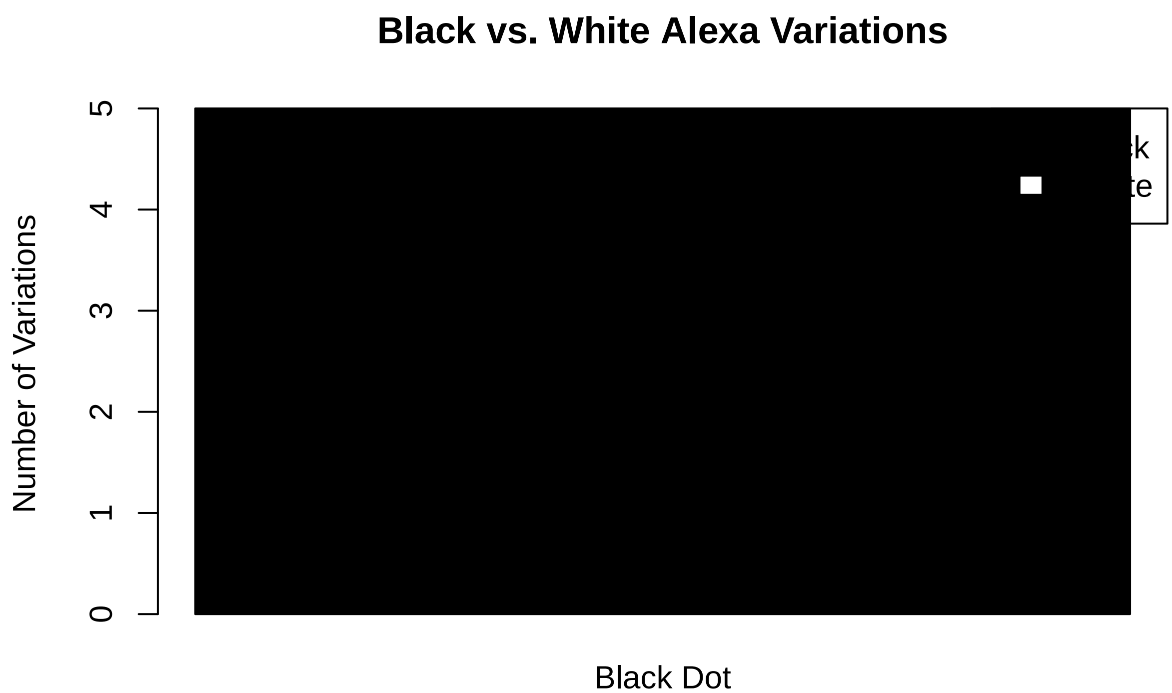
#7d. Create a barplot() for the black and white variations. Plot it in 1 frame, side by side. Complete the details of the chart.

```r
# 7d. Create a barplot() for the black and white variations. Plot it in 1 frame, side by side. Complete
load("variations.RData") # Load the variations_count object

# Separate black and white variations
black_variations <- variations_count[grep("Black", variations_count$variation), ]
white_variations <- variations_count[grep("White", variations_count$variation), ]

# Extract the variation names as character vectors
black_names <- black_variations$variation
white_names <- white_variations$variation

# Plot side by side
barplot(cbind(black_variations$n, white_variations$n),
        beside = TRUE,
        names.arg = c(black_names, white_names), # Use the extracted names
        main = "Black vs. White Alexa Variations",
        ylab = "Number of Variations",
        col = c("black", "white"),
        legend.text = c("Black", "White"),
        args.legend = list(x = "topright"))
```



Black vs. White Alexa Variations
Number of Variations
0
1
2
3
4
5
Black Dot