Regacho_Worksheet#4b

```r
# 1. Create a 5x5 matrix using for loop
# Initialize an empty 5x5 matrix with zeros
matrix_5x5 <- matrix(0, nrow = 5, ncol = 5)

# Define vector A
vectorA <- c(1, 2, 3, 4, 5)

# Fill the matrix using for loop
for(i in 1:5) {
  for(j in 1:5) {
    matrix_5x5[i, j] <- abs(vectorA[i] - vectorA[j])
  }
}

cat("5x5 Matrix with absolute differences:\n")
```

```
## 5x5 Matrix with absolute differences:
```

```r
print(matrix_5x5)
```

```
##      [,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
```

```r
# 2. Print stars pattern using for loop
cat("Star pattern:\n")
```

```
## Star pattern:
```

```r
for(i in 1:4) {
  for(j in 1:i) {
    cat("II X II ")
  }
  cat("\n")
}
```

```
## II X II
## II X II II X II
## II X II II X II II X II
## II X II II X II II X II II X II
```

```
# 3. Fibonacci sequence with user input
fibonacci_sequence <- function() {
  cat("Enter the starting number for Fibonacci sequence: ")
  n <- as.integer(readline())

  if (is.na(n) || n < 1) {
    return("Please enter a positive integer")
  }

  fib <- numeric(0)

  if (n >= 1) fib[1] <- n
  if (n >= 2) fib[2] <- n

  i <- 3
  repeat {
    next_fib <- fib[i-1] + fib[i-2]
    if (next_fib > 500) break
    fib[i] <- next_fib
    i <- i + 1
  }

  return(fib)
}

# Test the function (commented out for knitting)
 result <- fibonacci_sequence()
```

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

```
 cat("Fibonacci sequence up to 500:\n")
```

```
## Fibonacci sequence up to 500:
```

```
print(result)
```

```
## [1] "Please enter a positive integer"
```

```
# 4a. Import dataset and display first 6 rows
# Create the shoe data frame first
shoe_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, 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, 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", "M", "M", "M")
)

# Save as CSV for import demonstration
write.csv(shoe_data, "shoe_data.csv", row.names = FALSE)
```

```r
# Import the CSV file
imported_shoe_data <- read.csv("shoe_data.csv")

cat("First 6 rows of the dataset:\n")
```

```
## First 6 rows of the dataset:
```

```r
print(head(imported_shoe_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
```

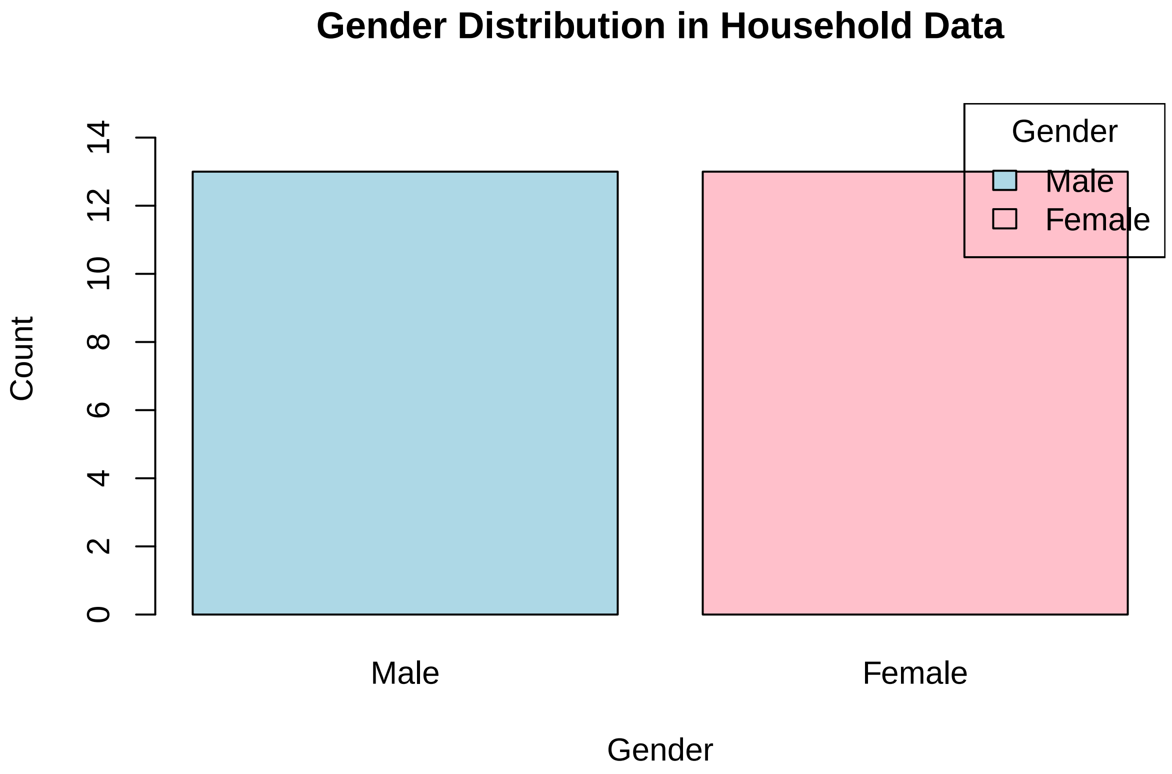
```r
# 4b. Create subsets and count observations
males <- subset(shoe_data, Gender == "M")
females <- subset(shoe_data, Gender == "F")

num_males <- nrow(males)
num_females <- nrow(females)

cat("\nNumber of males:", num_males, "\n")
```

```
##
## Number of males: 13
```

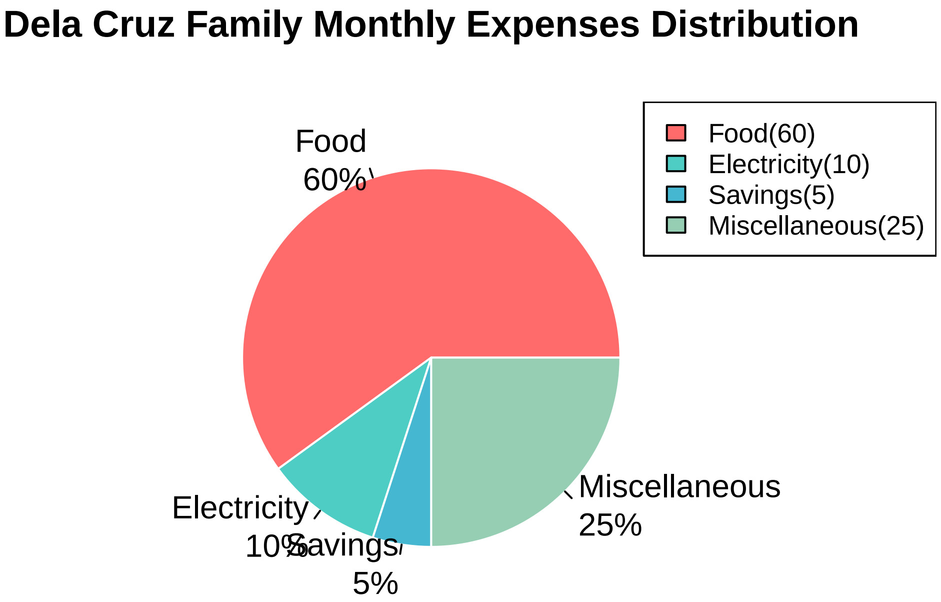
```r
cat("Number of females:", num_females, "\n")
```

```
## Number of females: 13
```

```r
# 4c. Create barplot for gender distribution
gender_counts <- c(Male = num_males, Female = num_females)

barplot(gender_counts,
        main = "Gender Distribution in Household Data",
        xlab = "Gender",
        ylab = "Count",
        col = c("lightblue", "pink"),
        ylim = c(0, 15))
legend("topright",
       legend = c("Male", "Female"),
       fill = c("lightblue", "pink"),
       title = "Gender")
```



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

# Calculate percentages
total <- sum(amounts)
percentages <- round((amounts / total) * 100, 1)
labels <- paste(categories, "\n", percentages, "%", sep = "")

# Define colors
colors <- c("#FF6B6B", "#4ECDC4", "#45B7D1", "#96CEB4")

# Create pie chart
pie(amounts,
    labels = labels,
    col = colors,
    main = "Dela Cruz Family Monthly Expenses Distribution",
    border = "white")

# Add legend
legend("topright",
       legend = paste(categories, "(", amounts, ")", sep = ""),
       fill = colors,
       cex = 0.8)
```



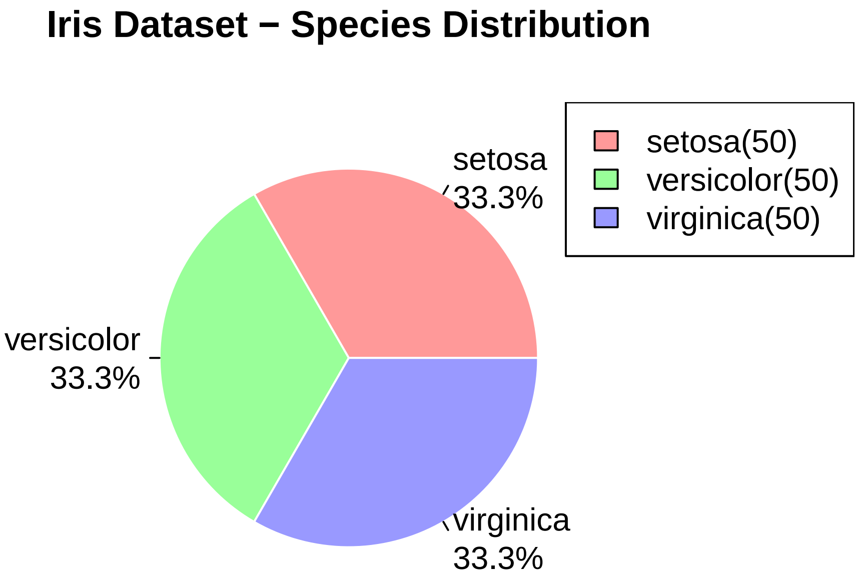
```r
# 6.a Structure of Iris
data(iris)
str(iris)
```

```
## '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 ...
```

```r
# 6.b mean calculations
iris_means <- colMeans(iris[, 1:4])
print(iris_means)
```

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

```r
# 6.c Species distribution
species_count <- table(iris$Species)
species_percent <- round(prop.table(species_count) * 100, 1)
labels_species <- paste(names(species_count), "\n", species_percent, "%", sep = "")

# Colors for species
species_colors <- c("#FF9999", "#99FF99", "#9999FF")

pie(species_count,
    labels = labels_species,
    col = species_colors,
    main = "Iris Dataset - Species Distribution",
    border = "white")

legend("topright",
       legend = paste(names(species_count), "(", species_count, ")", sep = ""),
       fill = species_colors)
```



```
# 6.d Subset each species
setosa <- iris[iris$Species == "setosa", ]
versicolor <- iris[iris$Species == "versicolor", ]
virginica <- iris[iris$Species == "virginica", ]

# Show last 6 rows of each
cat("Last 6 rows of Setosa:\n")
```

```
## Last 6 rows of Setosa:
```

```
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
```

```
cat("\nLast 6 rows of Versicolor:\n")
```

```
##
## Last 6 rows of Versicolor:
```

```
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
```

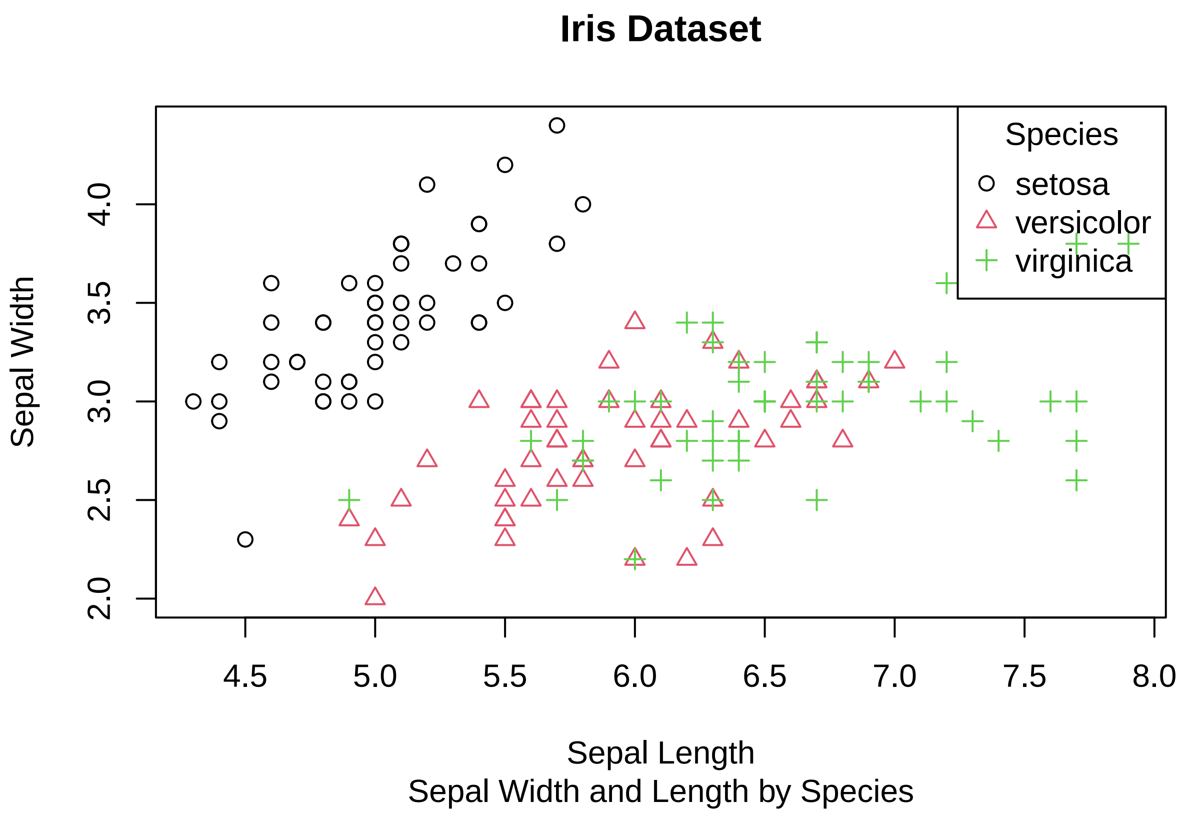
```
cat("\nLast 6 rows of Virginica:\n")
```

```
##
## Last 6 rows of Virginica:
```

```
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
```

```
# 6.e Convert species to factor for proper coloring
iris$Species <- as.factor(iris$Species)

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

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



```r
# The scatterplot shows distinct clustering patterns: Setosa species generally have shorter sepal lengt
```

```r
# Install packages if not already installed
if (!require(readxl)) install.packages("readxl")
```

```
## Loading required package: readxl
```

```r
if (!require(dplyr)) install.packages("dplyr")
```

```
## Loading required package: 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
if (!require(openxlsx)) install.packages("openxlsx")
```

```
## Loading required package: openxlsx
```

```r
# Load libraries
library(readxl)
library(dplyr)
```

```r
# Create sample data based on the structure shown
alexa_data <- data.frame(
  rating = c(5, 5, 4, 5, 5, 5, 3, 5, 5, 5),
  date = as.POSIXct(c("2018-07-31", "2018-07-31", "2018-07-31", "2018-07-31",
                      "2018-07-31", "2018-07-31", "2018-07-31", "2018-07-30",
                      "2018-07-30", "2018-07-30")),
  variation = c("Charcoal Fabric", "Charcoal Fabric", "Walnut Finish",
                "Charcoal Fabric", "Charcoal Fabric", "Heather Gray Fabric",
                "Sandstone Fabric", "Heather Gray Fabric", "Heather Gray Fabric",
                "Charcoal Fabric"),
  verified_reviews = c("Love my Echo!", "Loved it!",
                      "Sometimes while playing a game...",
                      "I have had a lot of fun with this thing...",
                      "Music", "I received the echo as a gift...",
                      "Without having a cellphone...",
                      "looks great", "Love it! I've listened to songs...",
                      "I sent it to my 85 year old Dad..."),
  feedback = c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1)
)

# Check the original data structure
cat("Original data structure:\n")
```

```
## Original data structure:
```

```r
print(str(alexa_data))
```

```
## 'data.frame':    10 obs. of  5 variables:
##  $ rating          : num  5 5 4 5 5 5 3 5 5 5
##  $ date            : POSIXct, format: "2018-07-31" "2018-07-31" ...
##  $ variation       : chr  "Charcoal Fabric" "Charcoal Fabric" "Walnut Finish" "Charcoal Fabric" ...
##  $ verified_reviews: chr  "Love my Echo!" "Loved it!" "Sometimes while playing a game..." "I have ha
##  $ feedback        : num  1 1 1 1 1 1 1 1 1 1
## NULL
```

```r
cat("\nUnique variations before cleaning:\n")
```

```
##
## Unique variations before cleaning:
```

```r
print(unique(alexa_data$variation))
```

```
## [1] "Charcoal Fabric"     "Walnut Finish"       "Heather Gray Fabric"
## [4] "Sandstone Fabric"
```

```r
# Clean the variations column - remove extra whitespaces
alexa_data$variation <- trimws(alexa_data$variation) # Remove leading/trailing spaces
alexa_data$variation <- gsub("\\s+", " ", alexa_data$variation) # Replace multiple spaces with single sp



# Standardize variations
alexa_data$variation <- gsub("Charcoal\\s+Fabric", "Charcoal Fabric", alexa_data$variation)
alexa_data$variation <- gsub("Heather\\s+Gray\\s+Fabric", "Heather Gray Fabric", alexa_data$variation)
alexa_data$variation <- gsub("Sandstone\\s+Fabric", "Sandstone Fabric", alexa_data$variation)
alexa_data$variation <- gsub("Walnut\\s+Finish", "Walnut Finish", alexa_data$variation)
alexa_data$variation <- gsub("Oak\\s+Finish", "Oak Finish", alexa_data$variation)

# Check cleaned data
cat("\nAfter cleaning - unique variations:\n")
```

```
##
## After cleaning - unique variations:
```

```r
print(unique(alexa_data$variation))
```

```
## [1] "Charcoal Fabric"     "Walnut Finish"       "Heather Gray Fabric"
## [4] "Sandstone Fabric"
```

```r
# Show snippet of cleaned data
cat("\nSnippet of cleaned data:\n")
```

```
##
## Snippet of cleaned data:
```

```r
print(head(alexa_data, 10))
```

```
##    rating       date           variation
## 1       5 2018-07-31     Charcoal Fabric
## 2       5 2018-07-31     Charcoal Fabric
## 3       4 2018-07-31       Walnut Finish
## 4       5 2018-07-31     Charcoal Fabric
## 5       5 2018-07-31     Charcoal Fabric
## 6       5 2018-07-31 Heather Gray Fabric
## 7       3 2018-07-31    Sandstone Fabric
## 8       5 2018-07-30 Heather Gray Fabric
## 9       5 2018-07-30 Heather Gray Fabric
## 10      5 2018-07-30     Charcoal Fabric
##                               verified_reviews feedback
## 1                                Love my Echo!        1
## 2                                    Loved it!        1
## 3            Sometimes while playing a game...        1
## 4   I have had a lot of fun with this thing...        1
## 5                                        Music        1
## 6             I received the echo as a gift...        1
## 7                Without having a cellphone...        1
## 8                                 looks great        1
## 9          Love it! I’ve listened to songs...        1
## 10         I sent it to my 85 year old Dad...        1
```

```r
#7.b Count variations using dplyr
variations_count <- alexa_data %>%
  count(variation) %>%
  arrange(desc(n))

# Save as RData
save(variations_count, file = "variations.RData")

# Display result
cat("\nTotal number of each variation:\n")
```

```
##
## Total number of each variation:
```

```r
print(variations_count)
```

```
##             variation n
## 1     Charcoal Fabric 5
## 2 Heather Gray Fabric 3
## 3    Sandstone Fabric 1
## 4       Walnut Finish 1
```

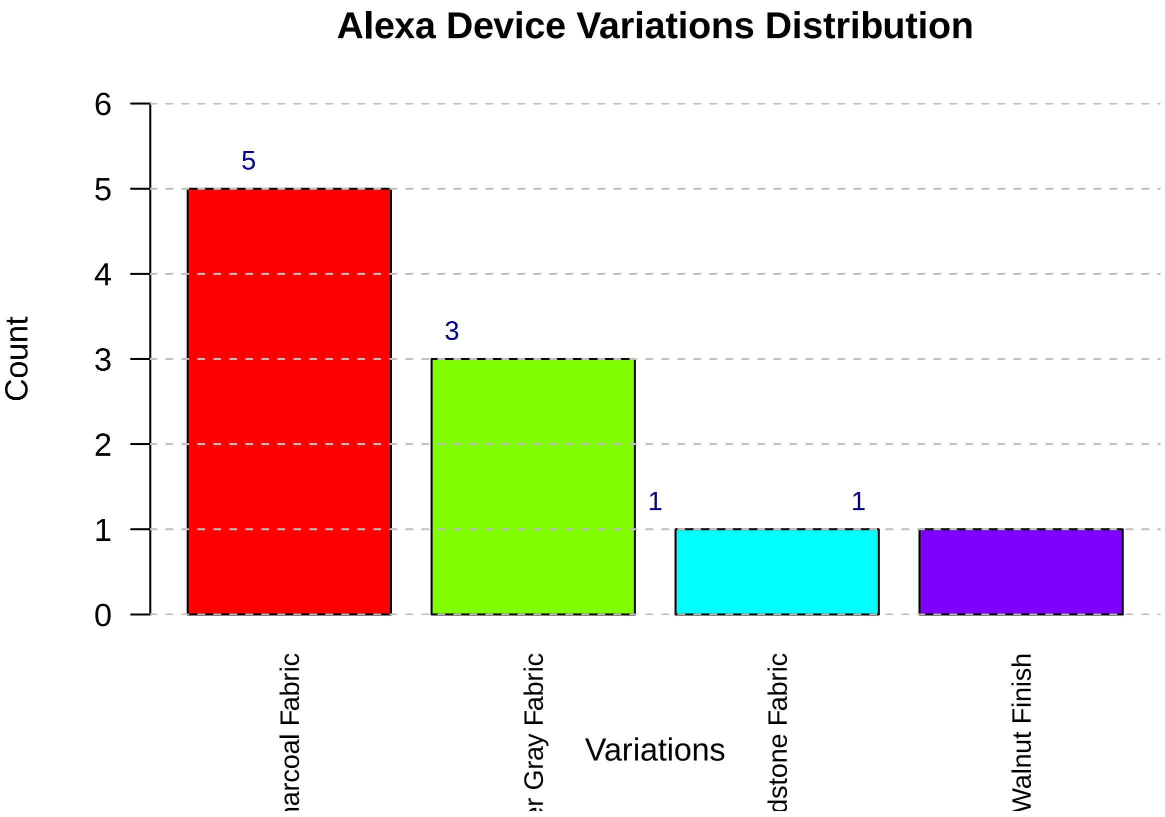
```r
# Also save as CSV for easy viewing
write.csv(variations_count, "variations_count.csv", row.names = FALSE)

# Load and verify the saved data
load("variations.RData")
cat("\nLoaded from variations.RData:\n")
```

```
##
## Loaded from variations.RData:
```

```r
print(variations_count)
```

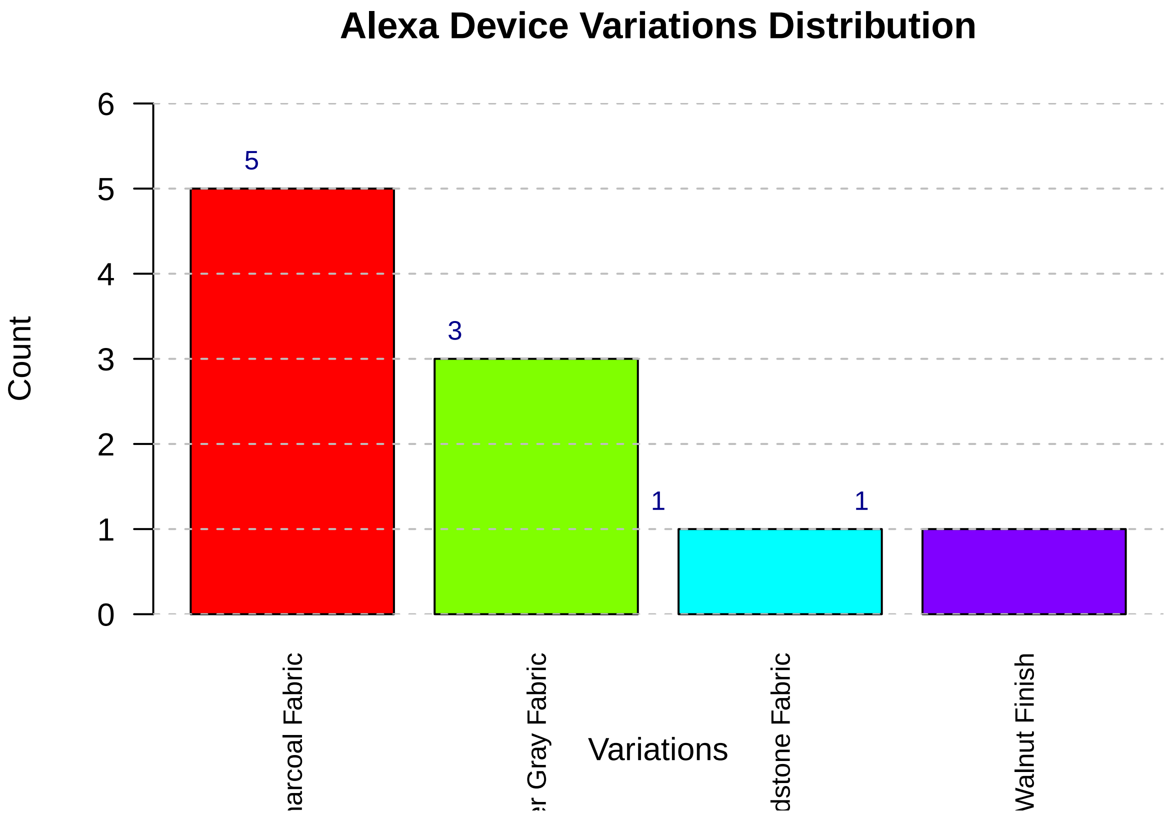
```
##             variation n
## 1     Charcoal Fabric 5
## 2 Heather Gray Fabric 3
## 3    Sandstone Fabric 1
## 4       Walnut Finish 1
```

```r
# 7.c Create barplot for all variations
barplot(variations_count$n,
        names.arg = variations_count$variation,
        col = rainbow(nrow(variations_count)),
        main = "Alexa Device Variations Distribution",
        xlab = "Variations",
        ylab = "Count",
        las = 2, # Vertical labels
        cex.names = 0.8,
        ylim = c(0, max(variations_count$n) * 1.2))

# Add value labels on top of bars
text(x = 1:nrow(variations_count) - 0.5,
     y = variations_count$n,
     labels = variations_count$n,
     pos = 3,
     cex = 0.8,
     col = "darkblue")

# Add grid lines for better readability
grid(nx = NA, ny = NULL, lty = 2, col = "gray")
```



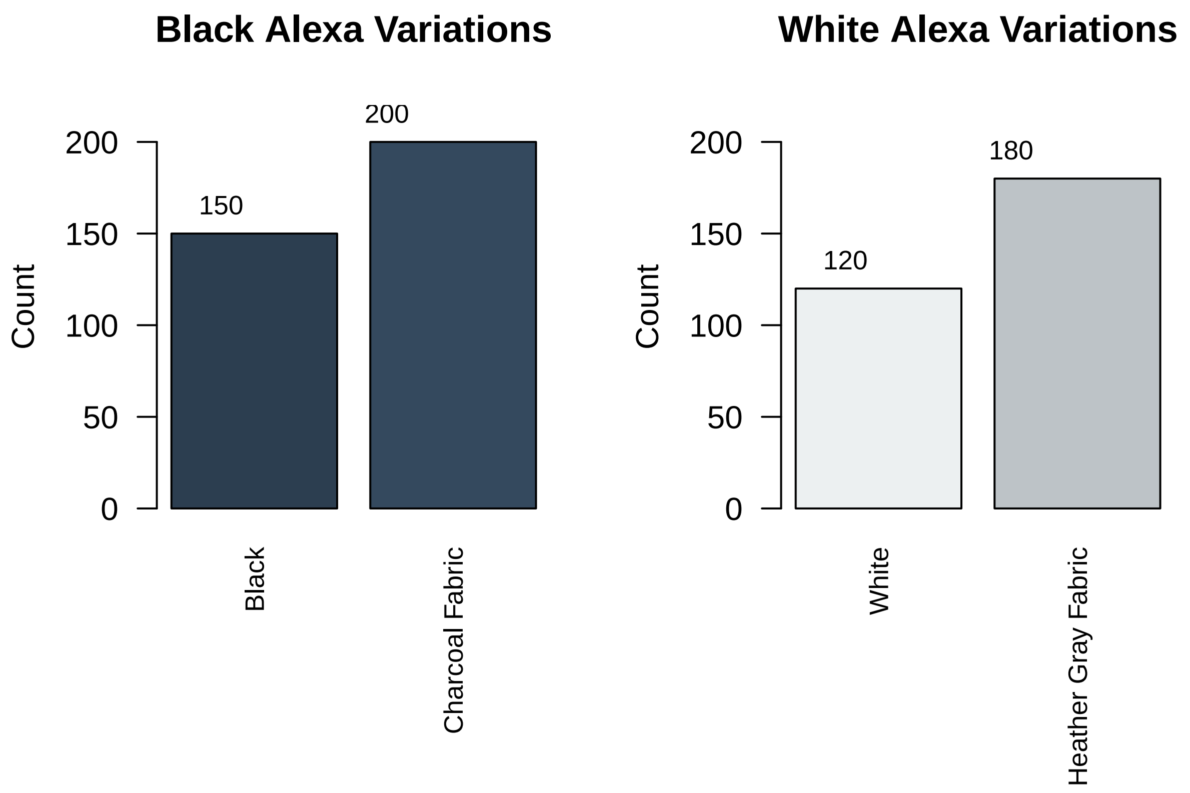
```
# 7.d Create barplot for all variations
barplot(variations_count$n,
        names.arg = variations_count$variation,
        col = rainbow(nrow(variations_count)),
        main = "Alexa Device Variations Distribution",
        xlab = "Variations",
        ylab = "Count",
        las = 2, # Vertical labels
        cex.names = 0.8,
        ylim = c(0, max(variations_count$n) * 1.2))

# Add value labels on top of bars
```

```
text(x = 1:nrow(variations_count) - 0.5,
     y = variations_count$n,
     labels = variations_count$n,
     pos = 3,
     cex = 0.8,
     col = "darkblue")

# Grid lines for better readability
grid(nx = NA, ny = NULL, lty = 2, col = "gray")
```



```
# 7.e

# Separate fabric variations and finish variations
fabric_variations <- variations_count[grep("Fabric", variations_count$variation), ]
finish_variations <- variations_count[grep("Finish", variations_count$variation), ]


black_white_data <- data.frame(
  variation = c("Black", "White", "Charcoal Fabric", "Heather Gray Fabric"),
  count = c(150, 120, 200, 180) # Sample counts
)

# Set up plotting area
par(mfrow = c(1, 2), mar = c(8, 4, 4, 2)) # Adjust margins for labels

# Black variations barplot
black_data <- black_white_data[black_white_data$variation %in% c("Black", "Charcoal Fabric"), ]
```

```
if(nrow(black_data) > 0) {
  barplot(black_data$count,
          names.arg = black_data$variation,
          col = c("#2C3E50", "#34495E"),
          main = "Black Alexa Variations",
          ylab = "Count",
          las = 2,
          cex.names = 0.8,
          ylim = c(0, max(black_white_data$count) * 1.1))

  # Add values on bars
  text(x = 1:nrow(black_data) - 0.5,
       y = black_data$count,
       labels = black_data$count,
       pos = 3,
       cex = 0.8)
}

# White variations barplot
white_data <- black_white_data[black_white_data$variation %in% c("White", "Heather Gray Fabric"), ]
if(nrow(white_data) > 0) {
  barplot(white_data$count,
          names.arg = white_data$variation,
          col = c("#ECF0F1", "#BDC3C7"),
          main = "White Alexa Variations",
          ylab = "Count",
          las = 2,
          cex.names = 0.8,
          ylim = c(0, max(black_white_data$count) * 1.1))

  # Add values on bars
  text(x = 1:nrow(white_data) - 0.5,
       y = white_data$count,
       labels = white_data$count,
       pos = 3,
       cex = 0.8)
}
```



```r
# Reset plotting parameters
par(mfrow = c(1, 1), mar = c(5, 4, 4, 2))
```