RWorksheet_parita#4b

Daven Parita

2025-12-11

Using Loop Function

1.

```
ROWS <- 5
COLS <- 5

matrix_A <- matrix(0, nrow = ROWS, ncol = COLS)

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

for (i in 1:ROWS) {
  for (j in 1:COLS) {

    row_index_0based <- i - 1
    col_index_0based <- j - 1

    matrix_A[i, j] <- abs(row_index_0based - col_index_0based)
  }
}

print(matrix_A)
```

```
##      [,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.

```
NUM_ROWS <- 5

for (i in 1:NUM_ROWS) {

  stars <- paste0(rep("*", i))
  cat(stars, "\n")
}
```

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

3.

```
start_num <- as.integer(1)

a <- 0
b <- 1

fib_sequence <- c()

repeat {
  if (a >= start_num) {
    current <- a
    previous <- b

    a_new <- 0
    b_new <- 1

    repeat {
      temp <- a_new + b_new
      a_new <- b_new
      b_new <- temp

      if (a_new >= start_num) {
        current <- a_new
        previous <- b_new - a_new
        break
      }
    }
    break
  }

  temp <- a + b
  a <- b
  b <- temp
}

repeat {

  if (current > 500) {
    break
  }

  fib_sequence <- c(fib_sequence, current)

  next_num <- previous + current
  previous <- current
  current <- next_num
}

cat("\n--- Output ---\n")
```

```
##
## --- Output ---
```

```
cat("Fibonacci sequence starting from the first number >= ", start_num, " and up to 500:\n")
```

```
## Fibonacci sequence starting from the first number >=  1  and up to 500:
```

```
print(fib_sequence)
```

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

Using Basic Graphics

4.

b.

```
shoes <- read.csv("Household.csv",
                  sep = ",",
                  header = TRUE)
head(shoes)
```

```
##   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.

```
shoes_male <- subset(shoes, Gender == "M")
shoes_female <- subset(shoes, Gender == "F")

str(shoes_male)
```

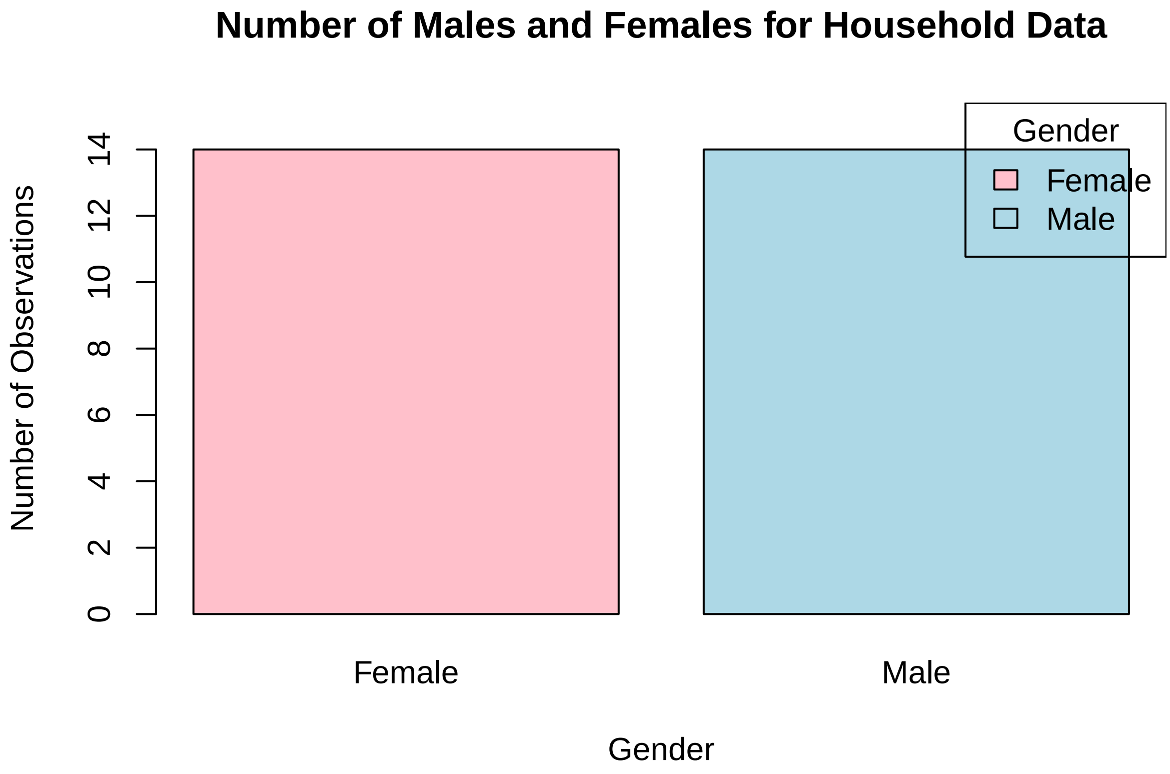
```
## ’data.frame':    14 obs. of  4 variables:
##  $ X        : int  5 9 11 13 14 15 16 19 22 23 ...
##  $ Shoe.size: num  10.5 13 10.5 12 10.5 13 11.5 10 8.5 10.5 ...
##  $ Height   : num  70 72 74.5 71 71 77 72 72 67 73 ...
##  $ Gender   : chr  "M" "M" "M" "M" ...
```

```
str(shoes_female)
```

```
## ’data.frame':    14 obs. of  4 variables:
##  $ X        : int  1 2 3 4 6 7 8 10 12 17 ...
##  $ Shoe.size: num  6.5 9 8.5 8.5 7 9.5 9 7.5 8.5 8.5 ...
##  $ Height   : num  66 68 64.5 65 64 70 71 64 67 59 ...
##  $ Gender   : chr  "F" "F" "F" "F" ...
```

c.

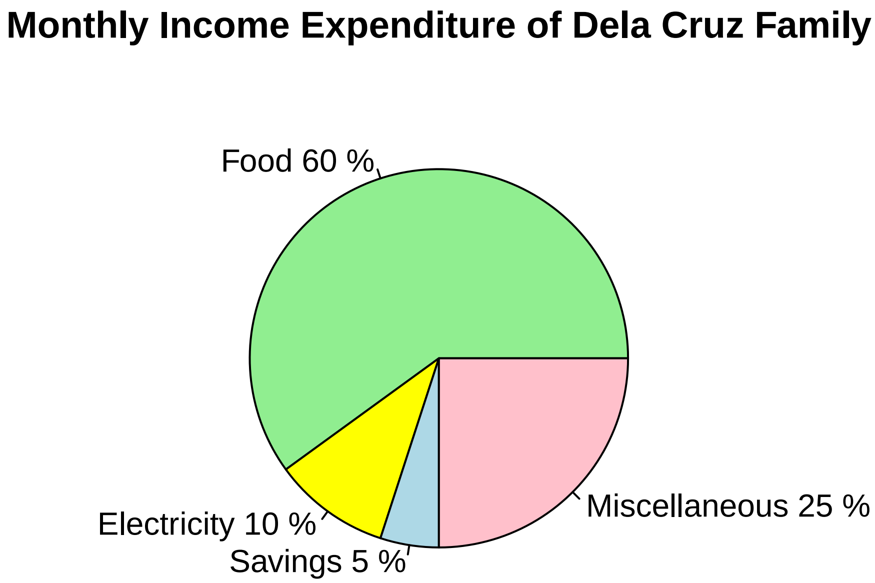
```
household_data <- read.csv("Household.csv")

gender_counts <- table(household_data$Gender)

names(gender_counts) <- c("Female", "Male")

barplot(
  gender_counts,
  main = "Number of Males and Females for Household Data",
  xlab = "Gender",
  ylab = "Number of Observations",
  col = c("pink", "lightblue"),
  ylim = c(0, max(gender_counts) * 1.1)
)

legend("topright",
       legend = names(gender_counts),
       fill = c("pink", "lightblue"),
       title = "Gender")
```



5.

a.

```
expenditures <- c(60, 10, 5, 25)
categories <- c("Food", "Electricity", "Savings", "Miscellaneous")

colors <- c("lightgreen", "yellow", "lightblue", "pink")

pct <- round(expenditures / sum(expenditures) * 100)

pie_labels <- paste(categories, pct, "%", sep=" ")

pie(expenditures,
    labels = pie_labels,
    main = "Monthly Income Expenditure of Dela Cruz Family",
    col = colors)
```



6.

```r
data("iris")
```

a.

```r
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 ...
```

The output of getting the structure of iris data shows the class, number of observations, number of variables, names of variables, and the first few observations of each variables.

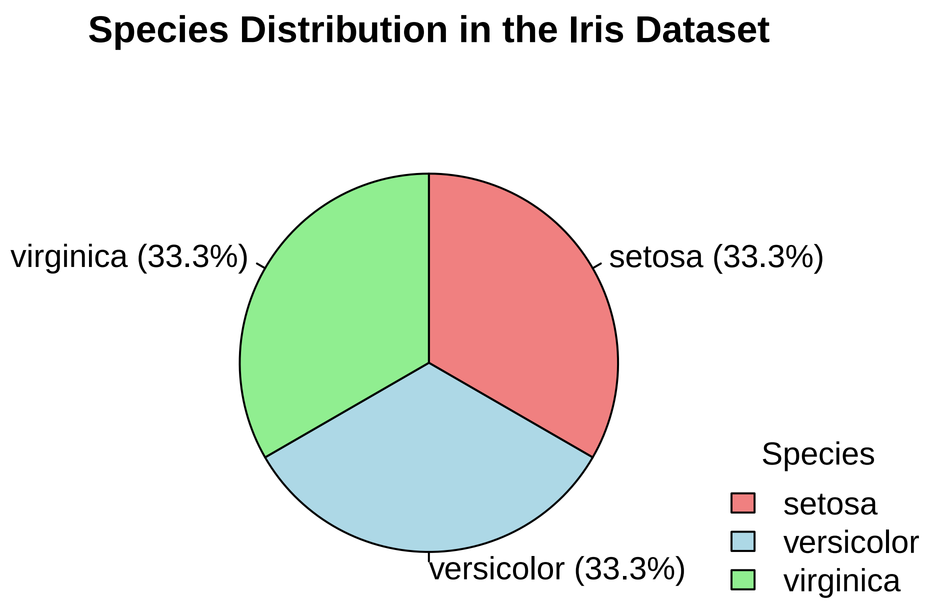
b.

```r
sepal_length_mean <- mean(iris$Sepal.Length)
sepal_width_mean <- mean(iris$Sepal.Width)
petal_length_mean <- mean(iris$Petal.Length)
petal_width_mean <- mean(iris$Petal.Width)

irisMean_df <- data.frame(sepal_length_mean,
                          sepal_width_mean,
                          petal_length_mean,
                          petal_width_mean)

print(irisMean_df)
```

```
##   sepal_length_mean sepal_width_mean petal_length_mean petal_width_mean
## 1          5.843333         3.057333             3.758         1.199333
```

c.

```r
data(iris)

species_counts <- table(iris$Species)

total_obs <- sum(species_counts)
percentages <- round(species_counts / total_obs * 100, 1)
pie_labels <- paste(names(species_counts), " (", percentages, "%)", sep="")

pie_colors <- c("lightcoral", "lightblue", "lightgreen")

pie(species_counts,
    labels = pie_labels,
    main = "Species Distribution in the Iris Dataset",
    col = pie_colors,
    clockwise = TRUE)

legend("bottomright",
       legend = names(species_counts),
       fill = pie_colors,
       title = "Species",
       bty = "n")
```



d.

```
data(iris)

setosa_df <- subset(iris, Species == "setosa")

versicolor_df <- subset(iris, Species == "versicolor")

virginica_df <- subset(iris, Species == "virginica")

print(tail(setosa_df, 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
```

```
print(tail(versicolor_df, 6))
```

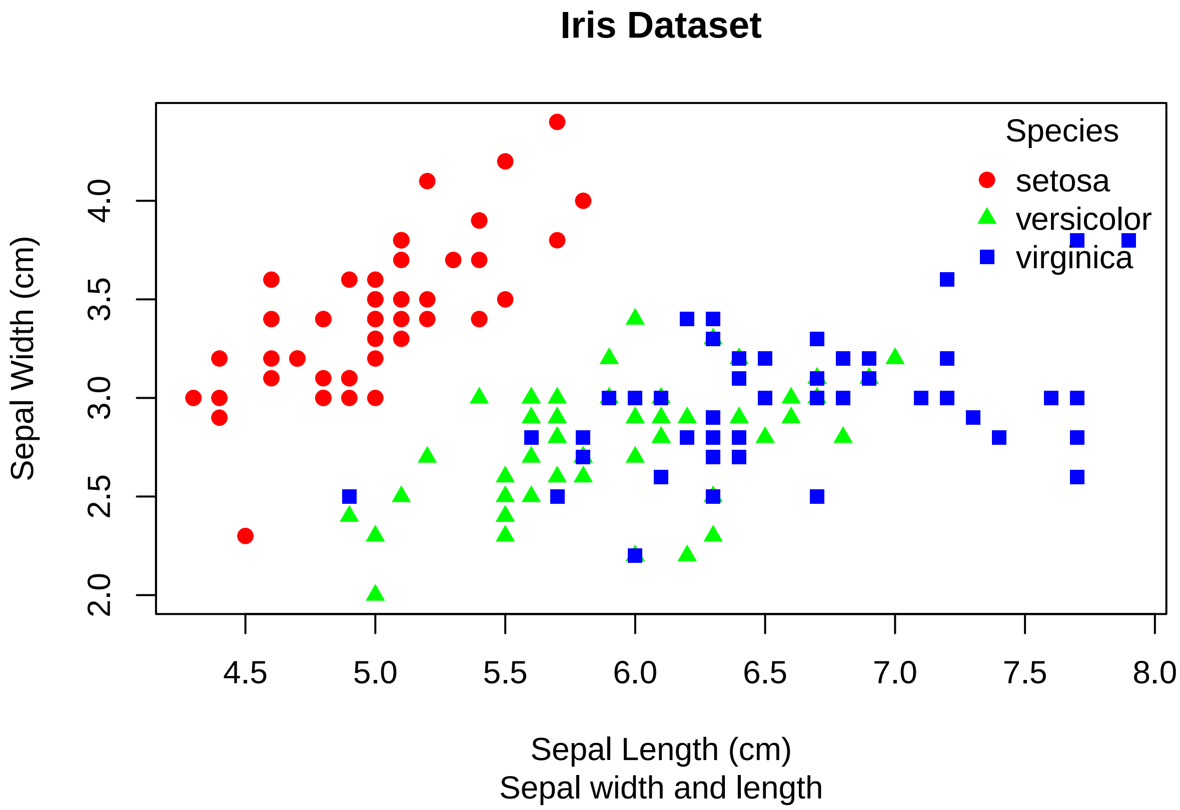
```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 95           5.6         2.7          4.2         1.3 versicolor
## 96           5.7         3.0          4.2         1.2 versicolor
## 97           5.7         2.9          4.2         1.3 versicolor
## 98           6.2     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

```
print(tail(virginica_df, 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.

```
color_map <- c("setosa" = "red", "versicolor" = "green", "virginica" = "blue")
iris$Color <- color_map[iris$Species]

pch_map <- c("setosa" = 19, "versicolor" = 17, "virginica" = 15)
iris$Pch <- pch_map[iris$Species]

plot(x = iris$Sepal.Length, y = iris$Sepal.Width,

     main = "Iris Dataset",

     sub = "Sepal width and length",

     xlab = "Sepal Length (cm)",
     ylab = "Sepal Width (cm)",

     col = as.character(iris$Color),
     pch = as.numeric(iris$Pch))

legend("topright",
       legend = unique(iris$Species),
       col = unique(as.character(iris$Color)),
       pch = unique(as.numeric(iris$Pch)),
       title = "Species",
       bty = "n")
```



f.

The result is a scatter plot of Sepal Length vs. Sepal Width of the three different species of flowers from the iris dataset. setosa is represented by a red circles in the scatterplot, versicolor by a green triangles, and virginica by blue squares.

Basic Cleaning and Transformation of Objects

7.

a.

```
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
```

```
library(stringr)
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: ’gridExtra’
```

```
## The following object is masked from ’package:dplyr’:
##
##     combine
```

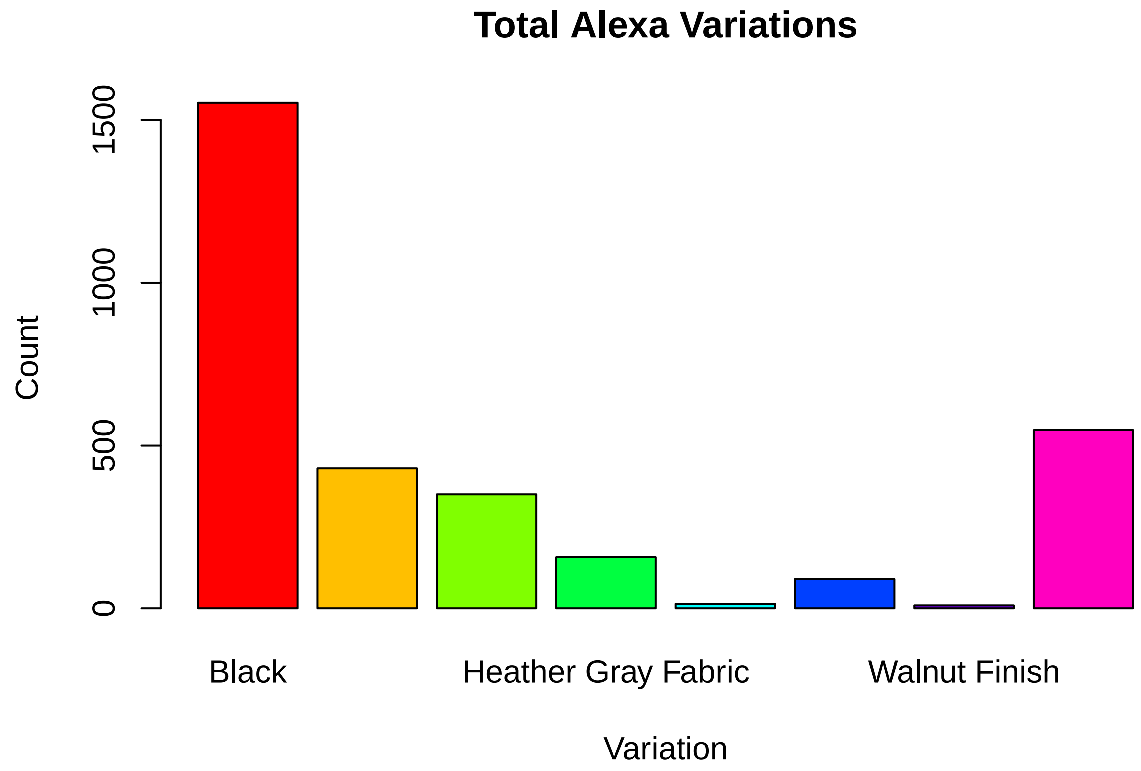
```
library(readxl)

alexa_df <- read_excel("alexa_file.xlsx")
alexa_df$variation <- str_squish(alexa_df$variation)
alexa_df$variation <- gsub("^White.*", "White", alexa_df$variation)
alexa_df$variation <- gsub("^Black.*", "Black", alexa_df$variation)
```

b.

```
variations <- alexa_df %>% count(variation)
vars <- variations$n
```

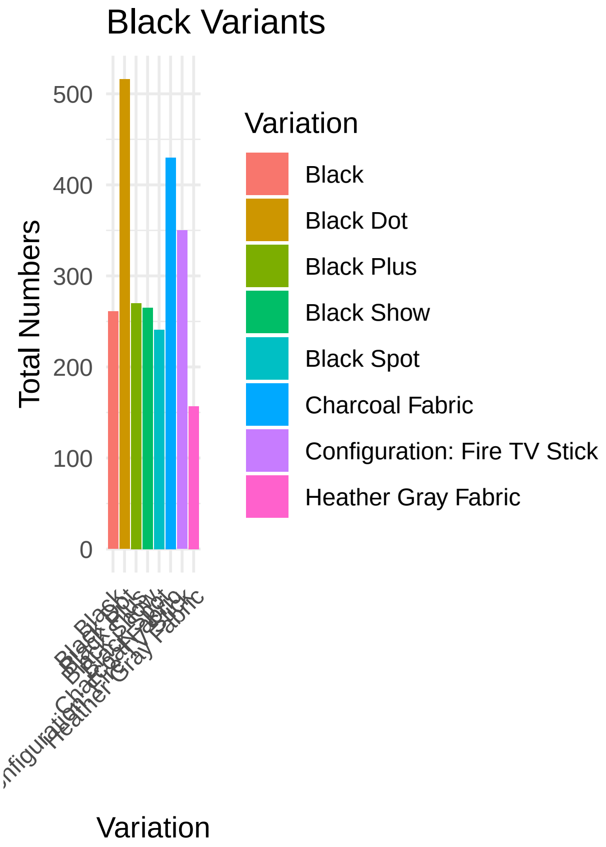
c.

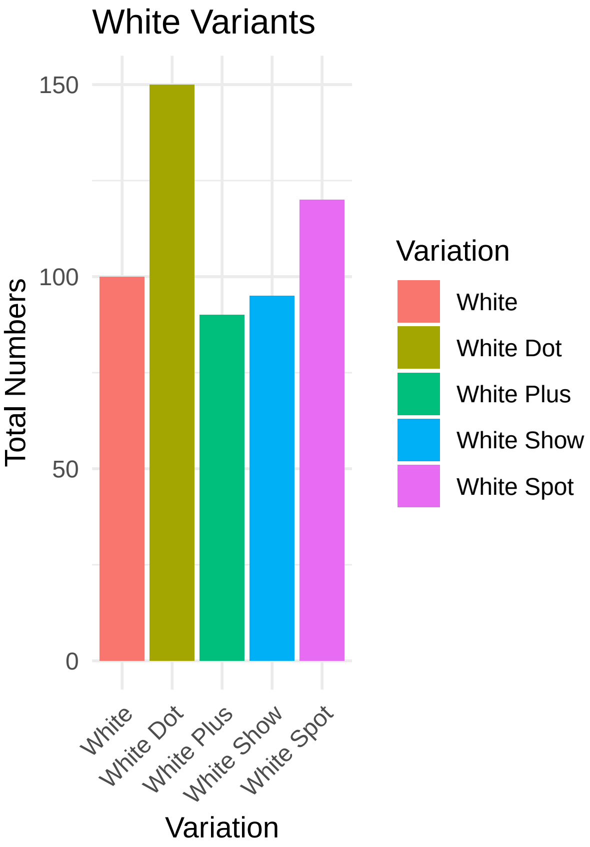
```
names(vars) <- variations$variation
barplot(vars, main="Total Alexa Variations", xlab="Variation", ylab="Count", col=rainbow(length(vars)))
```



d.

```
black_variants <- c(261, 270, 265, 241, 516, 430, 350, 157)
black_labels <- c("Black", "Black Plus", "Black Show", "Black Spot", "Black Dot", "Charcoal Fabric", "Co
white_variants <- c(100, 150, 90, 95, 120)
white_labels <- c("White", "White Dot", "White Plus", "White Show", "White Spot")

black_data <- data.frame(Variation = black_labels, Total = black_variants)
white_data <- data.frame(Variation = white_labels, Total = white_variants)

plot_black <- ggplot(black_data, aes(x = Variation, y = Total, fill = Variation)) + geom_bar(stat = "ide
plot_white <- ggplot(white_data, aes(x = Variation, y = Total, fill = Variation)) + geom_bar(stat = "ide

grid.arrange(plot_black, plot_white, ncol = 2)
```


Black Variants
Total Numbers
500
400
300
200
100
0
Variation
Variation
Black
Black Dot
Black Plus
Black Show
Black Spot
Charcoal Fabric
Configuration: Fire TV Stick
Heather Gray Fabric


White Variants
Total Numbers
150
100
50
0
White
White Dot
White Plus
White Show
White Spot
Variation
Variation
White
White Dot
White Plus
White Show
White Spot