RWorksheet_Animas#4b

Marvin Anthony Y. Animas

2024-10-28

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

vectorA <- c(1, 2, 3, 4, 5)
matrix_result <- matrix(0, nrow = 5, ncol = 5)
for (i in 1:5) {
  for (j in 1:5) {
    matrix_result[i, j] <- abs(vectorA[i] - vectorA[j])
  }
}

print(matrix_result)
```

```
##      [,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(rep("*", i), "\n")
}
```

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

```
#3. Get an input from the user to print the Fibonacci sequence starting from the 1st input up to 500. U

firstnum <- as.integer(readline(prompt = "Enter starting number for fibonacci sequence(up to 500): "))
```

```
## Enter starting number for fibonacci sequence(up to 500):
```

```r
fibonacci <- c(0,1)

repeat {
  next_fib <- sum(tail(fibonacci, 2))
  if (next_fib > 500) {
    break
  }
  fibonacci <- c(fibonacci, next_fib)
}

result <- fibonacci[fibonacci >= firstnum]
cat("Fibonacci sequence from", firstnum, "up to 500:", result, "\n")
```

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

```r
#4. Import the dataset as shown in Figure 1 you have created previously.
#a. What is the R script for importing an excel or a csv file? Display the first 6 rows of the dataset?

Data <- read.csv("Shoe_SHG.csv", header = TRUE, sep = ",")
Data[1:6,]
```

```
##   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
#b. Create a subset for gender(female and male). How many observations are there in Male? How about in
male_sub <- subset(Data, Gender == "M")
female_sub <- subset(Data, Gender == "F")

num_male <- nrow(male_sub)
num_female <- nrow(female_sub)

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

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

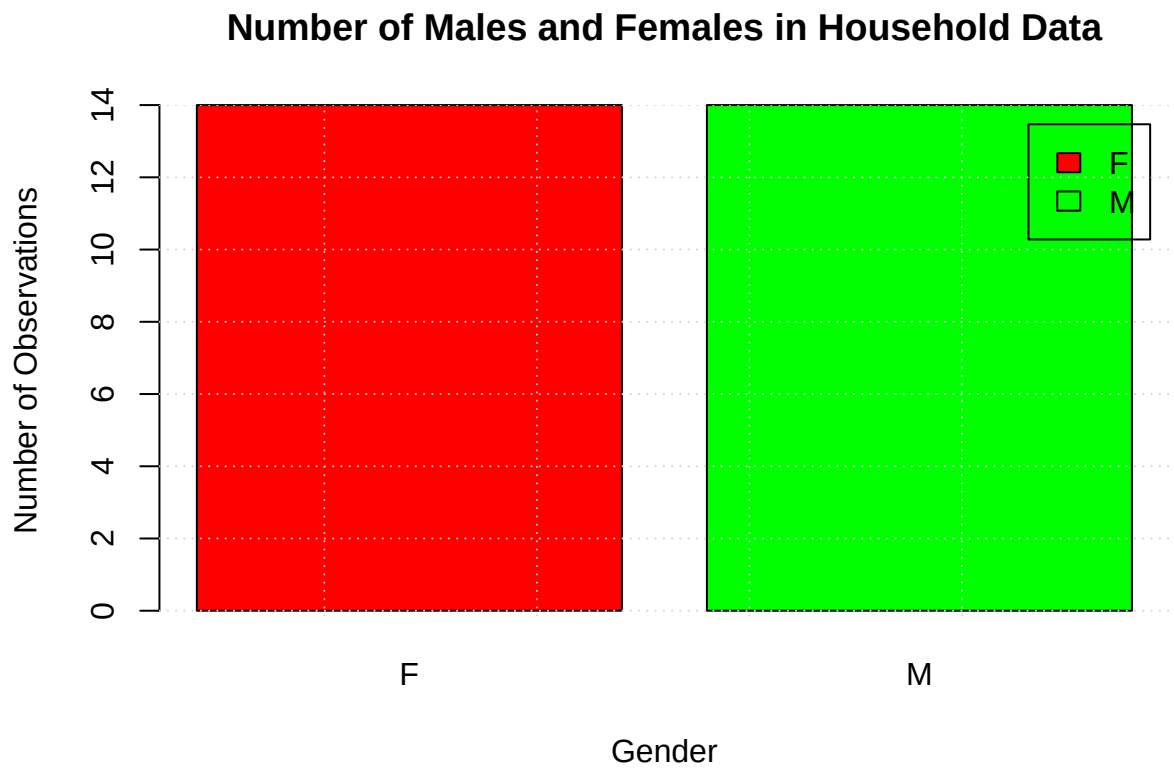
```r
cat("Number of Female observations:", num_female, "\n")
```

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

```r
#c. Create a graph for the number of males and females for Household Data. Use plot(),chart type = barp
gender_count <- table(Data$Gender)
bar_colors <- c("red", "green")
barplot(gender_count,
        main = "Number of Males and Females in Household Data",
        xlab = "Gender",
```

```
        ylab = "Number of Observations",
        col = bar_colors,
        legend = rownames(gender_count),
        beside = TRUE)

grid(nx = NULL, ny = NULL)
```



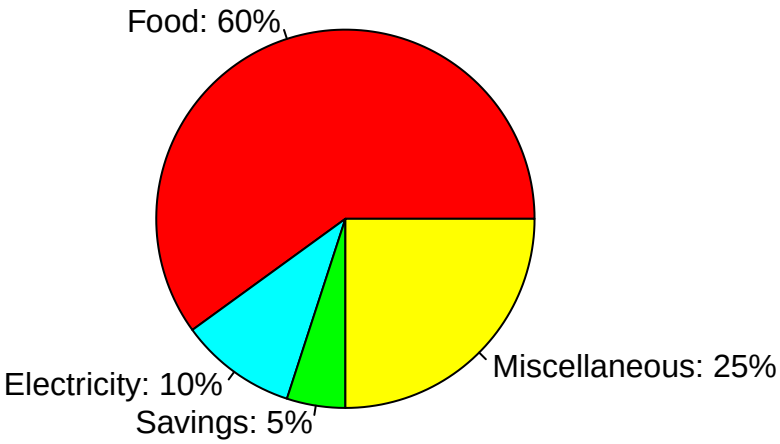

```
#5. The monthly income of Dela Cruz family was spent on the following:

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

percent_labels <- paste0(names(Expenses), ": ", round(Expenses / sum(Expenses) * 100, 1), "%")


colors <- c("red", "cyan", "green", "yellow")

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



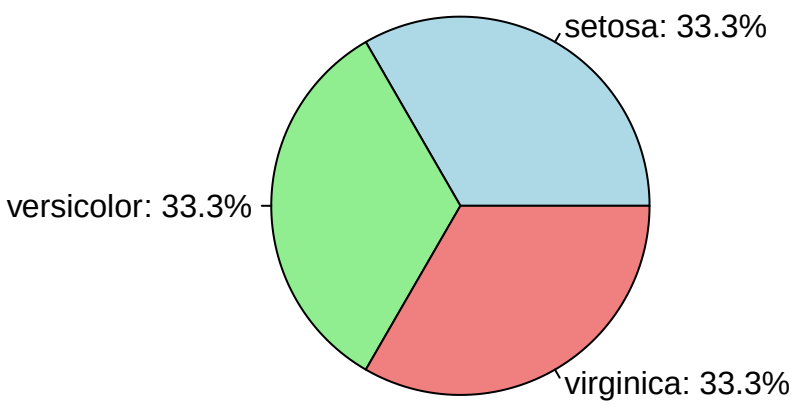

```r
#6.Use the iris dataset.
#a Check for the structure of the dataset using the str() function. Describe what you have seen in the
data(iris)
str(iris)
```

```
## '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
#b Create an R object that will contain the mean of the sepal.length, sepal.width,petal.length,and peta
ValMean <- colMeans(iris[, c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width")])
ValMean
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     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_quantity <- table(iris$Species)
pie(species_quantity,
    labels = paste(names(species_quantity), ": ", round(species_quantity / sum(species_quantity) * 100,
    col = c("lightblue", "lightgreen", "lightcoral"),
    main = "Species Distribution in Iris Dataset"
)
```




```r
#d Subset the species into setosa, versicolor, and virginica. Write the R scripts and show the last six

setosa_subset <- tail(subset(iris, Species == "setosa"), 6)
versicolor_subset <- tail(subset(iris, Species == "versicolor"), 6)
virginica_subset <- tail(subset(iris, Species == "virginica"), 6)

setosa_subset
```

```
##    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
```

```r
versicolor_subset
```

```
##     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
```

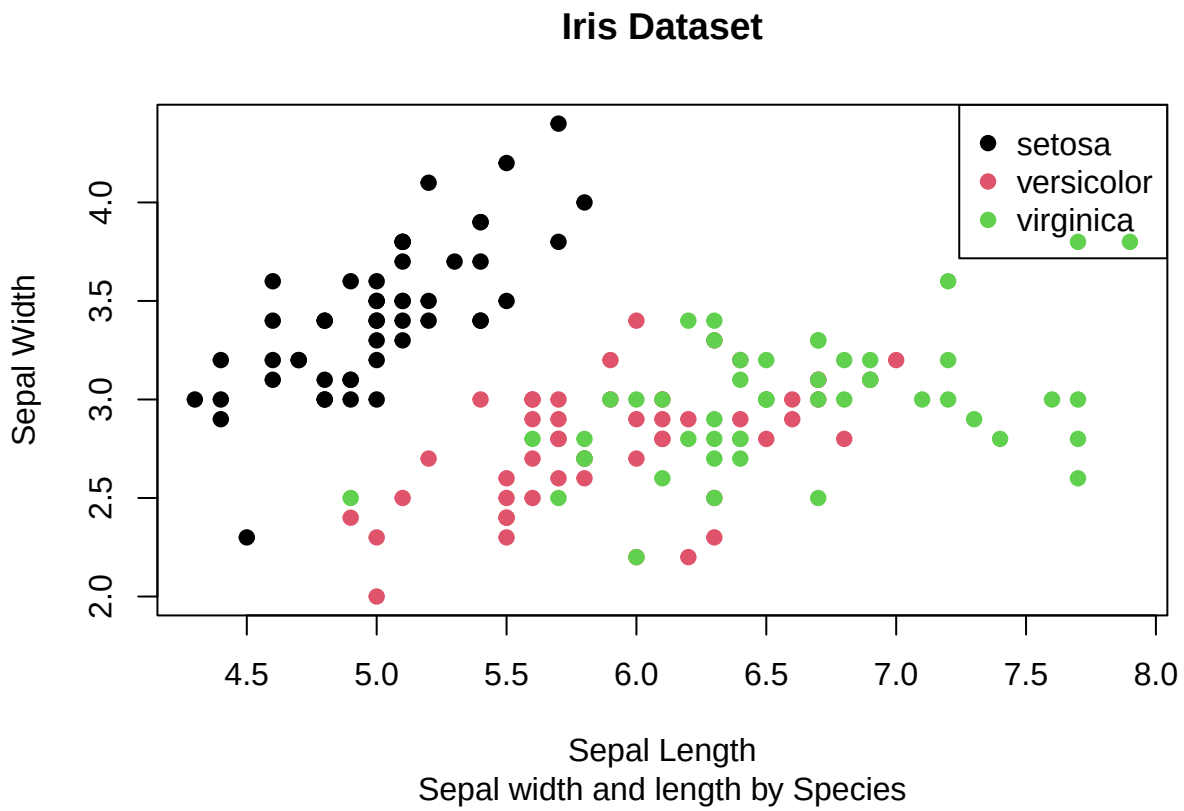
```
virginica_subset
```

```
##     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. Create a scatterplot of the sepal.length and sepal.width using the different species(setosa,versico

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 by Species"
)

legend("topright", legend = levels(iris$Species),
       col = 1:3, pch = 19)
```



```r
#f. Interpret the result.
#For part f, after creating the scatterplot, you can analyze the arrangement of the points based on spe
```

```r
#7. Import the alexa-file.xlsx. Check on the variations. Notice that there are extra whitespaces among
library(readxl)
library(readxl)
alexa_data <- read_excel("alexa_file.xlsx")

#a Rename the white and black variants by using gsub() function.
alexa_data$variation <- gsub("Black\\s+Dot", "Black Dot", alexa_data$variation)
alexa_data$variation <- gsub("Black\\s+Plus", "Black Plus", alexa_data$variation)
alexa_data$variation <- gsub("Black\\s+Show", "Black Show", alexa_data$variation)
alexa_data$variation <- gsub("Black\\s+Spot", "Black Spot", alexa_data$variation)
alexa_data$variation <- gsub("White\\s+Dot", "White Dot", alexa_data$variation)
alexa_data$variation <- gsub("White\\s+Plus", "White Plus", alexa_data$variation)
alexa_data$variation <- gsub("White\\s+Show", "White Show", alexa_data$variation)
alexa_data$variation <- gsub("White\\s+Spot", "White Spot", alexa_data$variation)


table(alexa_data$variation)
```

```
##
##                         Black                     Black Dot
##                           261                           516
##                    Black Plus                    Black Show
##                           270                           265
##                    Black Spot              Charcoal Fabric
##                           241                           430
## Configuration: Fire TV Stick          Heather Gray Fabric
##                           350                           157
##                    Oak Finish             Sandstone Fabric
##                            14                            90
##                 Walnut Finish                         White
##                             9                            91
##                     White Dot                    White Plus
##                           184                            78
##                    White Show                    White Spot
##                            85                           109
```

```r
#b. Get the total number of each variations and save it into another object. Save the object as variati
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_data %>%
  count(variation, name = "Total")


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


print(variations_count)
```

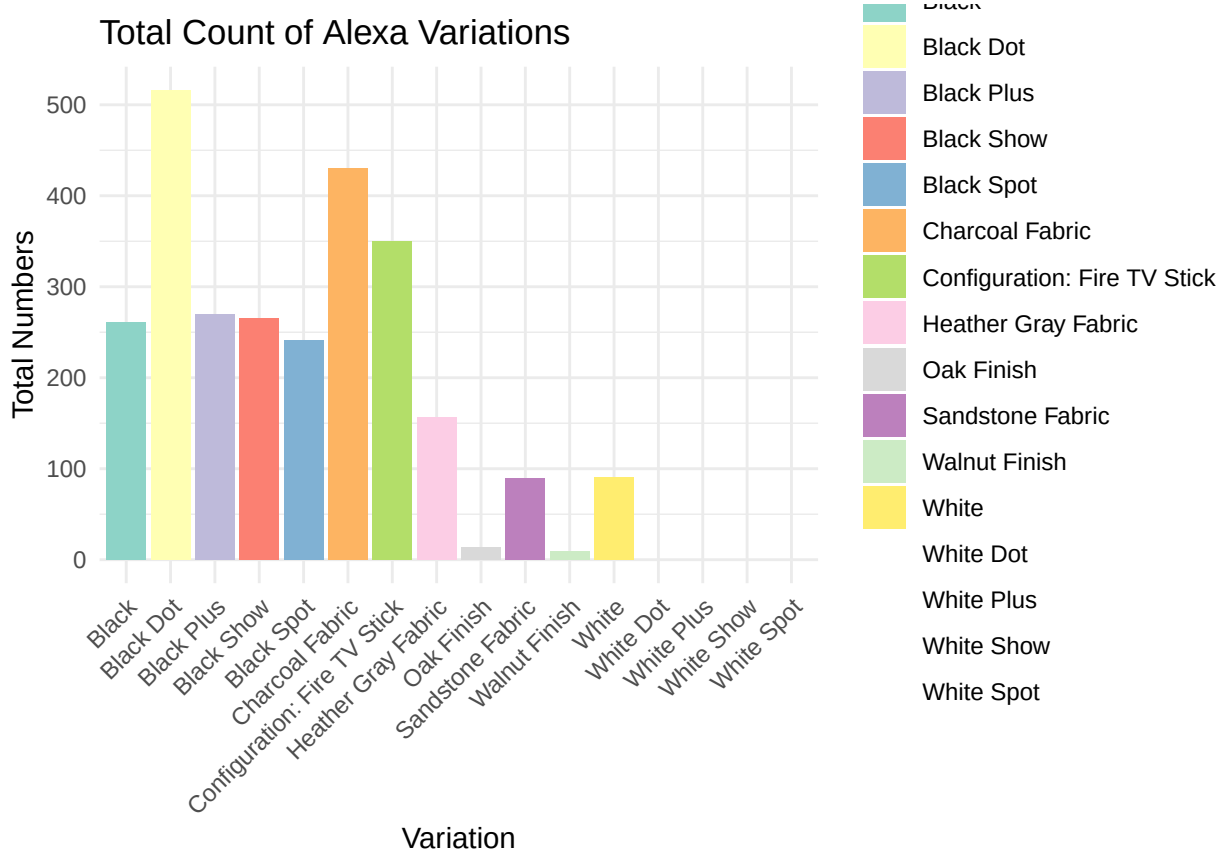
```
## # A tibble: 16 x 2
##    variation                    Total
##    <chr>                        <int>
##  1 Black                          261
##  2 Black Dot                      516
##  3 Black Plus                     270
##  4 Black Show                     265
##  5 Black Spot                     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(ggplot2)

load("variations.RData")


ggplot(variations_count, aes(x = variation, y = Total, fill = variation)) +
  geom_bar(stat = "identity") +
  ggtitle("Total Count of Alexa Variations") +
  xlab("Variation") +
  ylab("Total Numbers") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set3")
```

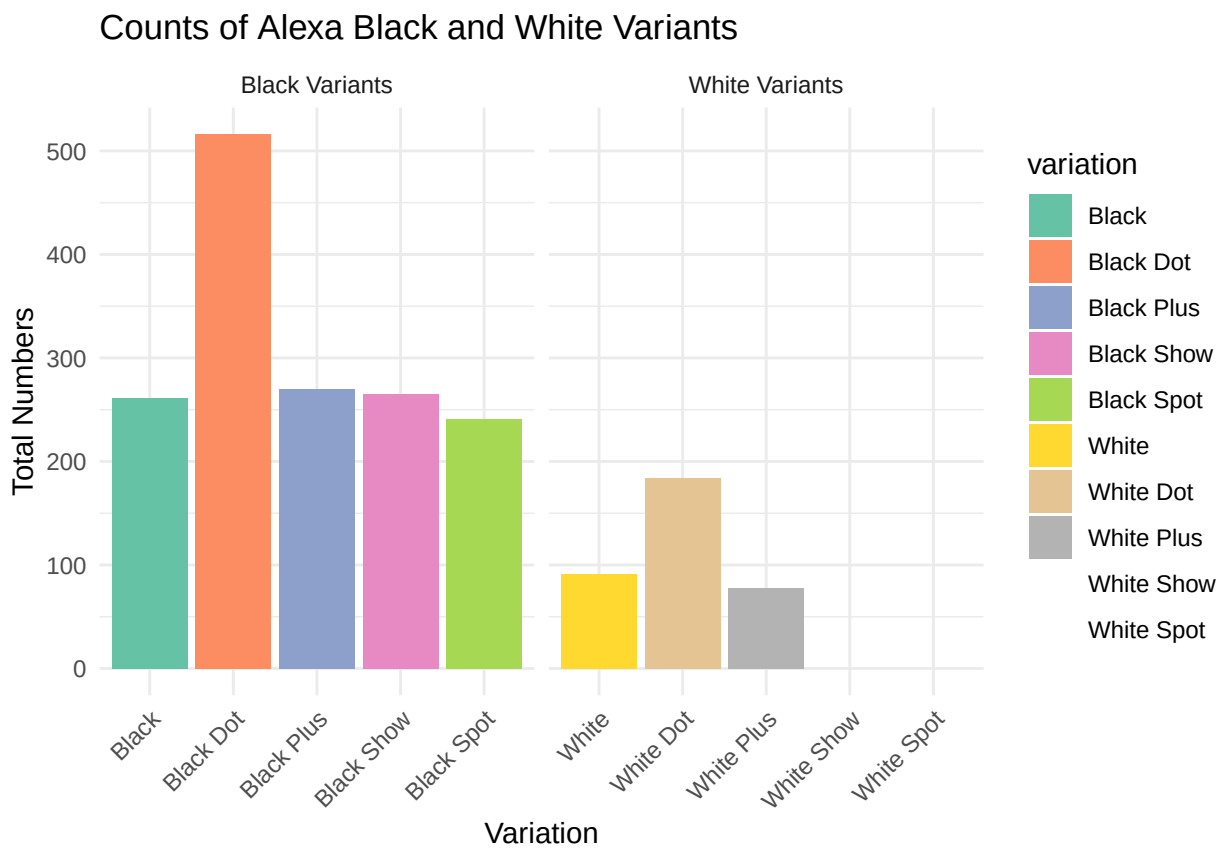
```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set3 is 12
## Returning the palette you asked for with that many colors
```




```
#d Create a barplot() for the black and white variations. Plot it in 1 frame, side byside. Complete the
variations_count$Category <- ifelse(grepl("Black", variations_count$variation), "Black Variants",
ifelse(grepl("White", variations_count$variation), "White Variants", NA))

black_white_variants <- variations_count %>% filter(!is.na(Category))
ggplot(black_white_variants, aes(x = variation, y = Total, fill = variation)) +
  geom_bar(stat = "identity") +
  facet_wrap(~ Category, scales = "free_x") +
  ggtitle("Counts of Alexa Black and White Variants") +
  xlab("Variation") +
  ylab("Total Numbers") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set2")
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors
```


Counts of Alexa Black and White Variants
Black Variants
White Variants
Total Numbers
500
400
300
200
100
0
Black
Black Dot
Black Plus
Black Show
Black Spot
White
White Dot
White Plus
White Show
White Spot
Variation
variation
Black
Black Dot
Black Plus
Black Show
Black Spot
White
White Dot
White Plus
White Show
White Spot