

worksheet#4

Missy Key Sadsad

2023-11-08

#Using 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.

```
vectorA <- c(1, 2, 3, 4, 5)
matrixA <- matrix(0, nrow = 5, ncol = 5)
```

matrixA

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
```

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

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

```
rightTriangle <- c()
for(i in 1:5) {
  for(j in 1:i+1) {
    rightTriangle = c(rightTriangle, "*")
  }
  print(rightTriangle)
  rightTriangle <- c()
}
```

```
## [1] "*"
## [1] "*" "*"
## [1] "*" "*" "*"
## [1] "*" "*" "*" "*"
## [1] "*" "*" "*" "*" "*"
```

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 <- as.integer(readline(prompt = "Enter the number of terms: "))
```

```
## Enter the number of terms:
```

```
a <- 0
b <- 1

cat("Fibonacci Sequence:", a, b)
```

```
## Fibonacci Sequence: 0 1
```

```
# Generate the sequence
repeat {
  c <- a + b
  if (c > 500) {
    break
  }
  cat(", ", c)
  a <- b
  b <- c
}
```

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

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

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? Show your codes and its result.

```
HouseholdData <-read.table("Household.csv",header = TRUE, sep = ",", as.is = TRUE)
View(HouseholdData)

head(HouseholdData,6)
```

```
##   Shoe.Size Height Gender
## 1      6.5   66.0      F
## 2      9.5   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
```

- b. Create a subset for gender(female and male). How many observations are there in Male? How about in Female? Write the R scripts and its output.

```
male_subset <- subset(HouseholdData, Gender == 'M')
female_subset <- subset(HouseholdData, Gender == 'F')

male_count <- nrow(male_subset)
male_count
```

```
## [1] 14
```

```
female_count <- nrow(female_subset)
female_count
```

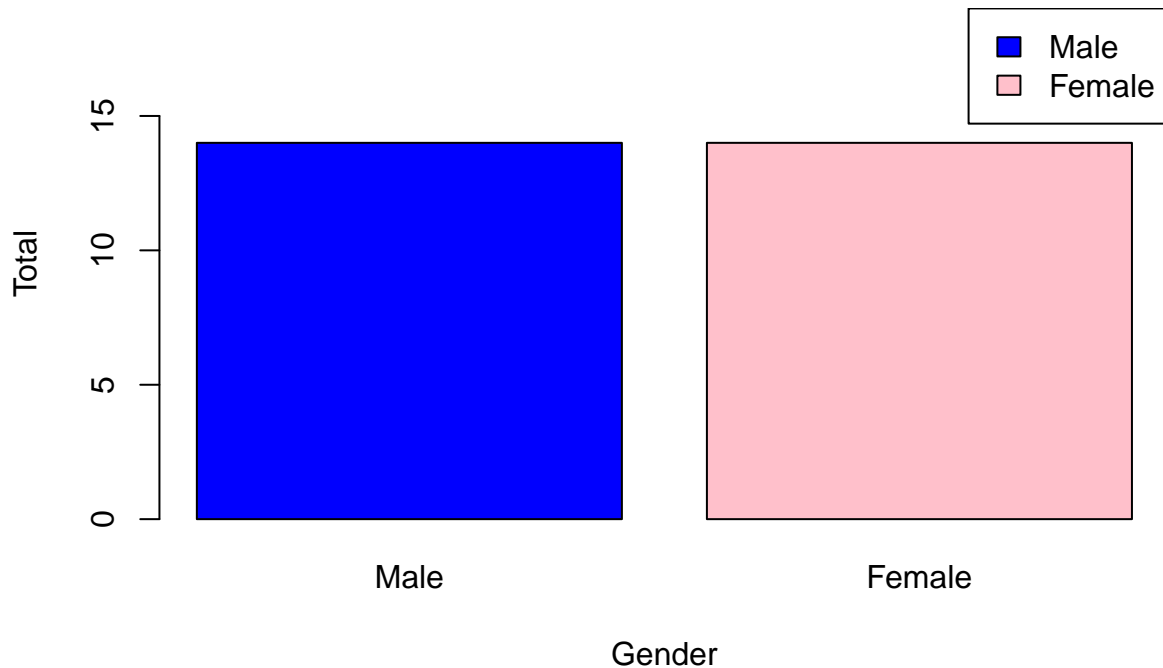
```
## [1] 14
```

- c. Create a graph for the number of males and females for Household Data. Use plot(), chart type = barplot. Make sure to place title, legends, and colors. Write the R scripts and its result.

```
total <- c(male_count, female_count)

barplot(total,
        names.arg = c("Male", "Female"),
        main = "The number of Males and Females in Household Data",
        xlab = "Gender",
        ylab = "Total",
        ylim = c(0, max(total)+ 5),
        col = c("blue", "pink"))
legend("topright",
       legend = c("Male", "Female"),
       fill = c("blue", "pink"))
```

The number of Males and Females in Household Data



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

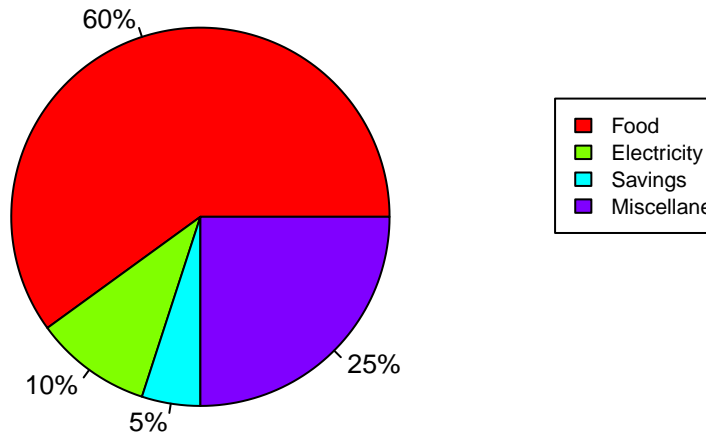
- Create a piechart that will include labels in percentage. Add some colors and title of the chart. Write the R scripts and show its output.

```
monthly_income <- c(60,10,5,25)

month_labels <- round(monthly_income/sum(monthly_income)*100,1)
month_labels <- paste(month_labels,"%", sep = "")

pie(monthly_income ,
    main = "The monthly income of Dela Cruz family",
    col = rainbow(length(monthly_income)),
    labels = month_labels,
    cex = 0.8)
legend(1.5,0.5,
    c("Food", "Electricity", "Savings", "Miscellaneous"),
    cex = 0.7,
    fill =rainbow(length(monthly_income)))
```

The monthly income of Dela Cruz family



6. Use the iris dataset.

- a. Check for the structure of the dataset using the `str()` function. Describe what you have seen in the output.

```
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 ...
```

#the str(iris) output gives us a clear overview of the structure of the Iris dataset, including the number of observations and the data types of the variables.

- b. Create an R object that will contain the mean of the sepal.length, sepal.width, petal.length, and petal.width. What is the R script and its result?

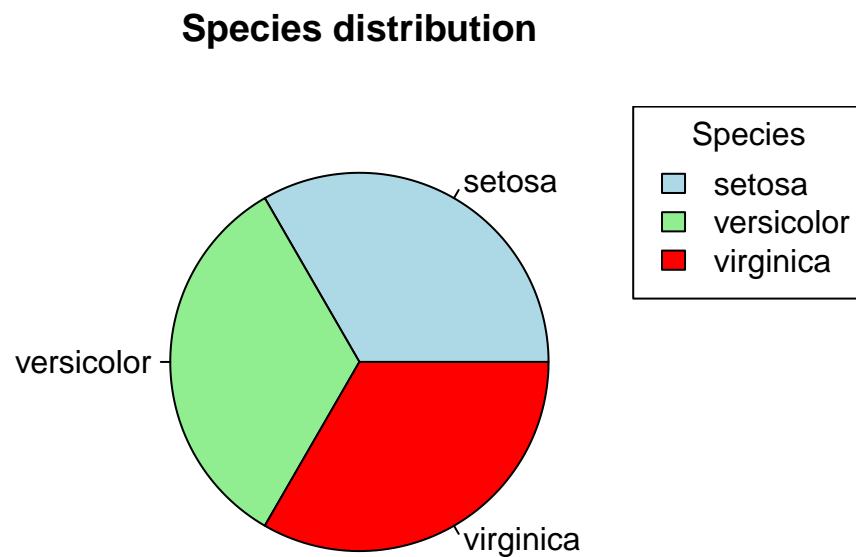
```
mean<- colMeans(iris[, c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width")])
mean
```

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

- c. Create a pie chart for the Species distribution. Add title, legends, and colors. Write the R script and its result.

```
pie(table(iris$Species),
     main = "Species distribution",
     labels = levels(iris$Species),
     col = c("lightblue","lightgreen","red"))

legend("topright",
      legend = levels(iris$Species),
      fill = c("lightblue","lightgreen","red") ,
      title = "Species")
```



- d. Subset the species into setosa, versicolor, and virginica. Write the R scripts and show the last six (6) rows of each species.

```
setosa_lastsix<- tail(subset(iris, Species == "setosa"), n = 6)
versicolor_lastsix <- tail(subset(iris, Species == "versicolor"), n = 6)
virginica_lastsix<- tail(subset(iris, Species == "virginica"), n = 6)
setosa_lastsix
```

```
##      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
```

```
versicolor_lastsix
```

```
##      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
```

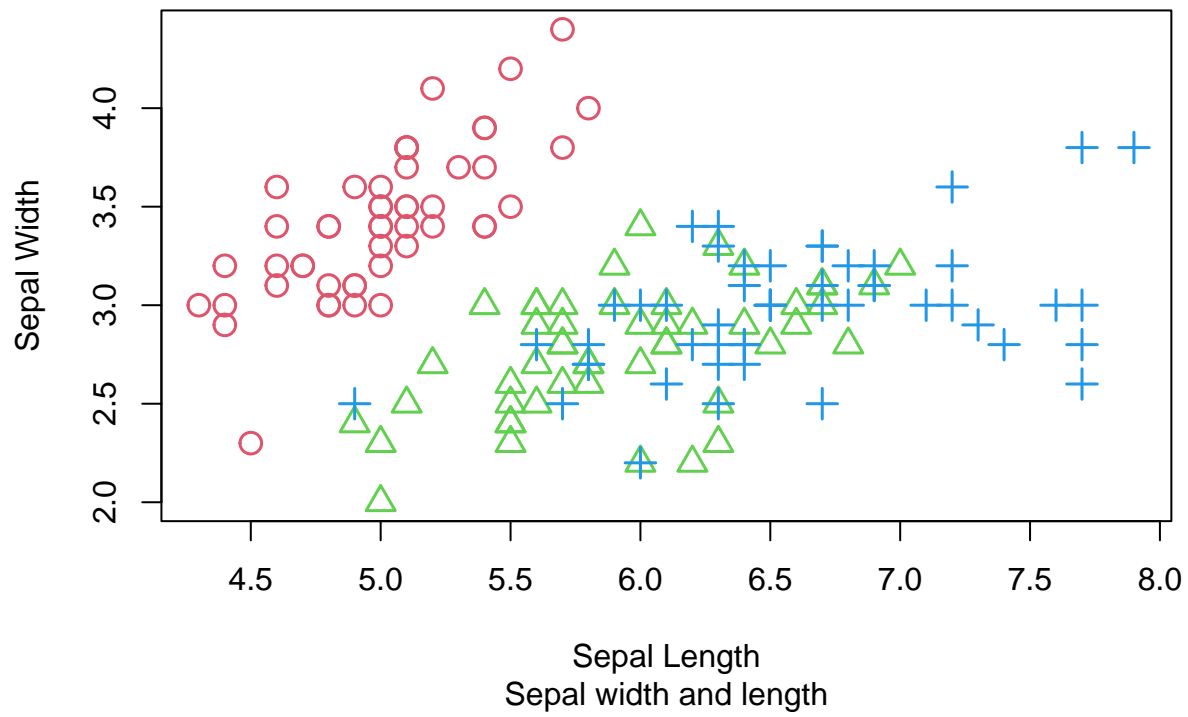
```
virginica_lastsix
```

```
##      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,versicolor,virginica). 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 species.

```
plot(iris$Sepal.Length, iris$Sepal.Width,
     pch = as.integer(iris$Species),
     col = as.integer(iris$Species) + 1,
     main = "Iris Dataset",
     sub = "Sepal width and length",
     xlab = "Sepal Length",
     ylab = "Sepal Width",
     cex = 1.5,
     lwd = 1.5)
```

Iris Dataset



```
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
```

#as.factor(iris\$Species) is a way of telling R that the "Species" variable should be treated as a category

Basic Cleaning and Transformation of Objects

7. Import the alexa-file.xlsx. Check on the variations. Notice that there are extra whitespaces among black variants (Black Dot, Black Plus, Black Show, Black Spot). Also on the white variants (White Dot, White Plus, White Show, White Spot).

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

```
library(readxl)
alexa_file <- read_excel("~/GitHub/RWorksheets_Sadsad/Worksheet#4/RWorksheet#4b/alexa_file.xlsx")
View(alexa_file)

alexa_file$variation <- gsub("Black Plus", "Black Plus", alexa_file$variation)
alexa_file$variation <- gsub("Black Show", "Black Show", alexa_file$variation)
alexa_file$variation <- gsub("Black Spot", "Black Spot", alexa_file$variation)
alexa_file$variation <- gsub("Black Dot", "Black Dot", alexa_file$variation)

alexa_file$variation <- gsub("White Dot", "White Dot", alexa_file$variation)
alexa_file$variation <- gsub("White Plus", "White Plus", alexa_file$variation)
alexa_file$variation <- gsub("White Show", "White Show", alexa_file$variation)
alexa_file$variation <- gsub("White Spot", "White Spot", alexa_file$variation)

knitr::include_graphics("C:/Users/missy/OneDrive/Documents/GitHub/RWorksheets_Sadsad/Worksheet#4/RWorksheets_Sadsad/Worksheet#4b/alexa_file.xlsx")
```

	alexa_file\$variation	n
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

- 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?

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.2
```

```
##
## 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
```

```
variations.RData <- alexa_file%>%count(alexa_file$variation)
variations.RData
```

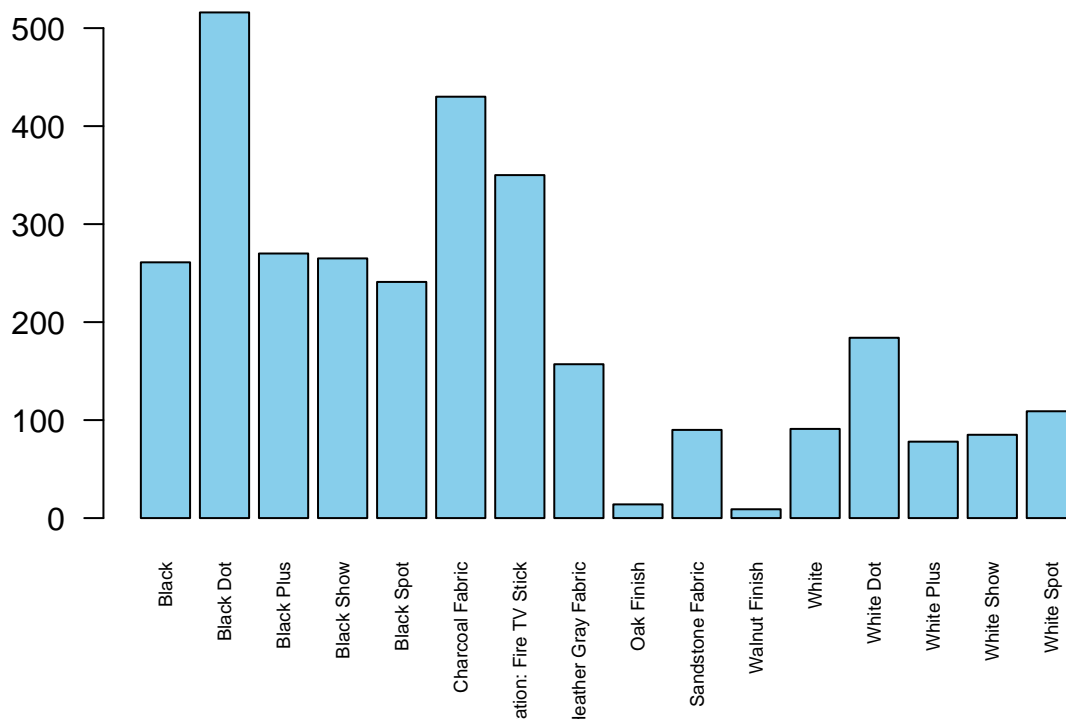
```
## # A tibble: 16 x 2
##   'alexa_file$variation'      n
##   <chr>                    <int>
## 1 Black                    261
## 2 Black Dot                516
## 3 Black Plus               270
## 4 Black Show               265
## 5 Black Spot               241
## 6 Charcoal Fabric          430
## 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
```

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

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

```
barplot(
  height = variations.RData$n,
  names.arg = variations.RData$`alexa_file$variation`,
  col = "skyblue",
  main = "Alexa Varations",
  las = 2,
  cex.names = 0.58
)
```

Alexa Variations



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

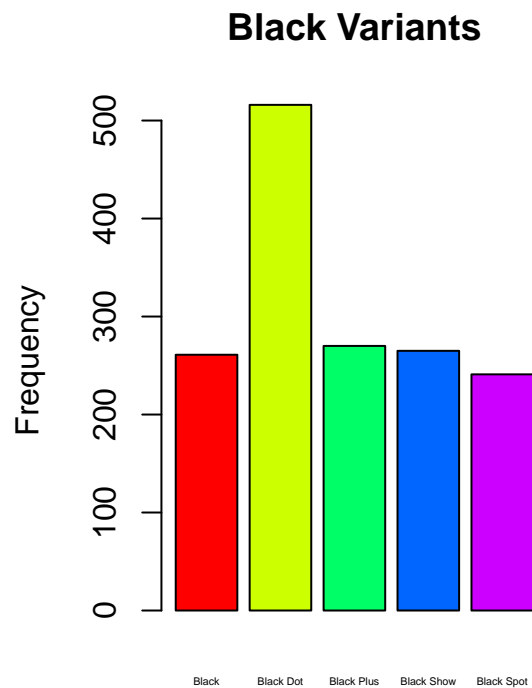
```
par(mfrow = c(1, 2))

black_variants <- variations.RData[1:5,]
white_variants <- variations.RData[12:16,]

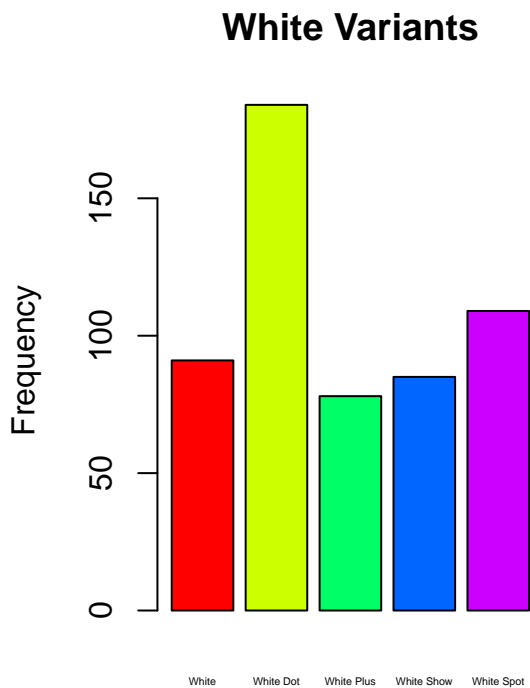
barplot(
  height = black_variants$n,
  names.arg = black_variants$`alexa_file$variation`,
  main = "Black Variants",
  col = rainbow(5),
  xlab = 'Total Numbers',
  ylab = 'Frequency',
  cex.names = 0.35,
)

barplot(
  height = white_variants$n,
  names.arg = white_variants$`alexa_file$variation`,
  main = "White Variants",
  col = rainbow(5),
  xlab = 'Total Numbers',
  ylab = 'Frequency',
)
```

```
cex.names = 0.35,  
)
```



Total Numbers



Total Numbers