

RWorksheets_Bagilidad#4b

2023-11-08

#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)

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 2.

```
for (i in 1:5) {
  cat(paste0("\n", rep("*", i), "\n"), "\n")
}
```

```
## "*"
## "*" "*"
## "*" "*" "*"
## "*" "*" "*" "*"
## "*" "*" "*" "*" "*"
```

#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.

```
begin <- as.integer(readline(prompt = "Enter the number to start the Fibonacci sequence: "))
```

```
## Enter the number to start the Fibonacci sequence:
```

```
#Assigning a value to the object begin, assuming that 4 is the user input. This is only for fixing the
begin <- 4
```

```
#Before running the code please remove ( begin <- 4) as you can see above this comment so that you can

a <- begin
b <- 0
```

```
cat("Fibonacci sequence starting from", begin, ":\n")
```

```
## Fibonacci sequence starting from 4 :
```

```
# Generate Fibonacci sequence
```

```
repeat
{
  nextFib <- a + b
  if (nextFib > 500)
  {
    break
  }
  cat(nextFib, " ")
  a <- b
  b <- nextFib
}
```

```
## 4 4 8 12 20 32 52 84 136 220 356
```

4. 4a. 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

```
# Two commonly used packages for this purpose are readxl for Excel files and readr for CSV files.
```

```
# For reading a csv file.
```

```
data_shoes <- read.csv("data_shoes.csv", header = TRUE)
head(data_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
```

- 4b. 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.

```
data_shoes <- read.csv("data_shoes.csv")
data_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
## 7 7         9.5   70.0     F
## 8 8         9.0   71.0     F
```

```
## 9 9 13.0 72.0 M
## 10 10 7.5 64.0 F
## 11 11 10.5 74.5 M
## 12 12 8.5 67.0 F
## 13 13 12.0 71.0 M
## 14 14 10.5 71.0 M
## 15 15 13.0 77.0 M
## 16 16 11.5 72.0 M
## 17 17 8.5 59.0 F
## 18 18 5.0 62.0 F
## 19 19 10.0 72.0 M
## 20 20 6.5 66.0 F
## 21 21 7.5 64.0 F
## 22 22 8.5 67.0 M
## 23 23 10.5 73.0 M
## 24 24 8.5 69.0 F
## 25 25 10.5 72.0 M
## 26 26 11.0 70.0 M
## 27 27 9.0 69.0 M
## 28 28 13.0 70.0 M
```

```
male_subset <- subset(data_shoes,Gender=="M")
male_subset
```

```
## X Shoe_Size Height Gender
## 5 5 10.5 70.0 M
## 9 9 13.0 72.0 M
## 11 11 10.5 74.5 M
## 13 13 12.0 71.0 M
## 14 14 10.5 71.0 M
## 15 15 13.0 77.0 M
## 16 16 11.5 72.0 M
## 19 19 10.0 72.0 M
## 22 22 8.5 67.0 M
## 23 23 10.5 73.0 M
## 25 25 10.5 72.0 M
## 26 26 11.0 70.0 M
## 27 27 9.0 69.0 M
## 28 28 13.0 70.0 M
```

```
female_subset <- subset(data_shoes,Gender=="F")
female_subset
```

```
## 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
## 6 6 7.0 64.0 F
## 7 7 9.5 70.0 F
## 8 8 9.0 71.0 F
## 10 10 7.5 64.0 F
## 12 12 8.5 67.0 F
## 17 17 8.5 59.0 F
## 18 18 5.0 62.0 F
```

```
## 20 20      6.5  66.0    F
## 21 21      7.5  64.0    F
## 24 24      8.5  69.0    F
```

```
totalMale <- nrow(male_subset)
totalMale
```

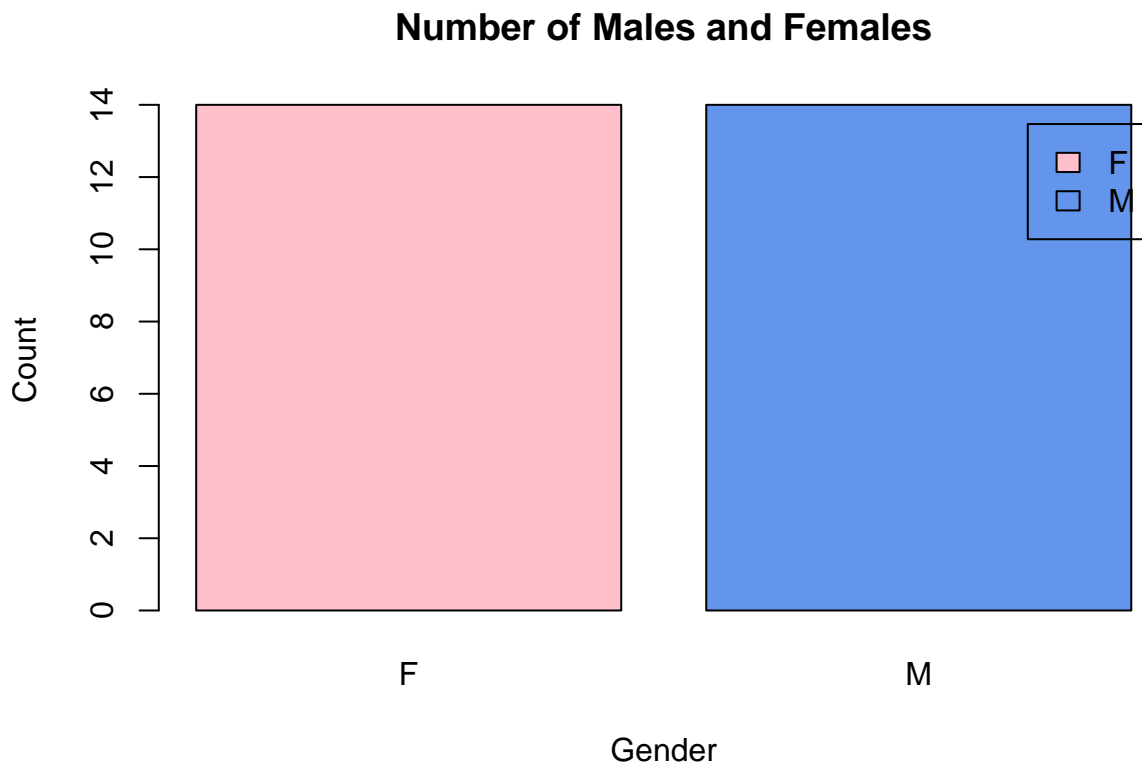
```
## [1] 14
```

```
totalFemale <- nrow(female_subset)
totalFemale
```

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

4c. 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.

```
totalMale_Female <- table(data_shoes$Gender)
barplot(totalMale_Female,
        main = "Number of Males and Females",
        xlab = "Gender",
        ylab = "Count",
        col = c("pink", "cornflowerblue"),
        legend.text = rownames(totalMale_Female),
        beside = TRUE)
```



5. The monthly income of Dela Cruz family was spent on the following: Food 60 Electricity 10 Savings 5 Miscellaneous 25

5a. Create a piechart that will include labels in percentage. Add some colors and title of chart. Write the R scripts and show its output.

```
# Define the data
data <- c(Food = 60, Electricity = 10, Savings = 5, Miscellaneous = 25)
```

```

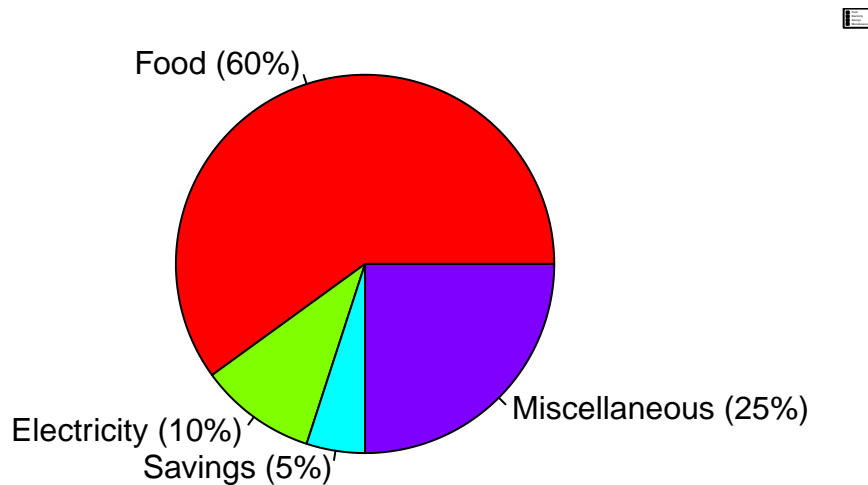
# Calculate percentages and format them as strings
percentages <- paste(round(100 * data / sum(data), 1), "%", sep = "")

# Create a pie chart
pie(data,
     labels = paste(names(data),
                    " (", percentages, ")", sep = ""),
     col = rainbow(length(data)),
     main = "Expense Distribution")

# Add a legend
legend("topright",
      names(data),
      cex = 0.10,
      fill = rainbow(length(data)))

```

Expense Distribution



6. 6a. Check for the structure of the data set 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 ...

```

6b. 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_Iris <- c(mean(iris$Sepal.Length), mean(iris$Sepal.Width), mean(iris$Petal.Length), mean(iris$Petal.Width))
mean_Iris

```

```

## [1] 5.843333 3.057333 3.758000 1.199333

```

```

mean_sepal_length <- mean(iris$Sepal.Length)
mean_sepal_width <- mean(iris$Sepal.Width)
mean_petal_length <- mean(iris$Petal.Length)
mean_petal_width <- mean(iris$Petal.Width)

mean_Iris <- data.frame(Sepal_Length = mean_sepal_length,
                        Sepal_Width = mean_sepal_width,
                        Petal_Length = mean_petal_length,
                        Petal_Width = mean_petal_width)

mean_Iris

```

```

##   Sepal_Length Sepal_Width Petal_Length Petal_Width
## 1      5.843333    3.057333      3.758      1.199333

```

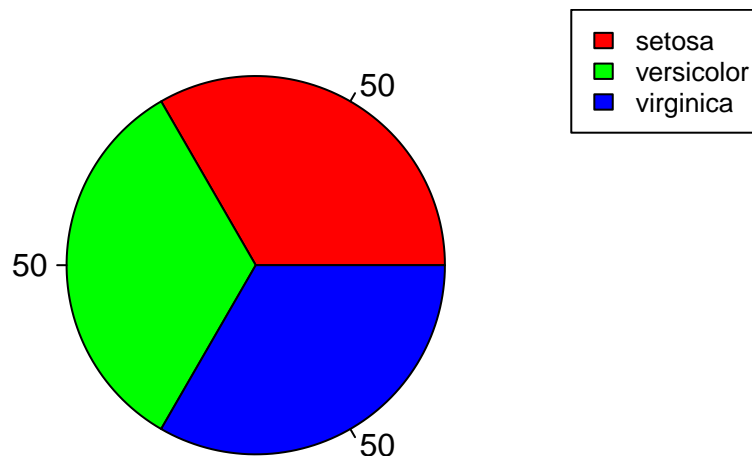
6c. Create a pie chart for the Species distribution. Add title, legends, and colors. Write the R script and its result.

```

species_count <- table(iris$Species)
pie(species_count, labels = species_count, col = rainbow(length(species_count)), main = "Species Distribution", legend("topright", names(species_count), cex = 0.8, fill = rainbow(length(species_count))))

```

Species Distribution



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

```

# Subset the iris data set into the three species.
setosa_subset <- subset(iris, Species == "setosa")
setosa_subset

```

```

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1         5.1         3.5         1.4         0.2   setosa
## 2         4.9         3.0         1.4         0.2   setosa
## 3         4.7         3.2         1.3         0.2   setosa
## 4         4.6         3.1         1.5         0.2   setosa
## 5         5.0         3.6         1.4         0.2   setosa
## 6         5.4         3.9         1.7         0.4   setosa
## 7         4.6         3.4         1.4         0.3   setosa
## 8         5.0         3.4         1.5         0.2   setosa
## 9         4.4         2.9         1.4         0.2   setosa
## 10        4.9         3.1         1.5         0.1   setosa
## 11        5.4         3.7         1.5         0.2   setosa

```

```
## 12      4.8      3.4      1.6      0.2 setosa
## 13      4.8      3.0      1.4      0.1 setosa
## 14      4.3      3.0      1.1      0.1 setosa
## 15      5.8      4.0      1.2      0.2 setosa
## 16      5.7      4.4      1.5      0.4 setosa
## 17      5.4      3.9      1.3      0.4 setosa
## 18      5.1      3.5      1.4      0.3 setosa
## 19      5.7      3.8      1.7      0.3 setosa
## 20      5.1      3.8      1.5      0.3 setosa
## 21      5.4      3.4      1.7      0.2 setosa
## 22      5.1      3.7      1.5      0.4 setosa
## 23      4.6      3.6      1.0      0.2 setosa
## 24      5.1      3.3      1.7      0.5 setosa
## 25      4.8      3.4      1.9      0.2 setosa
## 26      5.0      3.0      1.6      0.2 setosa
## 27      5.0      3.4      1.6      0.4 setosa
## 28      5.2      3.5      1.5      0.2 setosa
## 29      5.2      3.4      1.4      0.2 setosa
## 30      4.7      3.2      1.6      0.2 setosa
## 31      4.8      3.1      1.6      0.2 setosa
## 32      5.4      3.4      1.5      0.4 setosa
## 33      5.2      4.1      1.5      0.1 setosa
## 34      5.5      4.2      1.4      0.2 setosa
## 35      4.9      3.1      1.5      0.2 setosa
## 36      5.0      3.2      1.2      0.2 setosa
## 37      5.5      3.5      1.3      0.2 setosa
## 38      4.9      3.6      1.4      0.1 setosa
## 39      4.4      3.0      1.3      0.2 setosa
## 40      5.1      3.4      1.5      0.2 setosa
## 41      5.0      3.5      1.3      0.3 setosa
## 42      4.5      2.3      1.3      0.3 setosa
## 43      4.4      3.2      1.3      0.2 setosa
## 44      5.0      3.5      1.6      0.6 setosa
## 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_subset <- subset(iris, Species == "versicolor")
versicolor_subset
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
## 51      7.0      3.2      4.7      1.4 versicolor
## 52      6.4      3.2      4.5      1.5 versicolor
## 53      6.9      3.1      4.9      1.5 versicolor
## 54      5.5      2.3      4.0      1.3 versicolor
## 55      6.5      2.8      4.6      1.5 versicolor
## 56      5.7      2.8      4.5      1.3 versicolor
## 57      6.3      3.3      4.7      1.6 versicolor
## 58      4.9      2.4      3.3      1.0 versicolor
## 59      6.6      2.9      4.6      1.3 versicolor
## 60      5.2      2.7      3.9      1.4 versicolor
## 61      5.0      2.0      3.5      1.0 versicolor
```

```
## 62      5.9      3.0      4.2      1.5 versicolor
## 63      6.0      2.2      4.0      1.0 versicolor
## 64      6.1      2.9      4.7      1.4 versicolor
## 65      5.6      2.9      3.6      1.3 versicolor
## 66      6.7      3.1      4.4      1.4 versicolor
## 67      5.6      3.0      4.5      1.5 versicolor
## 68      5.8      2.7      4.1      1.0 versicolor
## 69      6.2      2.2      4.5      1.5 versicolor
## 70      5.6      2.5      3.9      1.1 versicolor
## 71      5.9      3.2      4.8      1.8 versicolor
## 72      6.1      2.8      4.0      1.3 versicolor
## 73      6.3      2.5      4.9      1.5 versicolor
## 74      6.1      2.8      4.7      1.2 versicolor
## 75      6.4      2.9      4.3      1.3 versicolor
## 76      6.6      3.0      4.4      1.4 versicolor
## 77      6.8      2.8      4.8      1.4 versicolor
## 78      6.7      3.0      5.0      1.7 versicolor
## 79      6.0      2.9      4.5      1.5 versicolor
## 80      5.7      2.6      3.5      1.0 versicolor
## 81      5.5      2.4      3.8      1.1 versicolor
## 82      5.5      2.4      3.7      1.0 versicolor
## 83      5.8      2.7      3.9      1.2 versicolor
## 84      6.0      2.7      5.1      1.6 versicolor
## 85      5.4      3.0      4.5      1.5 versicolor
## 86      6.0      3.4      4.5      1.6 versicolor
## 87      6.7      3.1      4.7      1.5 versicolor
## 88      6.3      2.3      4.4      1.3 versicolor
## 89      5.6      3.0      4.1      1.3 versicolor
## 90      5.5      2.5      4.0      1.3 versicolor
## 91      5.5      2.6      4.4      1.2 versicolor
## 92      6.1      3.0      4.6      1.4 versicolor
## 93      5.8      2.6      4.0      1.2 versicolor
## 94      5.0      2.3      3.3      1.0 versicolor
## 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 <- subset(iris, Species == "virginica")
virginica_subset
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
## 101      6.3      3.3      6.0      2.5 virginica
## 102      5.8      2.7      5.1      1.9 virginica
## 103      7.1      3.0      5.9      2.1 virginica
## 104      6.3      2.9      5.6      1.8 virginica
## 105      6.5      3.0      5.8      2.2 virginica
## 106      7.6      3.0      6.6      2.1 virginica
## 107      4.9      2.5      4.5      1.7 virginica
## 108      7.3      2.9      6.3      1.8 virginica
## 109      6.7      2.5      5.8      1.8 virginica
## 110      7.2      3.6      6.1      2.5 virginica
## 111      6.5      3.2      5.1      2.0 virginica
```



```
## 112      6.4      2.7      5.3      1.9 virginica
## 113      6.8      3.0      5.5      2.1 virginica
## 114      5.7      2.5      5.0      2.0 virginica
## 115      5.8      2.8      5.1      2.4 virginica
## 116      6.4      3.2      5.3      2.3 virginica
## 117      6.5      3.0      5.5      1.8 virginica
## 118      7.7      3.8      6.7      2.2 virginica
## 119      7.7      2.6      6.9      2.3 virginica
## 120      6.0      2.2      5.0      1.5 virginica
## 121      6.9      3.2      5.7      2.3 virginica
## 122      5.6      2.8      4.9      2.0 virginica
## 123      7.7      2.8      6.7      2.0 virginica
## 124      6.3      2.7      4.9      1.8 virginica
## 125      6.7      3.3      5.7      2.1 virginica
## 126      7.2      3.2      6.0      1.8 virginica
## 127      6.2      2.8      4.8      1.8 virginica
## 128      6.1      3.0      4.9      1.8 virginica
## 129      6.4      2.8      5.6      2.1 virginica
## 130      7.2      3.0      5.8      1.6 virginica
## 131      7.4      2.8      6.1      1.9 virginica
## 132      7.9      3.8      6.4      2.0 virginica
## 133      6.4      2.8      5.6      2.2 virginica
## 134      6.3      2.8      5.1      1.5 virginica
## 135      6.1      2.6      5.6      1.4 virginica
## 136      7.7      3.0      6.1      2.3 virginica
## 137      6.3      3.4      5.6      2.4 virginica
## 138      6.4      3.1      5.5      1.8 virginica
## 139      6.0      3.0      4.8      1.8 virginica
## 140      6.9      3.1      5.4      2.1 virginica
## 141      6.7      3.1      5.6      2.4 virginica
## 142      6.9      3.1      5.1      2.3 virginica
## 143      5.8      2.7      5.1      1.9 virginica
## 144      6.8      3.2      5.9      2.3 virginica
## 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
```

```
# Display the last six rows of each species.
```

```
tail(setosa_subset, 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
```

```
tail(versicolor_subset, 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
```

```
tail(virginica_subset, 6)
```

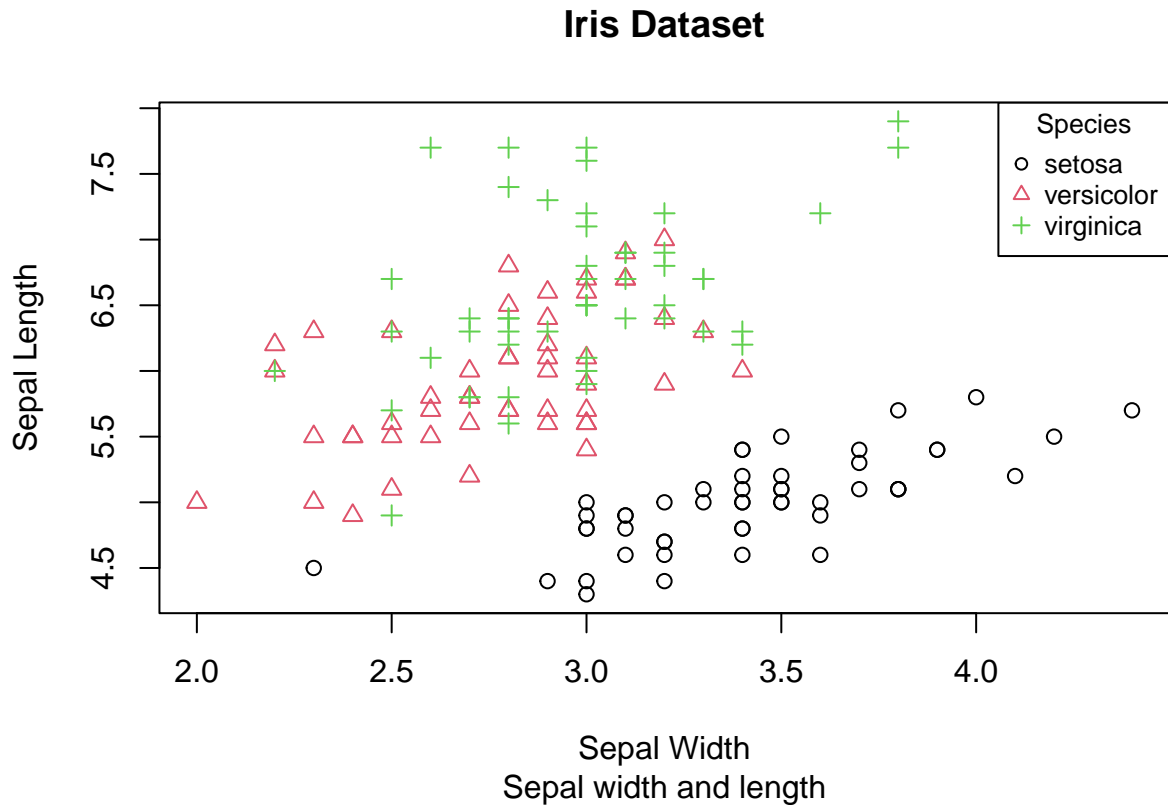
```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width  Species
## 145          6.7          3.3          5.7          2.5 virginica
## 146          6.7          3.0          5.2          2.3 virginica
## 147          6.3          2.5          5.0          1.9 virginica
## 148          6.5          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
```

6e. 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.

```
# Convert the "Species" column to a factor
iris$Species <- as.factor(iris$Species)

# Create a scatterplot
plot(
  Sepal.Length ~ Sepal.Width,
  data = iris,
  pch = as.integer(iris$Species), # Use different pch symbols for each species
  col = as.integer(iris$Species), # Use different colors for each species
  xlab = "Sepal Width",
  ylab = "Sepal Length",
  main = "Iris Dataset",
  sub = "Sepal width and length"
)

# Add a legend
legend("topright",
  legend = levels(iris$Species),
  col = 1:3, pch = 1:3,
  cex = 0.8,
  title = "Species")
```



6f. Interpret the result.

The dataset consists of five variables (columns) and 150 observations (rows) in a data frame format.

Petal.Length, Petal.Width, Sepal.Length, and Sepal. Width are the names of the four numerical variables. These stand for the measures of the length and width of the petals, respectively.

The factor variable Species, which represents the species of iris flowers, is the sixth variable. There are three tiers to it: “setosa,” “versicolor,” and “virginica,” denoting the many iris flower species included in the dataset.

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).

```
library(readxl)

alexa <- read_xlsx("alexa_file.xlsx")
alexa
```

A tibble: 3,150 x 5

##	rating	date	variation	verified_reviews	feedback
##	<dbl>	<dtm>	<chr>	<chr>	<dbl>
## 1	5	2018-07-31 00:00:00	Charcoal Fabric	Love my Echo!	1
## 2	5	2018-07-31 00:00:00	Charcoal Fabric	Loved it!	1
## 3	4	2018-07-31 00:00:00	Walnut Finish	Sometimes while play~	1
## 4	5	2018-07-31 00:00:00	Charcoal Fabric	I have had a lot of ~	1
## 5	5	2018-07-31 00:00:00	Charcoal Fabric	Music	1
## 6	5	2018-07-31 00:00:00	Heather Gray Fabric	I received the echo ~	1
## 7	3	2018-07-31 00:00:00	Sandstone Fabric	Without having a cel~	1
## 8	5	2018-07-31 00:00:00	Charcoal Fabric	I think this is the ~	1

```
## 9      5 2018-07-30 00:00:00 Heather Gray Fabric looks great      1
## 10     5 2018-07-30 00:00:00 Heather Gray Fabric Love it! I've listen~ 1
## # i 3,140 more rows
```

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

```
alexa$variation <- gsub("Black Dot",
                        "BlackDot",
                        alexa$variation)
alexa$variation <- gsub("Black Plus",
                        "BlackPlus",
                        alexa$variation)

alexa$variation <- gsub("Black Show",
                        "BlackShow",
                        alexa$variation)

alexa$variation <- gsub("Black Spot",
                        "BlackSpot",
                        alexa$variation)

alexa$variation <- gsub("White Dot",
                        "WhiteDot",
                        alexa$variation)

alexa$variation <- gsub("White Plus",
                        "WhitePlus",
                        alexa$variation)
alexa$variation <- gsub("White Show",
                        "WhiteShow",
                        alexa$variation)

alexa$variation <- gsub("White Spot",
                        "WhiteSpot",
                        alexa$variation)

alexa
```

```
## # A tibble: 3,150 x 5
##   rating date      variation      verified_reviews      feedback
##   <dbl> <dtm>      <chr>      <chr>      <dbl>
## 1      5 2018-07-31 00:00:00 Charcoal Fabric Love my Echo!      1
## 2      5 2018-07-31 00:00:00 Charcoal Fabric Loved it!      1
## 3      4 2018-07-31 00:00:00 Walnut Finish Sometimes while play~ 1
## 4      5 2018-07-31 00:00:00 Charcoal Fabric I have had a lot of ~ 1
## 5      5 2018-07-31 00:00:00 Charcoal Fabric Music      1
## 6      5 2018-07-31 00:00:00 Heather Gray Fabric I received the echo ~ 1
## 7      3 2018-07-31 00:00:00 Sandstone Fabric Without having a cel~ 1
## 8      5 2018-07-31 00:00:00 Charcoal Fabric I think this is the ~ 1
## 9      5 2018-07-30 00:00:00 Heather Gray Fabric looks great      1
## 10     5 2018-07-30 00:00:00 Heather Gray Fabric Love it! I've listen~ 1
## # i 3,140 more rows
```

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?

Hint: Use the dplyr package. Make sure to install it before loading the package.

```
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

variation_total <- alexa %>% count(alexas$variation)
variation_total

## # A tibble: 16 x 2
##   `alexas$variation`      n
##   <chr>                <int>
## 1 Black                261
## 2 BlackDot             516
## 3 BlackPlus            270
## 4 BlackShow            265
## 5 BlackSpot            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 WhiteDot             184
## 14 WhitePlus            78
## 15 WhiteShow            85
## 16 WhiteSpot           109

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

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

```
load("variations.RData")

variation_total

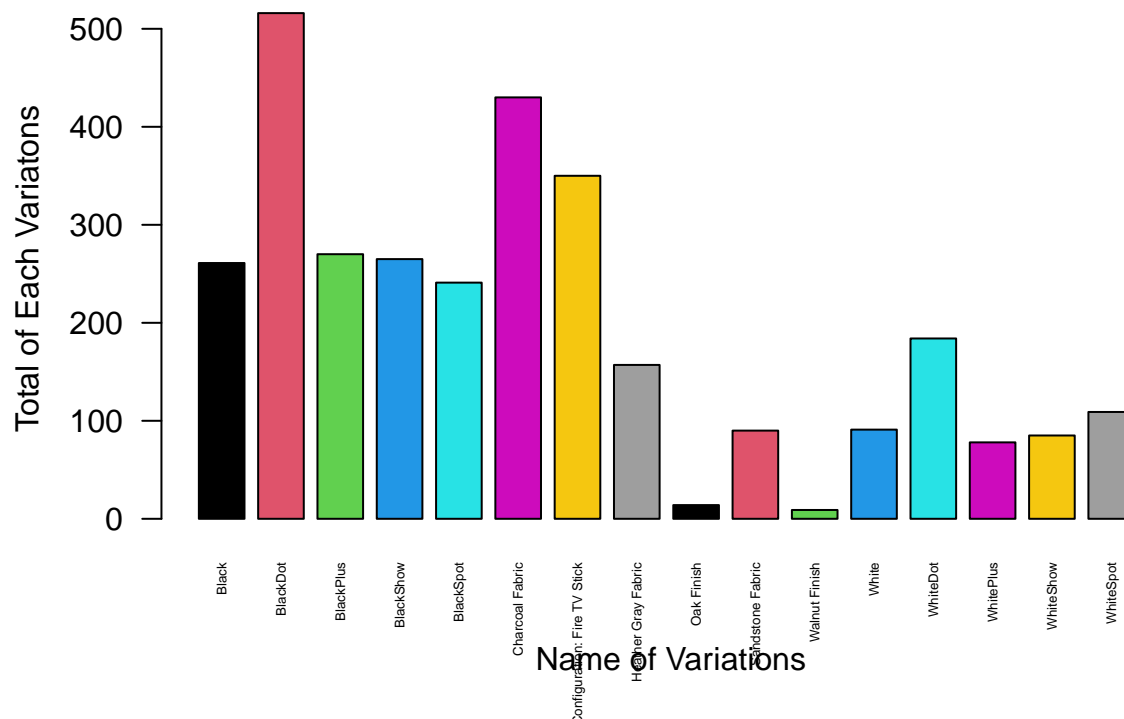
## # A tibble: 16 x 2
##   `alexas$variation`      n
##   <chr>                <int>
## 1 Black                261
## 2 BlackDot             516
## 3 BlackPlus            270
## 4 BlackShow            265
## 5 BlackSpot            241
## 6 Charcoal Fabric      430
## 7 Configuration: Fire TV Stick 350
## 8 Heather Gray Fabric  157
```

```
## 9 Oak Finish 14
## 10 Sandstone Fabric 90
## 11 Walnut Finish 9
## 12 White 91
## 13 WhiteDot 184
## 14 WhitePlus 78
## 15 WhiteShow 85
## 16 WhiteSpot 109
```

```
variable_Names <- variation_total$`alexa$variation`
```

```
totalVar_Plot <- barplot(variation_total$n,
  names.arg = variable_Names,
  main = " Total Number of Each Variations",
  xlab = " Name of Variations",
  ylab = "Total of Each Variatons",
  col = 1:16,
  space = 0.3,
  cex.names = 0.4,
  las = 2 )
```

Total Number of Each Variations



```
totalVar_Plot
```

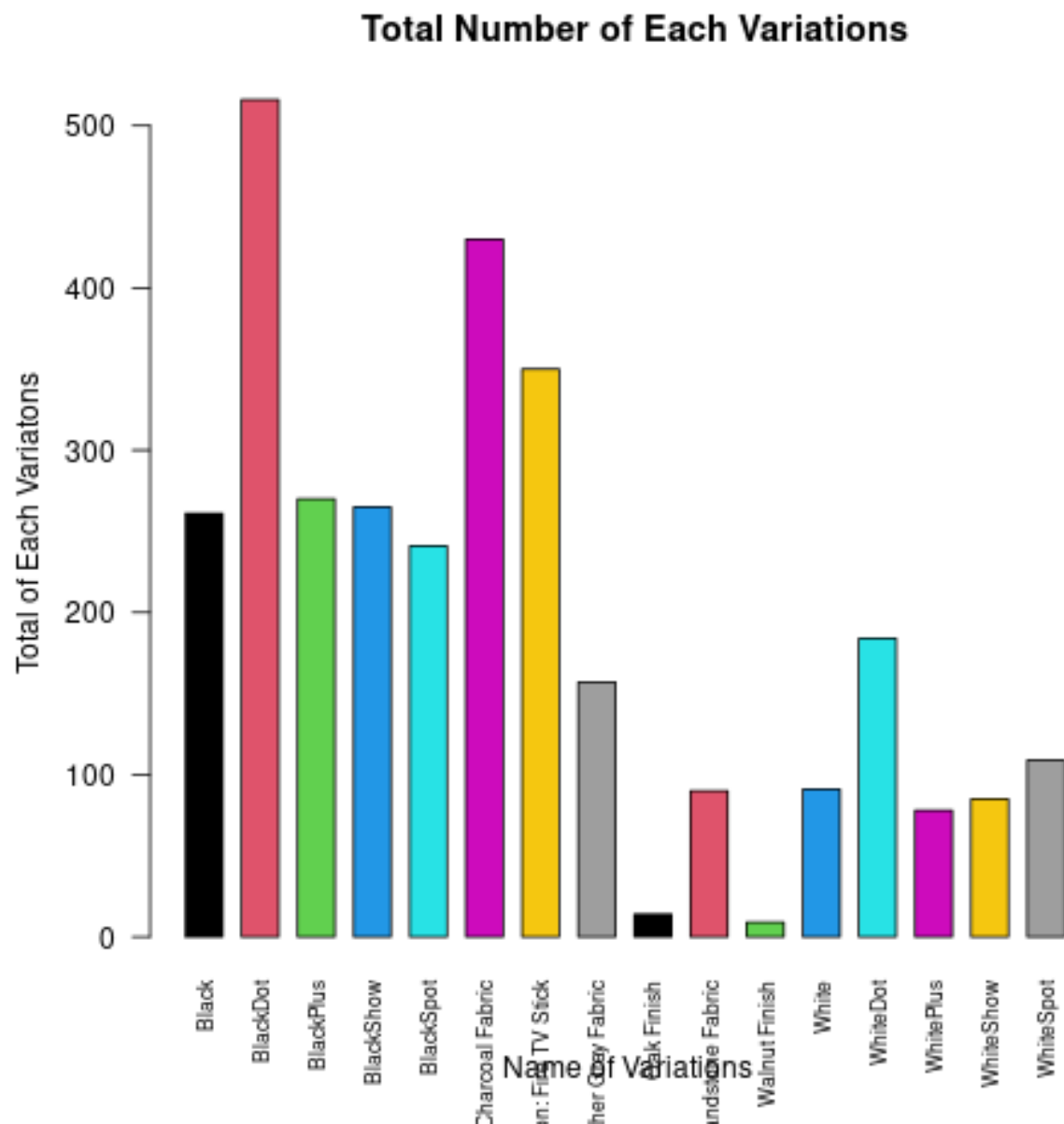
```
##      [,1]
## [1,] 0.8
## [2,] 2.1
## [3,] 3.4
## [4,] 4.7
## [5,] 6.0
```

```
## [6,] 7.3
## [7,] 8.6
## [8,] 9.9
## [9,] 11.2
## [10,] 12.5
## [11,] 13.8
## [12,] 15.1
## [13,] 16.4
## [14,] 17.7
## [15,] 19.0
## [16,] 20.3

png(filename = "/cloud/project/RWorksheets#4B/totalVar_Plot.png", width = 600, height = 400)
dev.copy(png, "totalVar_Plot.png")

## png
## 4

knitr::include_graphics("/cloud/project/RWorksheets#4B/totalVar_Plot.png")
```



```
dev.off()
```

```
## pdf
## 2
```

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

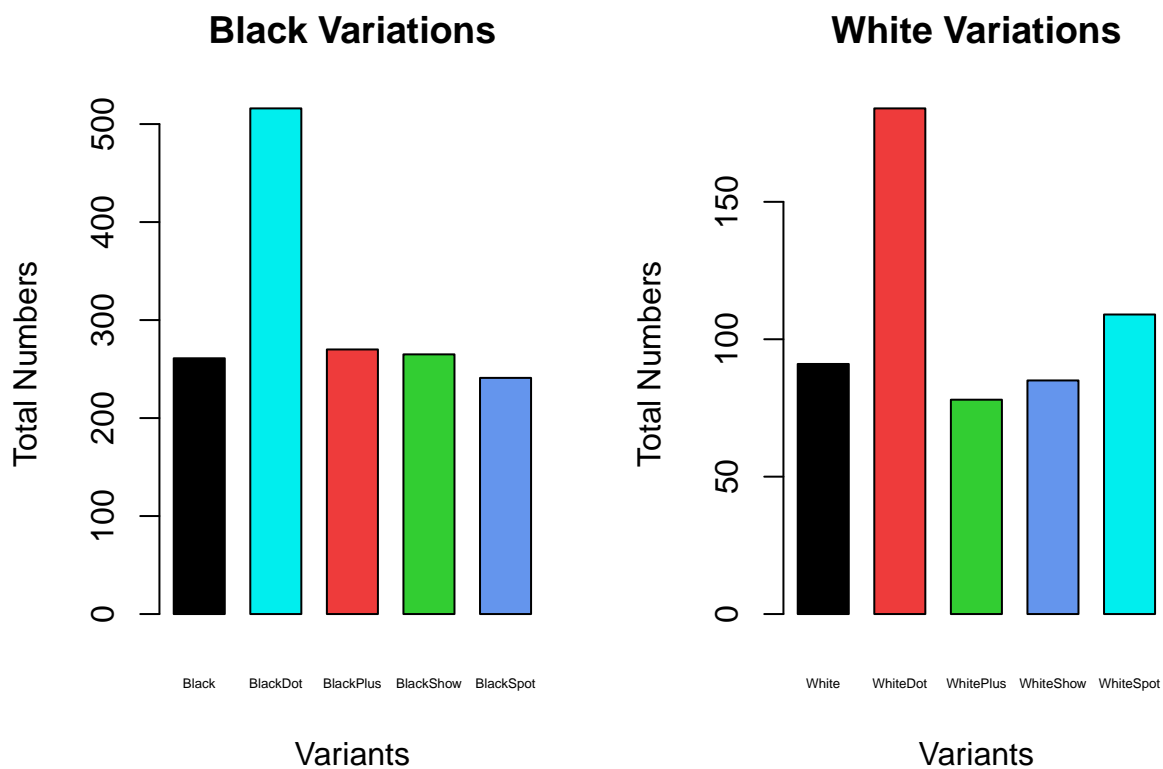
```
library(graphics)
black_variation <- variation_total[variation_total$`alexa$variation` %in% c("Black", "BlackDot", "BlackShow", "BlackSpot")]
white_variation <- variation_total[variation_total$`alexa$variation` %in% c("White", "WhiteDot", "WhitePlus", "WhiteShow", "WhiteSpot"), ]
```



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

```
black_Plot <- barplot(height = black_variation$n,
  names.arg = black_variation$`alexa$variation`,
  col = c("black","cyan2","brown2","limegreen","cornflowerblue"),
  main = "Black Variations",
  xlab = "Variants",
  ylab = "Total Numbers",
  border = "black",
  space = 0.5,
  cex.names = 0.4)
```

```
white_Plot <- barplot(height = white_variation$n,
  names.arg = white_variation$`alexa$variation`,
  col = c("black","brown2","limegreen","cornflowerblue","cyan2"),
  main = "White Variations",
  xlab = "Variants",
  ylab = "Total Numbers",
  border = "black",
  space = 0.5,
  cex.names = 0.4)
```



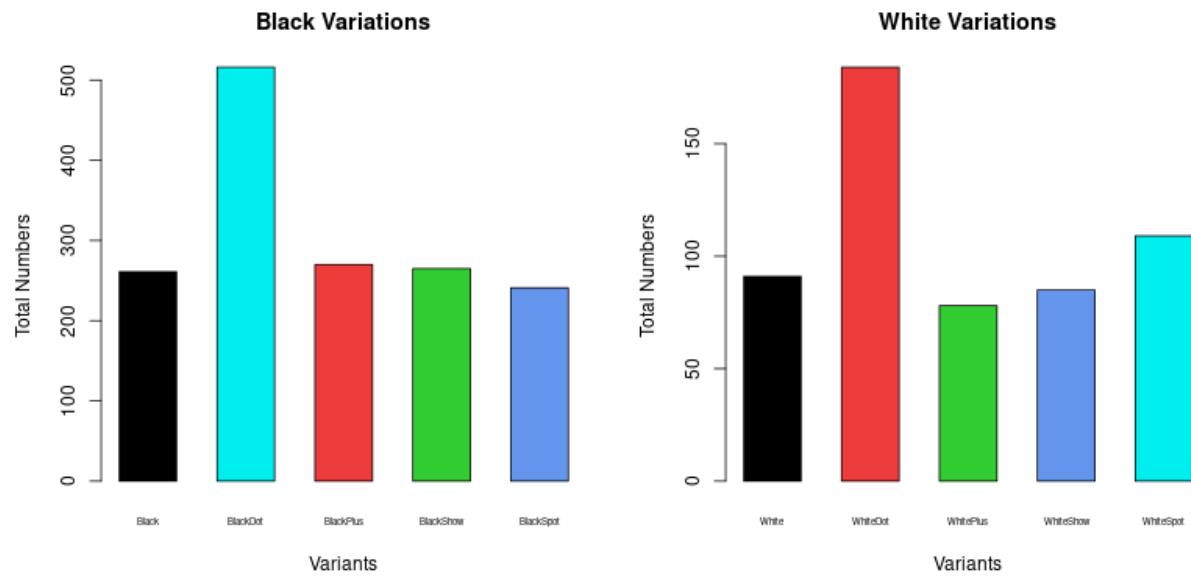
```
png("Black_White_Plot.png")
```

```
png(filename = "/cloud/project/RWorksheets#4B/Black_White_Plot.png", width = 600, height = 400)
```

```
dev.copy(png, "Black_White_Plot.png")
```

```
## png  
## 6
```

```
knitr::include_graphics("/cloud/project/RWorksheets#4B/Black_White_Plot.png")
```



```
dev.off()
```

```
## pdf  
## 2
```