Worksheet 4b in R

Erl Syron R. Espadon

#Using Loop Function: for() loop #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. Hint: Use abs() function to get the absolute value

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

# Create a 5x5 zero matrix
matrix <- matrix(0, nrow = 5, ncol = 5)

# Use for loop to populate the matrix
for (i in 1:5) {
  for (j in 1:5) {
    matrix[i, j] <- abs(vectorA[i] - vectorA[j])
  }
}
print(matrix)
```

```
##      [,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.

```
# Loop for each row
for (i in 1:5) {
  for (j in 1:i) {
    cat("* ")
  }
  cat("\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.

```
# Function to generate the Fibonacci sequence
fibonacci_sequence <- function(start_num) {
  # Initialize the sequence with the starting number
```

```
  a <- 0
  b <- 1
  c <- start_num  # Initialize 'c' to the starting number

  # Print the starting number
  cat(c, " ")

  # Loop to generate the sequence
  while (TRUE) {  # Loop indefinitely until break condition is met
    # Calculate the next Fibonacci number
    c <- a + b

    # Update the sequence variables (important order!)
    a <- b  # Update 'a' first
    b <- c  # Then update 'b'

    # Print the current Fibonacci number
    cat(c, " ")

    # Break the loop if the current number exceeds 500
    if (c >= 500) {  # Check before printing the next number
      break  # Exit the loop
    }
  }
}

# Get user input for the starting number
start_num <- as.numeric(readline("Enter the starting number for the Fibonacci sequence: "))
```

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

```
# Call the function to generate the sequence
fibonacci_sequence(start_num)
```

```
## NA  1  2  3  5  8  13  21  34  55  89  144  233  377  610
```

#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

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.4.2
```

```
shoes <- read_excel("C:\\WORKSHEETS\\Worksheet4a\\shoes.xlsx")

head(shoes)
```

```
## # A tibble: 6 x 3
##   Shoe_size Height Gender
##       <dbl>  <dbl> <chr>
## 1       6.5   66   F
## 2       9     68   F
## 3       8.5   64.5 F
## 4       8.5   65   F
## 5      10.5   70   M
## 6       7     64   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.

```
female_shoes <- shoes[shoes$Gender == "F", ]
male_shoes <- shoes[shoes$Gender == "M", ]

num_females <- nrow(female_shoes)
num_males <- nrow(male_shoes)

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

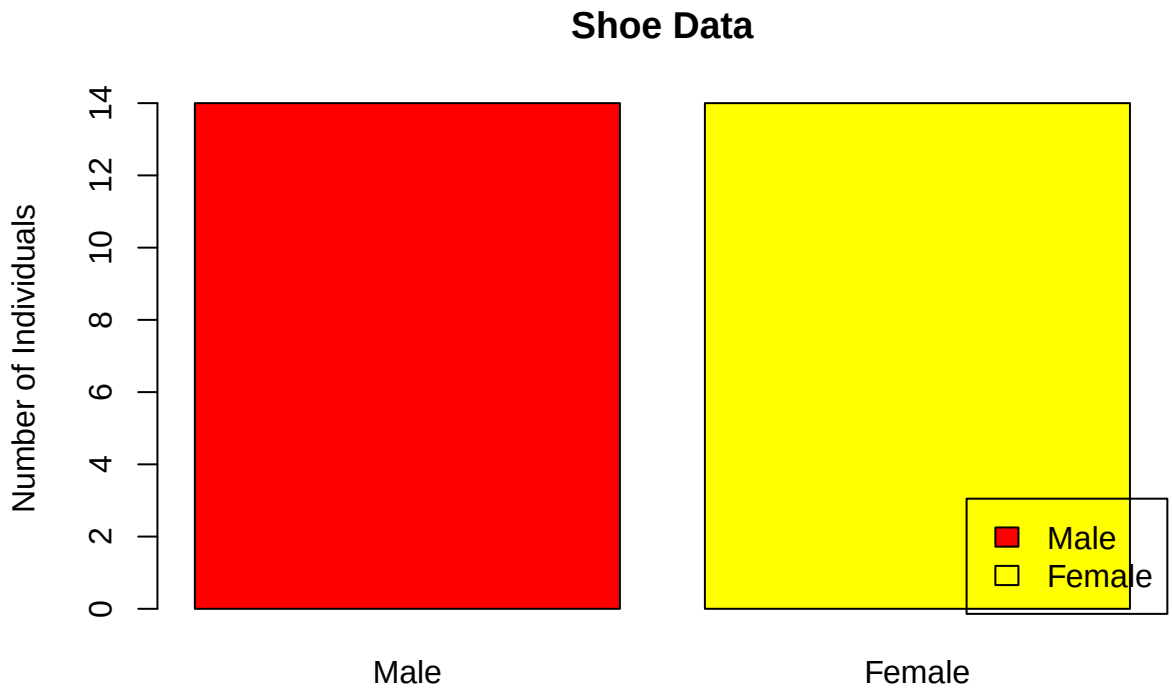
```
## Number of females: 14
```

```
cat("Number of males:", num_males)
```

```
## Number of males: 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.

```
barplot(c(num_males, num_females),
        names.arg = c("Male", "Female"),
        main = "Shoe Data",
        ylab = "Number of Individuals",
        col = c("red", "yellow"),
        legend.text = c("Male", "Female"),
        args.legend = list(x = "bottomright"))
```



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

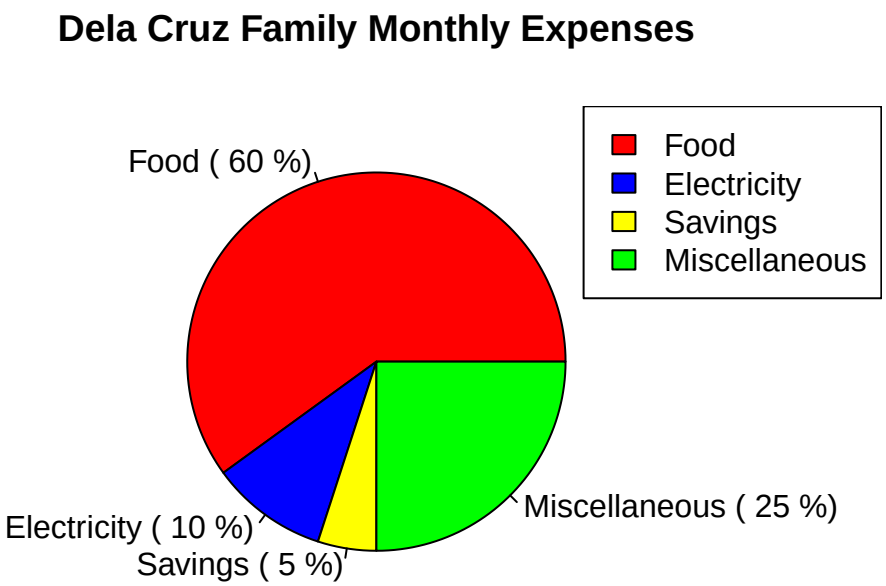
Miscellaneous-25 #.a. Create a piechart that will include labels in percentage.Add some colors and title of the chart. Write the R scripts and show its output.

```r
# data frame
expenses <- data.frame(
  Category = c("Food", "Electricity", "Savings", "Miscellaneous"),
  Amount = c(60, 10, 5, 25)
)

# Calculate percentages
expenses$Percentage <- round(expenses$Amount / sum(expenses$Amount) * 100, 2)

# pie chart
pie(expenses$Amount,
    labels = paste(expenses$Category, "(", expenses$Percentage, "%)"),
    main = "Dela Cruz Family Monthly Expenses",
    col = c("red", "blue", "yellow", "green"))

#legend
legend("topright", legend = expenses$Category, fill = c("red", "blue", "yellow", "green"))
```



#6. Use the iris dataset.

```r
data(iris)
```

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

```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 ...
```

#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?

```r
means <- c(mean(iris$Sepal.Length),
           mean(iris$Sepal.Width),
           mean(iris$Petal.Length),
           mean(iris$Petal.Width))

names(means) <- c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width")

print(means)
```

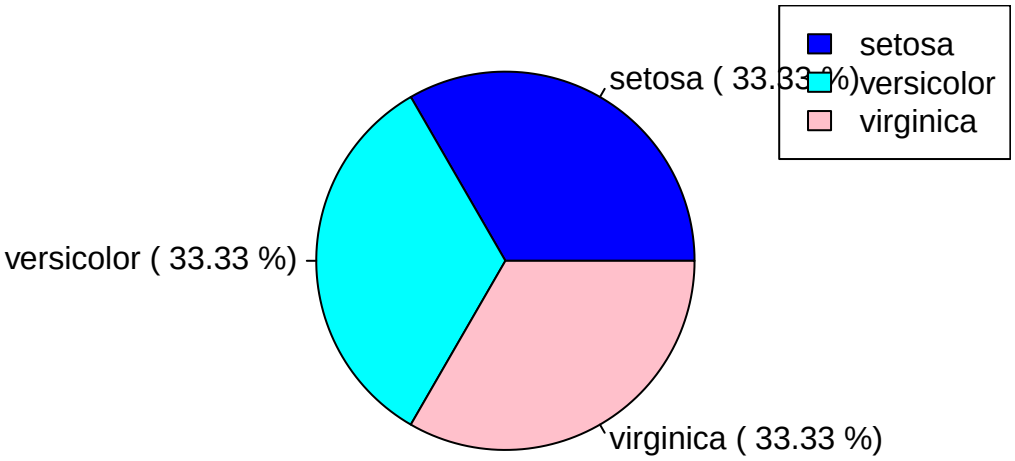
```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

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

```r
species_counts <- table(iris$Species)

pie(species_counts,
    labels = paste(names(species_counts), "(", round(species_counts / sum(species_counts) * 100, 2), "%
    main = "Iris Species Distribution",
    col = c("blue", "cyan", "pink"))

legend("topright", legend = names(species_counts), fill = c("blue", "cyan", "pink"))
```



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

```
setosa <- iris[iris$Species == "setosa", ]
versicolor <- iris[iris$Species == "versicolor", ]
virginica <- iris[iris$Species == "virginica", ]

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
```

```
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
```

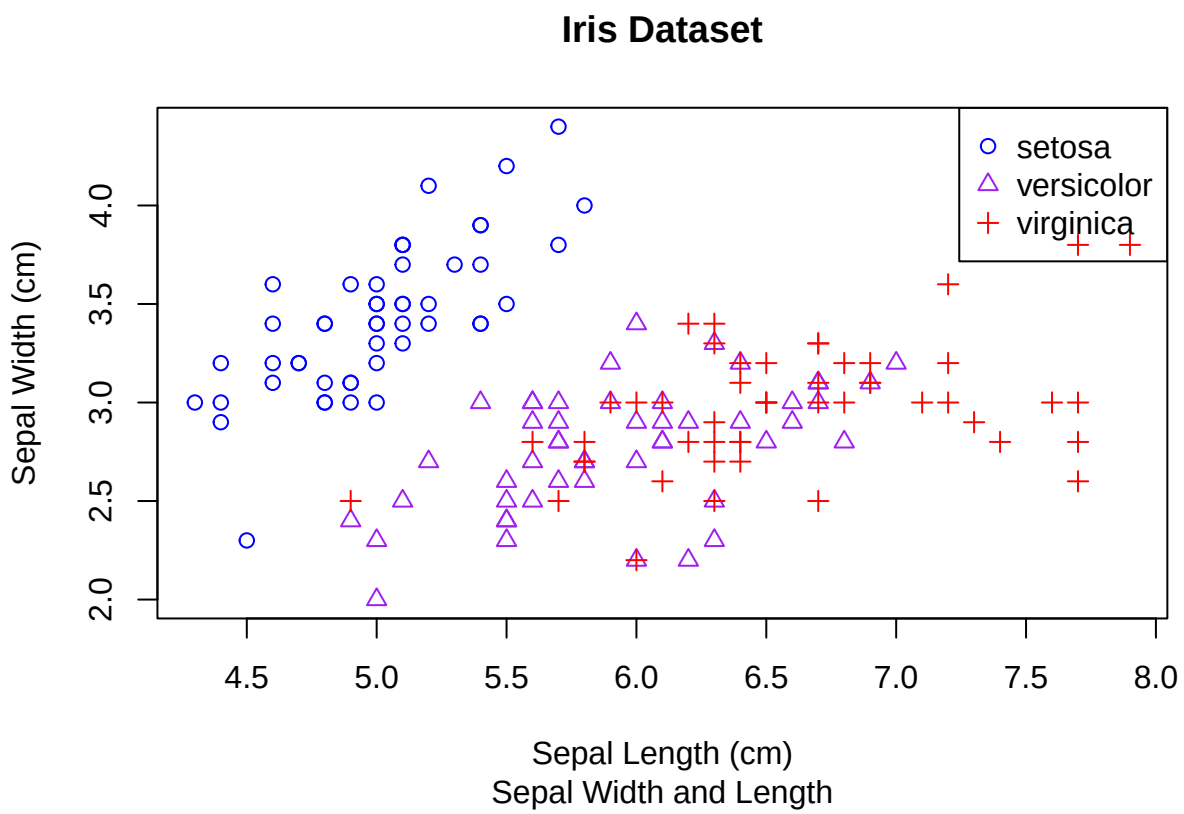
```
tail(virginica, 6)
```

```
##     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
```

#e. Scatterplot with Species Differentiation

```r
iris$Species <- as.factor(iris$Species)
plot(iris$Sepal.Length, iris$Sepal.Width,
     main = "Iris Dataset",
     sub = "Sepal Width and Length",
     xlab = "Sepal Length (cm)",
     ylab = "Sepal Width (cm)",
     pch = as.numeric(iris$Species),
     col = c("blue", "purple", "red")[iris$Species])

legend("topright", legend = levels(iris$Species),
       pch = 1:3, col = c("blue", "purple", "red"))
```



#f. Interpret the result.

```
"Species Distribution: The pie chart shows that the three species are roughly equally distributed in th
Sepal Length and Width: The scatterplot reveals some trends:
Setosa: Setosa species tend to have smaller sepal lengths and widths compared to the other two species.
Versicolor: Versicolor species fall in the middle range for both sepal length and width.
```

```
Virginica: Virginica species generally have the largest sepal lengths and widths."
```

```
## [1] "Species Distribution: The pie chart shows that the three species are roughly equally distributed
```

#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_file <- read_excel("C:/Users/Acer/Downloads/alexa_file.xlsx")
View(alexa_file)
```

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

```
# Rename white variants
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)

# Rename black variants
alexa_file$variation <- gsub("Black Dot", "Black_Dot", alexa_file$variation)
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)

# Example output (snippet)
head(alexa_file)
```

```
## # A tibble: 6 x 5
##   rating date                variation           verified_reviews       feedback
##    <dbl> <dttm>              <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 playi~        1
## 4      5 2018-07-31 00:00:00 Charcoal Fabric     I have had a lot of f~        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 a~        1
```

#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 4.4.2
```

```
##
## 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
```

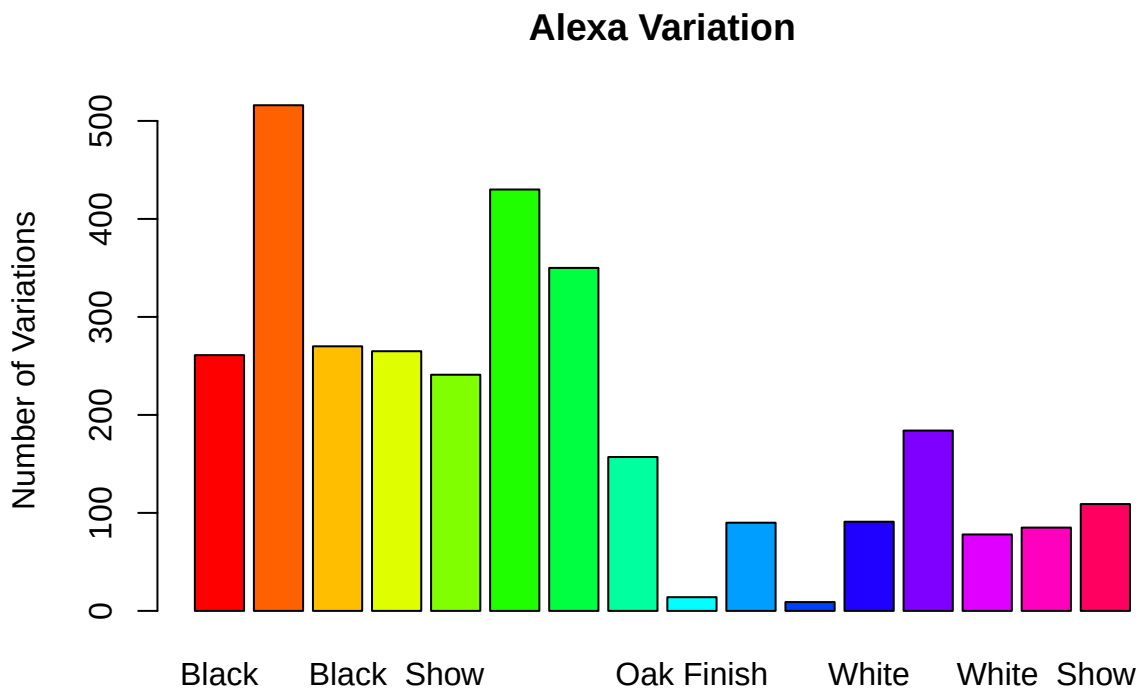
```
variations <- alexa_file %>%
  count(variation)
```

```r
save(variations, 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.

```r
load("variations.RData")

barplot(variations$n,
        names.arg = variations$variation,
        main = "Alexa Variation",
        ylab = "Number of Variations",
        col = rainbow(length(variations$variation)))
```



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

```r
# Separate black and white variations
black_variations <- variations[grep("Black", variations$variation), ]
white_variations <- variations[grep("White", variations$variation), ]

# Plot side by side
barplot(cbind(black_variations$n, white_variations$n),
        beside = TRUE,
        names.arg = c(black_variations$variation, white_variations$variation),
        main = "Black vs. White Alexa Variations",
        ylab = "Number of Variations",
        col = c("black", "white"),
```

```
        legend.text = c("Black", "White"),
        args.legend = list(x = "topright"))
```