RWorksheet_Ulgasan#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)

cat("Initial Matrix:\n")
```

```
## Initial Matrix:
```

```
print(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) {
  matrixA[i, ] <- vectorA * i
}


cat("\nFinal Matrix:\n")
```

```
##
## Final Matrix:
```

```
matrixA
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    2    4    6    8   10
## [3,]    3    6    9   12   15
## [4,]    4    8   12   16   20
## [5,]    5   10   15   20   25
```

2.2. Print the string "*" using for() function. The output should be the same as shown in Figure

```
for(i in 1:5){
  asterisk <- rep("*", i)
  print(asterisk)

}
```

```
## [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.

```
fibonacci <- function(n) {
 if (n <= 1)
    return(n)
 else
    return(fibonacci(n-1) + fibonacci(n-2))
}


cat("Enter a number: ")
```

```
## Enter a number:
```

```
input <- readLines(n=1)


input_num <- as.numeric(input)


if (is.na(input_num)) {
 cat("Invalid input. Please enter a valid number.")
} else {

 cat("Fibonacci sequence: \n")
 i <- 1
 while (i <= 500) {
    if (i < input_num) {
      i <- i + 1
    } else {
      cat(fibonacci(i), "\n")
      i <- i + 1
     break
    }
 }
}
```

```
## Invalid input. Please enter a valid number.
```

4.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

```
lastData <- read.csv("householdData.csv")
head(lastData)
```

```
##   X ShoeSize 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
```

```
getwd()
```

```
## [1] "/cloud/project/rWorksheet_ulgasan4a"
```

4.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_data <- subset(lastData, lastData$Gender == "M")
female_data <- subset(lastData, lastData$Gender == "F")


nrow(male_data)
```

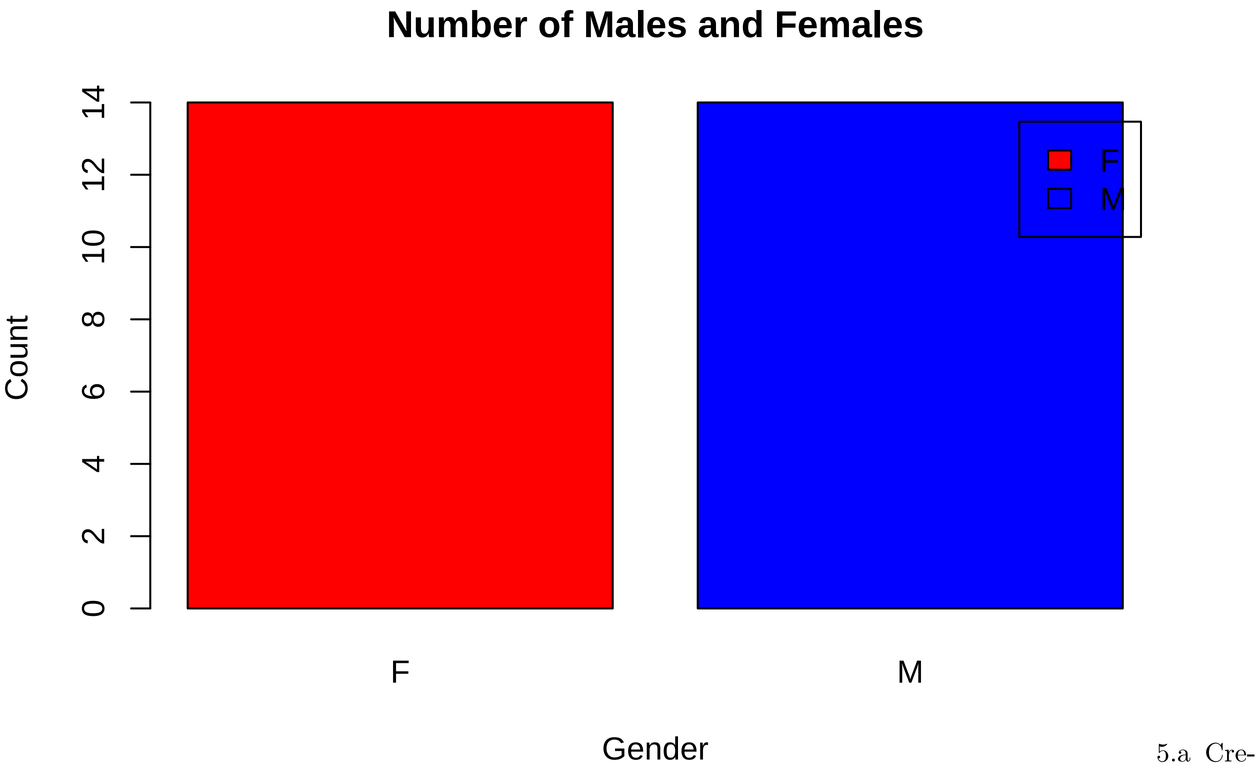
```
## [1] 14
```

```
nrow(female_data)
```

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

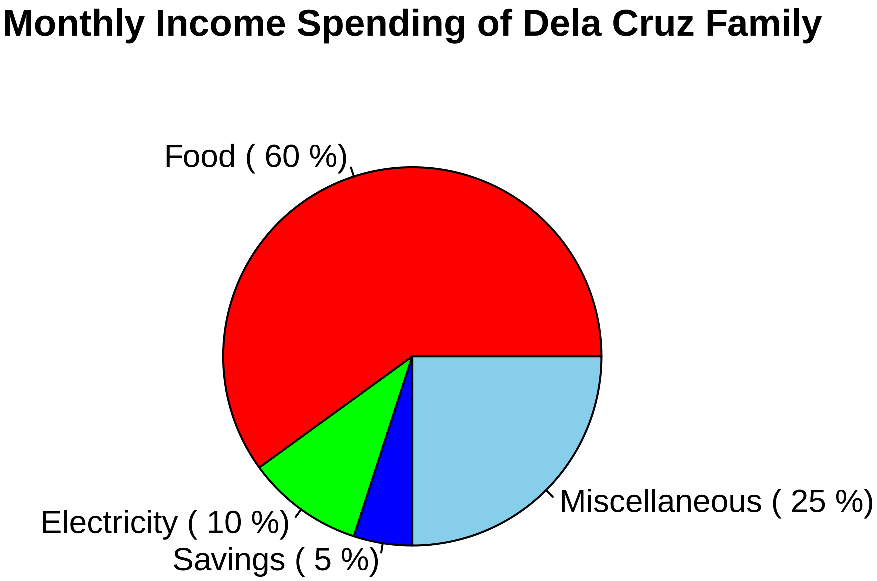
4.e Create a graph for the number of males and females for Household Data. Use plot(), chart type = barplot. Make sure to place title, legends, and colors. Write the R scripts and its result.

```
totalmf <- table(lastData$Gender)
barplot(totalmf,
        main = "Number of Males and Females",
        xlab = "Gender",
        ylab = "Count",
        col = c("red", "blue"),
        legend.text = rownames(totalmf),
        beside = TRUE)
```



5.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.

```
spending_data <- data.frame(
  Category = c("Food", "Electricity", "Savings", "Miscellaneous"),
  Value = c(60, 10, 5, 25)
)

spending_data$Percentage <- spending_data$Value / sum(spending_data$Value) * 100

colors <- c("red", "green", "blue", "skyblue")

pie(spending_data$Value,
    labels = paste(spending_data$Category,"(",spending_data$Percentage,"%)"),
    col = colors,
    main = "Monthly Income Spending of Dela Cruz Family")
```



6a. Check for the structure of the dataset using the str() function. Describe what you have seen in the output.

```
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 ...
```

```
#Information about iris blossoms is gathered in the iris dataset. It contains measurements for 150 dist
```

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?

```
meanOfiris <-colMeans(iris[,1:4])
meanOfiris
```

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

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

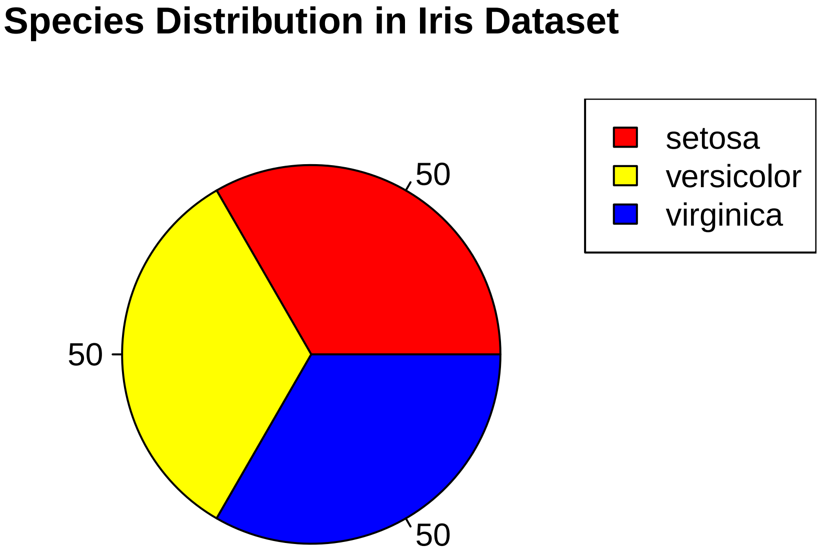
```
speciesOfiris <- table(iris$Species)
speciesOfiris
```

```
##
##     setosa versicolor  virginica
##         50         50         50
```

```
nameOfSpecies <- c("Setosa", "Versicolor","Vriginica")
nameOfSpecies
```

```
## [1] "Setosa"     "Versicolor" "Vriginica"
```

```
pie(speciesOfiris,
   labels = speciesOfiris,
   col = c("red","yellow","blue"),
   main = "Species Distribution in Iris Dataset")

legend("topright",
       legend = levels(iris$Species),
       fill = c("red","yellow","blue"),)
```


Species Distribution in Iris Dataset

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

```
subSetosa <- subset(iris, Species =="setosa")

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

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

tail(subSetosa)
```

```
##    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(subVer)
```

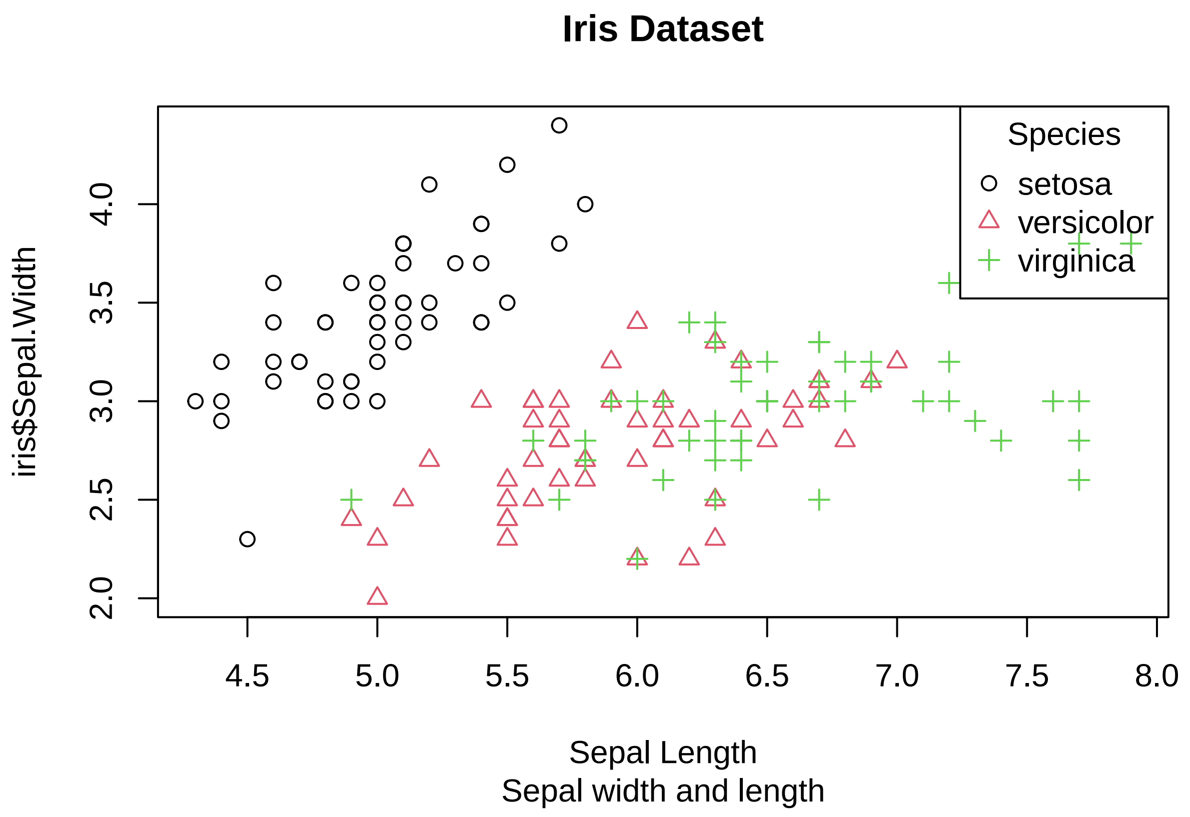
```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 95           5.6         2.7          4.2         1.3 versicolor
## 96           5.7         3.0          4.2         1.2 versicolor
## 97           5.7         2.9          4.2         1.3 versicolor
## 98           6.2     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

```r
tail(subVirg)
```

```
##     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.

```r
plot(iris$Sepal.Length,
iris$Sepal.Width,
xlab = "Sepal Length",
main = "Iris Dataset", sub = "Sepal width and length",
pch = as.numeric(iris$Species),
col = as.numeric(iris$Species))


legend("topright",
levels(iris$Species),
pch = 1:4,
col = 1:4,
title = "Species")
```



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

```
library(readxl)


alexa_file <- read_excel("alexa_file.xlsx")
alexa_file
```

```
## # A tibble: 3,150 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 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
```

```
alexa_file$variation <- gsub("Black  Dot", "BlackDot", alexa_file$variation)
alexa_file$variation <- gsub("Black  Plus", "BlackPlus", alexa_file$variation)
alexa_file$variation <- gsub("Black  Show", "BlackShow", alexa_file$variation)
alexa_file$variation <- gsub("White  Plus", "WhitePlus", alexa_file$variation)
```

```r
alexa_file$variation <- gsub("White  Show", "WhiteShow", alexa_file$variation)
alexa_file$variation <- gsub("White  Spot", "WhiteSpot", alexa_file$variation)
alexa_file
```

```
## # A tibble: 3,150 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 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
```

7.bGet 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?

```r
# Load required library
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
```

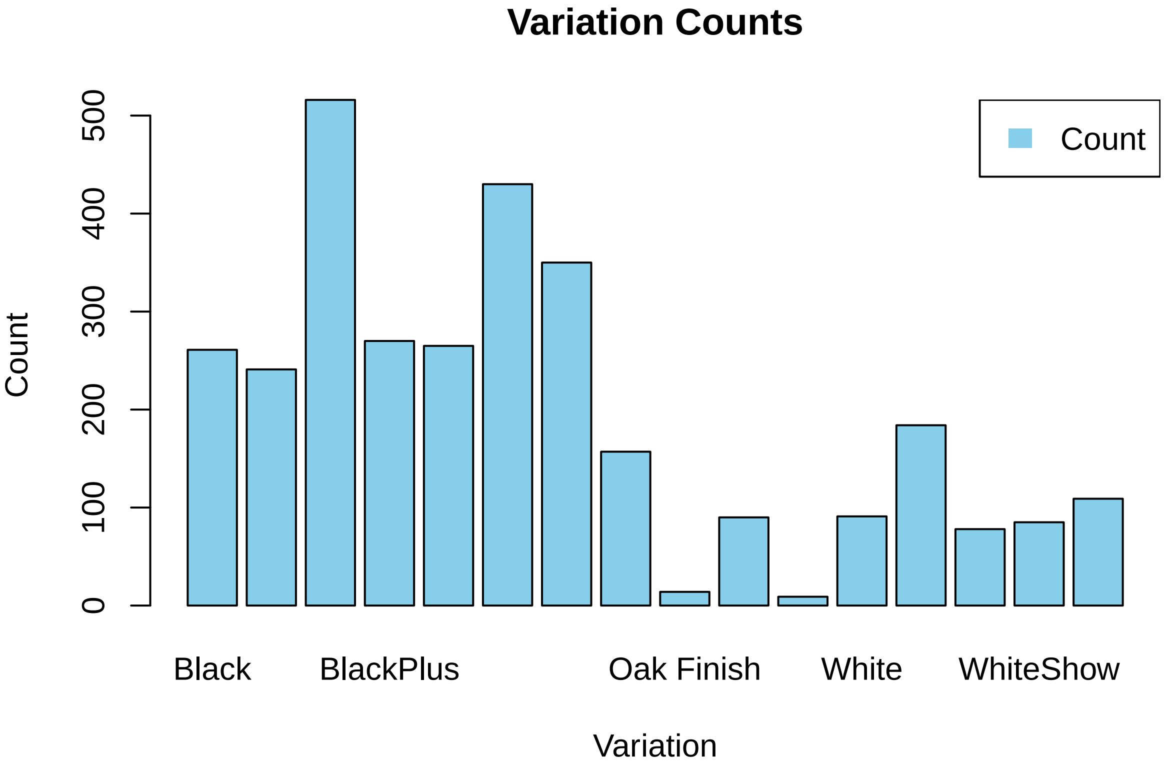
```r
#
variation_counts <- alexa_file %>%
  count(alexa_file$variation)

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


print(variation_counts)
```

```
## # A tibble: 16 x 2
##    `alexa_file$variation`           n
##    <chr>                        <int>
##  1 Black                          261
##  2 Black  Spot                    241
##  3 BlackDot                       516
##  4 BlackPlus                      270
##  5 BlackShow                      265
##  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 WhitePlus                       78
## 15 WhiteShow                       85
## 16 WhiteSpot                      109
```

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

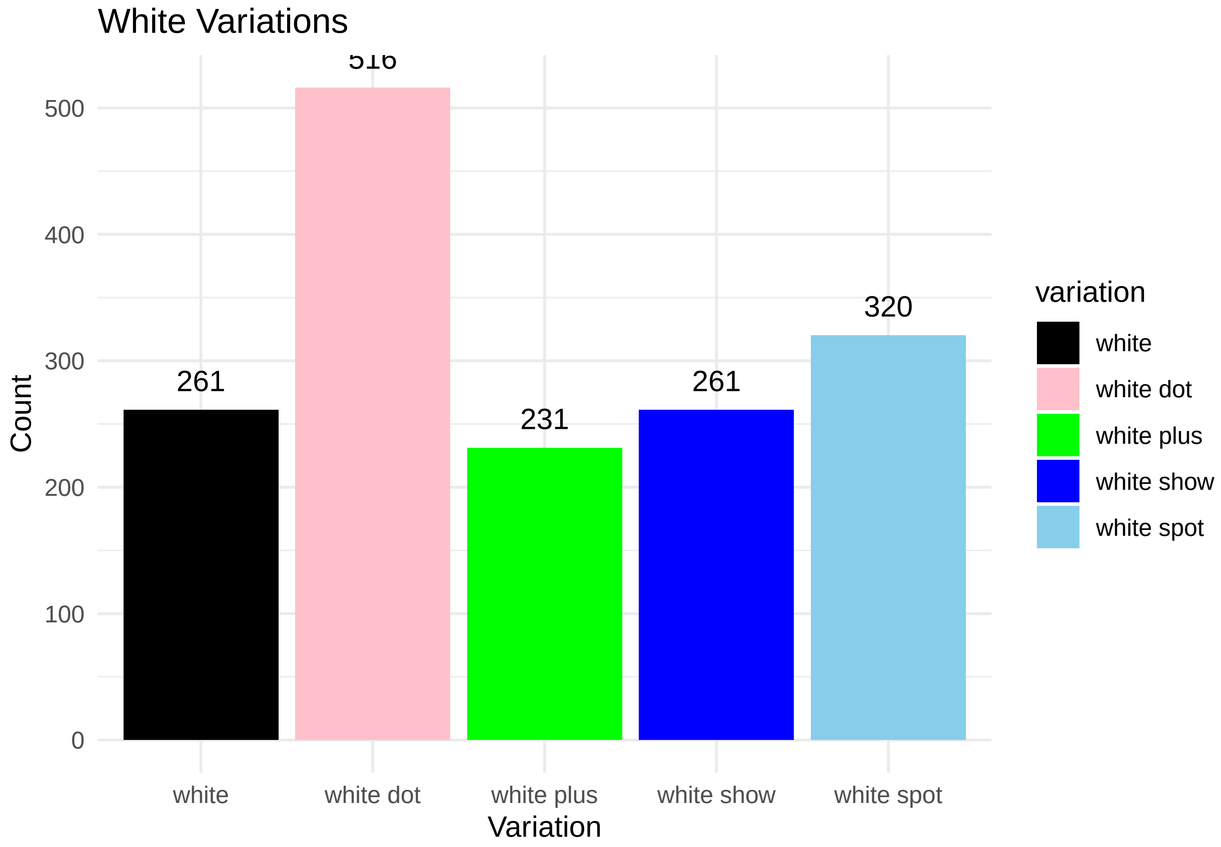
7c. From the variations.RData, create a barplot(). Complete the details of the chart which include the title, color, labels of each bar.

```
barplot(variation_counts$n,
        names.arg = variation_counts$`alexa_file$variation`,
        col = "skyblue",
        main = "Variation Counts",
        xlab = "Variation",
        ylab = "Count")


legend("topright", legend = "Count", fill = "skyblue", border = NA)
```

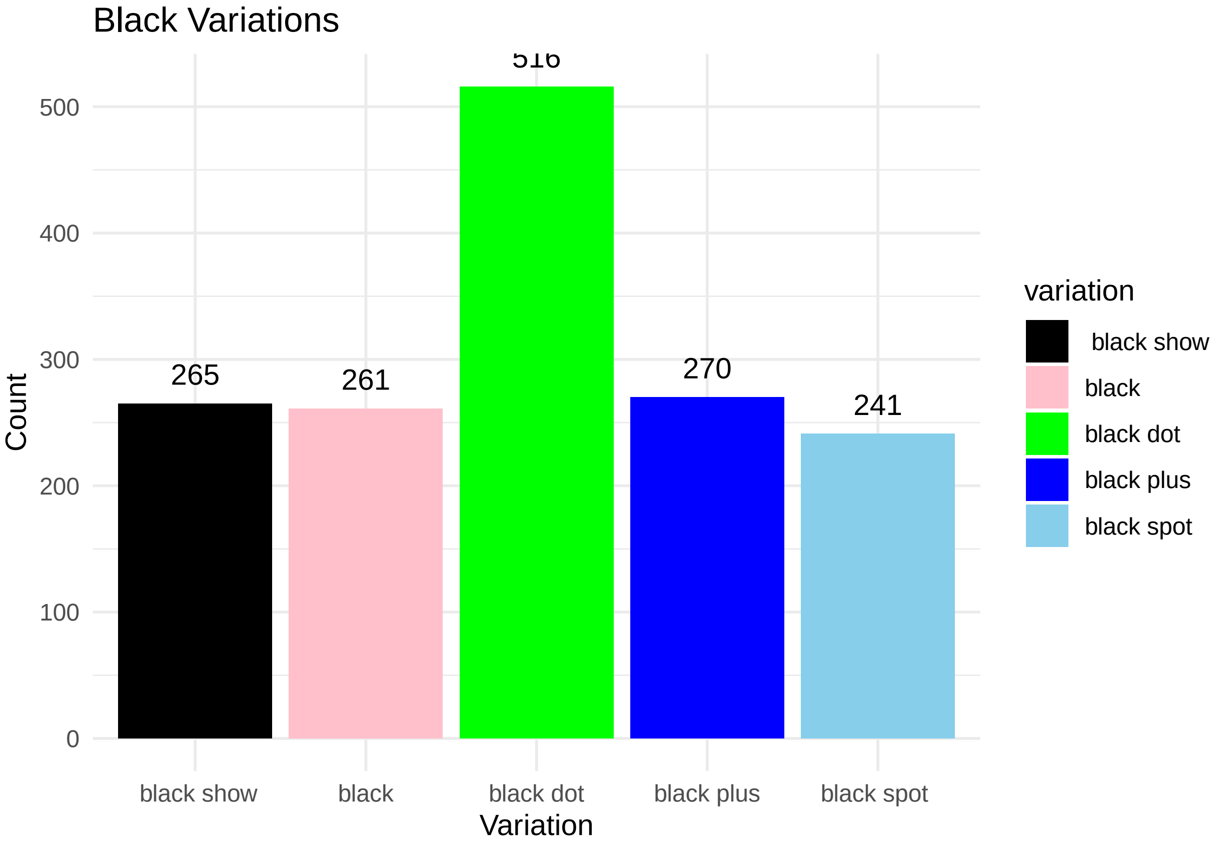


7d. Create a barplot() for the black and white variations. Plot it in 1 frame, side by side. Complete the details of the chart.

```
library(ggplot2)


par(mfrow = c(1,2))
black <- data.frame(
 variation = c("black", "black plus", " black show","black spot ", "black dot" ),
 n = c(261, 270, 265, 241, 516)
)


barplot_black <- ggplot(black, aes(x = variation, y = n, fill = variation)) +
 geom_bar(stat = "identity", position = "dodge") +
 geom_text(aes(label = n), position = position_dodge(width = 0.5), vjust = -0.9) +
 scale_fill_manual(values = c("black", "pink","green","blue", "skyblue")) +
 theme_minimal() +
 labs(title = "Black Variations", x = "Variation", y = "Count")


white <- data.frame(
  variation = c("white", "white dot", "white plus", "white show", "white spot"),
  n = c(261, 516, 231, 261,320 )
)

barplot_white <- ggplot(white, aes(x = variation, y = n, fill = variation)) +
 geom_bar(stat = "identity", position = "dodge") +
 geom_text(aes(label = n), position = position_dodge(width = 0.5), vjust = -0.9) +
 scale_fill_manual(values = c("black", "pink","green","blue", "skyblue")) +
 theme_minimal() +
 labs(title = "White Variations", x = "Variation", y = "Count")

  barplot_white
```



```
barplot_black
```


Black Variations
Count
black show
black
black dot
black plus
black spot
Variation
variation
black show
black
black dot
black plus
black spot