

RWorksheet_Songaling#4b.Rmd

Charles Daniel Songaling

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),5,5)

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 1.

```
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.

```
Num <- as.numeric(readline("Enter a starting number to start Fibonacci sequence: "))

## Enter a starting number to start Fibonacci sequence:

if(is.na(Num || Num < 0)){
  cat("Please enter something")
}else{

  num_input<-Num
  a <- num_input
  b <- 0
```

```

cat("Fibonacci sequence starting from",Num,"up to 500\n")

repeat{

  num = a + b
  if (num > 500){
    break
  }
  cat(num, " ")
  a <- b
  b <- num
}
cat("\n")
}

```

Please enter something

#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.

```

shoesizesdata <- read.csv("ShoeSizes.csv")
shoesizesdata

```

```

##      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

```

#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.

```
Malesub <- subset(shoesizesdata, Gender == "M")
Femalesub <- subset(shoesizesdata, Gender == "F")

cat("Number of obsevation in male:",nrow(Malesub),"\n")

## Number of obsevation in male: 14

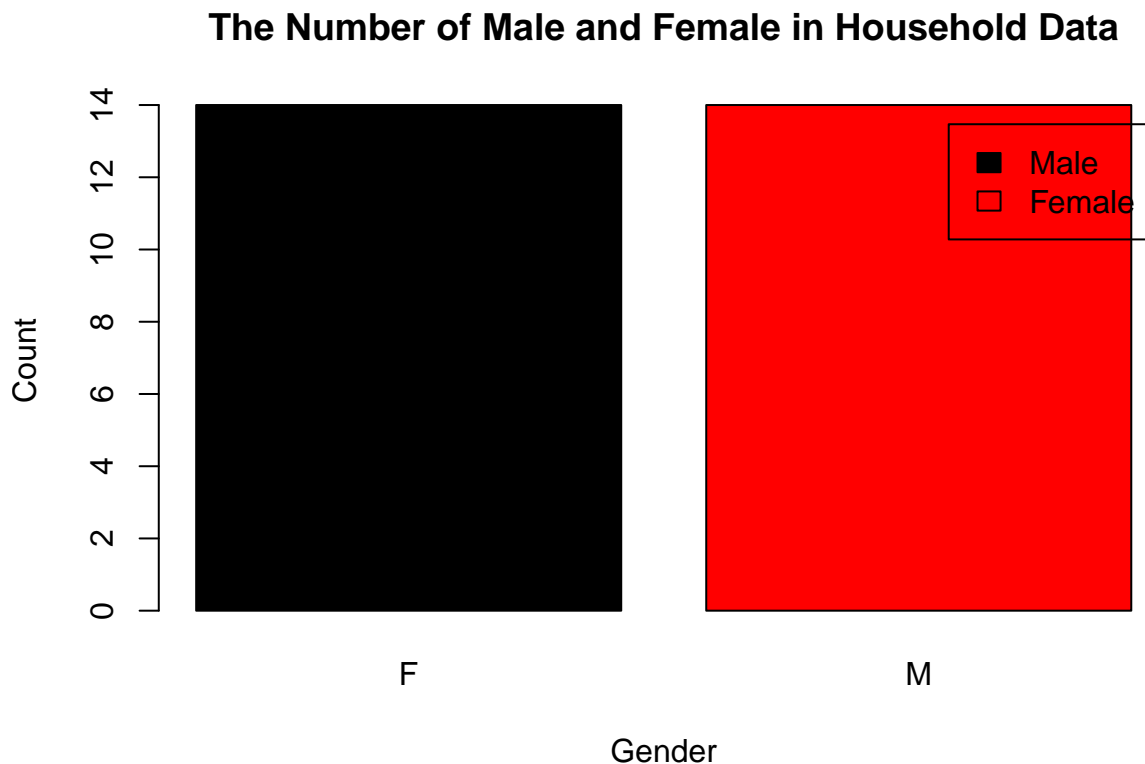
cat("Number of observation in female",nrow(Femalesub),"\n")

## Number of observation in female 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.

GendersMF <- table(shoesizesdata$Gender)

barplot(GendersMF,
        main = "The Number of Male and Female in Household Data",
        xlab = "Gender",
        ylab = "Count",
        col = c("black","red"),
        legend.text =c("Male", "Female"))
```



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

#5. The

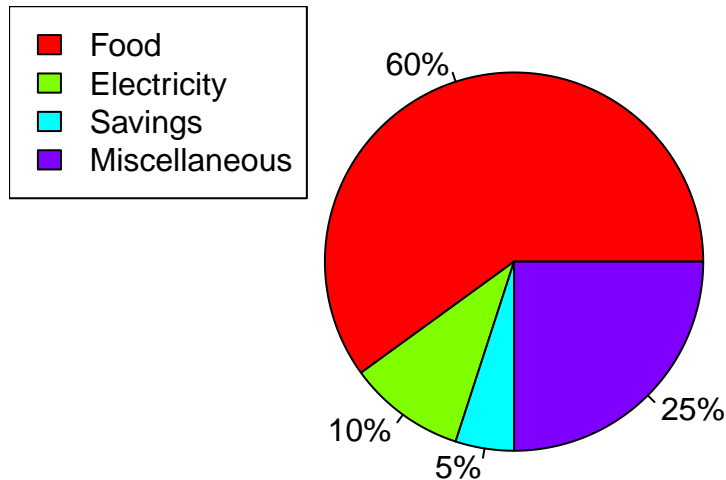
#a. Create a piechart that will include labels in percentage. Add some colors and title of the chart. Write the R scripts and show its output.

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

pie(monthlyincome_spent,labels = paste0(monthlyincome_spent,"%"),
    main = "Dela Cruz Family Expenses", col = rainbow(length(monthlyincome_spent)))
```

```
legend("topleft", legend = c("Food", "Electricity", "Savings", "Miscellaneous"),
      fill = rainbow(length(monthlyincome_spent)))
```

Dela Cruz Family Expenses



#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 ...
```

#There are over 150 obs. and 5 variables in the iris dataset. It also shows the numeric measurements in

#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?

```
data(iris)
```

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

```
irismean
```

```
## 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.

```
data(iris)
```

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

```
pie(iris_species, labels = names(iris_species),
    col = rainbow(length(iris_species)),
```

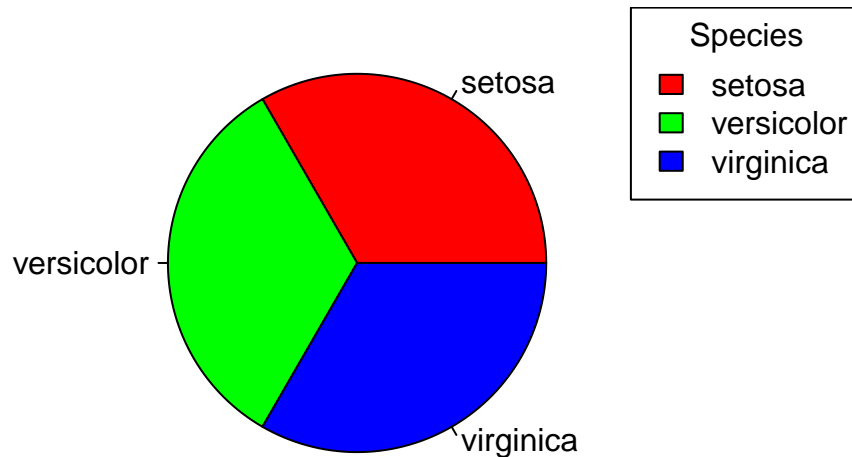
```

main = "Species Distribution")

legend("topright", legend = names(iris_species),
      fill = rainbow(length(iris_species)), title = "Species")

```

Species Distribution



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

```

data(iris)

setosasub <- subset(iris, Species == "setosa")
versicolsub <- subset(iris, Species == "versicolor")
virginicasub <- subset(iris, Species == "virginica")

#Displaying the last 6 rows of each species
tail(setosasub)

##      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(versicolsub)

##      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(virginicasub)

##      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.

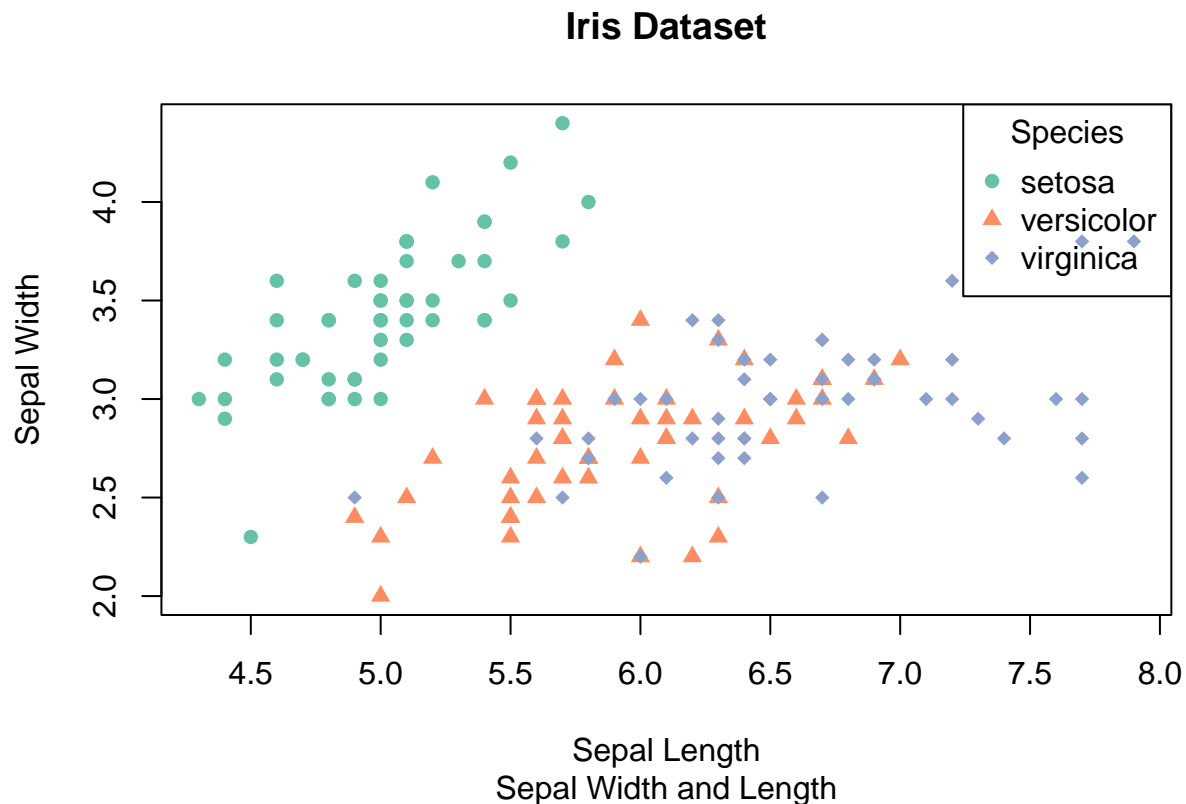
```
data(iris)

iris$Species <- as.factor(iris$Species)

colors <- c("setosa" = "#66c2a5", "versicolor" = "#fc8d62", "virginica" = "#8da0cb")
symbols <- c("setosa" = 16, "versicolor" = 17, "virginica" = 18)

plot(iris$Sepal.Length, iris$Sepal.Width,
     col = colors[iris$Species],
     pch = symbols[iris$Species],
     main = "Iris Dataset",
     sub = "Sepal Width and Length",
     xlab = "Sepal Length",
     ylab = "Sepal Width")

legend("topright", legend = levels(iris$Species), col = colors, pch = symbols, title = "Species")
```



#f. Interpret the result.

#By factoring the species, it will represent as categories in R.

#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, BlackSpot). Also on the white variants (White Dot, White Plus, White Show, WhiteSpot).

```
library(readxl)

alexafile <- read_excel("alexa_file.xlsx")

alexafile

## # 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
```

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

```
#black
alexafile$variation <- gsub("Black Dot", "BlackDot", alexafile$variation)
alexafile$variation <- gsub("Black Plus", "BlackPlus", alexafile$variation)
alexafile$variation <- gsub("Black Show", "BlackShow", alexafile$variation)
alexafile$variation <- gsub("Black Spot", "BlackSpot", alexafile$variation)

#white
alexafile$variation <- gsub("White Dot", "WhiteDot", alexafile$variation)
alexafile$variation <- gsub("White Plus", "WhitePlus", alexafile$variation)
alexafile$variation <- gsub("White Show", "WhiteShow", alexafile$variation)
alexafile$variation <- gsub("White Spot", "WhiteSpot", alexafile$variation)

alexafile
```

```
## # 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
```

#b. Get the total number of each variations and save it into another object. Save the object as varia-

tions.RData. Write the R scripts. What is its result?

```
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

Total_var <- alexafile %>%
  count(alexafile$variation)

Total_var

## # A tibble: 16 x 2
##   `alexafile$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(Total_var, 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.

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

## # A tibble: 16 x 2
##   `alexafile$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
```

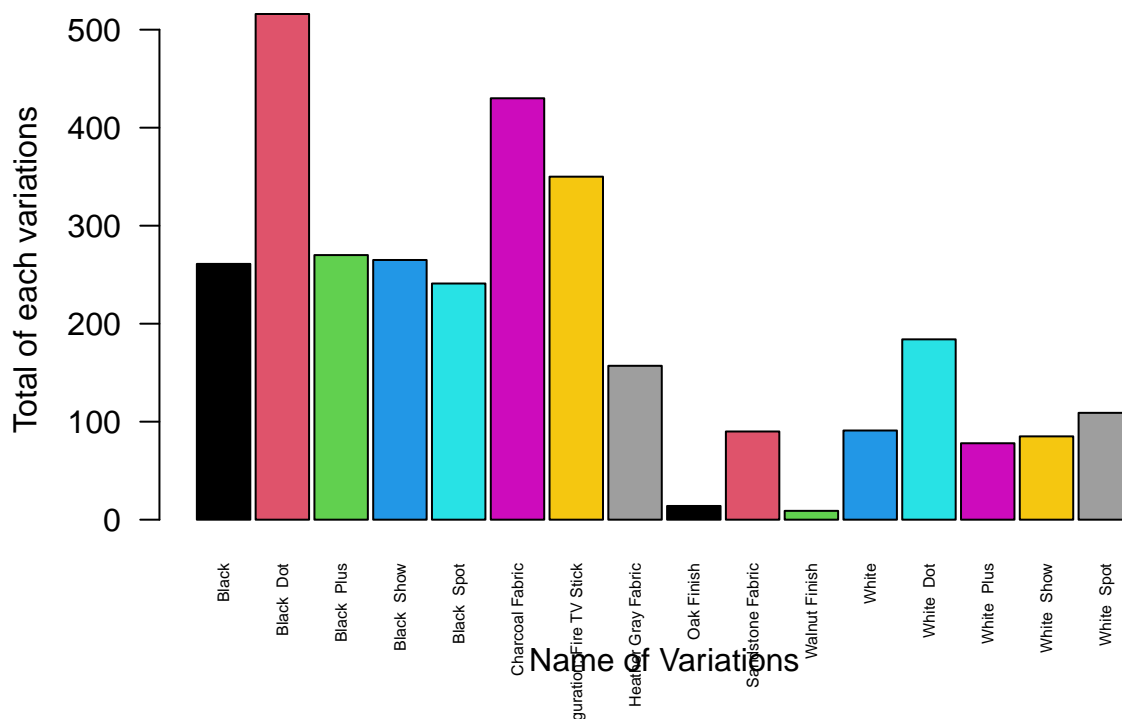


```
## 8 Heather Gray Fabric      157
## 9 Oak Finish               14
## 10 Sandstone Fabric       90
## 11 Walnut Finish          9
## 12 White                   91
## 13 White Dot              184
## 14 White Plus             78
## 15 White Show             85
## 16 White Spot            109
```

```
Name_var <- Total_var$`alexafile$variation`
```

```
Alexavar_plot <- barplot(Total_var$n,
  names.arg = Name_var,
  main = "Total number of variations",
  xlab = "Name of Variations",
  ylab = "Total of each variations",
  col = 1:16,
  space = 0.1,
  cex.names = 0.5,
  las = 2)
```

Total number of variations



```
png("Alexavar_plot.png")
dev.off()
```

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

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

```
library(RColorBrewer)

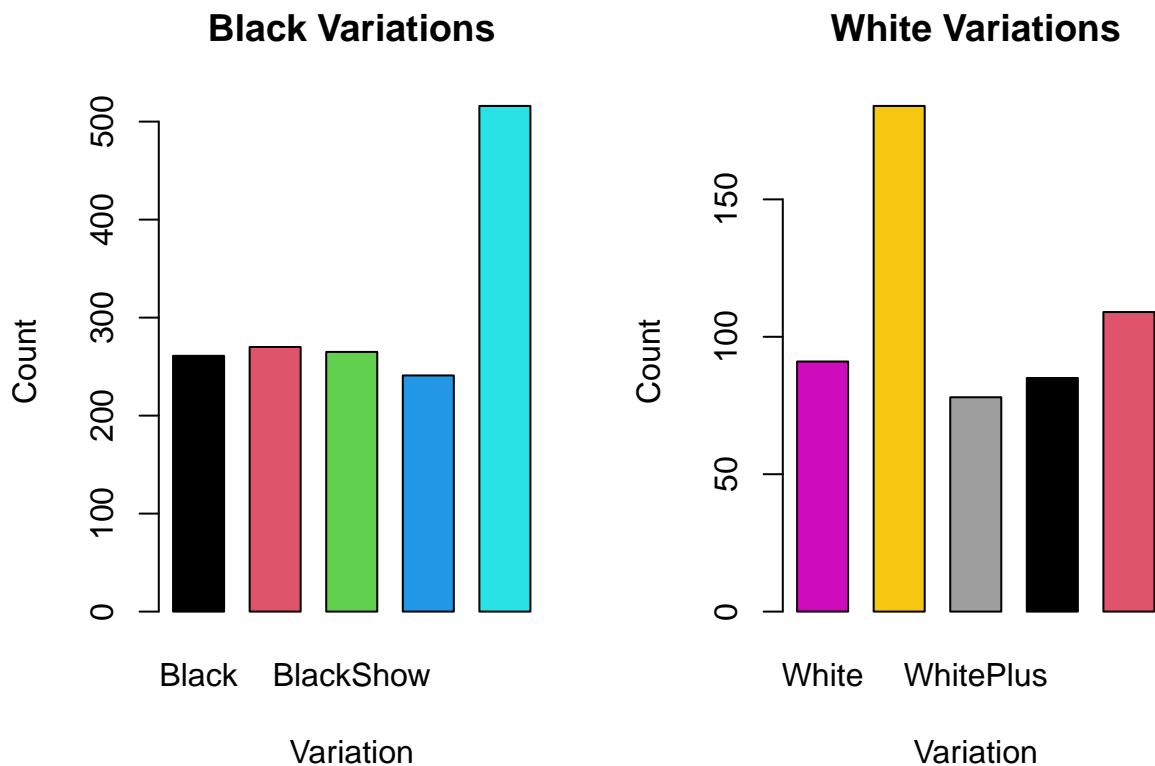
par(mfrow = c(1,2))

Blackvar_plot <- barplot(height = c(261,270,265,241,516),
                        names.arg = c("Black","BlackPlus","BlackShow","BlackSpot","BlackDot"), main = "Black Variations",
                        col = 1:5,
                        space = 0.5,
                        xlab = "Variation",
                        ylab = "Count")

png("Blackvar_plot.png")
dev.off()
```

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

```
Whiteplot <- barplot(height = c(91,184,78,85,109),
                    names.arg = c("White", "WhiteDot", "WhitePlus", "WhiteShow", "WhiteSpot"),
                    main = "White Variations",
                    space = 0.5,
                    col = 6:10,
                    xlab = "Variation",
                    ylab = "Count",)
```



```
png("Whiteplot.png")
dev.off()
```

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