RWorksheet_Canonicato#4b.

Dianah Marie Canonicato

2023-11-09

#1

```
#1
#Using Loop Function

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

```
#2 Print the string "*" using for() function.

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

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

#3

```
start_value <- as.integer(readline(prompt="Enter a positive integer: "))
```

```
## Enter a positive integer:
```

```
if (is.na(start_value) || start_value == 0) {
  cat("No input")
} else {
  a <- start_value
  b <- 0
```

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

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

  }
}
```

```
## No input
```

#Using Basic Graphics (plot(),barplot(),pie(),hist()) #4

```
#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?

library("readr")

accessData <- read_csv("/cloud/project/worksheet#4/shoes_size.csv")
```

```
## New names:
## Rows: 28 Columns: 4
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): Gender dbl (3): ...1, Shoe_Size, Height
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
head(accessData)
```

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

```
#b



malesub <- subset(accessData, Gender == "M")
malesub
```

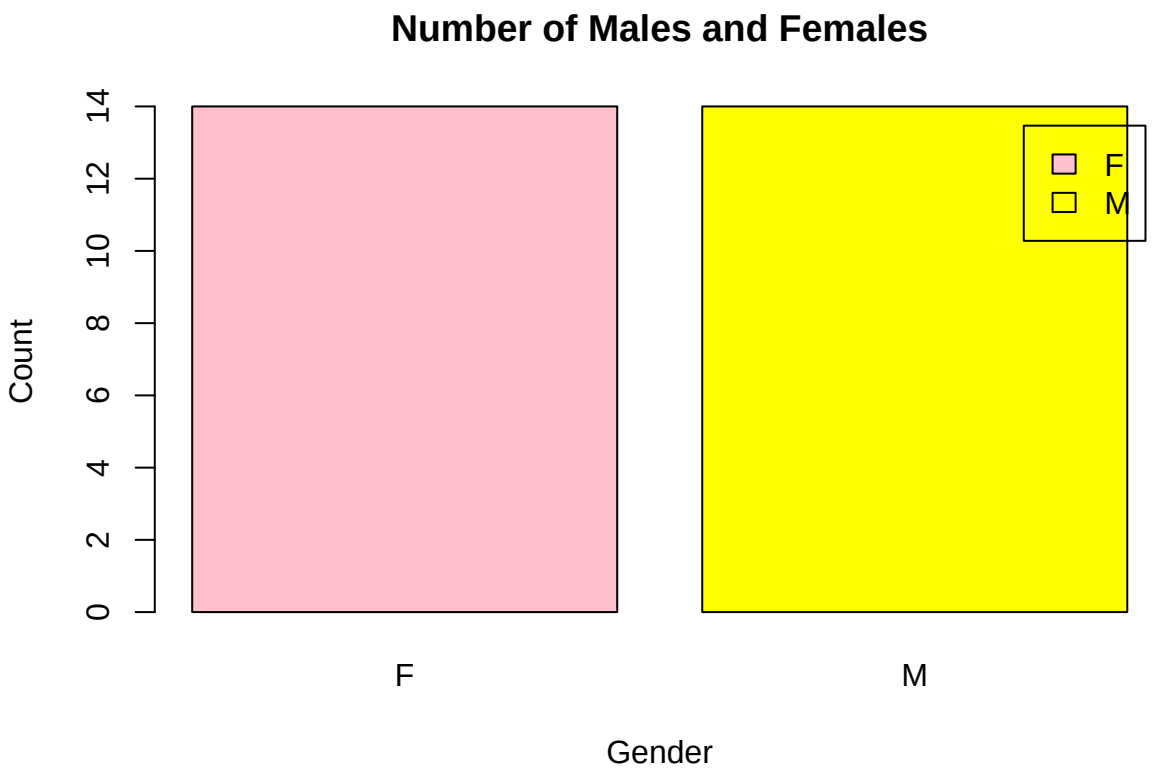
```
## # A tibble: 14 x 4
```

```
##     ...1 Shoe_Size Height Gender
##    <dbl>     <dbl>  <dbl> <chr>
##  1     5      10.5   70   M
##  2     9      13     72   M
##  3    11      10.5   74.5 M
##  4    13      12     71   M
##  5    14      10.5   71   M
##  6    15      13     77   M
##  7    16      11.5   72   M
##  8    19      10     72   M
##  9    22       8.5   67   M
## 10    23      10.5   73   M
## 11    25      10.5   72   M
## 12    26      11     70   M
## 13    27       9     69   M
## 14    28      13     70   M
```

```
femalesub <- subset(accessData, Gender == "F")
femalesub
```

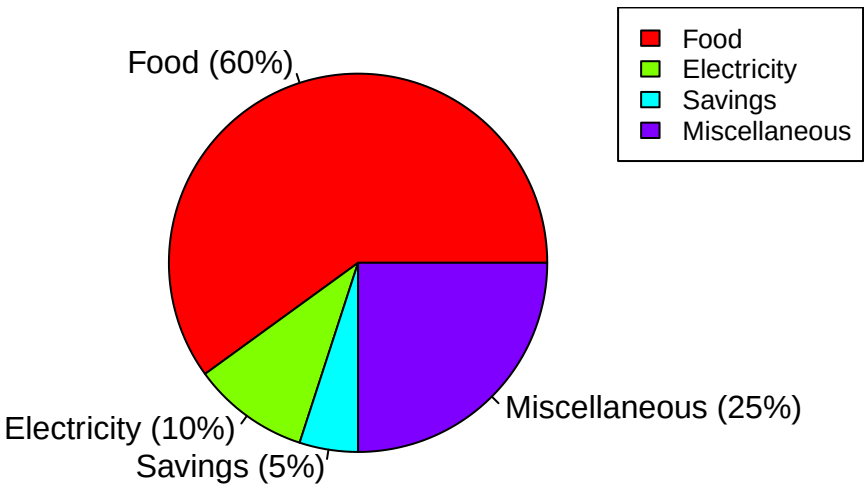
```
## # A tibble: 14 x 4
##     ...1 Shoe_Size Height Gender
##    <dbl>     <dbl>  <dbl> <chr>
##  1     1       6.5   66   F
##  2     2       9     68   F
##  3     3       8.5   64.5 F
##  4     4       8.5   65   F
##  5     6       7     64   F
##  6     7       9.5   70   F
##  7     8       9     71   F
##  8    10       7.5   64   F
##  9    12       8.5   67   F
## 10    17       8.5   59   F
## 11    18       5     62   F
## 12    20       6.5   66   F
## 13    21       7.5   64   F
## 14    24       8.5   69   F
```

```
#c

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



#5

```r
# 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)), ma

# Add a legend
legend("topright", names(data), cex = 0.8, fill = rainbow(length(data)))
```



#6

```
data(iris)

#6a
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 'Sepal.Length' column contains numerical values representing the length of the sepals of each flow
#The 'Sepal.Width' column contains numerical values representing the width of the sepals of each flower
#The 'Petal.Length' column contains numerical values representing the length of the petals of each flow
#The 'Petal.Width' column contains numerical values representing the width of the petals of each flower
#The 'Species' column contains categorical data, represented as a factor with 3 levels: "setosa", "vers
```

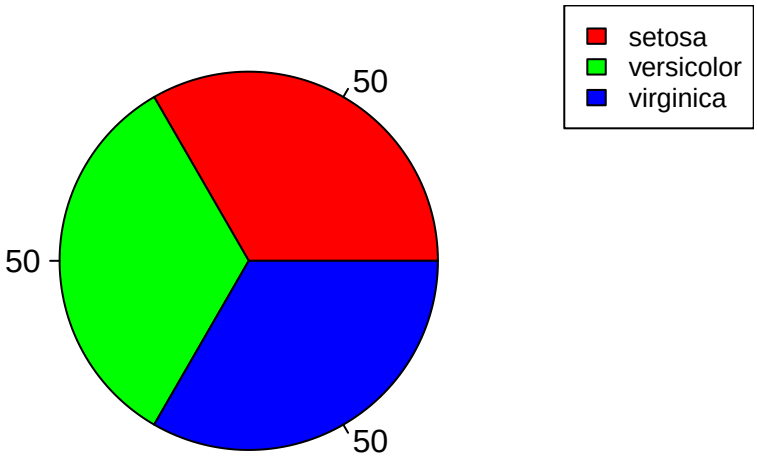
#b

```
#6b

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

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

#c

```
#6c

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



#d

```
#6d
# Subset the iris data set into the three species.
setosa_subset <- subset(iris, Species == "setosa")
versicolor_subset <- subset(iris, Species == "versicolor")
virginica_subset <- subset(iris, Species == "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)
```

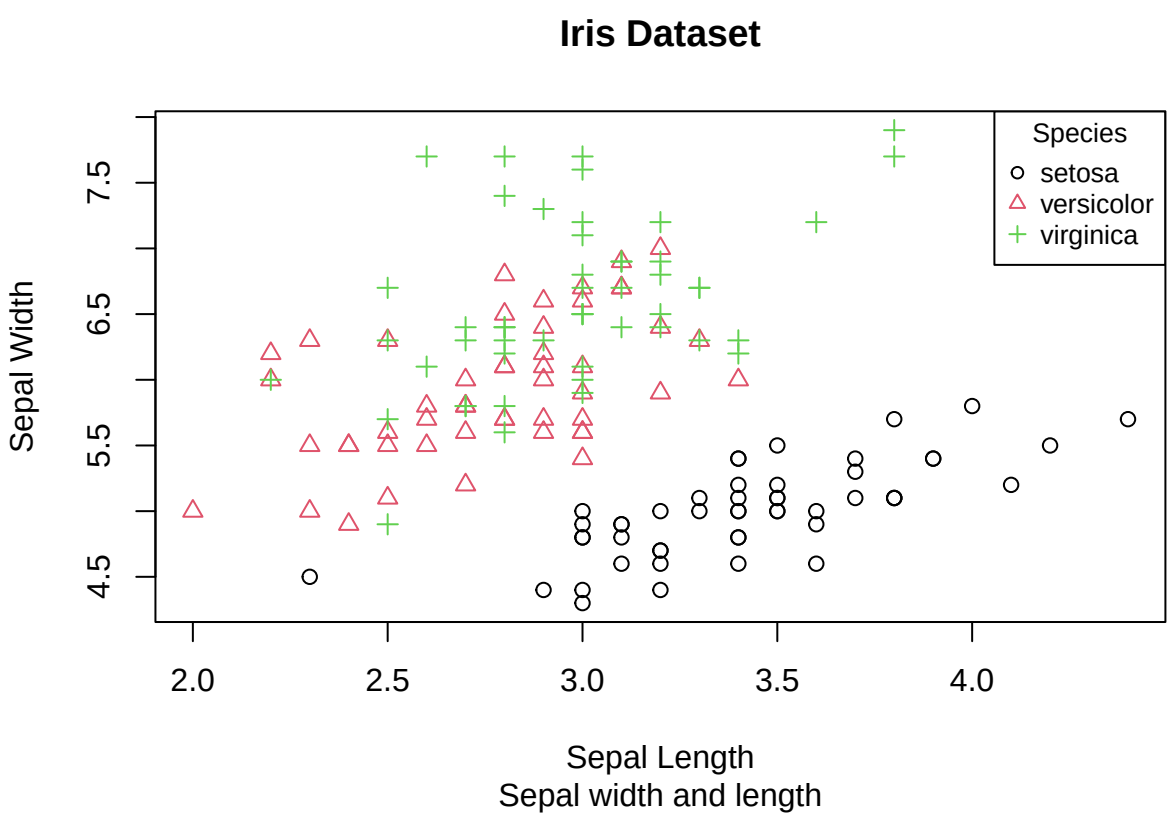
```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 95           5.6         2.7          4.2         1.3 versicolor
## 96           5.7         3.0          4.2         1.2 versicolor
## 97           5.7         2.9          4.2         1.3 versicolor
## 98           6.2     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

#e

```r
#6e

# 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 Length",
  ylab = "Sepal Width",
  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")
```



#f

```r
#f. Interpret the result.
#This R code converts the "Species" column in the iris dataset to a factor, and then creates a scatterp

#In the scatterplot, each species is represented by a different symbol (pch) and color. The pch paramet

#A legend is added to the top right corner of the plot to help interpret the colors and symbols. The ti

#When you run this code, you will see a scatterplot with sepal length on the x-axis and sepal width on
```

#7 Basic Cleaning and Transformation of Objects

```
library(readxl)
alexa_file <- read_excel("/cloud/project/worksheet#4/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
```

#7a

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

alexa_file$variation <- gsub("White  Dot", "WhiteDot", alexa_file$variation)
alexa_file$variation <- gsub("White  Plus", "WhitePlus", alexa_file$variation)
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
```

#7b

```
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
```

```
var_total <- alexa_file %>%
  count(alexa_file$variation)

var_total
```

```
## # A tibble: 16 x 2
##    `alexa_file$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(var_total, file = "VAR.RData")
```

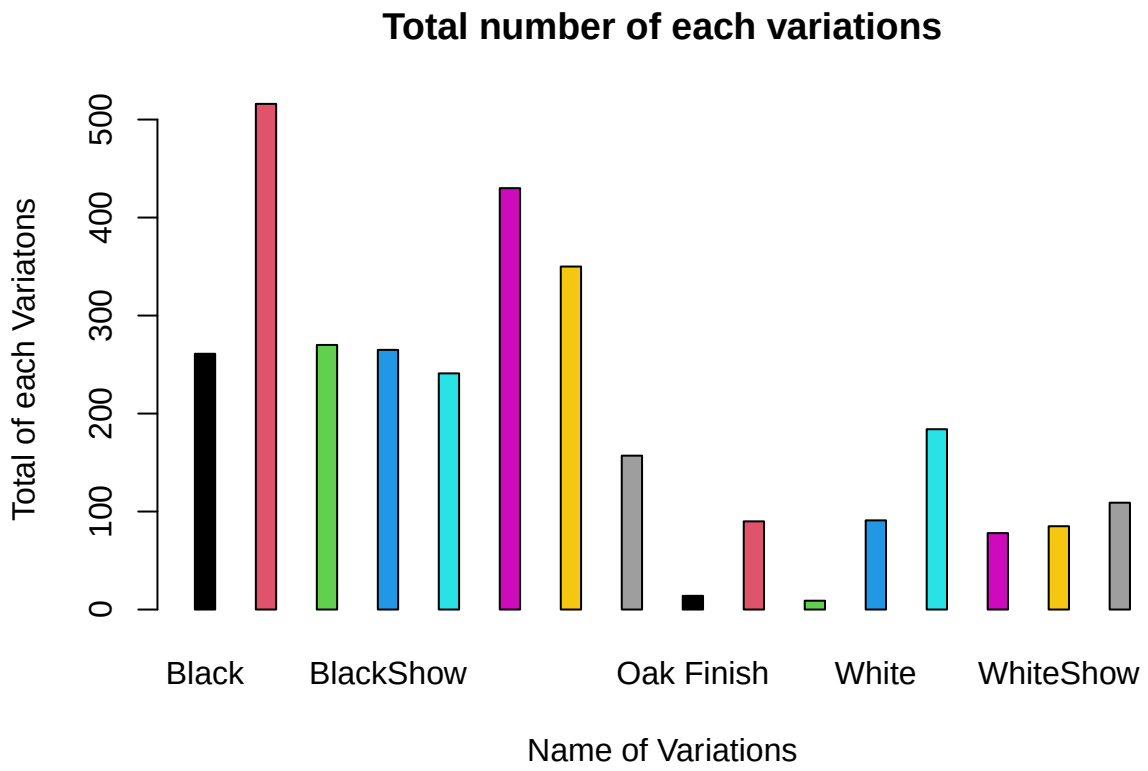
#7c

```
load("VAR.RData")
var_total
```

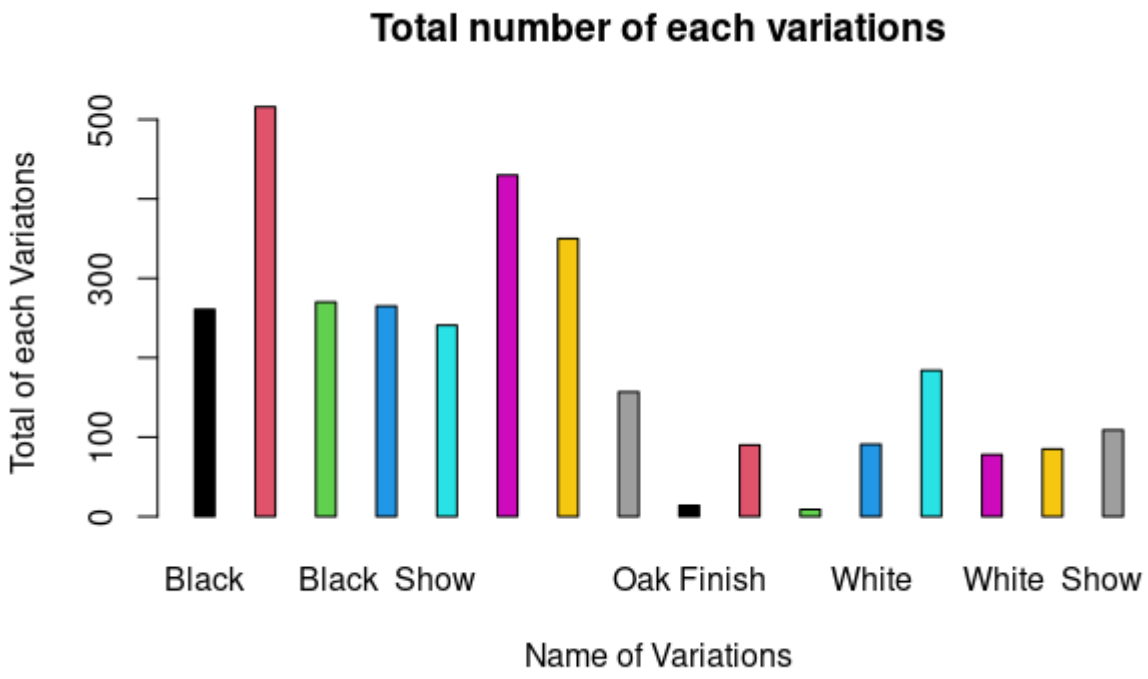
```
## # A tibble: 16 x 2
##    `alexa_file$variation`           n
##    <chr>                        <int>
##  1 Black                          261
##  2 BlackDot                       516
##  3 BlackPlus                      270
##  4 BlackShow                      265
##  5 BlackSpot                      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
```

```
varNames <- var_total$`alexa_file$variation`

totalPlot <- barplot(var_total$n,
        names.arg = varNames,
        main = "Total number of each variations",
        xlab = "Name of Variations",
        ylab = "Total of each Variatons",
        col = 1:16,
        space = 2)
```

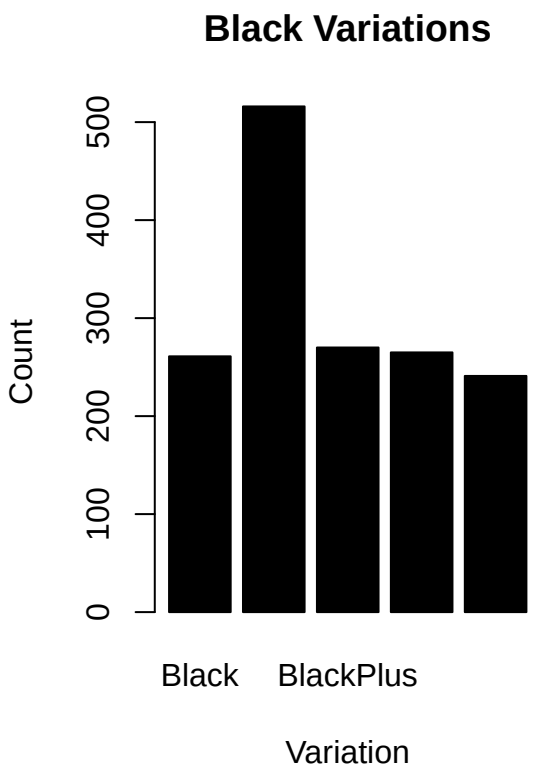


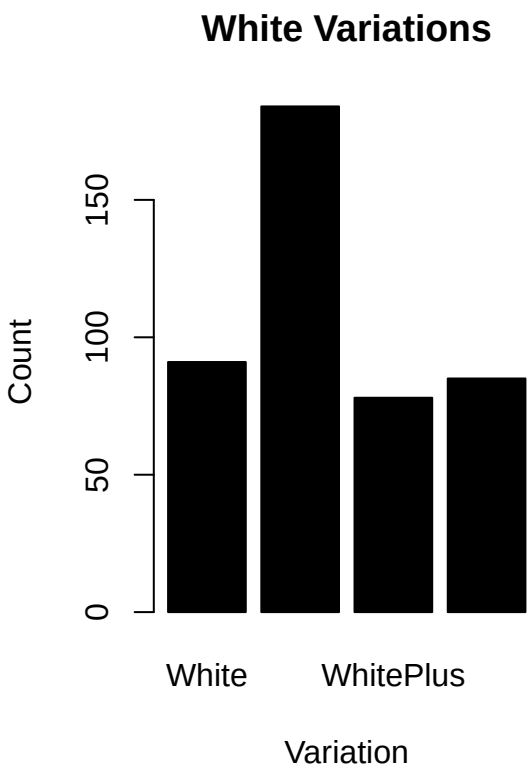

```
knitr::include_graphics("/cloud/project/worksheet#4/variations.png")
```



#7d

```
blackVars <- var_total[var_total$`alexa_file$variation` %in% c("Black", "BlackPlus" , "BlackShow" ,"Bla

whiteVars <- var_total[var_total$`alexa_file$variation` %in% c("White", "WhiteDot", "WhitePlus", "White

par(mfrow = c(1,2))

barplot(height = blackVars$n,
        names.arg = blackVars$`alexa_file$variation`,
        col = c("black"),
        main = "Black Variations",
        xlab = "Variation",
        ylab = "Count",
        border = "black")

barplot(height = whiteVars$n,
        names.arg = whiteVars$`alexa_file$variation`,
        col = c("black"),
        main = "White Variations",
        xlab = "Variation",
        ylab = "Count",
        border = "black")
```






```
knitr::include_graphics("/cloud/project/worksheet#4/bw.png")
```