RWorksheet_Condag#4b

Angel Blase

2024-10-18

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)
x <- matrix(vectorA, nrow = 5, ncol = 5)

for (i in 1:5) {
  for(j in 1:5) {
    x[i, j] <- abs(i - j)
  }
}

x
```

```
##      [,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.

```
for(i in 1:5) {
  line <- rep('"*"', i)
  cat(line, sep = " ")
  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.

```
#start <- as.integer(readline("Enter the starting term for the Fibonacci Sequence: "))
start <- 3
x <- 0
y <- 1
index <- 1
```

```
repeat {
  if(index >= start) {
    if (x > 500) {
      break
    }
    cat(x, " ")
  }

  nxt <- x + y
  x <- y
  y <- nxt
  index <- index + 1
}
```

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

```
cat("\n")
```

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

4. Import the dataset

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.

```
shoeSize <- read.csv("Shoe_sizes.csv")
head(shoeSize)
```

```
##   Show.Size Height Gender
## 1       6.5   66.0      F
## 2       9.0   68.0      F
## 3       8.5   64.5      F
## 4       8.5   65.0      F
## 5      10.5   70.0      M
## 6       7.0   64.0      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.

```
male <- subset(shoeSize, Gender == "M")
female <- subset(shoeSize, Gender == "F")

n_males <- nrow(male)
n_females <- nrow(female)

n_males
```

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

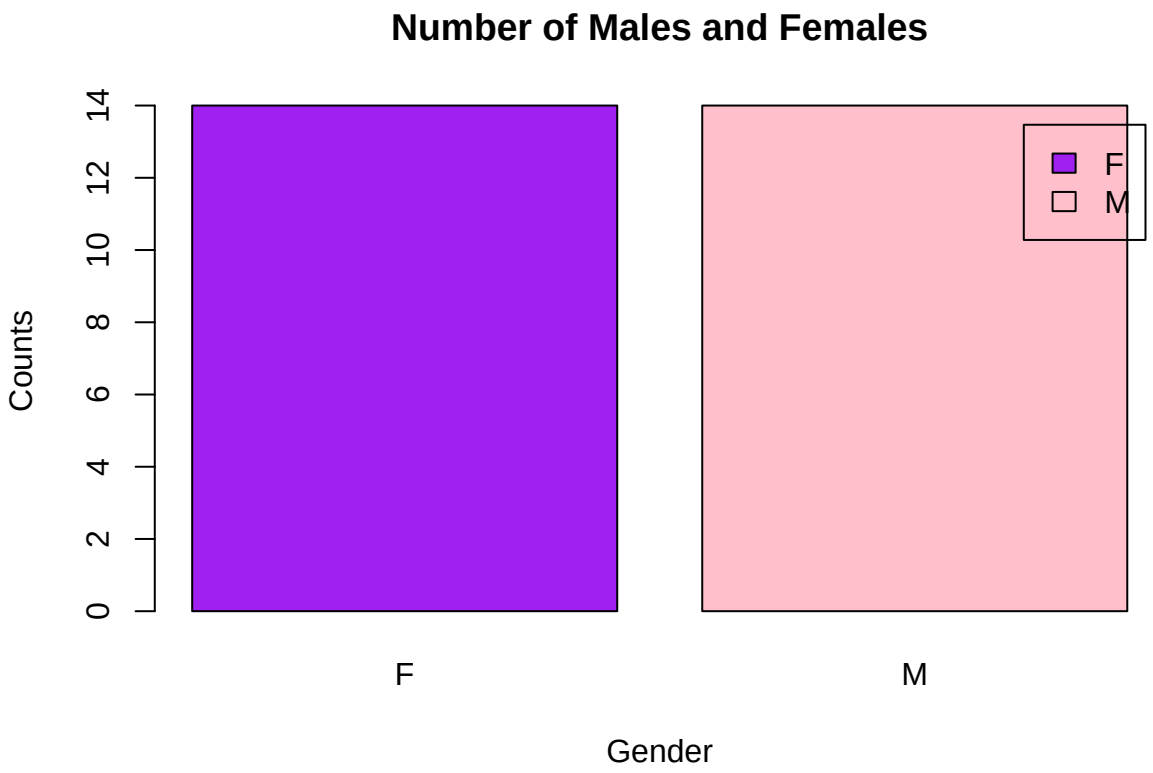
```
n_females
```

```
## [1] 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.

```
genderCounts <- table(shoeSize$Gender)
```

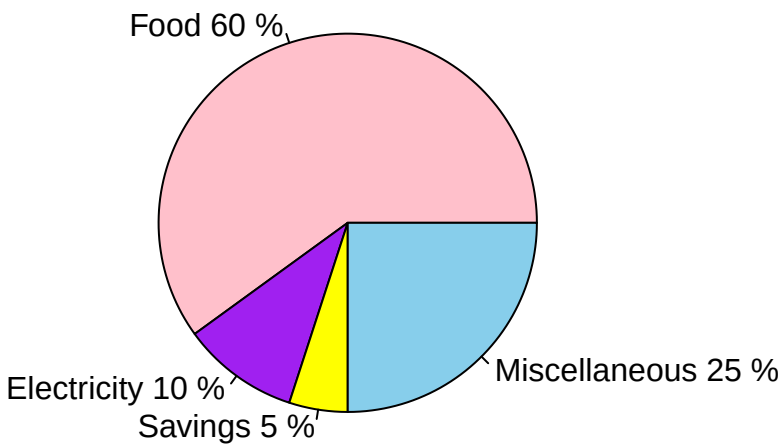
```r
barplot(genderCounts,
        main = "Number of Males and Females",
        xlab = "Gender",
        ylab = "Counts",
        col = c("purple", "pink"),
        legend = row.names(genderCounts)
        )
```



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

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
expenses <- c(Food = 60, Electricity = 10, Savings = 5, Miscellaneous = 25)
percentages <- round(expenses / sum(expenses) * 100)
labels <- paste(names(expenses), percentages, "%")
colors <- c("pink", "purple", "yellow", "skyblue")
pie(expenses,
    labels = labels,
    col = colors,
    main = "Dela Cruz Family Monthly Expenses")
```



6. Use the iris dataset. data(iris)

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 ...
```

Description: The image shows the output of the str(iris) command in R, which provides a summary of the iris dataset. This dataset has 150 rows and 5 columns. The columns include Sepal.Length and Sepal.Width, which represent the length and width of sepals, and Petal.Length and Petal.Width, which represent the length and width of petals. All these columns are numeric. The fifth column, Species, is a categorical (factor) variable with three levels: "setosa," "versicolor," and "virginica," representing different iris species. The summary displays sample values for each column to give an overview of the data structure.

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?

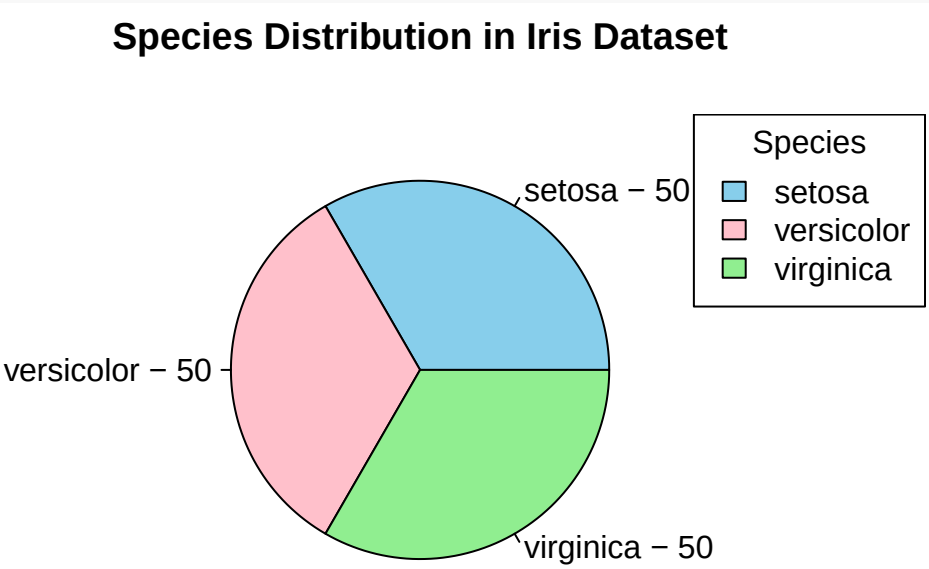
```
mean_values <- colMeans(iris[, c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width")])
mean_values
```

```
## 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.

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

pie(species_counts,
    main = "Species Distribution in Iris Dataset",
    col = c("skyblue", "pink", "lightgreen"),
```

```
    labels = paste(names(species_counts), "-", species_counts))

legend("topright",
       legend = names(species_counts),
       fill = c("skyblue", "pink", "lightgreen"),
       title = "Species"
)
```



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

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

cat("Last 6 rows of Setosa:\n")
```

```
## Last 6 rows of Setosa:
```

```
print(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
```

```
cat("\nLast 6 rows of Versicolor:\n")
```

```
##
## Last 6 rows of Versicolor:
```

```
print(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
```

```
cat("\nLast 6 rows of Virginica:\n")
```

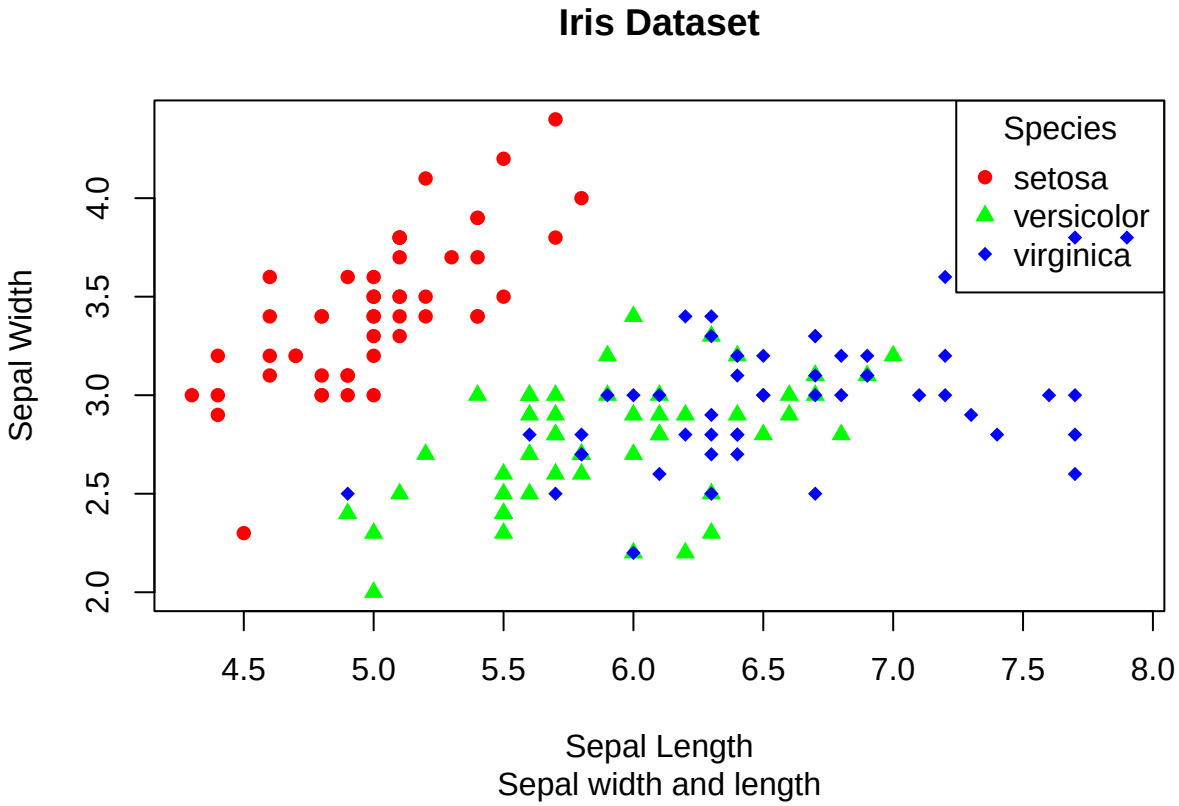
```
##
## Last 6 rows of Virginica:
```

```
print(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. 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.

```
species_colors <- c("setosa" = "red", "versicolor" = "green", "virginica" = "blue")
species_pch <- c("setosa" = 16, "versicolor" = 17, "virginica" = 18)

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

legend("topright", legend = levels(iris$Species),
       col = species_colors, pch = species_pch,
       title = "Species")
```



f. Interpretation: The scatterplot shows the relationship between sepal length and sepal width for the three species in the iris dataset: Setosa, Versicolor, and Virginica. Each species is represented by a different color and symbol, making it easy to see how they differ. Setosa (red points) generally has a wider sepal width compared to its sepal length and is well-separated from the other two species. Versicolor (green points) and Virginica (blue points) have more overlap in their sepal sizes, but Virginica tends to have longer sepals. This pattern suggests that Setosa is more distinct in terms of sepal measurements, while Versicolor and Virginica are more similar.

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, Black Spot). Also on the white variants (White Dot, White Plus, White Show, White Spot).

```
library(readxl)

alexa_file <- read_excel("alexa_file.xlsx")

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
```

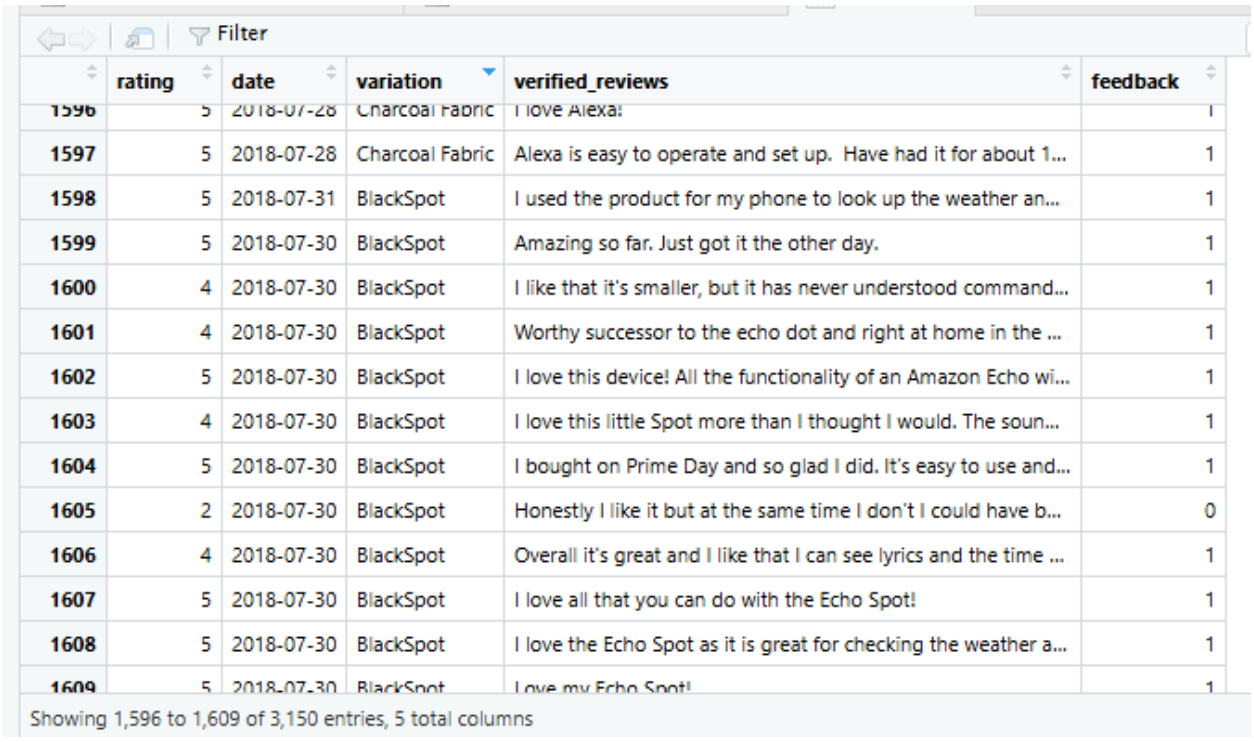
a. Rename the white and black variants by using gsub() function.

```
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$variation[1051:1100]
```

```
##  [1] "BlackSpot" "WhiteSpot" "BlackSpot" "BlackSpot" "BlackSpot" "BlackSpot"
##  [7] "WhiteSpot" "BlackSpot" "BlackSpot" "BlackSpot" "BlackSpot" "BlackSpot"
## [13] "WhiteSpot" "WhiteSpot" "BlackSpot" "WhiteSpot" "WhiteSpot" "WhiteSpot"
## [19] "WhiteSpot" "BlackSpot" "BlackSpot" "WhiteSpot" "BlackSpot" "BlackSpot"
## [25] "BlackSpot" "WhiteSpot" "BlackSpot" "BlackSpot" "WhiteSpot" "BlackSpot"
## [31] "BlackSpot" "BlackSpot" "BlackSpot" "WhiteSpot" "BlackSpot" "WhiteSpot"
## [37] "BlackSpot" "BlackSpot" "WhiteSpot" "WhiteSpot" "BlackSpot" "BlackSpot"
## [43] "BlackSpot" "BlackSpot" "WhiteSpot" "WhiteSpot" "BlackSpot" "BlackSpot"
## [49] "BlackSpot" "WhiteSpot"
```

Embed an image into Rmd.

```
knitr::include_graphics("alexa_modified.png")
```



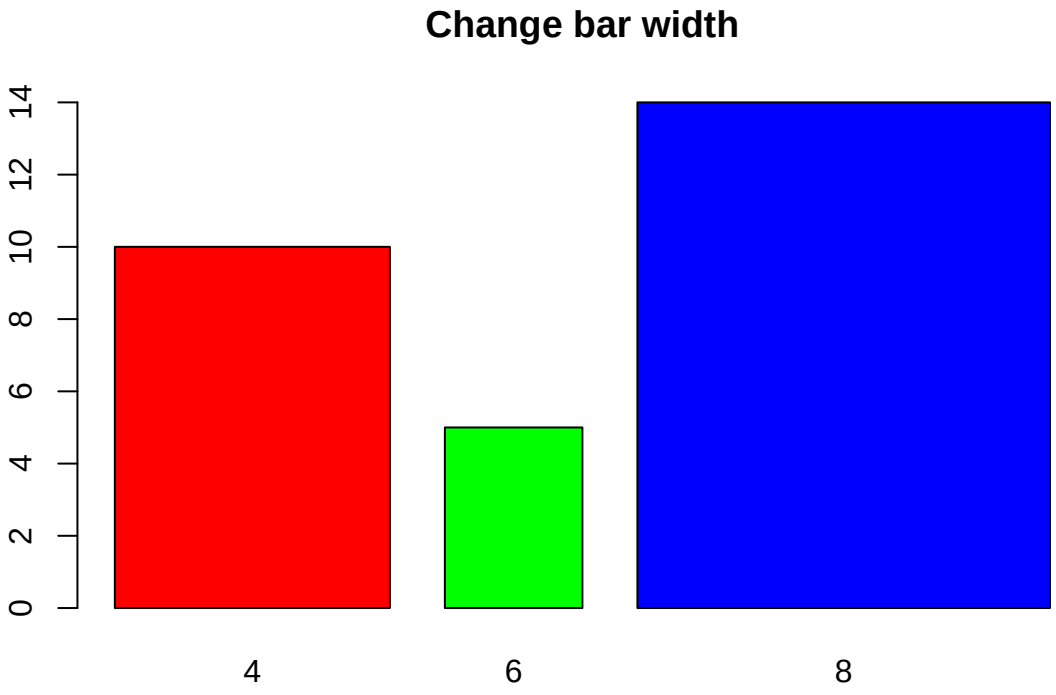
Filter

	rating	date	variation	verified_reviews	feedback
1596	5	2018-07-28	Charcoal Fabric	I love Alexa!	1
1597	5	2018-07-28	Charcoal Fabric	Alexa is easy to operate and set up. Have had it for about 1...	1
1598	5	2018-07-31	BlackSpot	I used the product for my phone to look up the weather an...	1
1599	5	2018-07-30	BlackSpot	Amazing so far. Just got it the other day.	1
1600	4	2018-07-30	BlackSpot	I like that it's smaller, but it has never understood command...	1
1601	4	2018-07-30	BlackSpot	Worthy successor to the echo dot and right at home in the ...	1
1602	5	2018-07-30	BlackSpot	I love this device! All the functionality of an Amazon Echo wi...	1
1603	4	2018-07-30	BlackSpot	I love this little Spot more than I thought I would. The soun...	1
1604	5	2018-07-30	BlackSpot	I bought on Prime Day and so glad I did. It's easy to use and...	1
1605	2	2018-07-30	BlackSpot	Honestly I like it but at the same time I don't I could have b...	0
1606	4	2018-07-30	BlackSpot	Overall it's great and I like that I can see lyrics and the time ...	1
1607	5	2018-07-30	BlackSpot	I love all that you can do with the Echo Spot!	1
1608	5	2018-07-30	BlackSpot	I love the Echo Spot as it is great for checking the weather a...	1
1609	5	2018-07-30	BlackSpot	Love my Echo Spot!	1

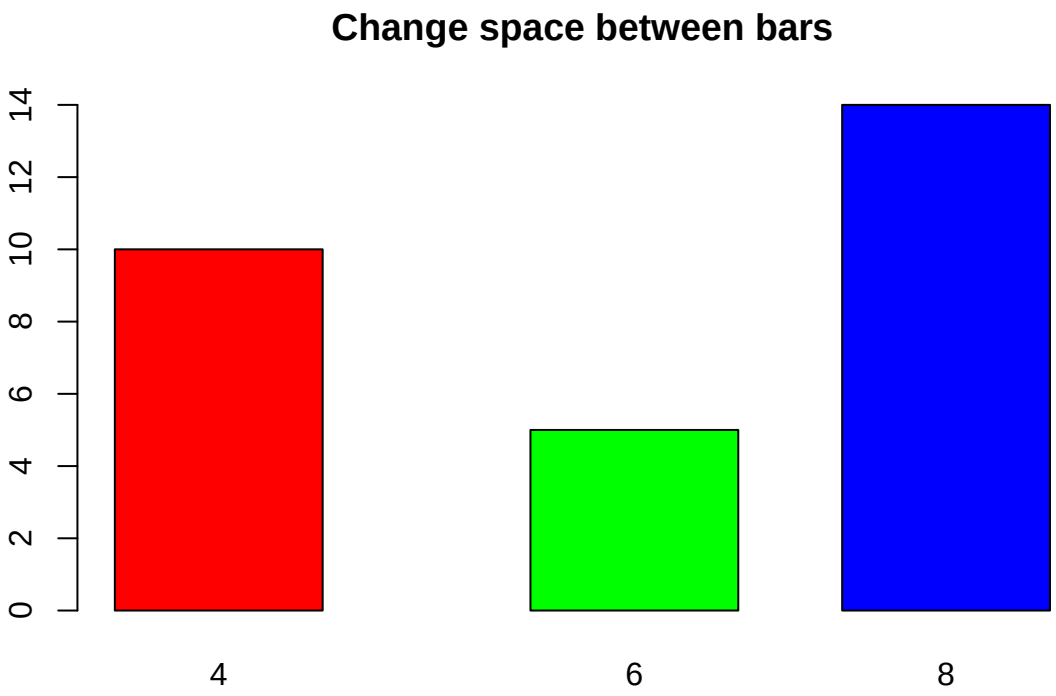
Showing 1,596 to 1,609 of 3,150 entries, 5 total columns

```
values <- c(10, 5, 14)
names <- c(4, 6, 8)
colors <- c("red", "green", "blue")
```

```
barplot(values, names.arg=names, col=colors, main="Change bar width", width=c(1, 0.5, 1.5))
```




```
barplot(values, names.arg=names, col=colors, main="Change space between bars", space=c(0.2, 1, 0.5))
```



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)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

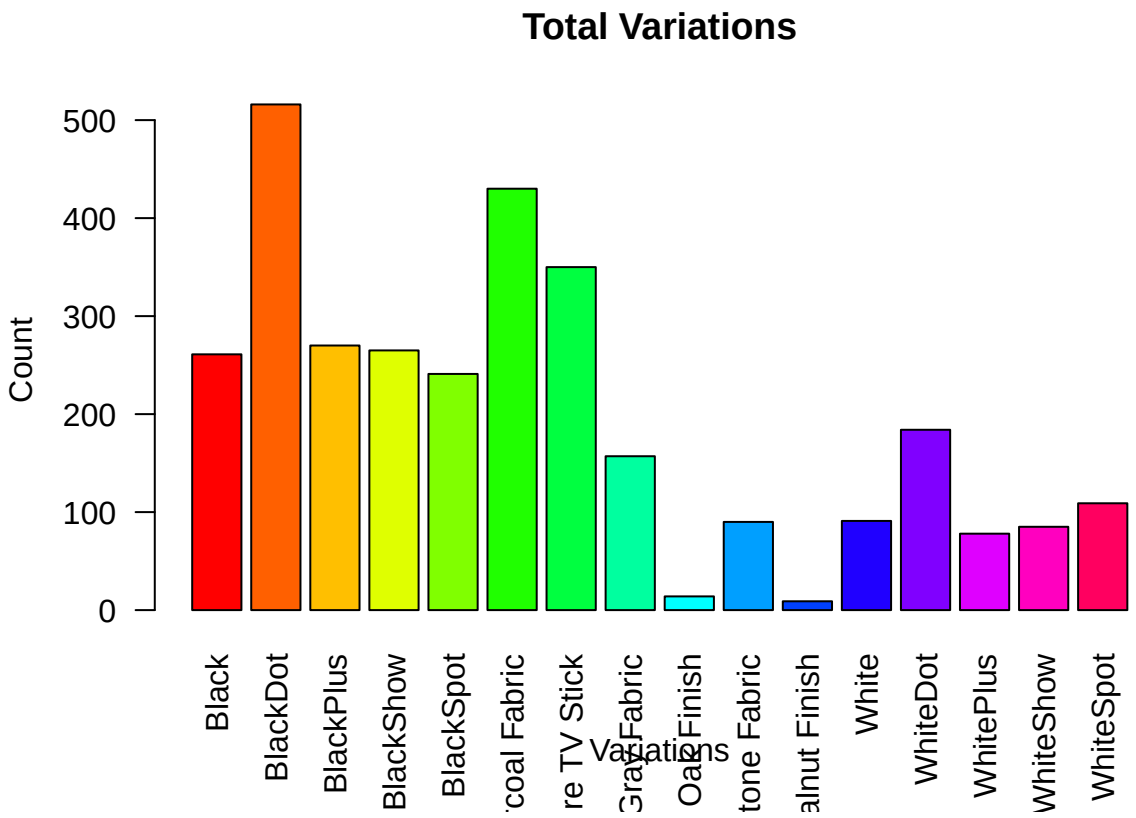
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

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

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

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

c. 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$n,
        names.arg = variation_counts$variation,
        col = rainbow(length(variation_counts$variation)),
        main = "Total Variations",
        xlab = "Variations",
        ylab = "Count",
        las = 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(dplyr)

variation_counts <- alexa_file %>%
  filter(grepl("Black|White", variation)) %>%
  count(variation)

black_counts <- variation_counts %>%
  filter(grepl("Black", variation))
white_counts <- variation_counts %>%
  filter(grepl("White", variation))

par(mfrow = c(1, 2))

variations <- variation_counts$variation
counts <- variation_counts$n

barplot(black_counts$n,
        names.arg = black_counts$variation,
        col = c("black", "pink", "green", "blue", "cyan"),
        main = "Black Variants",
        xlab = "Total Numbers",
        ylab = "Variants",
        ylim = c(0, max(black_counts$n) * 1.2))

barplot(white_counts$n,
        names.arg = white_counts$variation,
```

```
        col = c("black", "pink", "green", "blue", "cyan"),
        main = "White Variants",
        xlab = "Total Numbers",
        ylab = "Variants",
        ylim = c(0, max(white_counts$n) * 1.2))
```