RWorksheet_Palabrica#4b

Marvin Luiz Palabrica

2024-10-30

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

```
matrixA <- matrix(0, nrow = 5, ncol = 5)
vectorA <- c(1, 2, 3, 4, 5)

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

print(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
```

```
rows <- 5

for (i in 1:rows) {
  cat(rep('"*"', i), "\n")
}
```

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

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

```
rows <- 5
for (i in 1:rows) {
cat(rep("''*''", i), "\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.

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 data set? Show your codes and its result

```
library(readr)

data <- read_csv("/cloud/project/Worksheet#4/shoe_size2.csv")
```

```
## New names:
## Rows: 28 Columns: 6
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (1): Gender dbl (2): Shoe size, Height lgl (3): ...4, ...5, ...6
## 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.
## * `` -> `...4`
## * `` -> `...5`
## * `` -> `...6`
```

```
data
```

```
## # A tibble: 28 x 6
##    `Shoe size` Height Gender ...4  ...5  ...6
##          <dbl>  <dbl> <chr>  <lgl> <lgl> <lgl>
##  1         6.5   66   F      NA    NA    NA
##  2         9     68   F      NA    NA    NA
##  3         8.5   64.5 F      NA    NA    NA
##  4         8.5   65   F      NA    NA    NA
##  5        10.5   70   M      NA    NA    NA
##  6         7     68   F      NA    NA    NA
##  7         9.5   70   M      NA    NA    NA
##  8         9     71   F      NA    NA    NA
##  9        13     72   M      NA    NA    NA
## 10         7.5   64   M      NA    NA    NA
## # i 18 more rows
```

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_subset <- subset(data, Gender == "F")
male_subset <- subset(data, Gender == "M")

# Count observations in each subset
female_count <- nrow(female_subset)
male_count <- nrow(male_subset)
```

```r
# Print counts for each gender
cat("Female:", female_count, "\n")
```

```
## Female: 13
```

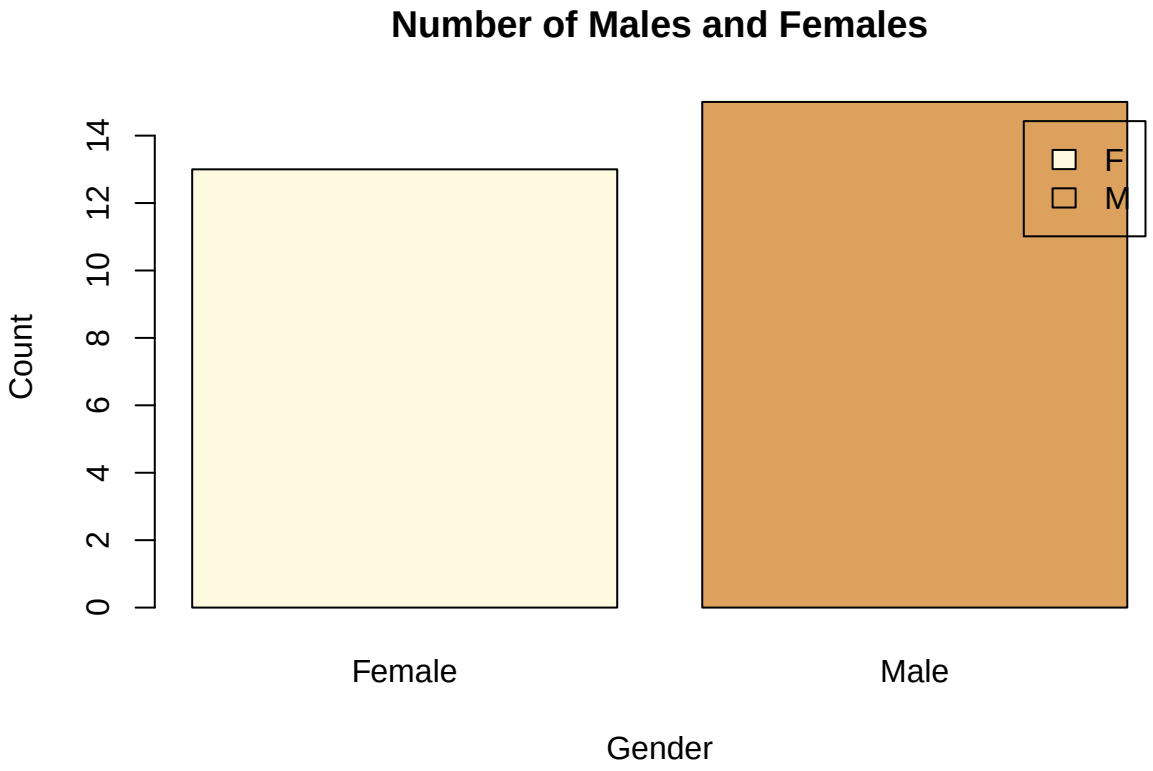
```r
cat("Male:", male_count, "\n")
```

```
## Male: 15
```

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.

```r
# Count the number of males and females
gender_counts <- table(data$Gender)

# Create a barplot
barplot(gender_counts, main = "Number of Males and Females",
        col = c("#FEFAE0", "#DDA15E"), names.arg = c("Female", "Male"),
        ylab = "Count", xlab = "Gender", legend = TRUE)
```



5. The monthly income of Dela Cruz family was spent on the following: Food Electricity Savings Miscellaneous.

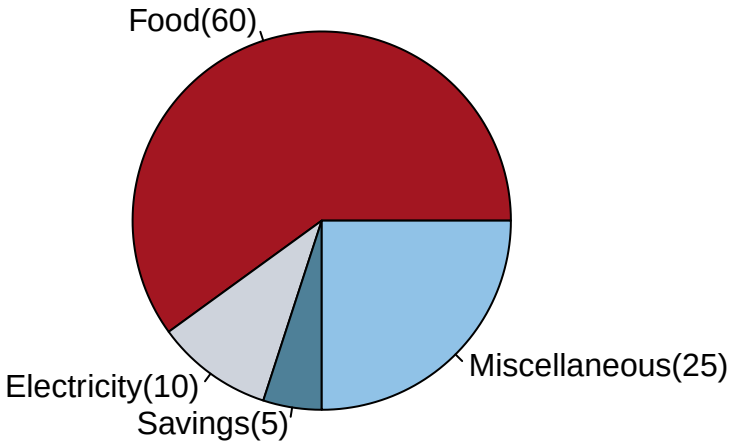
a. Create a piechart that will include labels in percentage.Add some colors and title of the chart. Write the R scripts and show its output.

```r
categories <- c("Food", "Electricity", "Savings", "Miscellaneous")
values <- c(60, 10, 5, 25)
pie_chart <- pie(values,

labels = paste(categories, "(", values, ")", sep = ""),
col = c("#A31621", "#CED3DC", "#4E8098", "#90C2E7"),
main = "Monthly Income Distribution of Dela Cruz Family")
```

```
percentages <- round(values / sum(values) * 100, 1)
text(0, 5, paste(percentages, "%"), cex = 1.2, pos = 3)
```

Monthly Income Distribution of Dela Cruz Family



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 ...
```

The output of the str(iris) function shows that the iris dataset is a data frame with 150 rows and 5 columns. Four of the columns contain numeric data, which correspond to measurements of various parts of the iris flower, while the fifth column is a factor that indicates the flower's species.

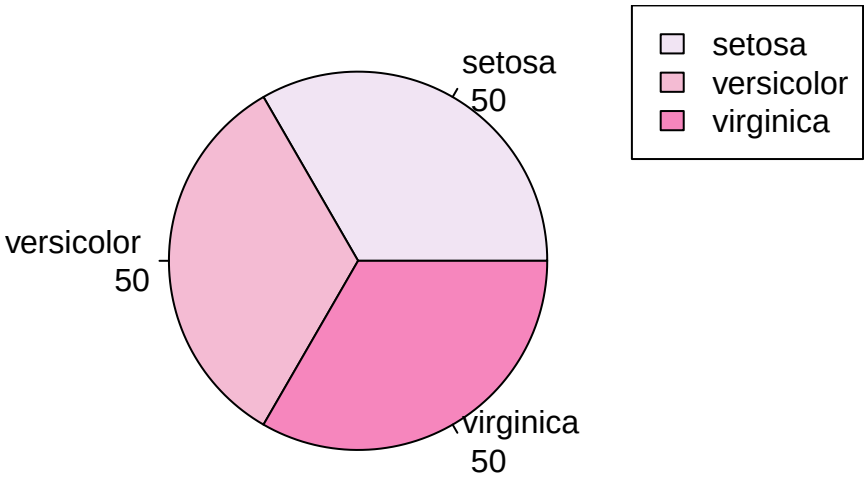
b. Create an R object that will contain the mean of the sepal.length, sepal.width,petal.length,and petal.width. What is the R script and its result?

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

```
## 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)
speciesCounts <- table(iris$Species)
pie(speciesCounts,
main = "Species Distribution in Iris Dataset",
col = c("#F1E4F3", "#F4BBD3", "#F686BD"), labels = paste(names(speciesCounts), "\n", speciesCounts))
legend("topright", legend = names(speciesCounts), fill = c("#F1E4F3", "#F4BBD3", "#F686BD"))
```



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

```
setosa_data <- iris[iris$Species == "setosa", ]
versicolor_data <- iris[iris$Species == "versicolor", ]
virginica_data <- iris[iris$Species == "virginica", ]
last_six_setosa_data <- tail(setosa_data, 6)
last_six_versicolor_data <- tail(versicolor_data, 6)
last_six_virginica_data <- tail(virginica_data, 6)
last_six_setosa_data
```

```
##    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
```

```
last_six_versicolor_data
```

```
##     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
```

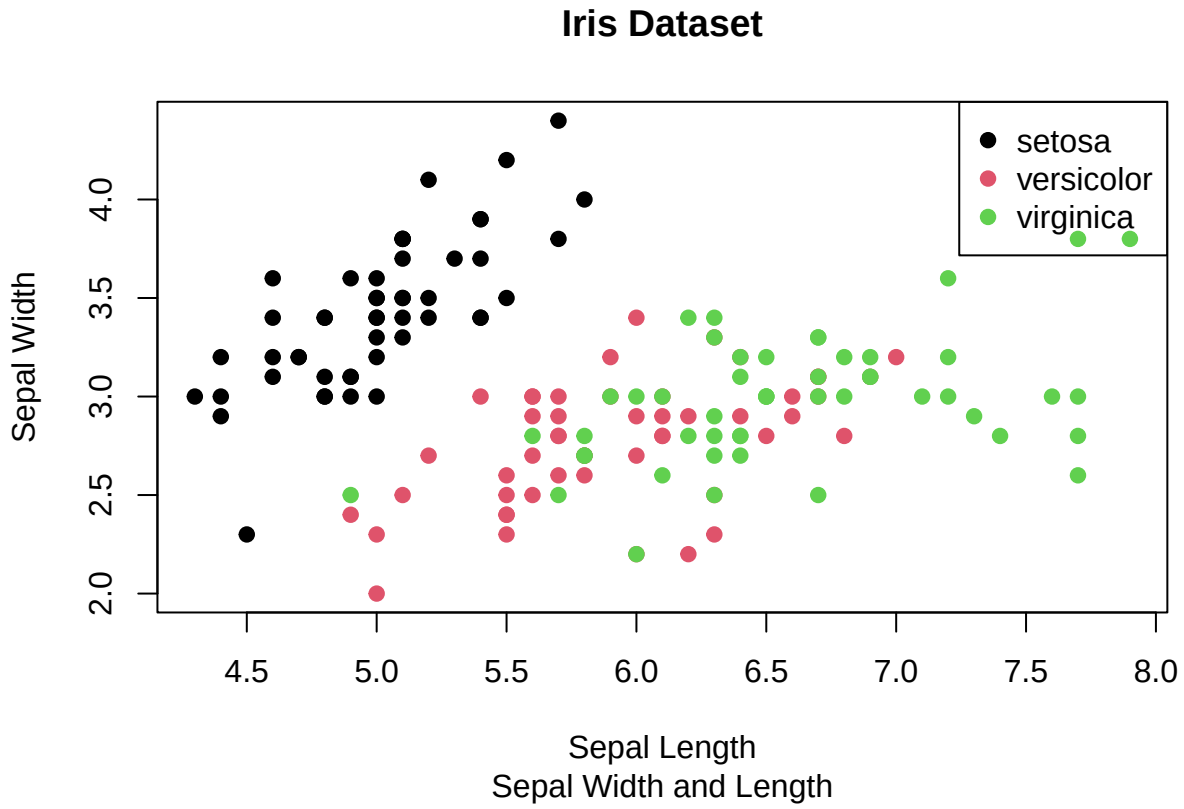
```
last_six_virginica_data
```

```
##     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.

```
plot(iris$Sepal.Length, iris$Sepal.Width,
main = "Iris Dataset",
sub = "Sepal Width and Length",
xlab = "Sepal Length",
ylab = "Sepal Width",
pch = 19,
col = iris$Species)
legend("topright", legend = levels(iris$Species),
col = 1:3, pch = 19)
```



f.Interpret the result.

The pie chart displays the distribution of iris species in the dataset, with each slice representing the proportion of each species. This helps identify the most common species. The subsets show the last six entries for each species, providing detailed measurements of sepal and petal dimensions, which helps compare how these measurements vary across species. The scatterplot reveals the relationship between sepal length and width. If setosa forms a cluster at lower values, it indicates that this species generally has smaller flowers. On the other hand, if versicolor and virginica overlap, it suggests that their flower sizes are quite similar, making it difficult to distinguish between them based on these measurements.

BASIC CLEANING AND TRANSFORMATION OF OBJECTS

7. Import the alexa-file.xlsx. Check on the variations. Notice that there are ex- tra 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).

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

```
library(readxl)
alexa_data <- read_excel("/cloud/project/Worksheet#4/alexa_file.xlsx")
unique(alexa_data$variation)
```

```
##  [1] "Charcoal Fabric"              "Walnut Finish"
##  [3] "Heather Gray Fabric"          "Sandstone Fabric"
##  [5] "Oak Finish"                   "Black"
##  [7] "White"                        "Black  Spot"
##  [9] "White  Spot"                  "Black  Show"
## [11] "White  Show"                  "Black  Plus"
## [13] "White  Plus"                  "Configuration: Fire TV Stick"
## [15] "Black  Dot"                   "White  Dot"
```

```
alexa_data$variation <- gsub("Black Dot", "BlackDot", alexa_data$variation)
alexa_data$variation <- gsub("Black Plus", "BlackPlus", alexa_data$variation)
alexa_data$variation <- gsub("Black Show", "BlackShow", alexa_data$variation)
alexa_data$variation <- gsub("Black Spot", "BlackSpot", alexa_data$variation)
alexa_data$variation <- gsub("White Dot", "WhiteDot", alexa_data$variation)
alexa_data$variation <- gsub("White Plus", "WhitePlus", alexa_data$variation)
alexa_data$variation <- gsub("White Show", "WhiteShow", alexa_data$variation)
alexa_data$variation <- gsub("White Spot", "WhiteSpot", alexa_data$variation)
unique(alexa_data$variation)
```

```
##  [1] "Charcoal Fabric"              "Walnut Finish"
##  [3] "Heather Gray Fabric"          "Sandstone Fabric"
##  [5] "Oak Finish"                   "Black"
##  [7] "White"                        "Black  Spot"
##  [9] "White  Spot"                  "Black  Show"
## [11] "White  Show"                  "Black  Plus"
## [13] "White  Plus"                  "Configuration: Fire TV Stick"
## [15] "Black  Dot"                   "White  Dot"
```

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? Hint: Use the dplyr package. Make sure to install it before loading the package.

```
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
```

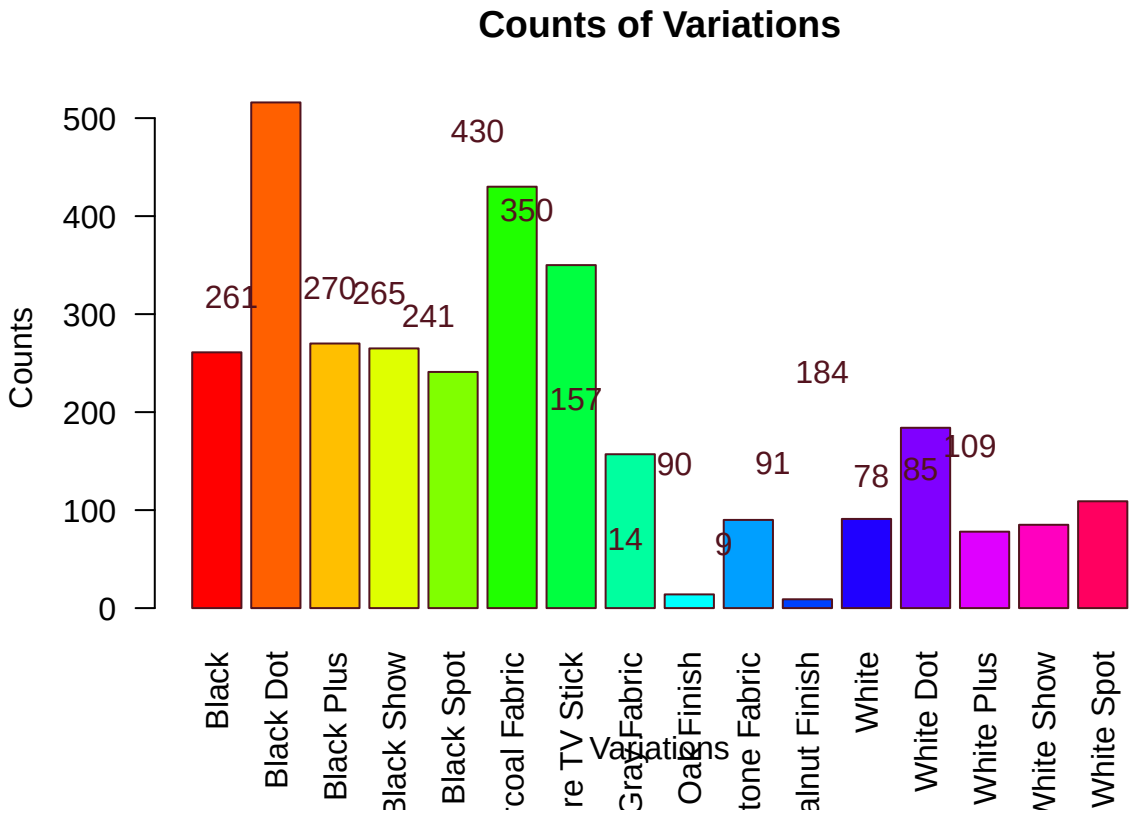
```
variations <- alexa_data %>%
count(variation)
print(variations)
```

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

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

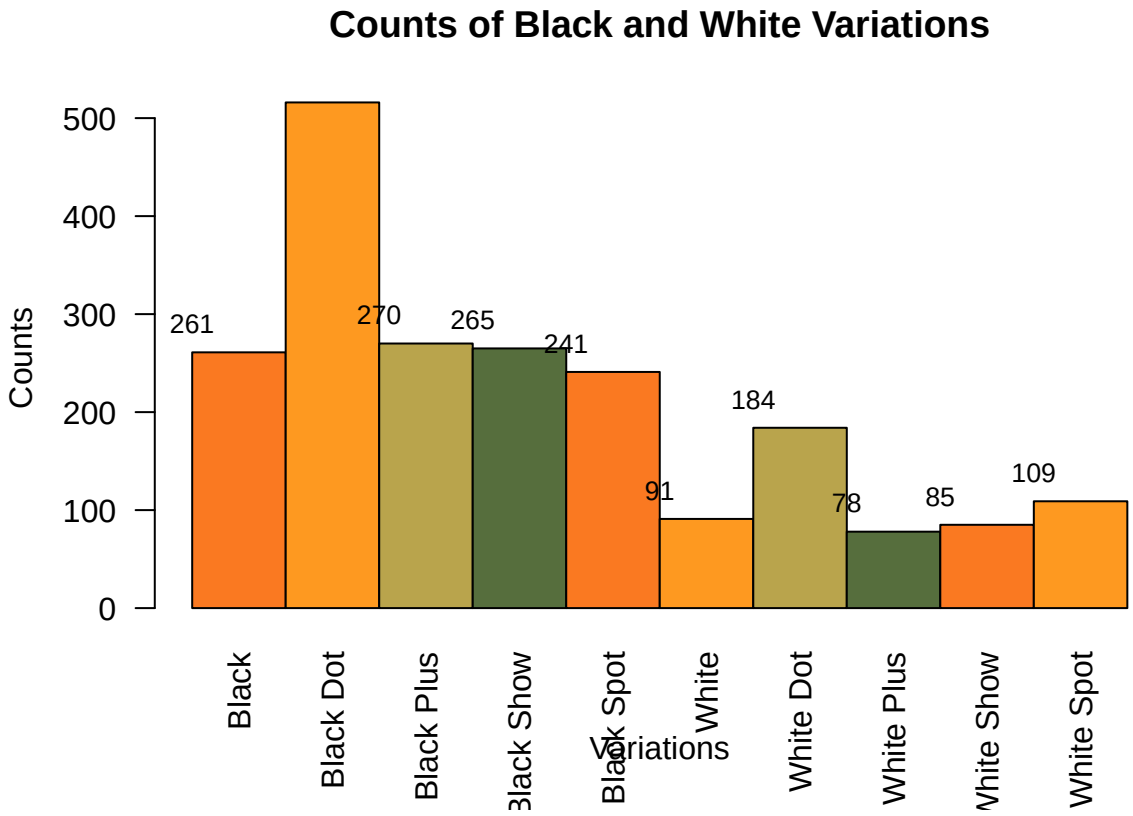
```
library(dplyr)
variations$variation <- gsub(" +", " ", variations$variation)
variations$variation <- trimws(variations$variation)
bardata <- variations$n
barnames <- variations$variation
barplot(
bardata,
main = "Counts of Variations",
col = rainbow(length(bardata)),
names.arg = barnames,
xlab = "Variations",
ylab = "Counts",
las = 2,
border = "#53131e"
)
text(
x = seq_along(bardata),
y = bardata + max(bardata) * 0.05,
labels = bardata,
pos = 3,
cex = 1,
col = "#53131e"
)
```



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(ggplot2)
library(dplyr)

variations$variation <- gsub(" +", " ", variations$variation)
variations$variation <- trimws(variations$variation)
bwvariations <- variations %>%
filter(grepl("Black|White", variation))
bardata <- as.matrix(bwvariations$n)
barnames <- bwvariations$variation
barplot(
bardata,
beside = TRUE,
main = "Counts of Black and White Variations",
col = c("#FA7921", "#FE9920", "#B9A44C", "#566E3D"),
names.arg = barnames,
xlab = "Variations",
ylab = "Counts",
las = 2,
border = "black"
)
text(x = seq_along(bardata), y = bardata, labels = bardata, pos = 3, cex = 0.8, col = "black")
```


Counts of Black and White Variations
Counts
500
400
300
200
100
0
261
270
265
241
91
184
78
85
109
Black
Black Dot
Black Plus
Black Show
Black Spot
White
White Dot
White Plus
White Show
White Spot
Variations