RWorksheet_Sorenio#4b

2024-10-28

```
# 1
vectA <- c(1, 2, 3, 4, 5)

matrixB <- matrix(0, nrow = 5, ncol = 5)

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

```
##      [,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
for (i in 1:5) {
  for (j in 1:i) {
    cat("* ")
  }
  cat("\n")
}
```

```
## *
## * *
## * * *
## * * * *
## * * * * *
```

```
# 3
start1 <- as.integer(readline(prompt="Enter the first number of the Fibonacci sequence: "))
```

```
## Enter the first number of the Fibonacci sequence:
```

```
start2 <- as.integer(readline(prompt="Enter the second number of the Fibonacci sequence: "))
```

```
## Enter the second number of the Fibonacci sequence:
```

```r
fibonacci <- c(start1, start2)
repeat {
  next_value <- tail(fibonacci, 2)[1] + tail(fibonacci, 2)[2]
  if (is.na(next_value) || next_value > 500) break
  fibonacci <- c(fibonacci, next_value)
}
print(fibonacci)
```

```
## [1] NA NA
```

```r
# 4
library(readr)

data <- read_csv("C:/Users/User/Documents/sample_data.csv", show_col_types = FALSE)
data
```

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

```r
# 4 a
data <- read_csv("C:/Users/User/Documents/sample_data.csv", show_col_types = FALSE)
print(head(data, 6))
```

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

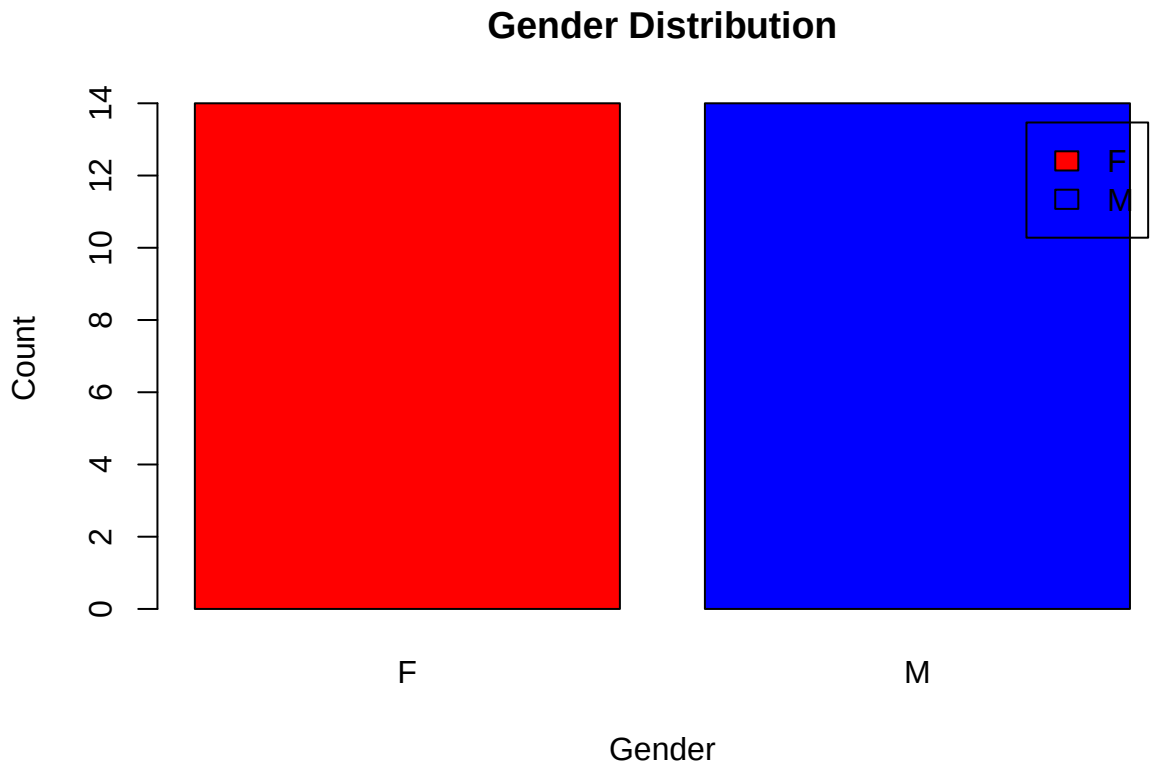
```r
# 4b
fem <- subset(data, Gender == "F")
male <- subset(data, Gender == "M")
cat("Female count:", nrow(fem),"\n")
```

```
## Female count: 14
```

```r
cat("Male count:", nrow(male),"\n")
```

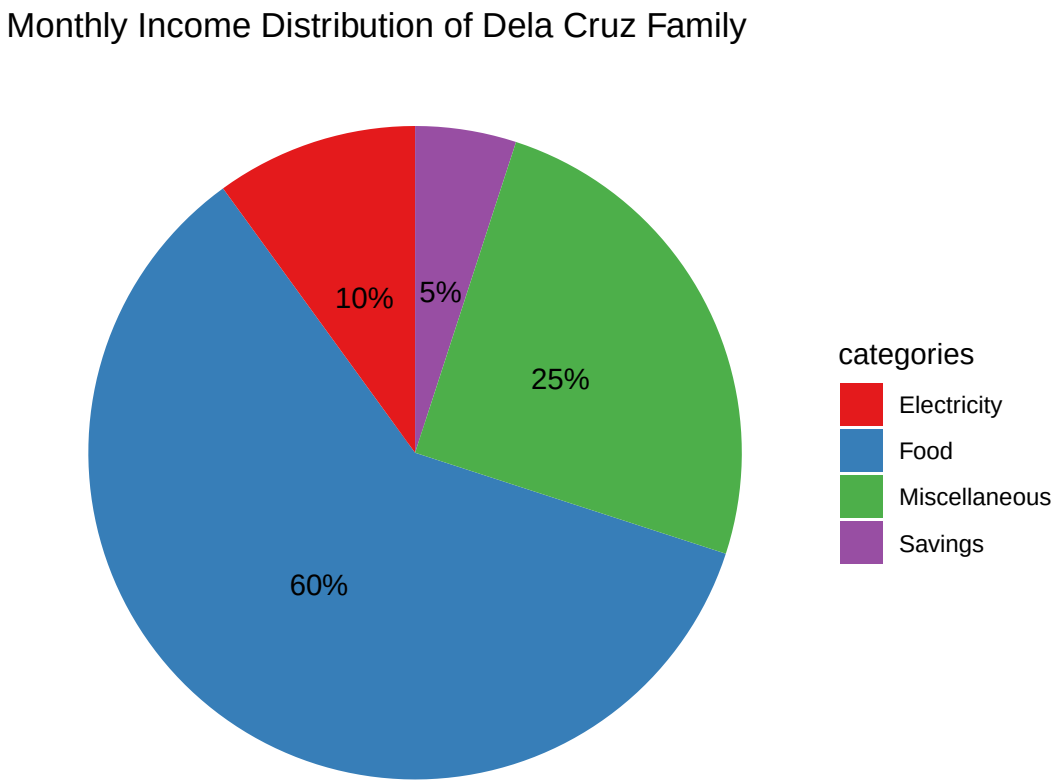
```
## Male count: 14
```

```r
# 4c
gencount <- table(data$Gender)
barplot(gencount, main= "Gender Distribution", col = c("red", "blue"),
        xlab="Gender", ylab="Count", legend=TRUE)
```




```r
# 5 a
library(ggplot2)

categories <- c("Food", "Electricity", "Savings", "Miscellaneous")
spending <- c(60, 10, 5, 25)


data <- data.frame(categories, spending)


data$percentage <- data$spending / sum(data$spending) * 100


ggplot(data, aes(x = "", y = percentage, fill = categories)) +
  geom_bar(stat = "identity", width = 1) +
```

```
  coord_polar("y") +
  geom_text(aes(label = paste0(round(percentage), "%")),
            position = position_stack(vjust = 0.5)) +
  labs(title = "Monthly Income Distribution of Dela Cruz Family") +
  theme_void() +
  scale_fill_brewer(palette = "Set1")
```



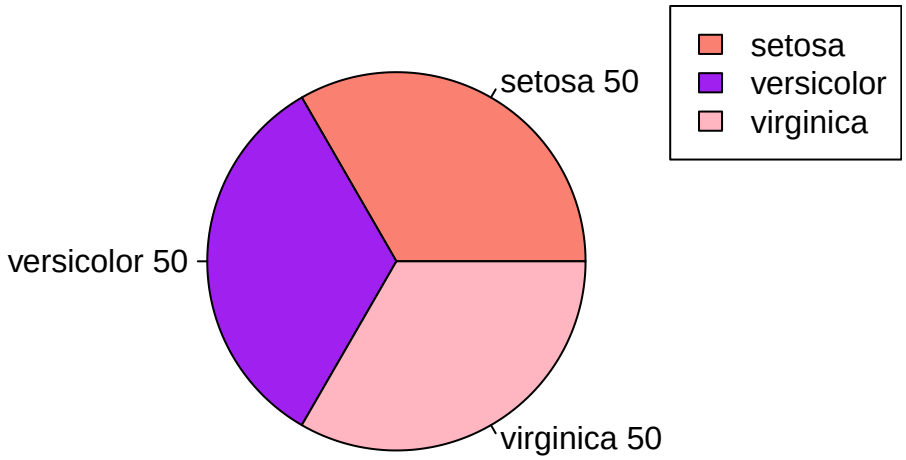

```
# 6 a
data(iris)
str(iris)
```

```
## '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 ...
```

```
# 6 b
means <- colMeans(iris[, 1:4])
means
```

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

```r
# 6 c
specscounts <- table(iris$Species)
pie(specscounts,
    main = "Species Distribution in Iris Dataset",
    col = c("salmon", "purple", "lightpink"),
    labels = paste(names(specscounts), specscounts))
legend("topright", legend = names(specscounts), fill = c("salmon", "purple", "lightpink"))
```




```r
# 6 d
setosa <- iris[iris$Species == "setosa", ]
versicolor <- iris[iris$Species == "versicolor", ]
virginica <- iris[iris$Species == "virginica", ]

tail(setosa)
```

```
##    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
```

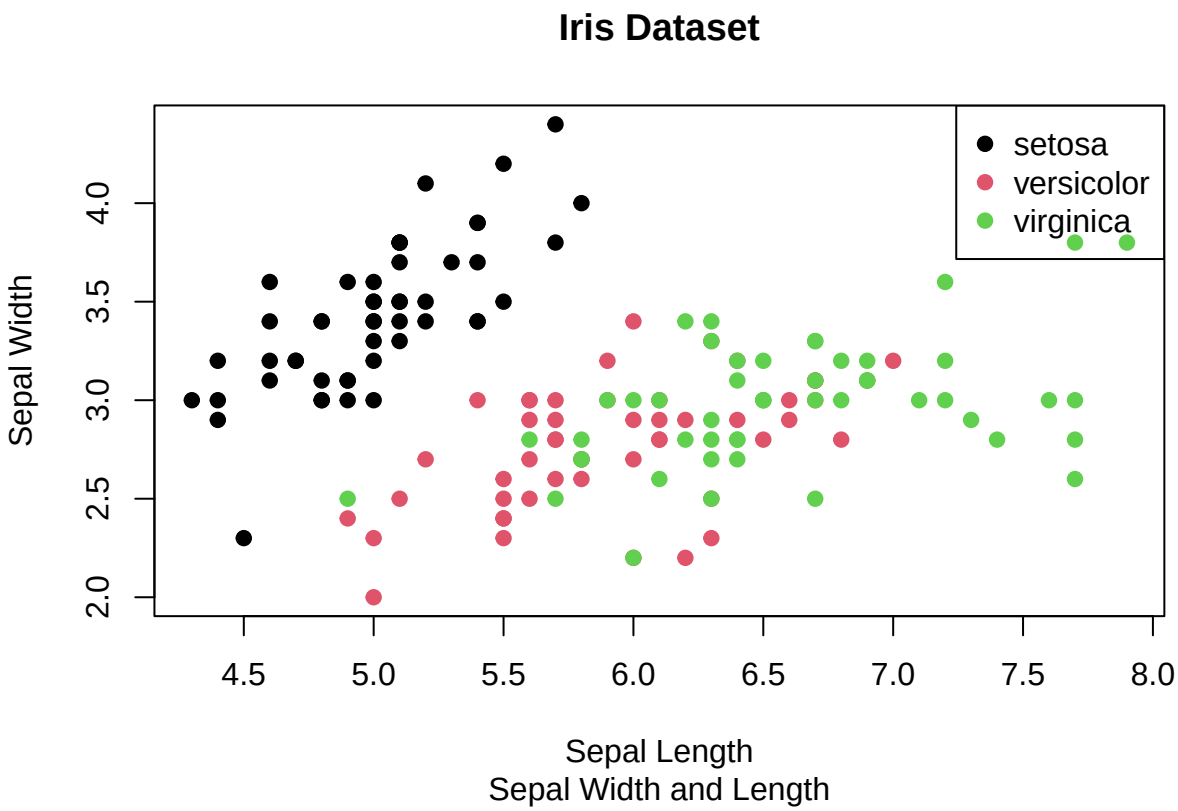
```r
tail(versicolor)
```

```
##     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(virginica)
```

```
##     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
```

```r
# 6 e
iris$Species <- as.factor(iris$Species)

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



6 f. Interpretation # In the analysis of the iris dataset, we found that it has 150 samples and five variables: sepal length, sepal width, petal length, petal width, and species. The mean sepal length is about 5.84 cm, which tells us the average size of the flowers.

The created pie chart shows the distribution of species, revealing that setosa is the most common. We also looked at the last six rows of each species to see their specific measurements.

Finally, the scatterplot of sepal length versus sepal width showed that setosa flowers generally have shorter sepals compared to versicolor and virginica, which are more similar. Overall, this analysis helps us understand the differences among the iris species.

```
# 7

library(readxl)

alexa_data <- read_excel("C:\\Users\\User\\Downloads\\alexa_file.xlsx")
alexa_data
```

```
## # 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
```

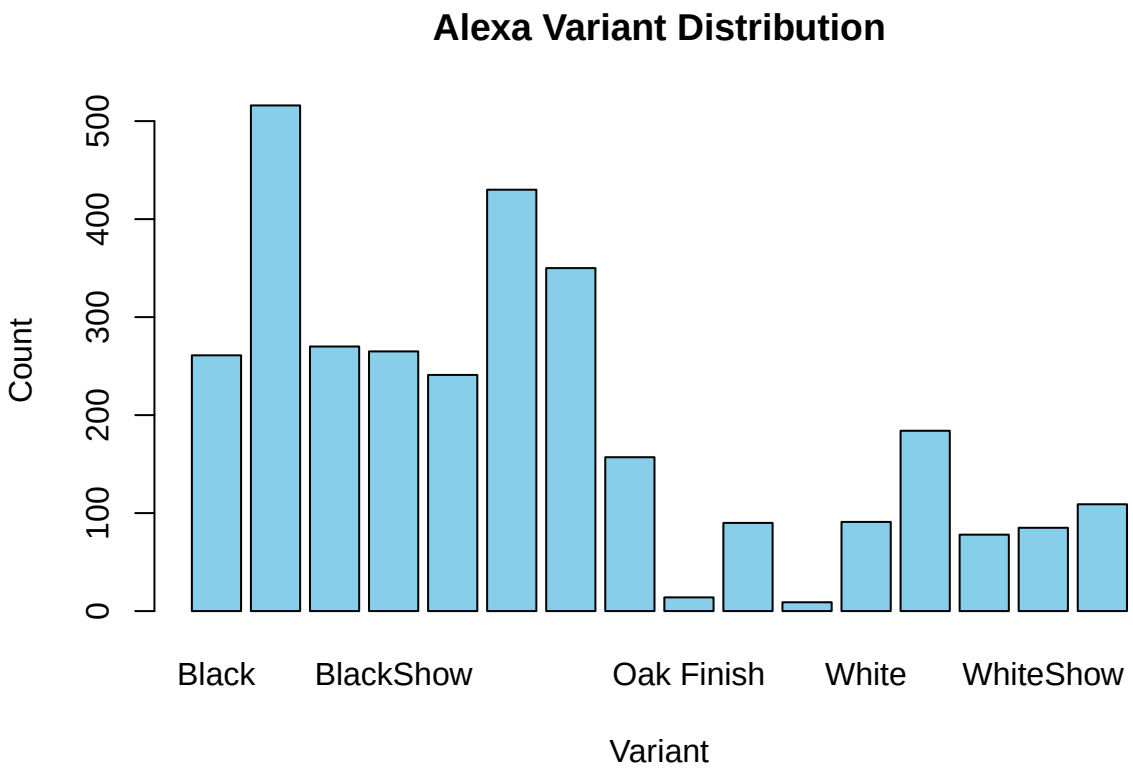
```
# 7. a

alexa_data$variation <- gsub("Black\\s+", "Black", alexa_data$variation)
alexa_data$variation <- gsub("White\\s+", "White", alexa_data$variation)
```

```
# 7. b

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
```

```
variationCount <- alexa_data %>% count(variation)
save(variationCount, file = "variations.RData")
```

```
# 7. c

load("variations.RData")
barplot(variationCount$n, names.arg = variationCount$variation, col = "skyblue", main = "Alexa Variant
```




```
# 7.d

variantCount <- alexa_data %>%
  group_by(variation) %>%
  summarize(count = n())
```

```
black_variants <- variantCount %>%
  filter(grepl("Black", variation))
white_variants <- variantCount %>%
  filter(grepl("White", variation))

print(black_variants)
```

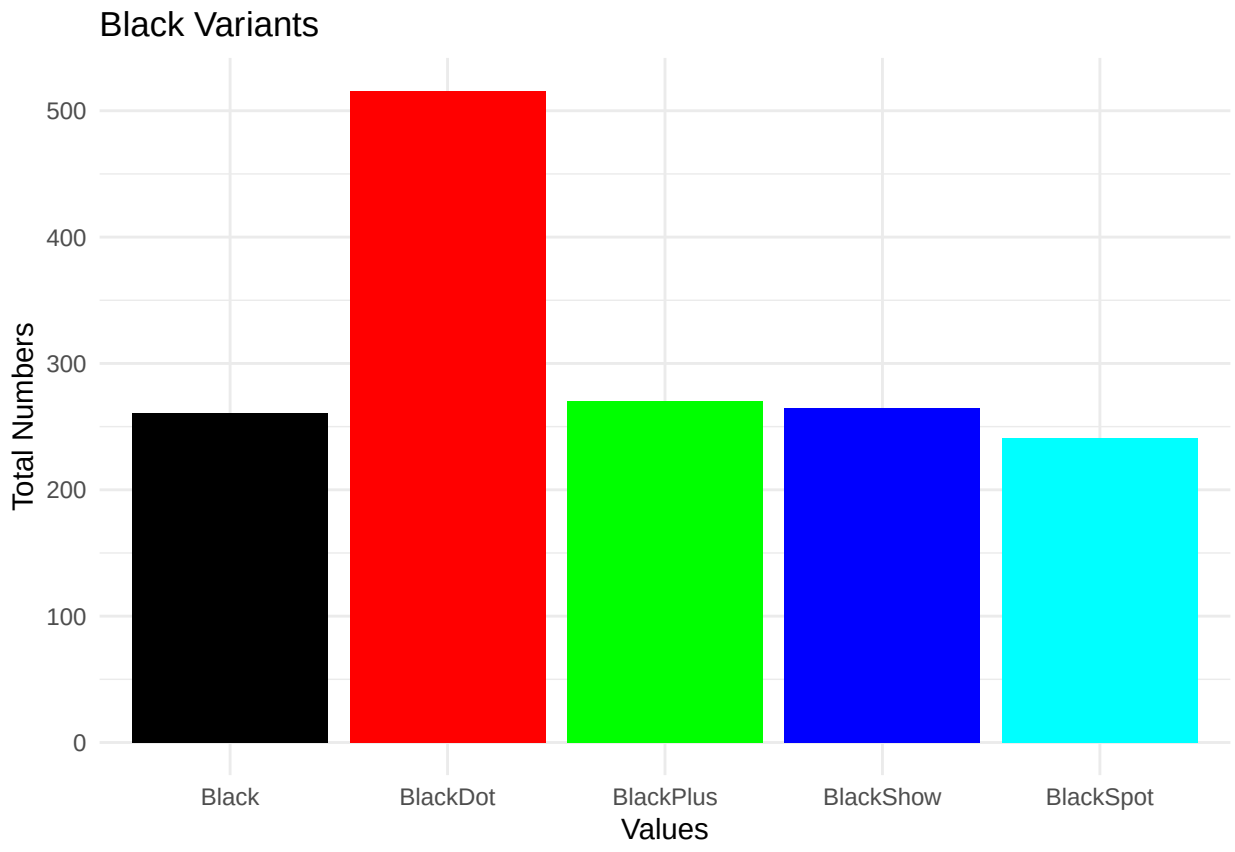
```
## # A tibble: 5 x 2
##   variation count
##   <chr>     <int>
## 1 Black       261
## 2 BlackDot    516
## 3 BlackPlus   270
## 4 BlackShow   265
## 5 BlackSpot   241
```

```
print(white_variants)
```

```
## # A tibble: 5 x 2
```

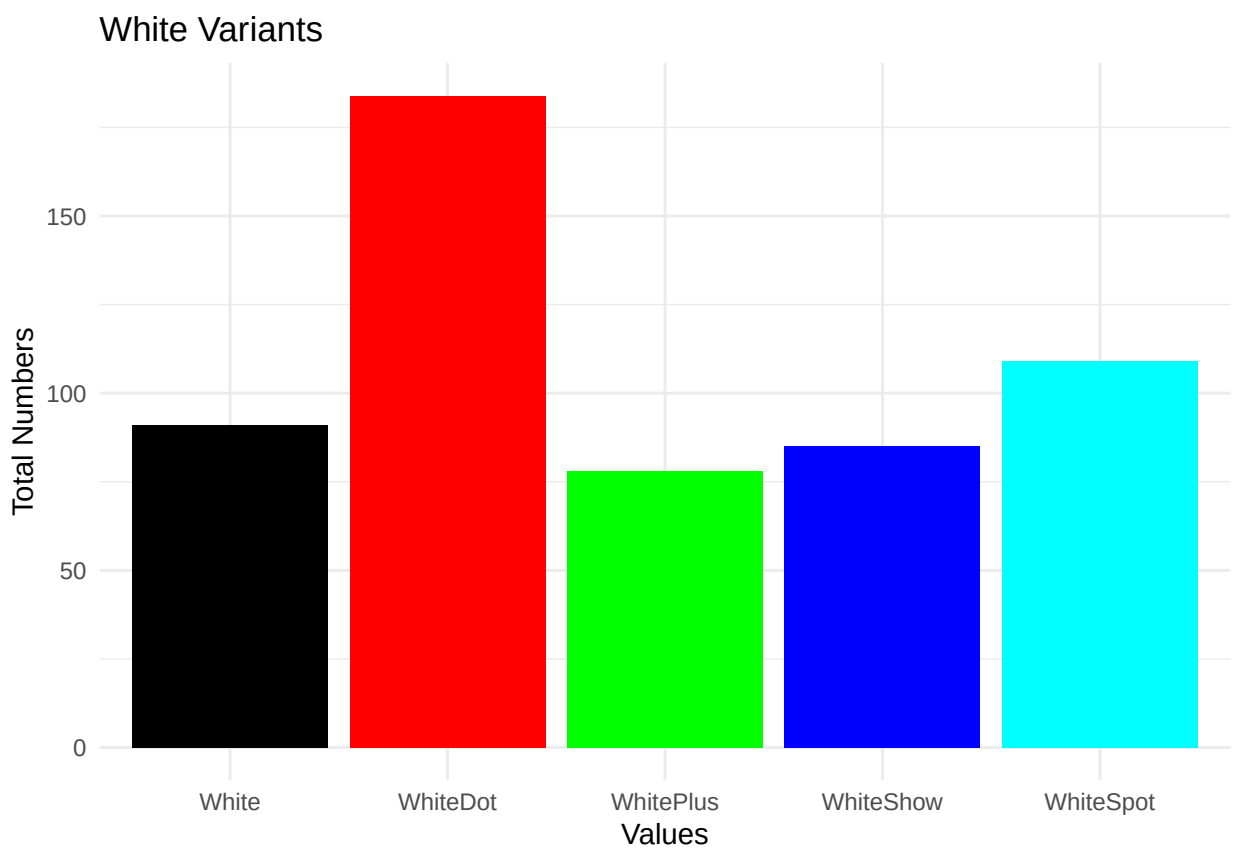
```
##   variation count
##   <chr>     <int>
## 1 White        91
## 2 WhiteDot    184
## 3 WhitePlus    78
## 4 WhiteShow    85
## 5 WhiteSpot   109
```

```r
library(ggplot2)

black_plot <- ggplot(black_variants, aes(x = variation, y = count, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(title = "Black Variants", x = "Values", y = "Total Numbers") +
  theme_minimal() +
  theme(legend.position = "none") +
  scale_fill_manual(values = c("black", "red", "green", "blue", "cyan"))

print(black_plot)
```




```r
white_plot <- ggplot(white_variants, aes(x = variation, y = count, fill = variation)) +
  geom_bar(stat = "identity") +
  labs(title = "White Variants", x = "Values", y = "Total Numbers") +
  theme_minimal() +
  theme(legend.position = "none") +
  scale_fill_manual(values = c("black", "red", "green", "blue", "cyan"))
```

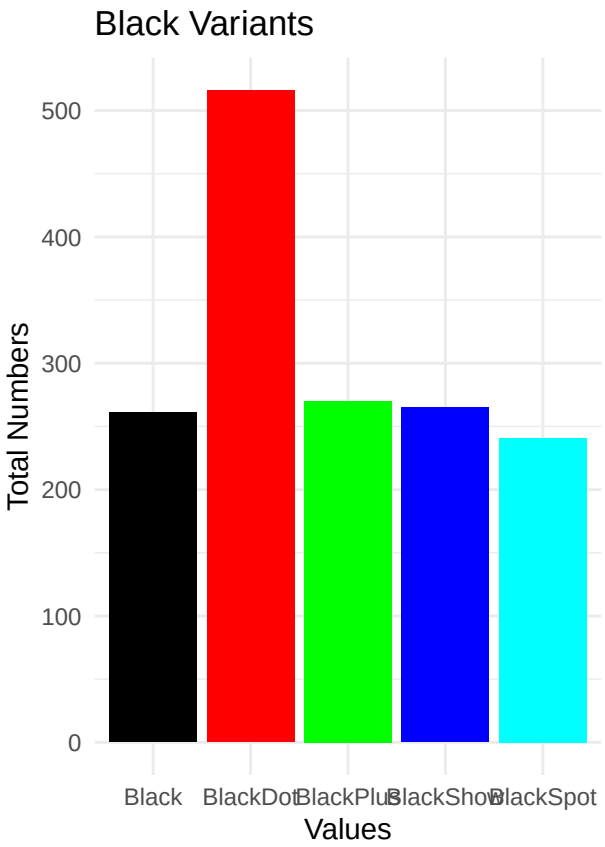
```
print(white_plot)
```

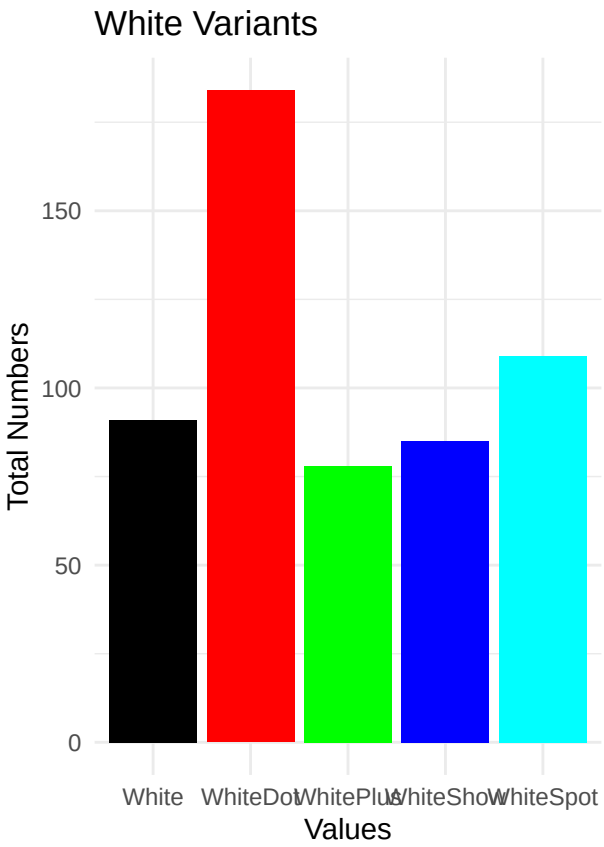



```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
grid.arrange(black_plot, white_plot, ncol = 2)
```


Black Variants
Total Numbers
500
400
300
200
100
0
Black
BlackDot
BlackPlus
BlackShow
BlackSpot
Values


White Variants
Total Numbers
150
100
50
0
White
WhiteDot
WhitePlus
WhiteShow
WhiteSpot
Values