

# RWorksheet\_Pabriaga#4b

John Martin S. Pabriaga

2024-10-28

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

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

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

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

result_matrix
```

```
##      [,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.
matrix_size <- 5
matrix <- matrix(0, nrow = matrix_size, ncol = matrix_size)

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

cat("Transformed Matrix:\n")
```

```
## Transformed Matrix:
```

```
print(matrix)
```

```
##      [,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
```

```

cat("\n")

rows <- 5
cat("Star Pattern:\n")

## Star Pattern:
for (i in 1:rows) {
  cat(rep("*", i), sep = " ", "\n")
}

## *
## * *
## * * *
## * * * *
## * * * * *

first <- as.integer(readline(prompt = "Enter the first number: "))

## Enter the first number:
if (is.na(first)) {
  cat("Please enter a valid integer.\n")
} else {
  cat(first, "", sep = "")
  fibonacci <- c(first)

  repeat {
    if (length(fibonacci) < 2) {
      next_num <- first
    } else {
      next_num <- sum(tail(fibonacci, 2))
    }

    if (next_num > 500) {
      break
    }

    cat(" ", next_num, sep = "")
    fibonacci <- c(fibonacci, next_num)
  }
}

## Please enter a valid integer.
#4.a
household <- read.table("DataFrame.csv", header = TRUE, sep = ",")
print("Data loaded successfully:")

## [1] "Data loaded successfully:"
print(head(household))

##   ShoeSize Height Gender   X X.1 X.2
## 1      6.5   66.0      F NA  NA  NA
## 2      9.0   68.0      F NA  NA  NA
## 3      8.5   64.5      F NA  NA  NA

```

```
## 4      8.5    65.0      F NA  NA  NA
## 5     10.5    70.0      M NA  NA  NA
## 6      7.0    64.0      F NA  NA  NA
```

```
str(household)
```

```
## 'data.frame':  28 obs. of  6 variables:
## $ ShoeSize: num  6.5 9 8.5 8.5 10.5 7 9.5 9 13 7.5 ...
## $ Height  : num  66 68 64.5 65 70 64 70 71 72 64 ...
## $ Gender  : chr  "F" "F" "F" "F" ...
## $ X       : logi  NA NA NA NA NA NA ...
## $ X.1     : logi  NA NA NA NA NA NA ...
## $ X.2     : logi  NA NA NA NA NA NA ...
```

```
#4.b
```

```
male_data <- subset(household, Gender == "M")
female_data <- subset(household, Gender == "F")

male_count <- nrow(male_data)
female_count <- nrow(female_data)

cat("Number of observations in Male:", male_count, "\n")
```

```
## Number of observations in Male: 14
```

```
cat("Number of observations in Female:", female_count, "\n")
```

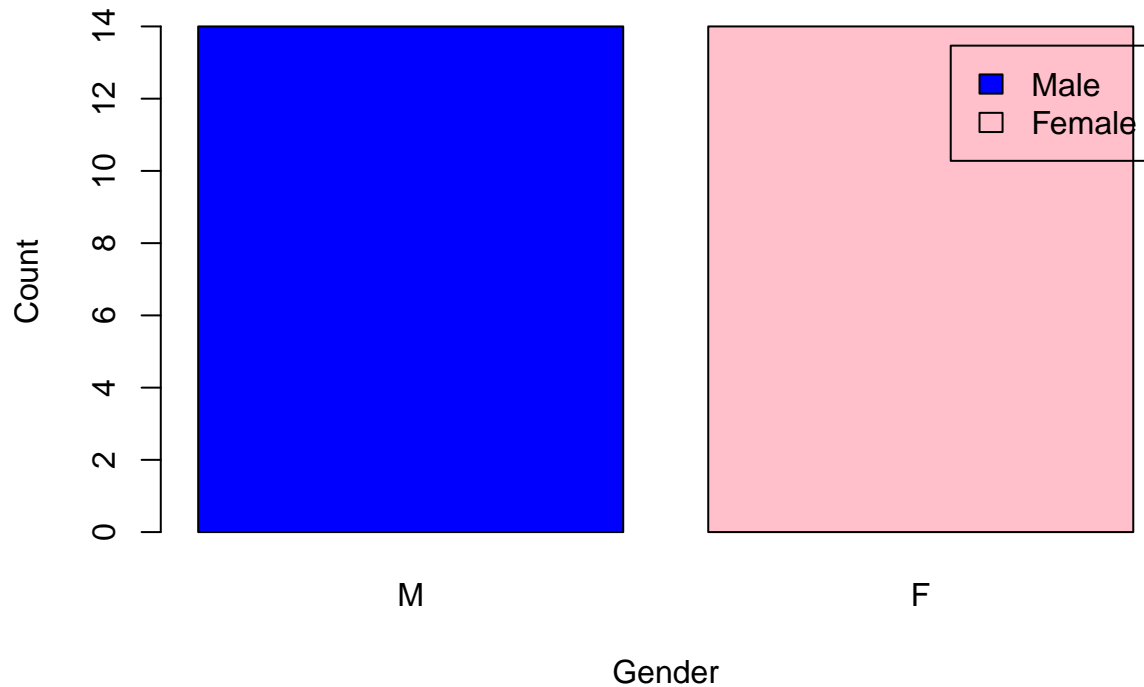
```
## Number of observations in Female: 14
```

```
#4.c
```

```
gender_counts <- c(male_count, female_count)
names(gender_counts) <- c("M", "F")

barplot(gender_counts,
        main = "Number of Males and Females in Household Data",
        xlab = "Gender",
        ylab = "Count",
        col = c("blue", "pink"),
        legend = c("Male", "Female"))
```

## Number of Males and Females in Household Data



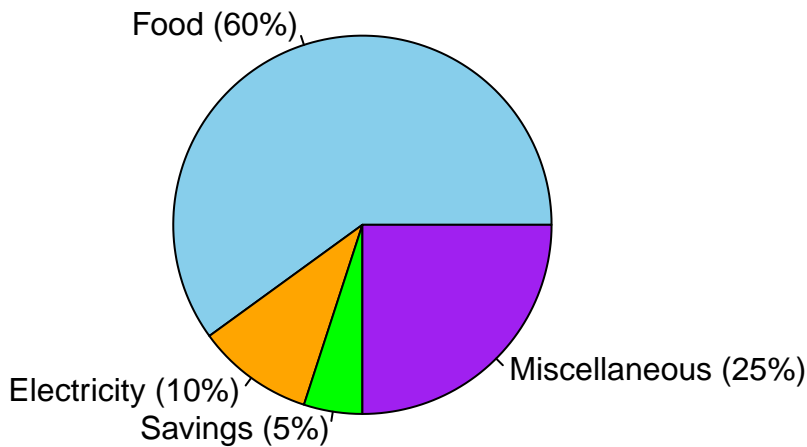
```
#5
expenses <- c(Food = 60, Electricity = 10, Savings = 5, Miscellaneous = 25)
expense_labels <- names(expenses)

percent_labels <- paste0(expense_labels, " (", round(expenses / sum(expenses) * 100), "%)")

colors <- c("skyblue", "orange", "green", "purple")

pie(expenses,
    labels = percent_labels,
    main = "Monthly Expenses of the Dela Cruz Family",
    col = colors)
```

## Monthly Expenses of the Dela Cruz Family



```
#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

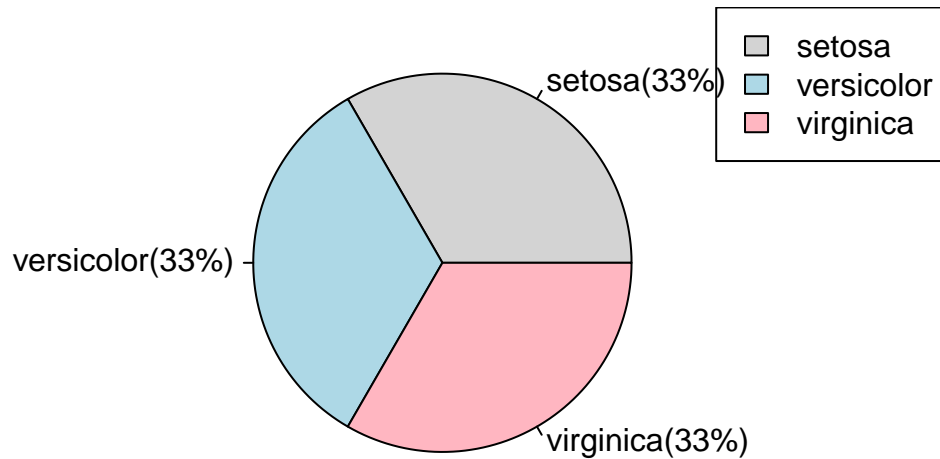
#6.c
species_counts <- table(iris$Species)

colors <- c("lightgray", "lightblue", "lightpink")

pie(species_counts,
    main = "Species Distribution in Iris Dataset",
    col = colors,
    labels = paste(names(species_counts),
                    "(", round(species_counts / sum(species_counts) * 100), "%)", sep = "))

legend("topright", legend = names(species_counts), fill = colors)
```

## Species Distribution in Iris Dataset



#6.d

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

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

```
## Last six rows of Setosa:
```

```
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
```

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

```
##
```

```
## Last six rows of Versicolor:
```

```
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
```

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

```
##
```

```
## Last six rows of Virginica:
```

```
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
```

```
#6.e
```

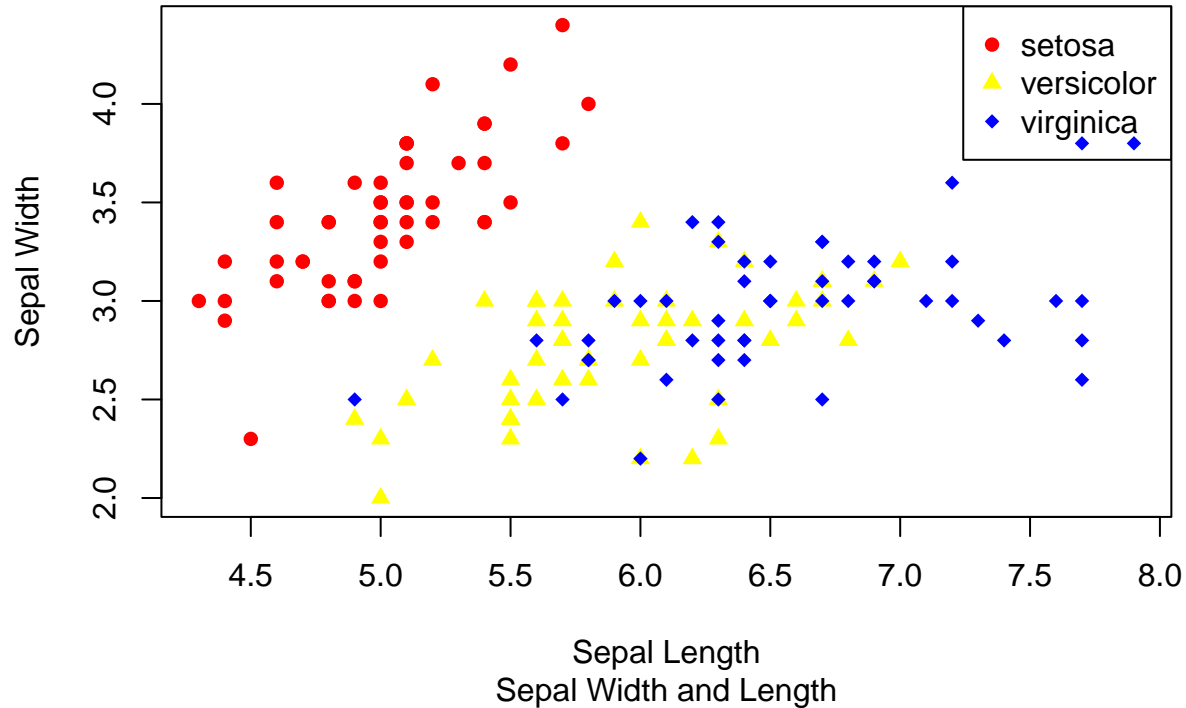
```
iris$Species <- as.factor(iris$Species)
```

```
plot(iris$Sepal.Length, iris$Sepal.Width,
     main = "Iris Dataset",
     sub = "Sepal Width and Length",
     xlab = "Sepal Length",
     ylab = "Sepal Width",
     col = c("red", "yellow", "blue")[iris$Species],
     pch = c(16, 17, 18)[iris$Species])
```

```
# Add legend
```

```
legend("topright", legend = levels(iris$Species),
      col = c("red", "yellow", "blue"), pch = c(16, 17, 18))
```

## Iris Dataset



```
#6.f
```

In the iris dataset, the Sepal.Length versus Sepal.Width scatterplot displays separate groups for the setosa, versicolor, and virginica species. Setosa may be easily distinguished from the other species by its smaller sepal

measures, which often have lower values for both length and width. While Virginica has the biggest sepal dimensions, especially in terms of length, Versicolor falls in the middle, with average sepal diameters. Because each species tends to cluster within particular ranges of these measurements, showing natural groupings based on sepal size, this separation shows that Sepal.Length and Sepal.Width are good criteria for categorizing the species.

```
#7.a
library(readxl)
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

alexa_data <- read_excel("alexa_file.xlsx")

print(head(alexa_data))

## # A tibble: 6 x 5
##   rating date                Variant      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 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

print(colnames(alexa_data))

## [1] "rating"      "date"        "Variant"     "verified_reviews"
## [5] "feedback"

if ("Variant" %in% colnames(alexa_data)) {
  alexa_data$Variant <- gsub("^\\s+|\\s+$", "", alexa_data$Variant)
  alexa_data$Variant <- gsub("Black Dot", "BlackDot", alexa_data$Variant)
  alexa_data$Variant <- gsub("Black Plus", "BlackPlus", alexa_data$Variant)
  alexa_data$Variant <- gsub("Black Show", "BlackShow", alexa_data$Variant)
  alexa_data$Variant <- gsub("Black Spot", "BlackSpot", alexa_data$Variant)
  alexa_data$Variant <- gsub("White Dot", "WhiteDot", alexa_data$Variant)
  alexa_data$Variant <- gsub("White Plus", "WhitePlus", alexa_data$Variant)
  alexa_data$Variant <- gsub("White Show", "WhiteShow", alexa_data$Variant)
  alexa_data$Variant <- gsub("White Spot", "WhiteSpot", alexa_data$Variant)

  print(head(alexa_data))
} else {
  stop("The specified column 'Variant' does not exist in the dataframe.")
}

## # A tibble: 6 x 5
##   rating date                Variant      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 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
```

#7.b

```
if ("Variant" %in% colnames(alexa_data)) {
  alexa_data$Variant <- gsub("^\\s+|\\s+$", "", alexa_data$Variant)
  variations_count <- alexa_data %>%
    count(Variant)
  print(variations_count)

  save(variations_count, file = "variations.RData")
} else {
  stop("The specified column 'variant' does not exist in the dataframe.")
}
```

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

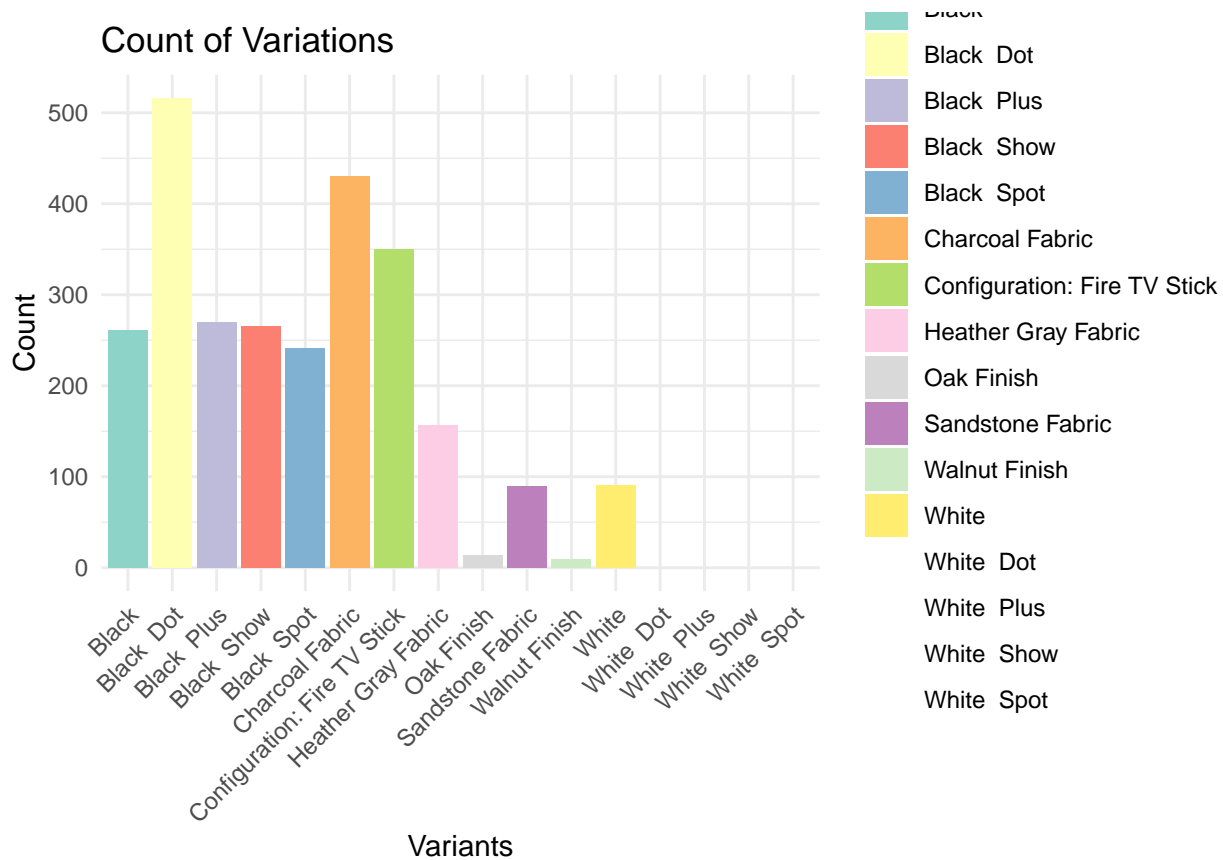
#7.c

```
library(ggplot2)

load("variations.RData")

ggplot(variations_count, aes(x = Variant, y = n, fill = Variant)) +
  geom_bar(stat = "identity") +
  labs(title = "Count of Variations",
       x = "Variants",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_brewer(palette = "Set3")
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set3 is 12
## Returning the palette you asked for with that many colors
```



```
#7.d
library(gapminder)
library(dplyr)
library(ggplot2)

load("variations.RData")

black_Variations <- variations_count %>%
  filter(grepl("Black", Variant)) %>%
  summarise(n = sum(n)) %>%
  mutate(color = "Black")

white_Variations <- variations_count %>%
  filter(grepl("White", Variant)) %>%
  summarise(n = sum(n)) %>%
  mutate(color = "White")

combined_variations <- rbind(black_Variations, white_Variations)

ggplot(combined_variations, aes(x = color, y = n, fill = color)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Count of Black and White Variations",
       x = "Variants",
       y = "Count") +
  theme_minimal() +
  scale_fill_manual(values = c("Black" = "black", "White" = "gray")) +
```

```
theme(axis.text.x = element_text(angle = 0, hjust = 0.5))
```

