RWorksheet_Punay#4b

#1

```
matrixA <- matrix(0, nrow = 5, ncol = 5)

vectorA <- 1:5

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

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

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

#3

```
Fibo_seq <- function() {
  cat("Enter the starting number for Fibonacci sequence: ")
  x <- as.integer(readline())

  if (is.na(x) || x < 1) {
    return("Please enter a positive integer")
  }

  fib <- numeric(0)

  if (x >= 1) fib[1] <- x
  if (x >= 2) fib[2] <- x
```

```r
  i <- 3
  repeat {
    next_fib <- fib[i-1] + fib[i-2]
    if (next_fib > 500) break
    fib[i] <- next_fib
    i <- i + 1
  }

  return(fib)
}


 result <- Fibo_seq()
```

```
## Enter the starting number for Fibonacci sequence:
```

```r
 cat("Fibonacci sequence up to 500:\n")
```

```
## Fibonacci sequence up to 500:
```

```r
print(result)
```

```
## [1] "Please enter a positive integer"
```

#4

```r
shoe_size <- c(6.5, 9.0, 8.5, 8.5, 10.5, 7.0, 9.5, 9.0, 13.0, 7.5, 10.5, 8.5, 12.0, 10.5, 13.0 ,11.5, 8

height <- c(66.0, 68.0, 64.5, 65.0, 70.0, 64.0, 70.0, 71.0, 72.0, 64.0, 74.5, 67.0, 71.0, 71.0, 77.0, 7

gender <- c("F", "F", "F", "F", "M", "F", "F", "F", "M", "F", "M", "F", "M", "M", "M", "M", "F", "F", "



daf <- data.frame(shoe_size, height, gender)
daf
```

```
##    shoe_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
## 7        9.5   70.0      F
## 8        9.0   71.0      F
## 9       13.0   72.0      M
## 10       7.5   64.0      F
## 11      10.5   74.5      M
## 12       8.5   67.0      F
## 13      12.0   71.0      M
## 14      10.5   71.0      M
## 15      13.0   77.0      M
## 16      11.5   72.0      M
## 17       8.5   59.0      F
## 18       5.0   62.0      F
```

```
## 19      10.0   72.0      M
## 20      65.0   66.0      F
## 21      75.0   64.0      F
## 22       8.5   67.0      M
## 23      10.5   73.0      M
## 24       8.5   69.0      F
## 25      10.5   72.0      M
## 26      11.0   70.0      M
## 27       9.0   69.0      M
## 28      13.0   70.0      M
```

```r
write.csv(daf, "daf.csv", row.names = FALSE)

read.csv("daf.csv")
```

```
##    shoe_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
## 7        9.5   70.0      F
## 8        9.0   71.0      F
## 9       13.0   72.0      M
## 10       7.5   64.0      F
## 11      10.5   74.5      M
## 12       8.5   67.0      F
## 13      12.0   71.0      M
## 14      10.5   71.0      M
## 15      13.0   77.0      M
## 16      11.5   72.0      M
## 17       8.5   59.0      F
## 18       5.0   62.0      F
## 19      10.0   72.0      M
## 20      65.0   66.0      F
## 21      75.0   64.0      F
## 22       8.5   67.0      M
## 23      10.5   73.0      M
## 24       8.5   69.0      F
## 25      10.5   72.0      M
## 26      11.0   70.0      M
## 27       9.0   69.0      M
## 28      13.0   70.0      M
```

```r
daf
```

```
##    shoe_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
## 7        9.5   70.0      F
## 8        9.0   71.0      F
```

```
## 9       13.0   72.0      M
## 10       7.5   64.0      F
## 11      10.5   74.5      M
## 12       8.5   67.0      F
## 13      12.0   71.0      M
## 14      10.5   71.0      M
## 15      13.0   77.0      M
## 16      11.5   72.0      M
## 17       8.5   59.0      F
## 18       5.0   62.0      F
## 19      10.0   72.0      M
## 20      65.0   66.0      F
## 21      75.0   64.0      F
## 22       8.5   67.0      M
## 23      10.5   73.0      M
## 24       8.5   69.0      F
## 25      10.5   72.0      M
## 26      11.0   70.0      M
## 27       9.0   69.0      M
## 28      13.0   70.0      M
```

```r
#b.
male_dat <- subset(daf, gender == "M")
female_dat <- subset(daf, gender == "F")

male_dat
```

```
##    shoe_size height gender
## 5       10.5   70.0      M
## 9       13.0   72.0      M
## 11      10.5   74.5      M
## 13      12.0   71.0      M
## 14      10.5   71.0      M
## 15      13.0   77.0      M
## 16      11.5   72.0      M
## 19      10.0   72.0      M
## 22       8.5   67.0      M
## 23      10.5   73.0      M
## 25      10.5   72.0      M
## 26      11.0   70.0      M
## 27       9.0   69.0      M
## 28      13.0   70.0      M
```

```r
female_dat
```

```
##    shoe_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
## 6        7.0   64.0      F
## 7        9.5   70.0      F
## 8        9.0   71.0      F
## 10       7.5   64.0      F
## 12       8.5   67.0      F
## 17       8.5   59.0      F
```

```
## 18       5.0   62.0       F
## 20      65.0   66.0       F
## 21      75.0   64.0       F
## 24       8.5   69.0       F
```

```
count_f <- nrow(female_dat)
count_m <- nrow(male_dat)

print(paste("Number of Female observations:", count_f, "\n"))
```

```
## [1] "Number of Female observations: 14 \n"
```

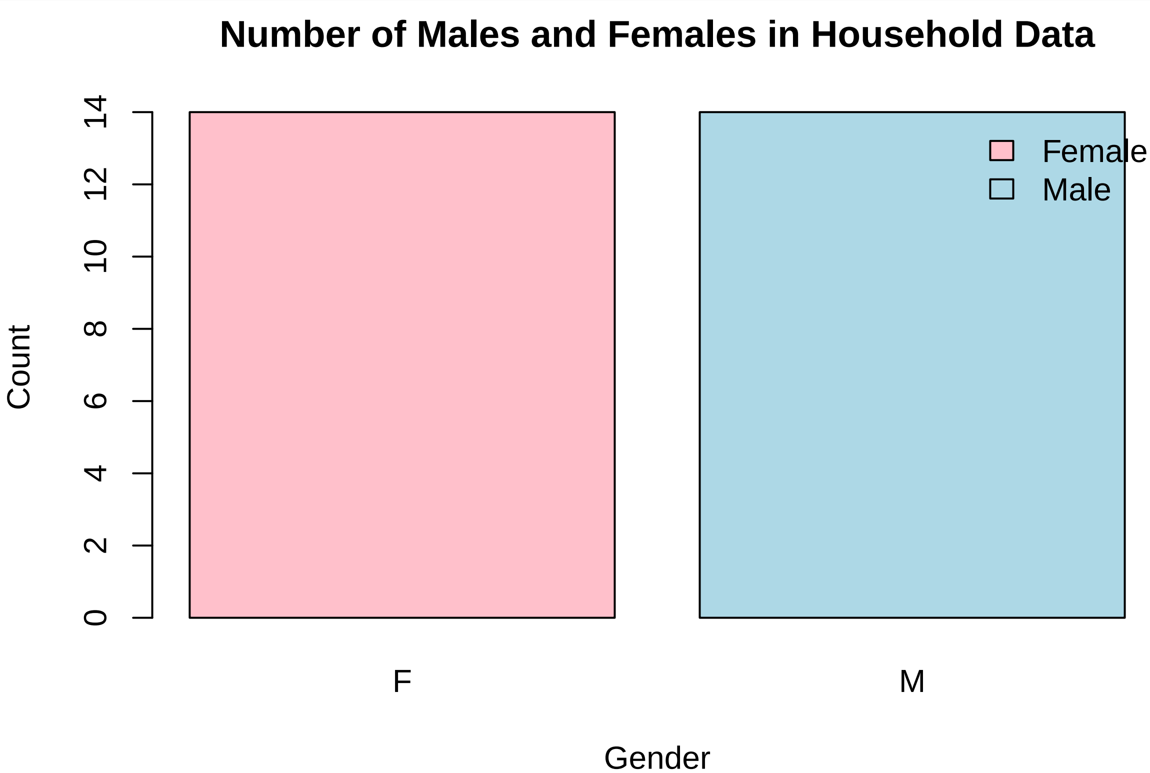
```
print(paste("Number of Male observations:", count_m, "\n"))
```

```
## [1] "Number of Male observations: 14 \n"
```

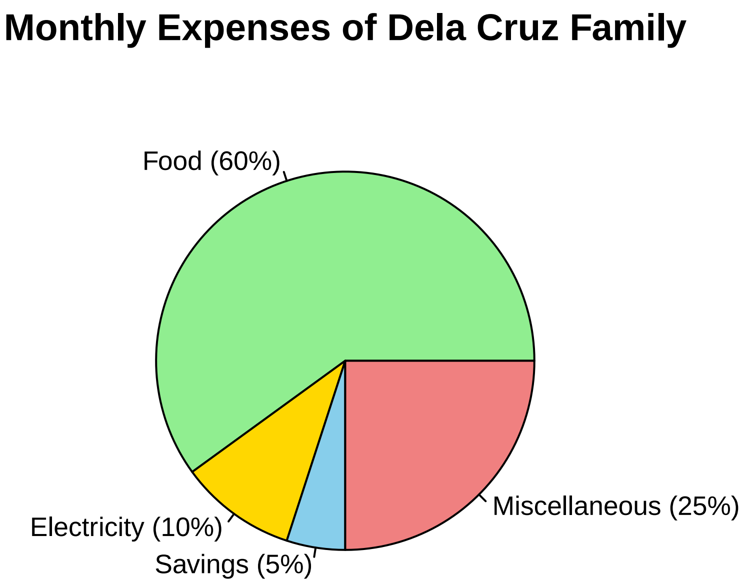
```
#c.
gender_count <- table(daf$gender)

gender_count
```

```
##
##  F  M
## 14 14
```

```
barplot(gender_count,
        main = "Number of Males and Females in Household Data",
        xlab = "Gender",
        ylab = "Count",
        col = c("pink", "lightblue"),
        legend.text = c("Female", "Male"),
        args.legend = list(x = "topright", bty = "n"))
```



```r
#5
expenses <- c(60, 10, 5, 25)
categories <- c("Food", "Electricity", "Savings", "Miscellaneous")

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


pie(expenses,
    labels = percent_labels,
    col = c("lightgreen", "gold", "skyblue", "lightcoral"),
    main = "Monthly Expenses of Dela Cruz Family",
    cex = 0.8)
```



#6

```r
#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 ...
```

The iris dataset is a data frame with 150 observations and 5 variables.

Variables include:

Sepal.Length (numeric)

Sepal.Width (numeric)

Petal.Length (numeric)

Petal.Width (numeric)

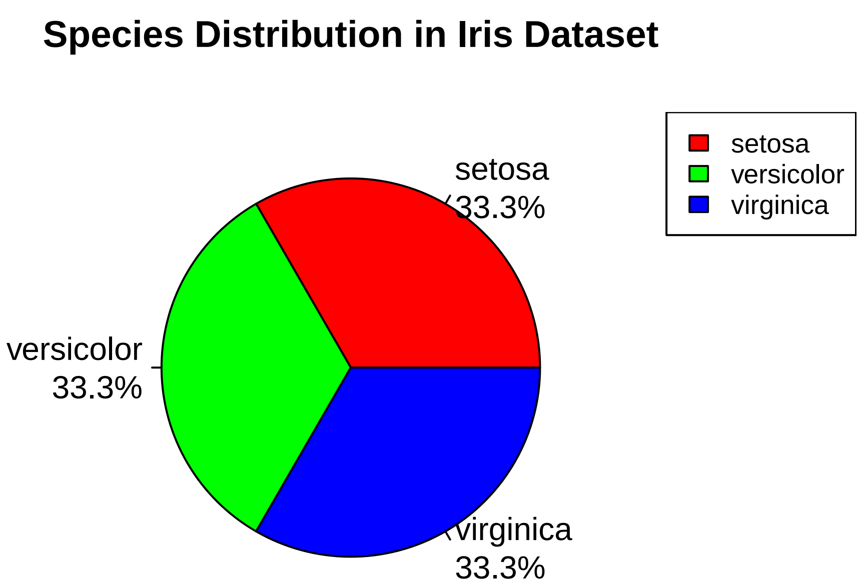
Species (factor with 3 levels: setosa, versicolor, virginica)

Each row represents one flower observation with measurements and species category.

```r
#b.
mean_values <- colMeans(iris[, 1:4])
mean_values
```

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

```r
#c.

species_count <- table(iris$Species)

# Percentages
species_percent <- round(prop.table(species_count) * 100, 1)


labels_species <- paste(names(species_count), "\n", species_percent, "%", sep = "")


colors <- c("red", "green", "blue")

# Pie chart
pie(species_count,
    main = "Species Distribution in Iris Dataset",
    col = colors,
    labels = labels_species)

legend("topright", legend = names(species_count), fill = colors, cex = 0.8)
```




```r
#d.
setosa <- subset(iris, Species == "setosa")
versicolor <- subset(iris, Species == "versicolor")
virginica <- subset(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
```

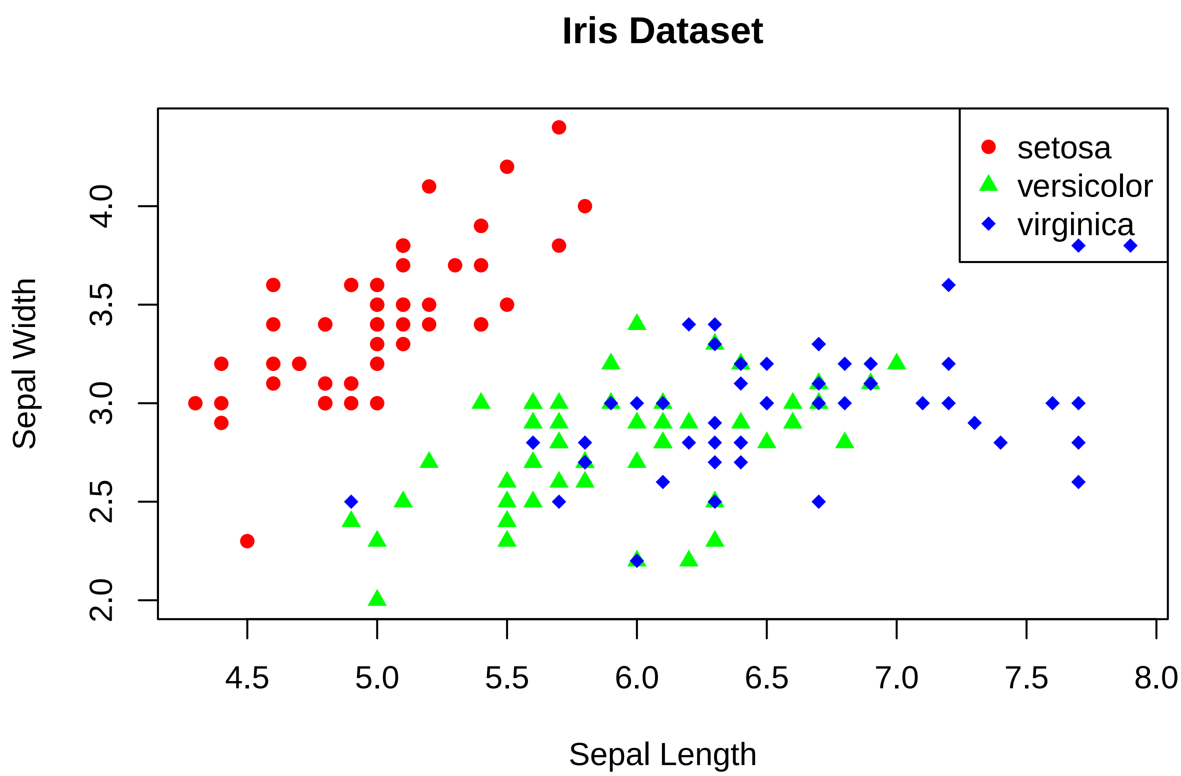
```
tail(versicolor)
```

```
##     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
```

```
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
```

```
#e.

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


colors <- c("setosa" = "red", "versicolor" = "green", "virginica" = "blue")
pch_values <- c("setosa" = 16, "versicolor" = 17, "virginica" = 18)

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

legend("topright", legend = levels(iris$Species), col = colors, pch = pch_values)
```



Sepal width and length

#f.

The scatterplot shows clear grouping based on species.

Setosa (red circles) has generally smaller sepal length and larger sepal width.

Versicolor (green triangles) and Virginica (blue diamonds) show some overlap but tend to have larger sepal length.

The different symbols and colors make it easier to distinguish species visually.

This supports that sepal measurements can be useful for classifying iris species.

```
#7
library(readxl)


alexa <- read_excel("alexa_file.xlsx")


print(alexa)
```

```
## # 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
```

```r
#a
unique(alexa$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"
```

```r
#black variants

# Fix leading/trailing/multiple spaces
alexa$variation <- trimws(gsub("\\s+", " ", alexa$variation))

# Standardize Black and White variants
alexa$variation <- gsub("^Black.*Dot$",  "Black Dot",  alexa$variation)
alexa$variation <- gsub("^Black.*Plus$", "Black Plus", alexa$variation)
alexa$variation <- gsub("^Black.*Show$", "Black Show", alexa$variation)
alexa$variation <- gsub("^Black.*Spot$", "Black Spot", alexa$variation)

alexa$variation <- gsub("^White.*Dot$",  "White Dot",  alexa$variation)
alexa$variation <- gsub("^White.*Plus$", "White Plus", alexa$variation)
alexa$variation <- gsub("^White.*Show$", "White Show", alexa$variation)
alexa$variation <- gsub("^White.*Spot$", "White Spot", alexa$variation)


unique(alexa$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"
```

```r
#Snippet of output

knitr::include_graphics("image_2025-11-30_153211021.png")
```

Source Visual Outline

```
 [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"
```

```r
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.5'
## (as 'lib' is unspecified)
```

```r
library(dplyr)
```

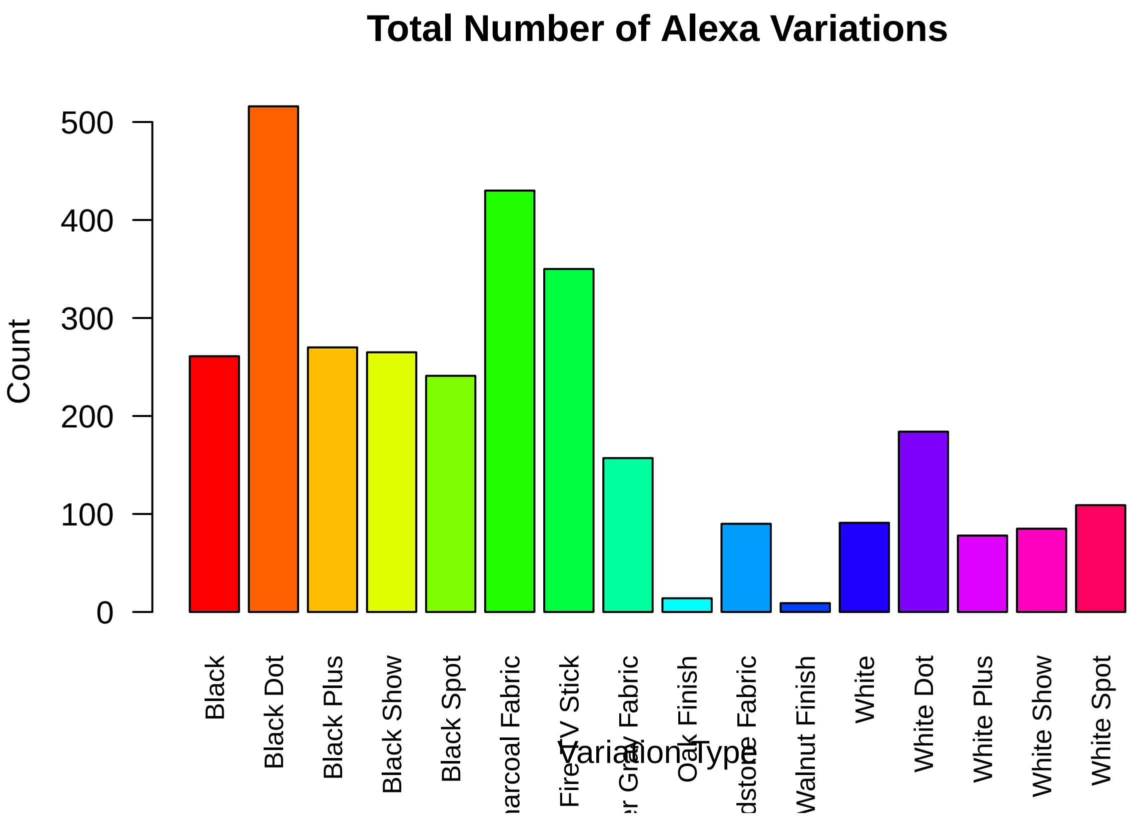
```
##
## Attaching package: 'dplyr'

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

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

```r
variations <- alexa %>%
  count(variation)

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

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
```

```r
#c
load("variations.RData")
counts <- variations$n
labels <- variations$variation

barplot(
  counts,
  names.arg = labels,
  col = rainbow(length(counts)),
  main = "Total Number of Alexa Variations",
  xlab = "Variation Type",
  ylab = "Count",
  las = 2,           # rotate labels for readability
  cex.names = 0.8    # shrink label text if long
)
```




```r
#d.
black_data <- variations[grep("^Black", variations$variation), ]

white_data <- variations[grep("^White", variations$variation), ]


par(mfrow = c(1, 2))

barplot(
  black_data$n,
```

```
  names.arg = black_data$variation,
  col = c("black", "firebrick2", "chartreuse3", "dodgerblue2", "cyan3"),
  main = "Black Variants",
  xlab = "Total Numbers",
  ylab = "Variants",
  las = 2,
  cex.names = 0.8
)

barplot(
  white_data$n,
  names.arg = white_data$variation,
  col = c("black", "firebrick2", "chartreuse3", "dodgerblue2", "cyan3"),
  main = "White Variants",
  xlab = "Total Numbers",
  ylab = "Variants",
  las = 2,
  cex.names = 0.8
)
```