

worksheet__Arlante#4b

Stephanie Lois Amber Arlante

2024-10-31

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

for (i in 1:5) {
  matrix_result[i, ] <- abs(vectorA - i)
}
print(matrix_result)
```

```
##      [,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) {
  cat(rep("*", i), "\n")
}
```

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

```
#3.
start <- 5 # Change this number as needed

a <- 0
b <- 1

fibonacci_sequence <- c()

for (i in 1:500) {
  if (b >= start) {
    fibonacci_sequence <- c(fibonacci_sequence, b)
  }

  next_term <- a + b

  if (next_term > 500) {
    break
  }
}
```

```

}

a <- b
b <- next_term
}

cat("Fibonacci sequence starting from", start, "up to 500:\n")

## Fibonacci sequence starting from 5 up to 500:
print(fibonacci_sequence)

## [1] 5 8 13 21 34 55 89 144 233 377

#a.
shoes <- read.csv("shoes.csv")

head(shoes)

## Shoe.Size Height Gender Shoe.Size.1 Height.1 Gender.1
## 1 6.5 66.0 F 13.0 77 M
## 2 9.0 68.0 F 11.5 72 M
## 3 8.5 64.5 F 8.5 59 F
## 4 8.5 65.0 F 5.0 62 F
## 5 10.5 70.0 M 10.0 72 M
## 6 7.0 64.0 F 6.5 66 F

#b.
male_data <- subset(shoes, Gender == "M")
female_data <- subset(shoes, Gender == "F")

num_male <- nrow(male_data)
num_female <- nrow(female_data)

cat("Number of Male observations:", num_male, "\n")

## Number of Male observations: 5

cat("Number of Female observations:", num_female, "\n")

## Number of Female observations: 9

#c.
num_male <- nrow(subset(shoes, Gender == "M"))
num_female <- nrow(subset(shoes, Gender == "F"))

gender_counts <- c(num_male, num_female)

gender_labels <- c("Male", "Female")
colors <- c("skyblue", "pink")

barplot(
  gender_counts,
  names.arg = gender_labels,
  col = colors,
  main = "Number of Males and Females in Household Data",
  xlab = "Gender",

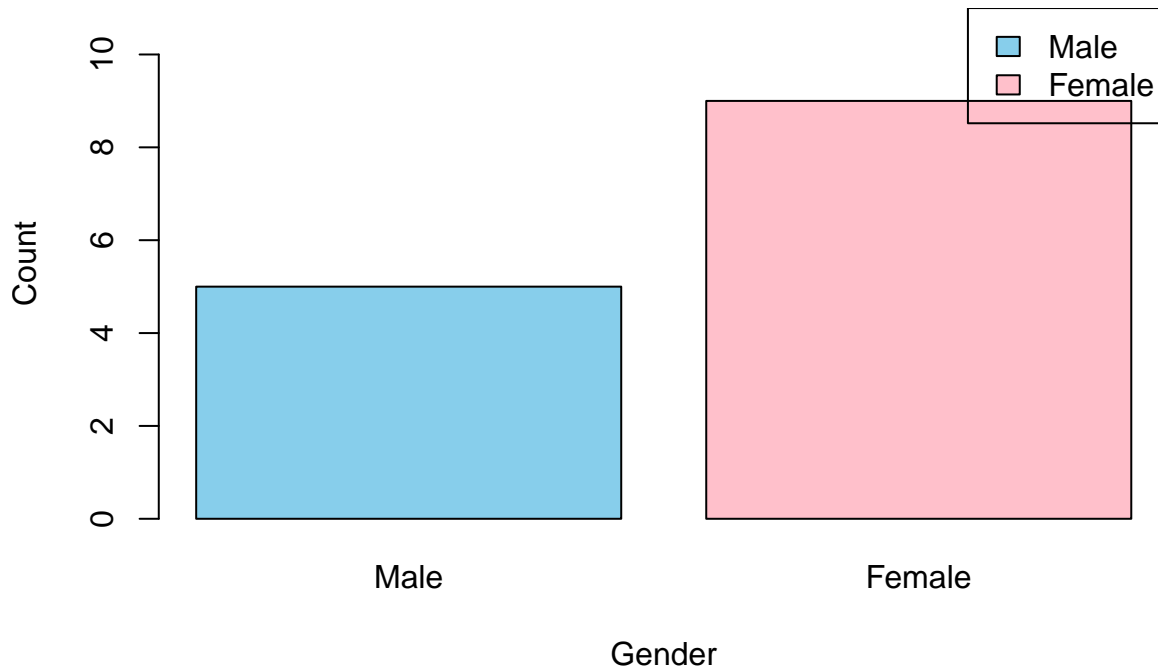
```

```

ylab = "Count",
ylim = c(0, max(gender_counts) + 2)
)
legend("topright", legend = gender_labels, fill = colors)

```

Number of Males and Females in Household Data

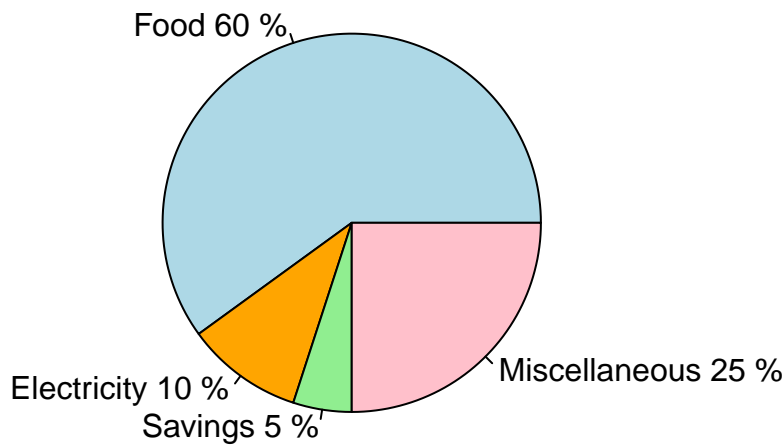


```

#5.
expenses <- c(60, 10, 5, 25)
categories <- c("Food", "Electricity", "Savings", "Miscellaneous")
colors <- c("lightblue", "orange", "lightgreen", "pink")
percentages <- round(expenses / sum(expenses) * 100, 1)
labels <- paste(categories, percentages, "%")
pie(
  expenses,
  labels = labels,
  col = colors,
  main = "Dela Cruz Family Monthly Expenditure"
)

```

Dela Cruz Family Monthly Expenditure



#6a.

Load the iris dataset

```
data(iris)
```

Check the structure of the dataset

```
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 of the Output:The iris dataset is one of the most widely used datasets for machine learning.

#1.Sepal.Length: The length of the sepal in centimeters (numeric).

#2.Sepal.Width: The width of the sepal in centimeters (numeric).

#3.Petal.Length: The length of the petal in centimeters (numeric).

#4.Petal.Width: The width of the petal in centimeters (numeric).

#5.Species: A categorical variable (factor) representing the flower species, with three possible values

#-setosa

#-versicolor

#-virginica

#Each observation corresponds to a unique flower, and the measurements capture the physical dimensions

#6b.

```
data(iris)
```

```
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
```

#Result: The iris dataset is built into R, so you can use it directly and the colMeans() function compu

```
#6c.
data(iris)

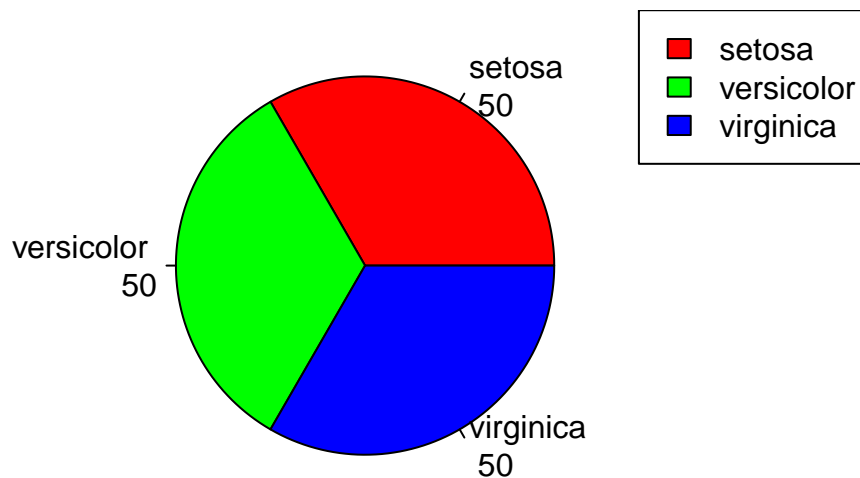
species_count <- table(iris$Species)

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

pie(species_count,
    main = "Species Distribution in Iris Dataset",
    col = colors,
    labels = paste(names(species_count), "\n", species_count))

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

Species Distribution in Iris Dataset



```
#6d.
data(iris)

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

# Display the last six rows of each species
last_six_setosa <- tail(setosa, 6)
last_six_versicolor <- tail(versicolor, 6)
last_six_virginica <- tail(virginica, 6)

# Print the results
cat("Last six rows of Setosa:\n")

## Last six rows of Setosa:
print(last_six_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:
```

```
print(last_six_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:
```

```
print(last_six_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
```

```
#6e.
```

```
data(iris)
```

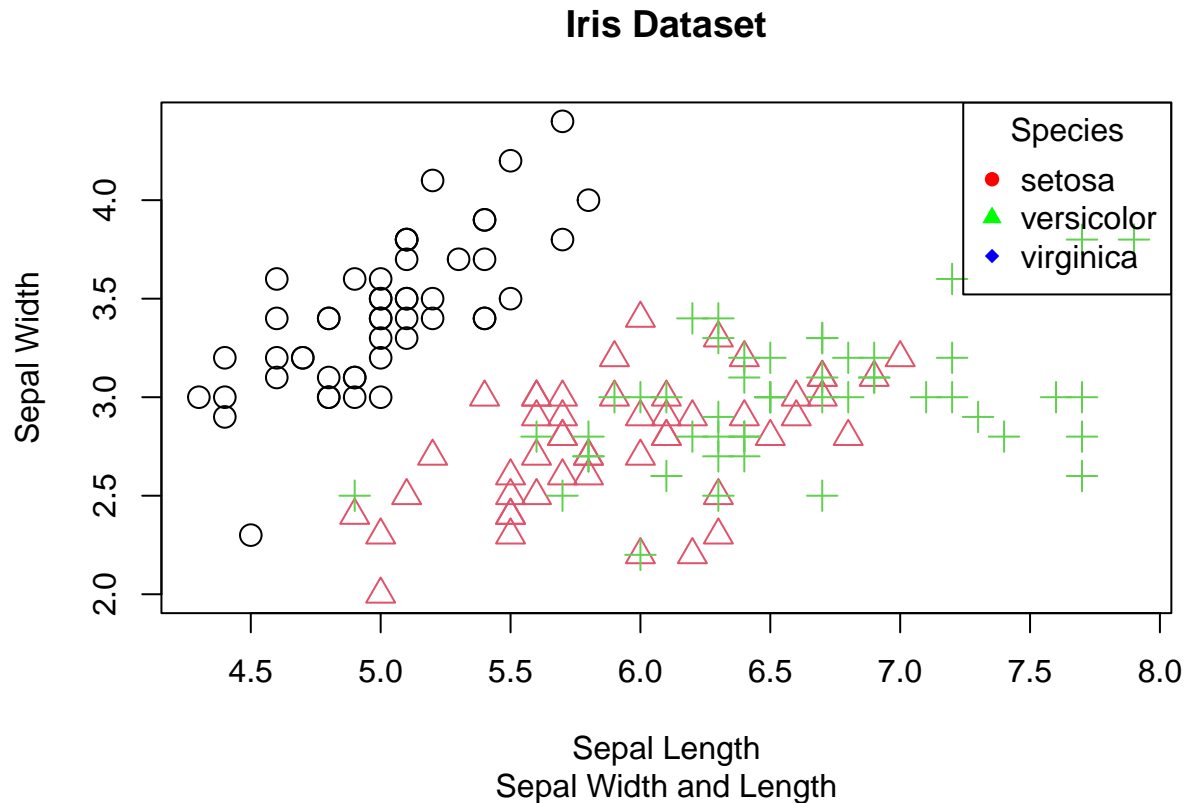
```
iris$Species <- as.factor(iris$Species)
```

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

```
pch_values <- c(16, 17, 18) # Different pch symbols
```

```
plot(iris$Sepal.Length, iris$Sepal.Width,
     main = "Iris Dataset",
     sub = "Sepal Width and Length",
     xlab = "Sepal Length",
     ylab = "Sepal Width",
     col = iris$Species, # Color by species
     pch = as.numeric(iris$Species), # Point shapes by species
     cex = 1.5) # Size of points
```

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



#6f. Interpretation

#1. Species Separation:

#Setosa: The red dots represent the Setosa species, and they are clustered together in the lower left p

#Versicolor: The green triangles represent Versicolor. These points are spread out in the middle of the

#Virginica: The blue diamonds represent Virginica, and they are located in the upper right area. This m

#2. Trends:

#You can see a clear trend where both the sepal length and width get larger from Setosa to Virginica. T

#3. Variability:

#The spread of the points shows that there is some variation in the sepal sizes within each species. Fo

#4. Implications for Classification:

#The clear separation of the Setosa species suggests that it could be easily identified just by looking

#Conclusion

#In summary, the scatterplot shows how sepal length and width differ among the three species of irises.

#I gained help from ChatGPT in typing and creating this text. I can assure that I understand the contex

#Basic Cleaning and Transformation of Objects.

#7a.

```
if (!require("readxl")) install.packages("readxl")
```

```
## Loading required package: readxl
```

```

library(readxl)

df <- read_excel("alexa_file.xlsx")

head(df)

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

#7b.
if (!require("dplyr")) install.packages("dplyr")

## Loading required package: 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

library(dplyr)

variation_counts <- df %>%
  count(variation)

print(variation_counts)

## # 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

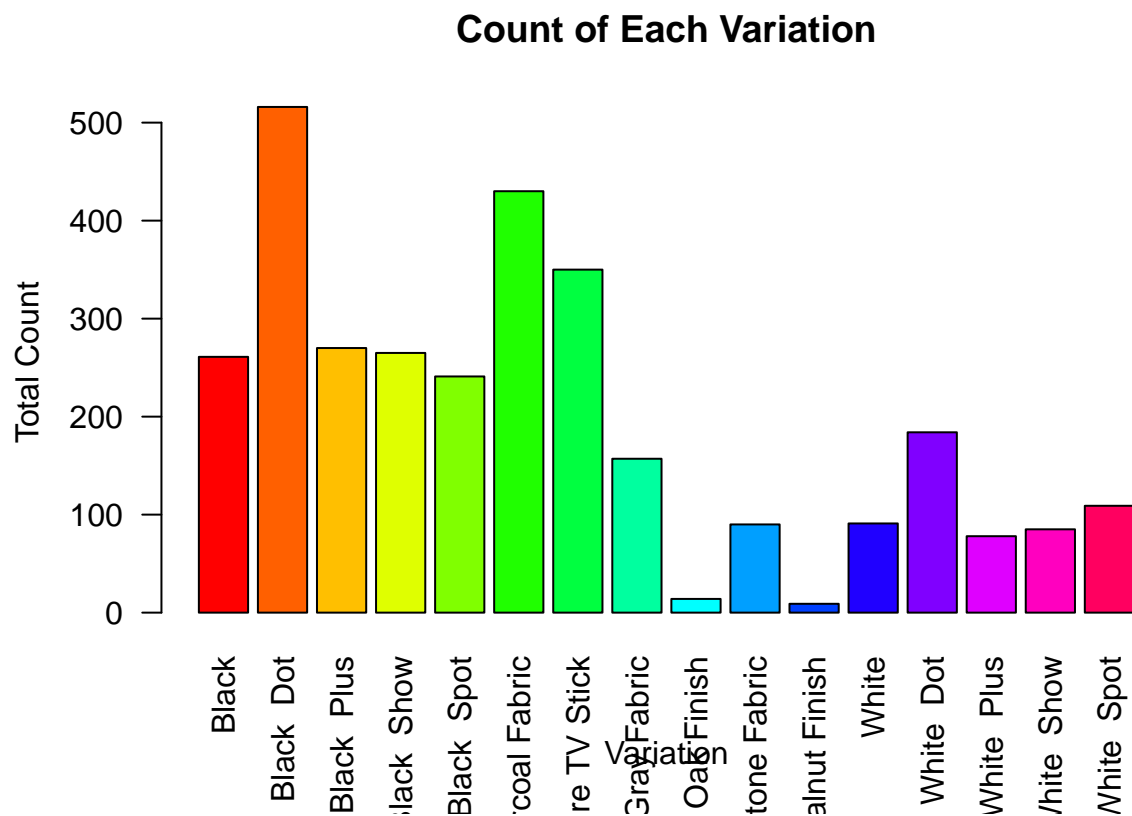
109

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

#7c.

```
load("variations.RData")
```

```
barplot(
  variation_counts$n,
  names.arg = variation_counts$variation,
  main = "Count of Each Variation",
  xlab = "Variation",
  ylab = "Total Count",
  col = rainbow(length(variation_counts$n)),
  las = 2
)
```



#7d.

```
black_white_counts <- subset(variation_counts, grepl("Black|White", variation_counts$variation))
```

```
barplot(
  black_white_counts$n,
  names.arg = black_white_counts$variation,
  main = "Count of Black and White Variations",
  xlab = "Variation",
  ylab = "Total Count",
  col = c("black", "gray"),
  beside = TRUE
)
```

