

RWorksheet_Liza#4b

Jowelyn Liza

2024-12-18

```
# 1
matrix_result <- matrix(0, nrow = 5, ncol = 5)
A <- c(1, 2, 3, 4, 5)
for (i in 1:5) {
  for (j in 1:5) {
    matrix_result[i, j] <- abs(i - j)
  }
}
```

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")
}
```

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

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

Enter the first number:

```
if (is.na(first)) {
  cat("Enter an 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) {
```

```

1
break
}
cat(" ", next_num, sep = "")
fibonacci <- c(fibonacci, next_num)
}
}

## Enter an integer.

# 4a
library(readr)
sample_data <- read_csv("sample_data.csv")

## Rows: 28 Columns: 3
## -- Column specification -----
## Delimiter: ","
## chr (1): Gender
## dbl (2): ShoeSize, Height
##
## 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.
data <- read_csv("sample_data.csv")
head(data)

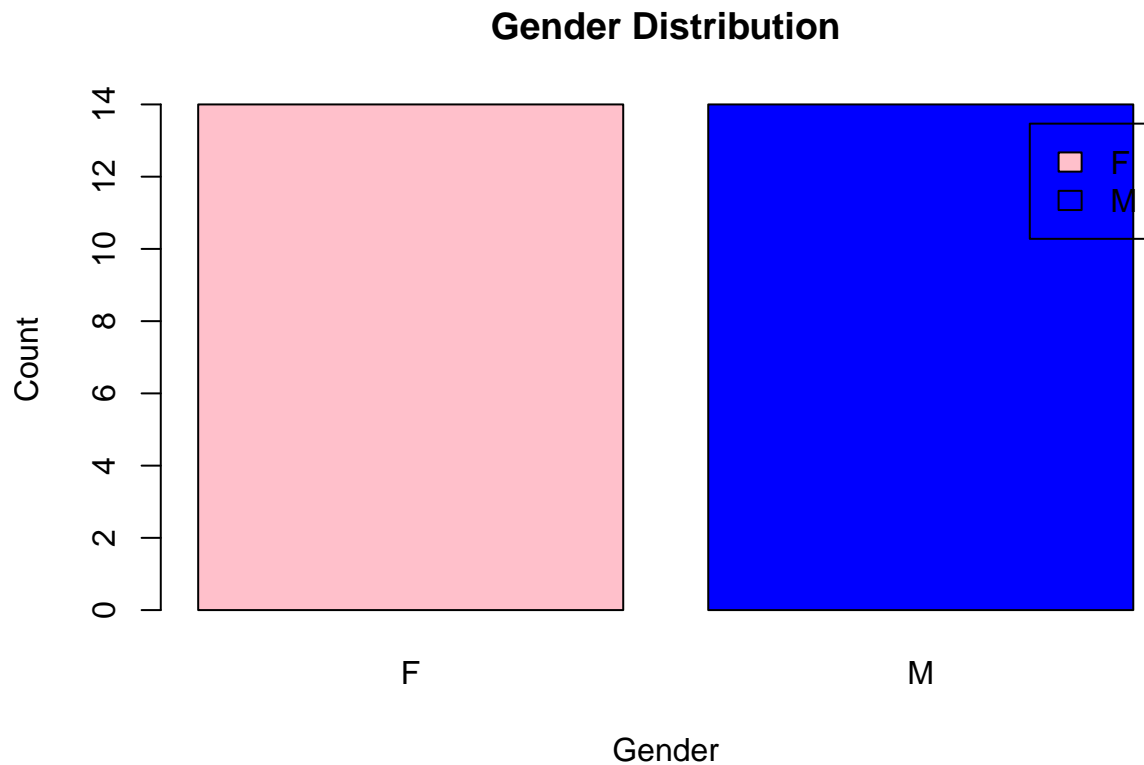
##   ShoeSize Height Gender
## 1      6.5   66.0      F
## 2      9.0   68.0      F
## 3      8.5   64.5      F
## 4      8.5   65.0      F
## 5     10.5   70.0      M
## 6      7.0   64.0      F

# 4b
femdata <- subset(data, Gender == "F")
maledata <- subset(data, Gender == "M")
cat("Female count:", nrow(femdata), "\n")

## Female count: 14

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

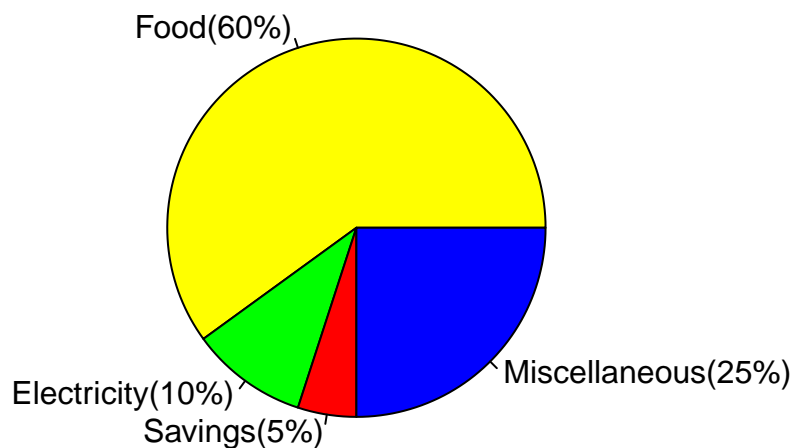
```



5

```
expenses <- c(Food = 60, Electricity = 10, Savings = 5, Miscellaneous = 25)
percent <- paste0(names(expenses), "(", round(100*expenses / sum(expenses), 1), "%)")
pie(expenses, labels= percent, col = c("yellow", "green", "red", "blue"), main = "Monthly Family Expenses")
```

Monthly Family Expenses



6a

```
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 ...
```

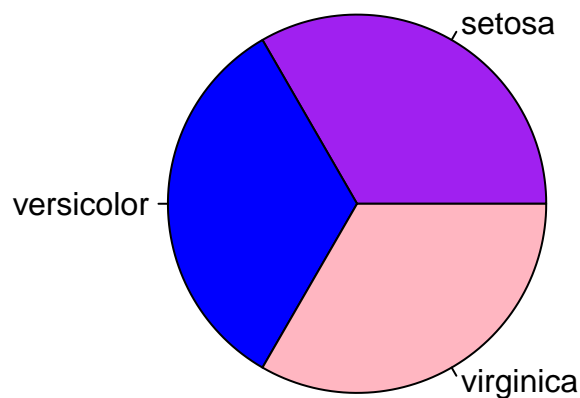
```
## $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
## $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
## $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...

# 6b
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
```

```
# 6c
species_counts <- table(iris$Species)
pie(species_counts, main="Species Distribution", col=c("purple", "blue", "lightpink"),
    labels=names(species_counts))
```

Species Distribution



```
# 6d
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:
```

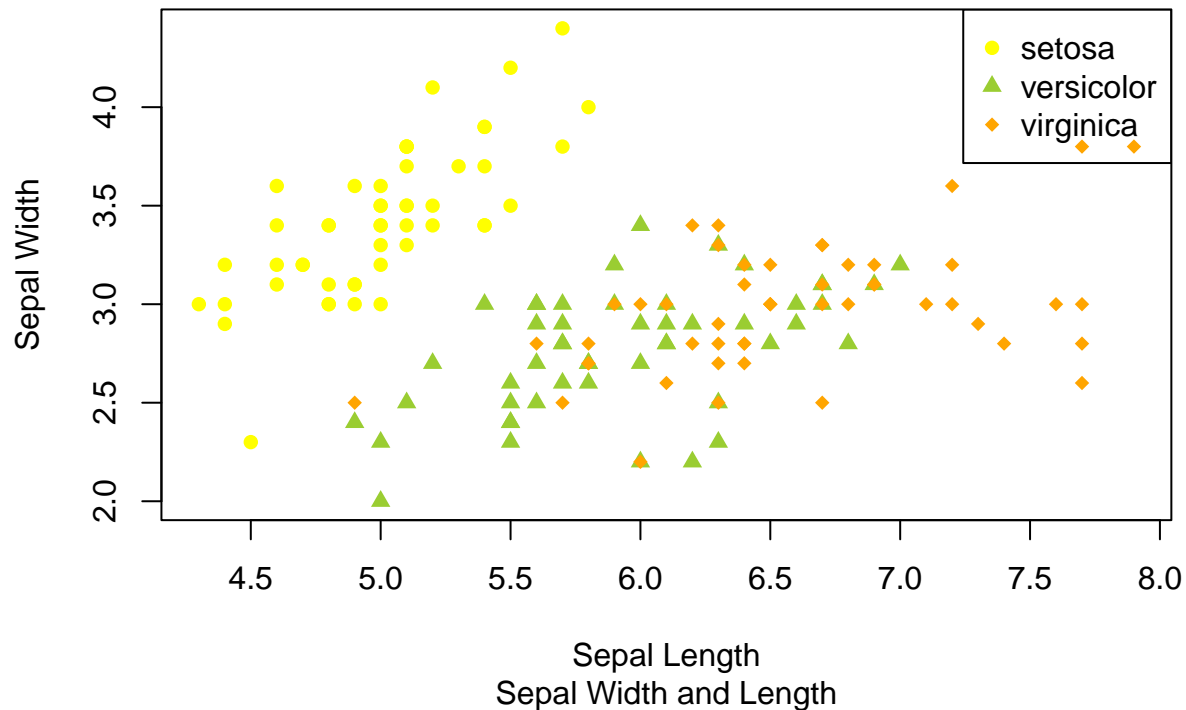
```
tail(virginica)
```

```
##      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
```

```
# 6e
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("yellow", "yellowgreen", "orange")[iris$Species],
     pch = c(16, 17, 18)[iris$Species])

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

Iris Dataset



6f

#In this plot, we can observe the classification of species based on sepal dimensions. Example, Setosa .

7a

```
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
```

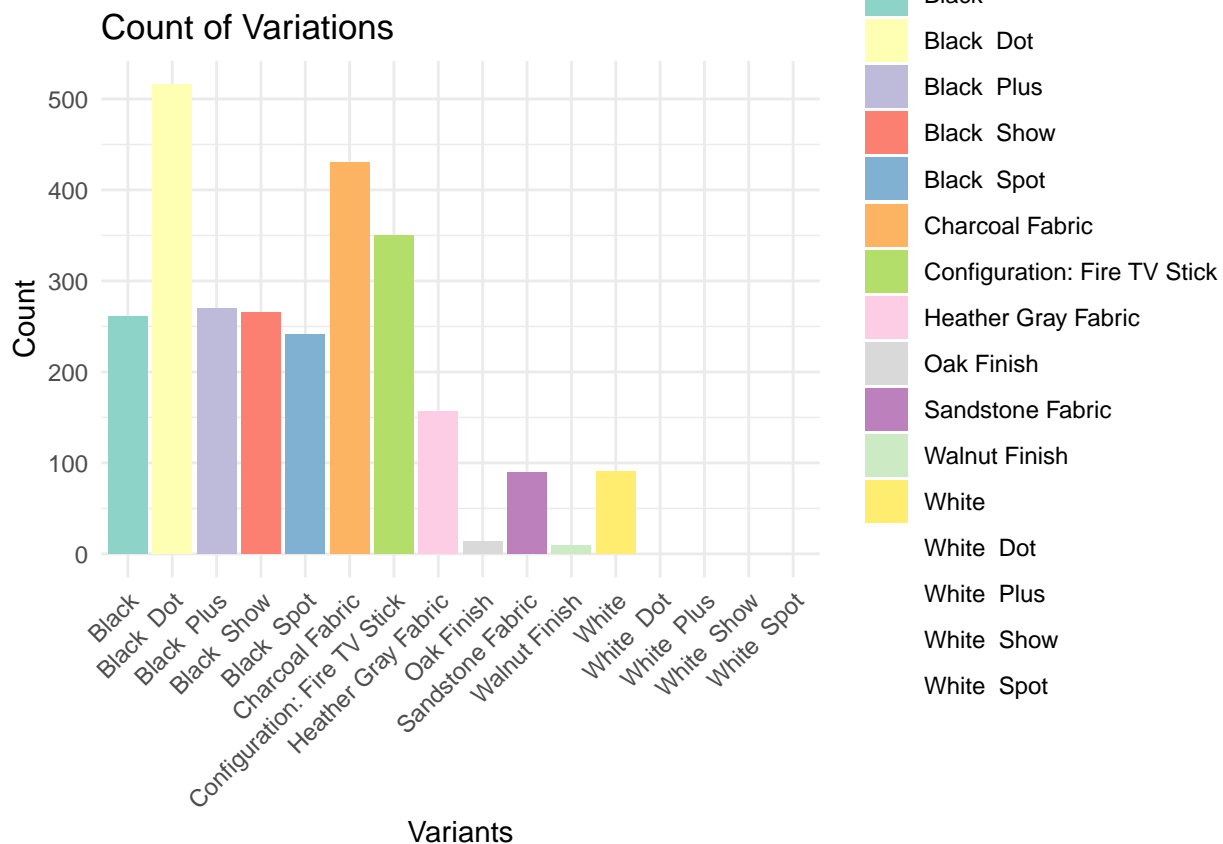
#7.c

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

```

