

Work sheet 4b

Mikee Lorraine S. Benjamin

2025-11-28

1. 5×5 Matrix

```
vectorA <- 1:5
matrix_result <- matrix(0, nrow = 5, ncol = 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. Print “*” in pattern

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

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

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

3. Fibonacci sequence from a chosen starting index up to 500

(Revised: No user input — knit-friendly)

```
fib_sequence <- function(start = 1) {

  a <- 0
  b <- 1
  count <- 0
  result <- c()

  repeat {
    if (count >= start) {
      if (b > 500) break
      result <- c(result, b)
    }

    temp <- a + b
    a <- b
    b <- temp
    count <- count + 1
  }

  return(result)
}

fib_sequence(1)

## [1] 1 2 3 5 8 13 21 34 55 89 144 233 377
```

4a. Import Excel/CSV file

```
library(readxl)
data <- read_excel("import_march.xlsx")

head(data)

## # A tibble: 6 x 4
##   Students `Strategy 1` `Strategy 2` `Strategy 3`
##   <chr>       <dbl>       <dbl>       <dbl>
```

```

## 1 Male          8          10          8
## 2 <NA>         4          8          6
## 3 <NA>         0          6          4
## 4 Female        14         4          15
## 5 <NA>         10         2          12
## 6 <NA>         6          0          9

```

4b. Subset male and female

```

male_data <- subset(data, Students == "Male")
female_data <- subset(data, Students == "Female")

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

## Number of Male observations: 1

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

## Number of Female observations: 1

```

4c. Barplot of Male & Female

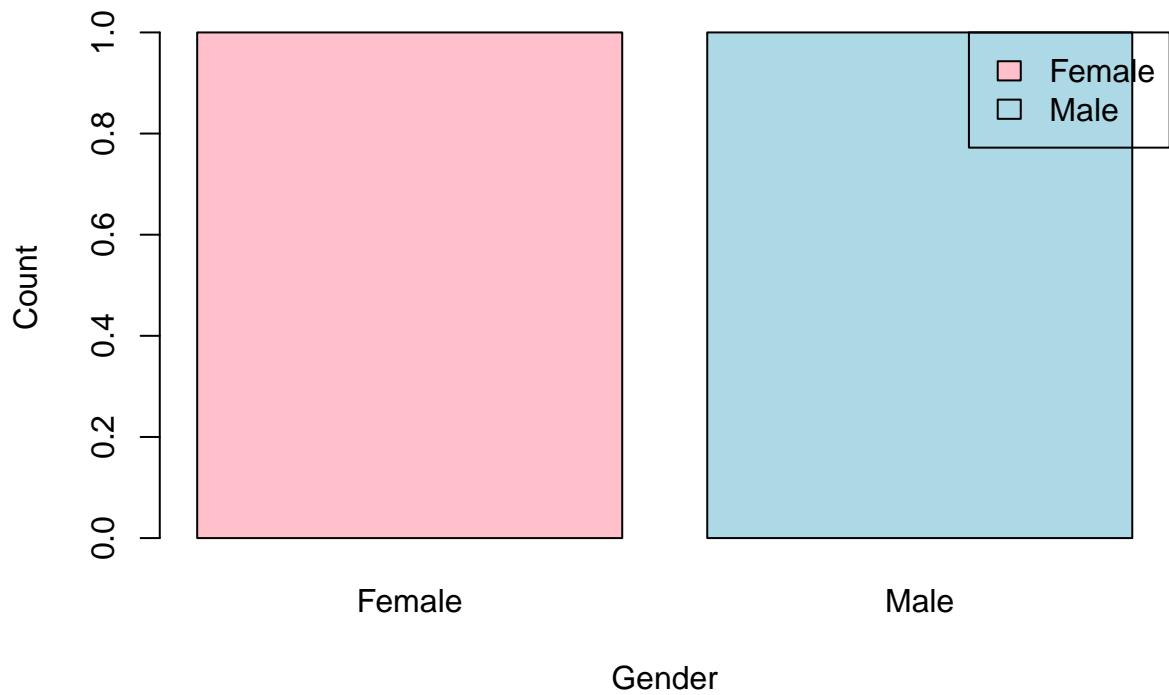
```

gender_counts <- table(data$Students)

barplot(
  gender_counts,
  main = "Number of Male and Female Observations",
  xlab = "Gender",
  ylab = "Count",
  col = c("pink", "lightblue"),
  legend.text = TRUE,
  args.legend = list(x = "topright", legend = names(gender_counts))
)

```

Number of Male and Female Observations



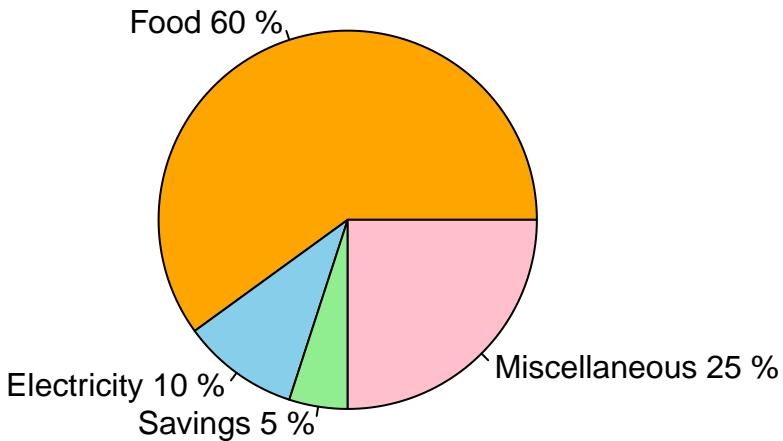
5. Pie Chart of Expenses

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

percent <- round(expenses / sum(expenses) * 100)
labels <- paste(categories, percent, "%")

pie(
  expenses,
  labels = labels,
  col = c("orange", "skyblue", "lightgreen", "pink"),
  main = "Monthly Income Distribution of Dela Cruz Family"
)
```

Monthly Income Distribution of Dela Cruz Family



6a. Structure of iris dataset

```
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 0.4 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 of 4 variables

```
iris_means <- data.frame(
  Sepal.Length = mean(iris$Sepal.Length),
  Sepal.Width  = mean(iris$Sepal.Width),
  Petal.Length = mean(iris$Petal.Length),
  Petal.Width  = mean(iris$Petal.Width)
)

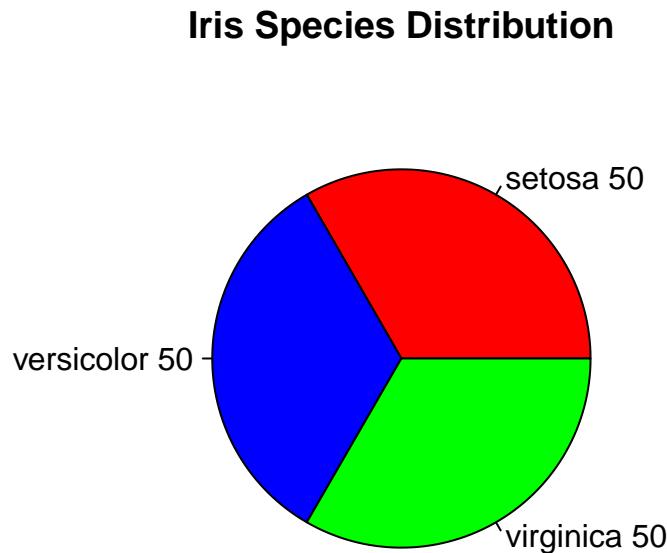
iris_means

##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1      5.843333    3.057333     3.758     1.199333
```

6c. Pie chart of Species

```
species_counts <- table(iris$Species)

pie(
  species_counts,
  main = "Iris Species Distribution",
  col = c("red", "blue", "green"),
  labels = paste(names(species_counts), species_counts)
)
```



6d. Subsets for each species

```
setosa      <- iris[iris$Species == "setosa", ]
versicolor <- iris[iris$Species == "versicolor", ]
virginica  <- iris[iris$Species == "virginica", ]

print("Last 6 rows: Setosa")

## [1] "Last 6 rows: 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
```

```

print("Last 6 rows: Versicolor")

## [1] "Last 6 rows: 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

print("Last 6 rows: Virginica")

## [1] "Last 6 rows: 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. Scatterplot of Sepal Length vs Width

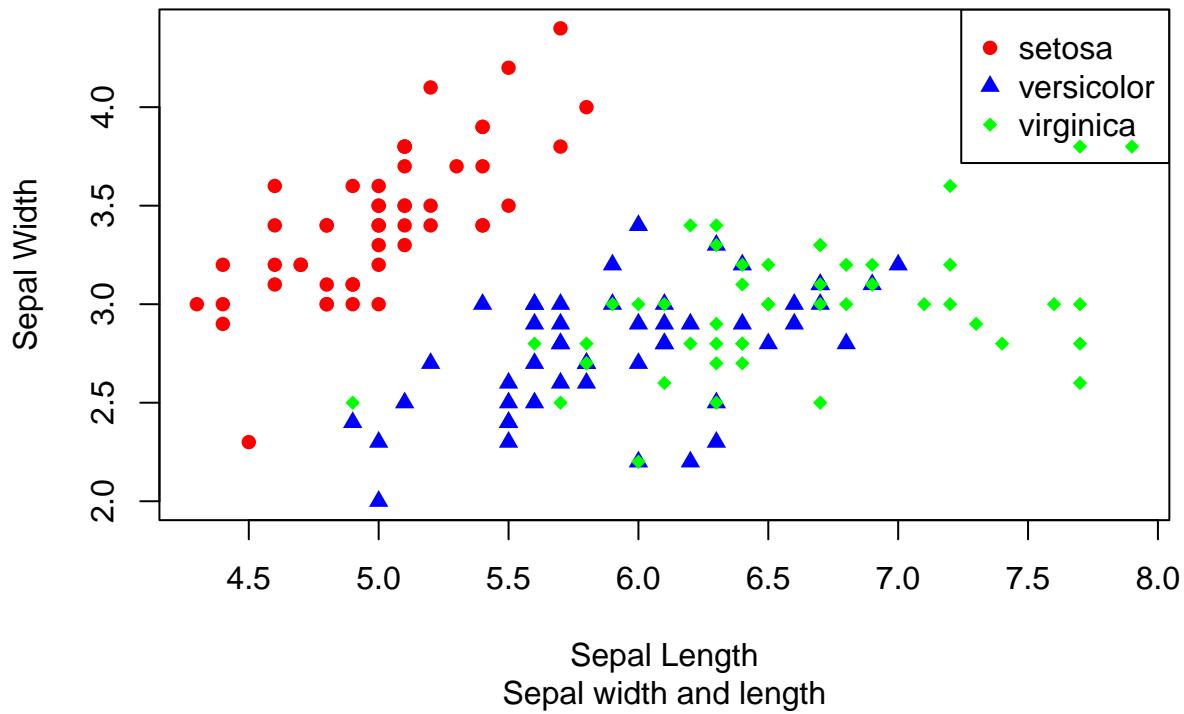
```

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

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

```

Iris Dataset



6f. Interpretation

"The scatterplot shows that Setosa forms a distinct cluster with smaller sepal length and relatively wide sepal width.

```
## [1] "The scatterplot shows that Setosa forms a distinct cluster with smaller sepal length and relatively wide sepal width."
```

7a. Cleaning Alexa variations

```
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 <- read_xlsx("alexa-file.xlsx")

# Clean double whitespaces
alexa$variation <- gsub("  ", " ", alexa$variation)

head(alexa, 10)

## # A tibble: 8 x 5
##   rating date      variation verified_reviews feedback
##   <dbl> <chr>    <chr>        <chr>
## 1     5 2018-07-30 Black Dot  It works great!!      1
## 2     5 2018-07-30 Black Plus PHENOMENAL          1
## 3     4 2018-07-30 Black Show I used it to control my smart devices. 1
## 4     3 2018-07-30 Black Spot Very convenient       1
## 5     5 2018-07-31 White Dot  I love it!            1
## 6     4 2018-07-31 White Plus Nice sound quality    1
## 7     5 2018-07-31 White Show Great screen!          1
## 8     4 2018-07-31 White Spot Easy to use           1

```

7b. Count variations and save to RData

```

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

## # A tibble: 8 x 2
##   variation     n
##   <chr>     <int>
## 1 Black Dot     1
## 2 Black Plus    1
## 3 Black Show    1
## 4 Black Spot    1
## 5 White Dot     1
## 6 White Plus    1
## 7 White Show    1
## 8 White Spot    1

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

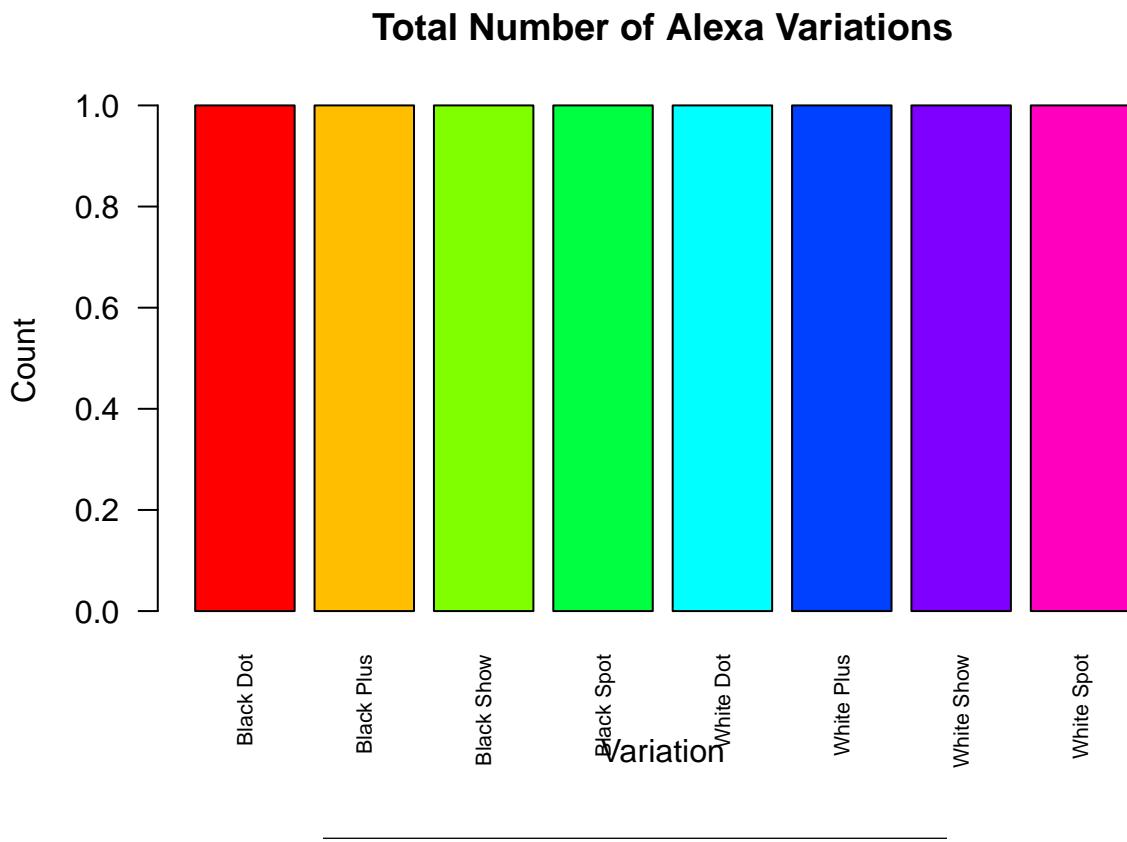
```

7c. Barplot of all Alexa variations

```

barplot(
  variations$n,
  names.arg = variations$variation,
  main = "Total Number of Alexa Variations",
  xlab = "Variation",
  ylab = "Count",
  col = rainbow(nrow(variations)),
  las = 2,
  cex.names = 0.7
)

```



7d. Side-by-side Black & White barplots

```

black_vars <- variations[grep("Black", variations$variation), ]
white_vars <- variations[grep("White", variations$variation), ]

par(mfrow = c(1, 2))

barplot(
  black_vars$n,
  names.arg = black_vars$variation,
  main = "Black Variations",
  xlab = "Variation",
  ylab = "Count",

```

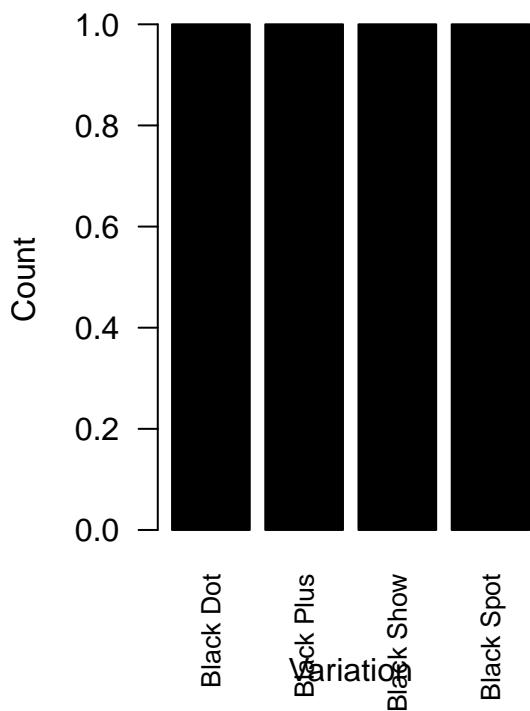
```

    col = "black",
    las = 2,
    cex.names = 0.8
)

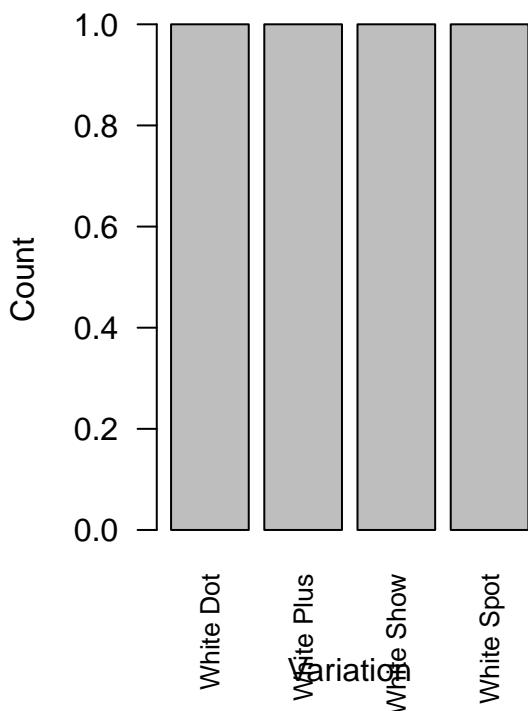
barplot(
  white_vars$n,
  names.arg = white_vars$variation,
  main = "White Variations",
  xlab = "Variation",
  ylab = "Count",
  col = "grey",
  las = 2,
  cex.names = 0.8
)

```

Black Variations



White Variations



```
par(mfrow = c(1, 1))
```