

RWorksheet_lauron_4b.Rmd

Mary Ghale C. Lauron

2025-11-07

```
#1

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

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

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

#2

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

## "*"
## ** **
## *** ***
## **** ****
## ***** *****

#3 fibonacci

start <- as.numeric(readline(prompt = "Enter starting number:"))

## Enter starting number:
if (is.na(start)) {
  cat("Invalid input! Please enter a numeric value.\n")
} else {
  # Initialize first two Fibonacci numbers
  a <- 0
  b <- 1

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

  repeat {
    # Only print numbers >= start
```

```

if (a >= start) {
  cat(a, " ")
}

# Stop if the next number will be greater than 500
if (a > 500) {
  break
}

# Generate next Fibonacci number
temp <- a + b
a <- b
b <- temp
}

cat("\nDone!\n")
}

```

Invalid input! Please enter a numeric value.

#4

#a For CSV and display first 6 rows
shoe_data <- read.csv("shoe_data.csv")

head(shoe_data)

	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

#b Subset for males and females

```

male_subset <- shoe_data[shoe_data$Gender == "M", ]
female_subset <- shoe_data[shoe_data$Gender == "F", ]
male_subset

```

	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

```

female_subset

##      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        6.5    66.0     F
## 21        7.5    64.0     F
## 24        8.5    69.0     F

# Count observations in each
nrow(male_subset)

## [1] 14

nrow(female_subset)

## [1] 14

gender_count <- table(shoe_data$Gender)

# Display counts
gender_count

## 
##   F   M
## 14 14

# Create a bar plot
barplot(gender_count,
        main = "Number of Males and Females in Household Data",
        xlab = "Gender",
        ylab = "Number of Individuals",
        col = c("pink", "lightblue"),
        legend.text = TRUE,
        args.legend = list(x = "topright", title = "Gender"))

```

Number of Males and Females in Household Data



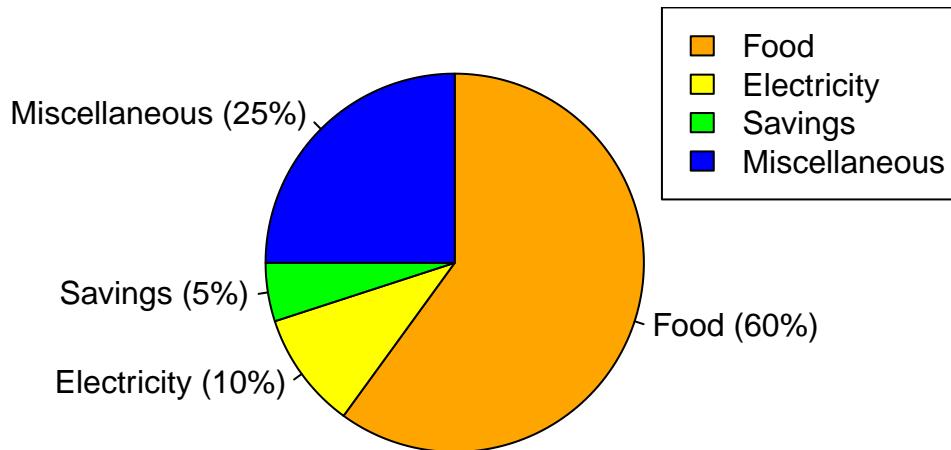
```
# 5
expenses <- c(Food = 60, Electricity = 10, Savings = 5, Miscellaneous = 25)

labels <- paste0(names(expenses),
                 " (", round(100 * expenses / sum(expenses), 1), "%)")

# Create pie chart
pie(expenses,
     main = "Monthly Expenses of Dela Cruz Family",
     col = c("orange", "yellow", "green", "blue"),
     labels = labels,
     clockwise = TRUE)

legend("topright", legend = names(expenses), fill = c("orange", "yellow", "green", "blue"))
```

Monthly Expenses of Dela Cruz Family



```
#6 iris dataset
data(iris)

#a Check structure
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 ...
#b mean of the sepal.length,sepal.width,petal.length, and petal.width.
iris_mean <- colMeans(iris[, 1:4])
iris_mean

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

#c Count species
species_distribution <- table(iris$Species)
species_distribution

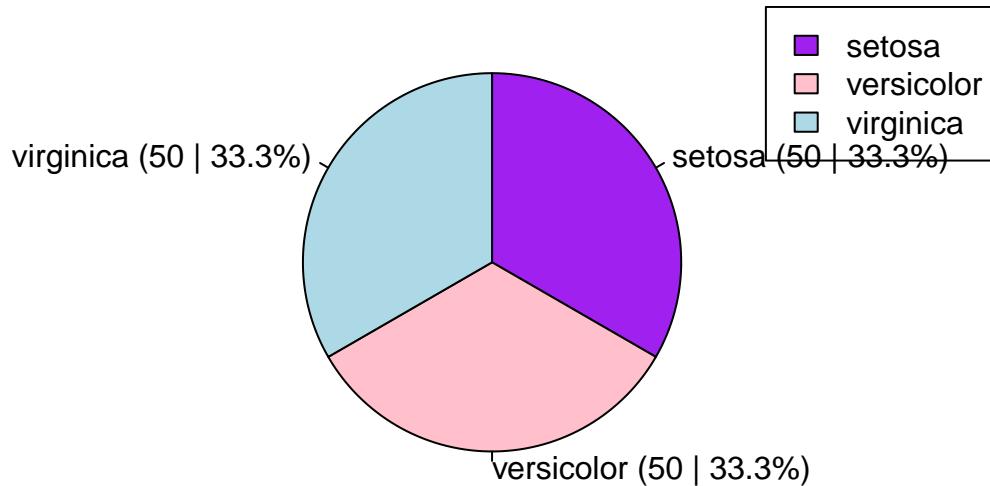
##
##      setosa versicolor virginica
##      50        50        50

labels <- paste0(names(species_distribution),
                 " (", species_distribution, " | ",
                 round(100 * species_distribution / sum(species_distribution), 1), "%)")

# Create pie chart
pie(species_distribution,
     main = "Species Distribution in Iris Dataset",
     col = c("purple", "pink", "lightblue"),
     labels = labels,
     clockwise = TRUE)
```

```
legend("topright", legend = names(species_distribution), fill = c("purple", "pink", "lightblue"))
```

Species Distribution in Iris Dataset



```
#d Subset each species
setosa <- subset(iris, Species == "setosa")
versicolor <- subset(iris, Species == "versicolor")
virginica <- subset(iris, Species == "virginica")

#last 6 rows of each
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 scatter_plot

data(iris)

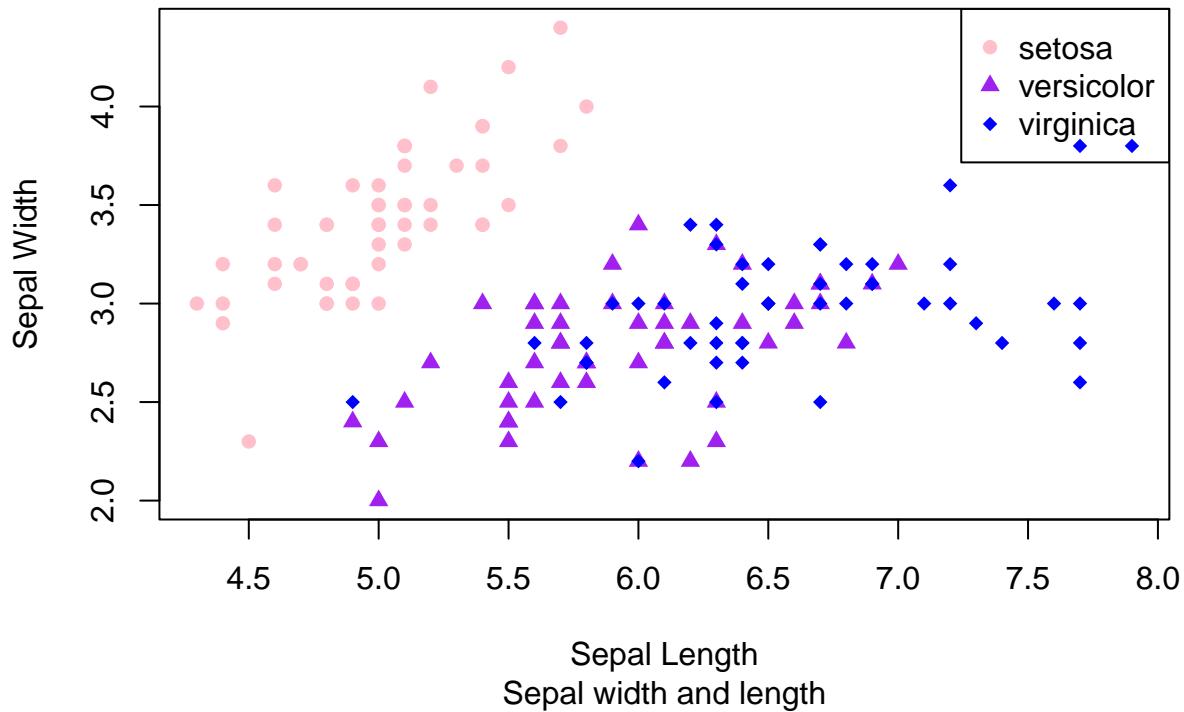
iris$Species <- as.factor(iris$Species)

colors <- c("setosa" = "pink", "versicolor" = "purple", "virginica" = "blue")
symbols <- c("setosa" = 16, "versicolor" = 17, "virginica" = 18)

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

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

Iris Dataset



#f. Interpret the result.

#Setosa species has wider sepal but shorter in length, the versicolor is in between or in average in size
#The clear separation among clusters suggests that Sepal.Length and Sepal.Width can help distinguish species

```

##7 import alexa.xlsx
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_excel("/cloud/project/alexa_file.xlsx")
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

#a. Rename the white and black variants by using gsub() function.
alexa$variation <- gsub("Black +", "Black ", alexa$variation)

alexa$variation <- gsub("White +", "White ", alexa$variation)

head(alexa$variation, 10)

## [1] "Charcoal Fabric"      "Charcoal Fabric"      "Walnut Finish"
## [4] "Charcoal Fabric"      "Charcoal Fabric"      "Heather Gray Fabric"
## [7] "Sandstone Fabric"     "Charcoal Fabric"      "Heather Gray Fabric"
## [10] "Heather Gray Fabric"

#b. Variations.Rdata
# b. Count variations and save
variations <- alexa %>% count(variation)
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

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

```

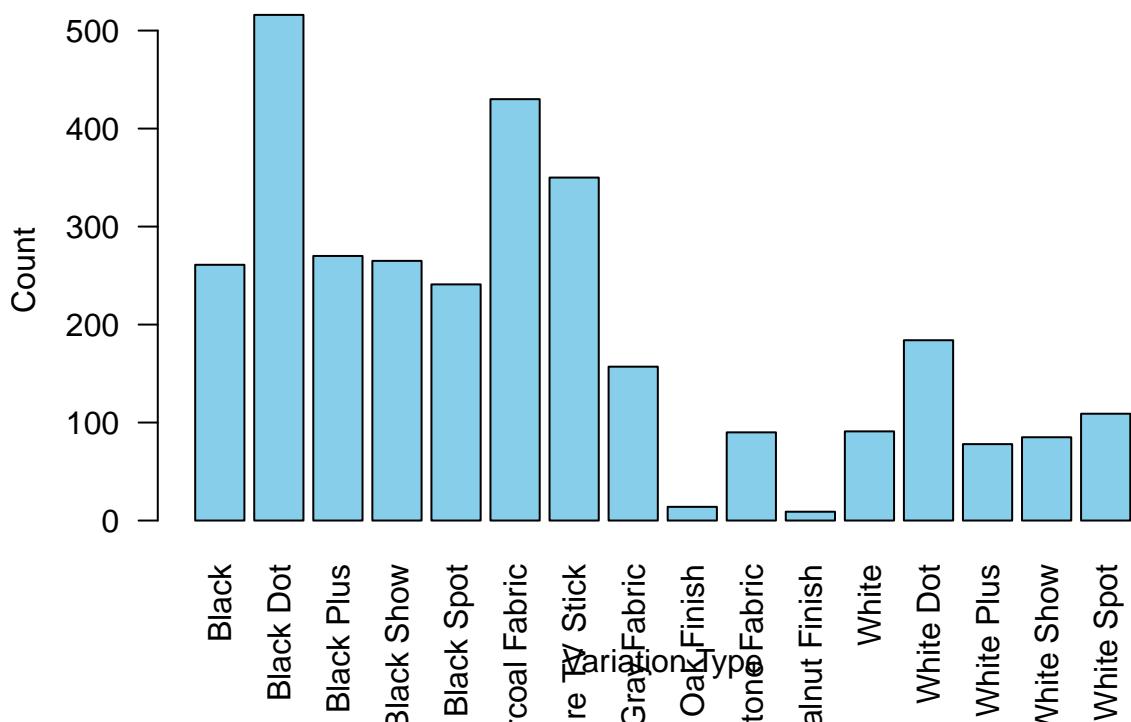
#c. From the variations.RData, create a barplot().

```

barplot(variations$n,
        names.arg = variations$variation,
        col = "skyblue",
        main = "Variation Counts",
        xlab = "Variation Type",
        ylab = "Count",
        las = 2)

```

Variation Counts



```

black <- variations %>% filter(grep("Black", variation))
white <- variations %>% filter(grep("White", variation))

```

Combine counts into matrix

```

bw_matrix <- rbind(black$n, white$n)

#d. Create a barplot() for the black and white variations. Plot it in 1 frame, side by side. Complete the code below.
barplot(bw_matrix,
        beside = TRUE,
        names.arg = gsub("Black |White ", "", black$variation),
        col = c("black", "gray90"),
        main = "Black vs White Variation Counts",
        xlab = "Variation Type",
        ylab = "Count")
legend("topright",
       legend = c("Black", "White"),
       fill = c("black", "gray90"))

```

