Worksheet-4b in R

Riza Angelique Pelaez

2024-10-28

```
# Task 1: Create a 5x5 Matrix Using a for Loop
matrix_5x5 <- matrix(0, nrow=5, ncol=5)
vectorA <- c(1, 2, 3, 4, 5)
for (i in 1:5) {
  matrix_5x5[i, ] <- abs(vectorA)
}
print("5x5 Matrix with vectorA:")
```

```
## [1] "5x5 Matrix with vectorA:"
```

```
print(matrix_5x5)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    1    2    3    4    5
## [3,]    1    2    3    4    5
## [4,]    1    2    3    4    5
## [5,]    1    2    3    4    5
```

```
# Task 2: Print the String "*" Using a for Loop
print("Pyramid of '*' using for loop:")
```

```
## [1] "Pyramid of '*' using for loop:"
```

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

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

```
# Task 3: Get user input for the first two numbers in the Fibonacci sequence
n1 <- as.integer(readline(prompt="Enter the first number: "))
```

```
## Enter the first number:
```

```
n2 <- as.integer(readline(prompt="Enter the second number: "))
```

```
## Enter the second number:
```

```
if (is.na(n1) || is.na(n2)) {
  cat("Please enter valid integers.\n")
} else {
  cat("Fibonacci sequence up to 500:\n")
```

```
  repeat {
    cat(n1, " ")
    fib <- n1 + n2
    if (fib > 500) break
    n1 <- n2
    n2 <- fib
  }

  cat("\n")
}
```

```
## Please enter valid integers.
```

```
# Task 4: Import File and Perform Operations
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
```

```
#a.R script for importing an excel or a csv file
ShoeSize <- read.csv("ShoeSizes.csv")
cat("First 6 rows of the ShoeSize:\n")
```

```
## First 6 rows of the ShoeSize:
```

```
print(head(ShoeSize))
```

```
##   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 male and female
female <- subset(ShoeSize, Gender == "F")
male <- subset(ShoeSize, Gender == "M")
cat("Number of females:", nrow(female), "\n")
```

```
## Number of females: 14
```

```
cat("Number of males:", nrow(male), "\n")
```

```
## Number of males: 14
```

```
#c. Graph for the number of males and females
str(ShoeSize)
```

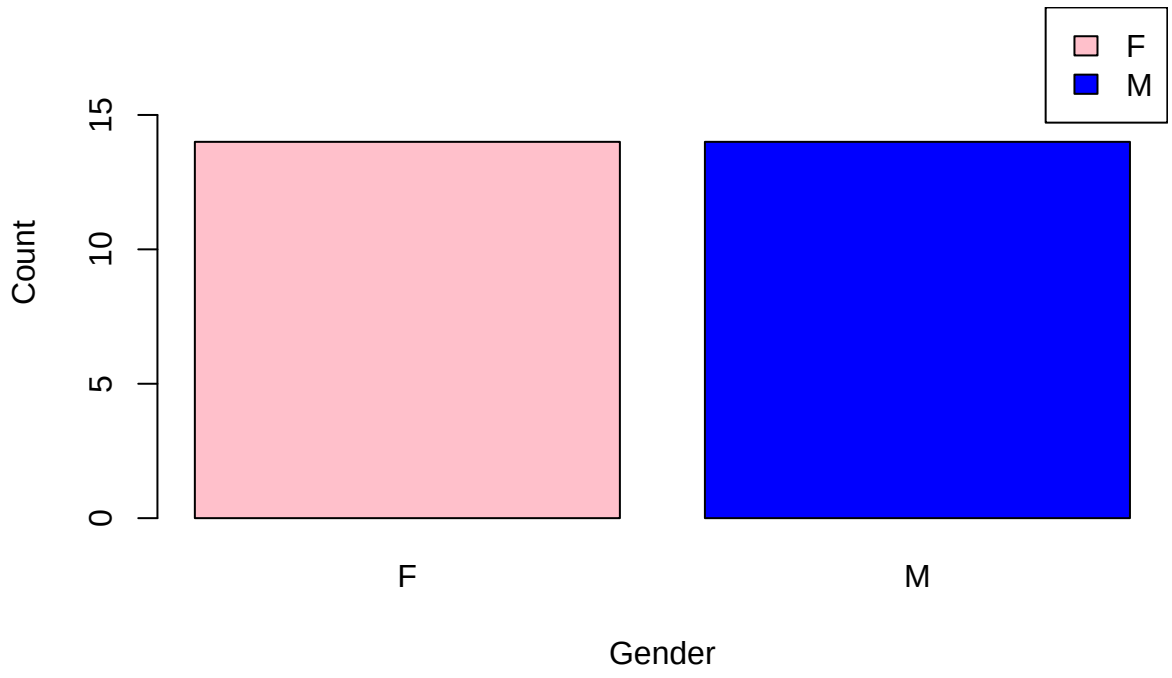
```
## 'data.frame':    28 obs. of  3 variables:
##  $ Shoe.size: num  6.5 9 8.5 8.5 10.5 7 9.5 9 13 7.5 ...
```

```
##  $ Height   : num  66 68 64.5 65 70 64 70 71 72 64 ...
##  $ Gender   : chr  "F" "F" "F" "F" ...
```

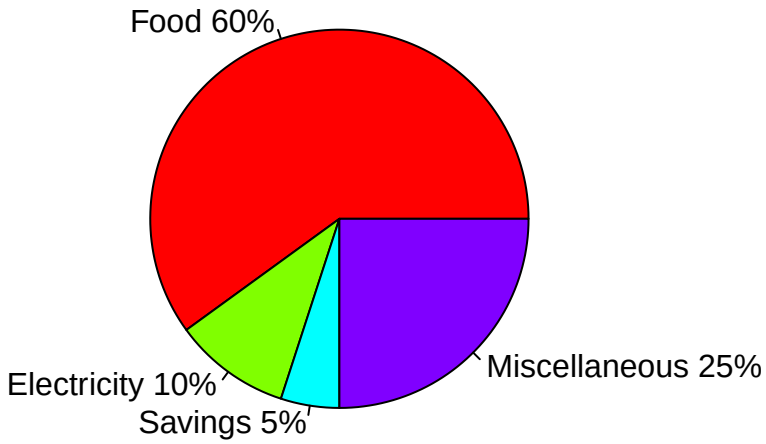
```
head(ShoeSize)
```

```
##   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
```

```
ShoeSize$Gender <- as.factor(ShoeSize$Gender)

gender_counts <- ShoeSize %>%
  group_by(Gender) %>%
  summarise(Count = n(), .groups = 'drop')

barplot(gender_counts$Count,
        names.arg = gender_counts$Gender,
        col = c("pink", "blue" ),
        main = "Number of Females and Males in Shoe Sizes",
        xlab = "Gender",
        ylab = "Count",
        ylim = c(0, max(gender_counts$Count) + 5))

legend("topright", legend = levels(ShoeSize$Gender), fill = c("pink", "blue"))
```




```r
# Task 5: Monthly Income Pie Chart
expenses <- c(Food=60, Electricity=10, Savings=5, Miscellaneous=25)
pie(expenses, main="Monthly Income Distribution", col=rainbow(length(expenses)),
    labels=paste0(names(expenses), " ", round(100 * expenses / sum(expenses), 1), "%"))
```




```r
# Task 6: Iris Dataset Operations
data(iris)

# a. Check the structure
cat("Structure of the iris dataset:\n")
```

```
## Structure of the iris dataset:
```

```r
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 ...
```

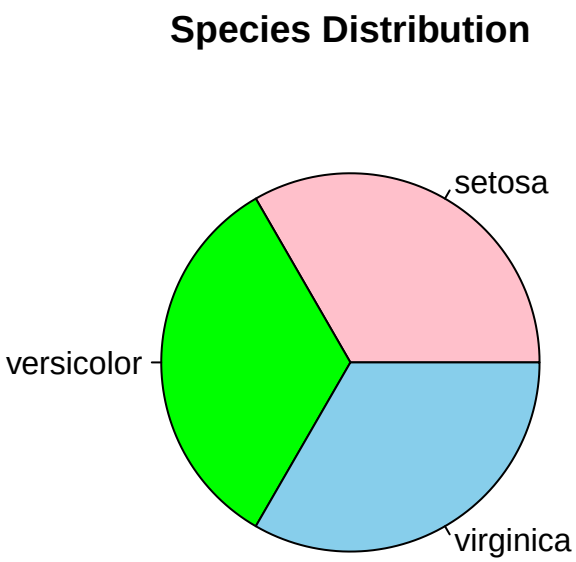
```r
# b. Calculate means of Sepal and Petal measurements
iris_means <- colMeans(iris[, 1:4])
cat("Means of Sepal and Petal measurements:\n")
```

```
## Means of Sepal and Petal measurements:
```

```r
print(iris_means)
```

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

```r
# c. Pie chart for Species distribution
species_counts <- table(iris$Species)
pie(species_counts, main="Species Distribution", col=c("pink", "green", "skyblue"),
    labels=names(species_counts))
```




```r
# d. Subset each species and show the last six rows
setosa <- subset(iris, Species == "setosa")
versicolor <- subset(iris, Species == "versicolor")
virginica <- subset(iris, Species == "virginica")
cat("Last 6 rows of Setosa species:\n")
```

```
## Last 6 rows of Setosa species:
```

```r
print(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
```

```r
cat("Last 6 rows of Versicolor species:\n")
```

```
## Last 6 rows of Versicolor species:
```

```r
print(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
```

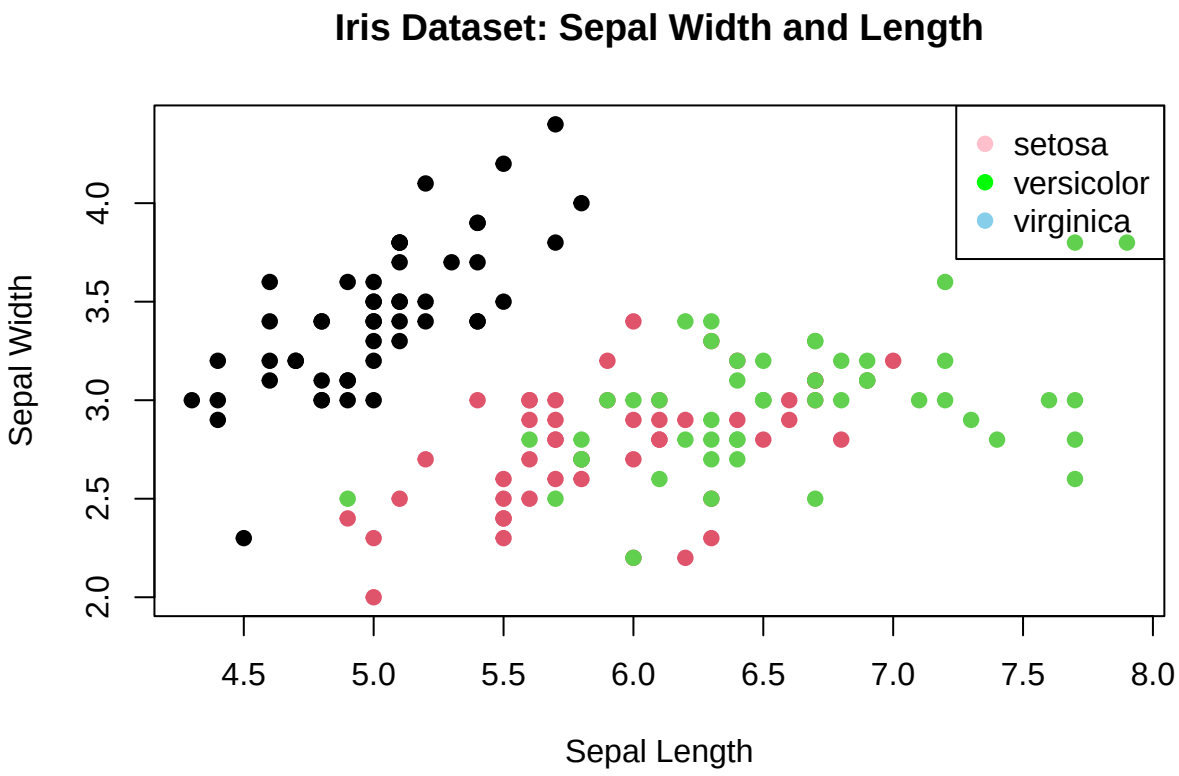
```r
cat("Last 6 rows of Virginica species:\n")
```

```
## Last 6 rows of Virginica species:
```

```r
print(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
```

```r
# e. Scatterplot of Sepal measurements by species
plot(iris$Sepal.Length, iris$Sepal.Width, col=iris$Species, pch=19,
     main="Iris Dataset: Sepal Width and Length",
     xlab="Sepal Length", ylab="Sepal Width")
legend("topright", legend=levels(iris$Species), col=c("pink", "green", "skyblue"), pch=19)
```




```
# Task 7: Alexa Variations Dataset Cleaning and Plotting

# a. Rename white and black variants
library(readxl)
alexafile <- read_excel("alexa_file.xlsx")
alexafile$variation <- gsub("Black ", "Black_", alexafile$variation)
alexafile$variation <- gsub("White ", "White_", alexafile$variation)
print(head(alexafile))
```

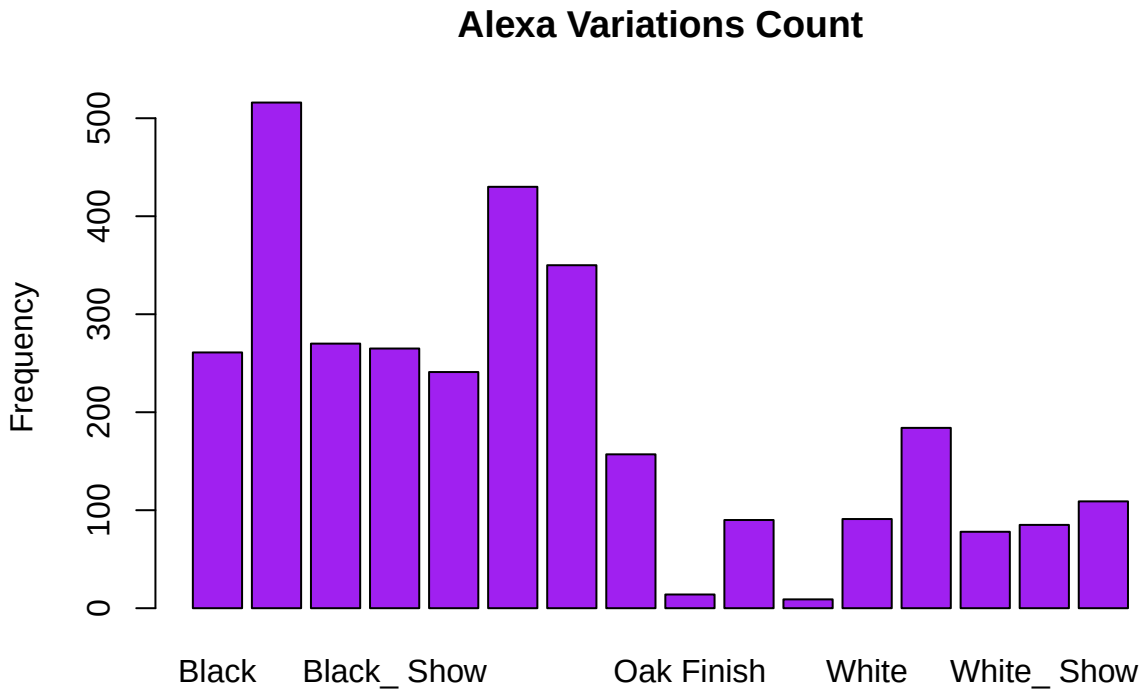
```
## # A tibble: 6 x 5
##   rating date                variation          verified_reviews       feedback
##    <dbl> <dttm>              <chr>              <chr>                     <dbl>
## 1      5 2018-07-31 00:00:00 Charcoal Fabric    Love my Echo!                 1
## 2      5 2018-07-31 00:00:00 Charcoal Fabric    Loved it!                     1
## 3      4 2018-07-31 00:00:00 Walnut Finish      Sometimes while 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
```

```
# b. Count each variation and save as variations.RData
library(dplyr)
variation_counts <- alexafile %>%
  count(variation)
save(variation_counts, file = "variations.RData")
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
```

```r
# c. Barplot for each variation count
load("variations.RData")
barplot(variation_counts$n,
        names.arg = variation_counts$variation,
        col = "purple",
        main = "Alexa Variations Count",
        ylab = "Frequency")
```



```r
# d. Barplot for black and white variants side by side
black_white_counts <- variation_counts %>%
  filter(grepl("Black|White", variation))
barplot(black_white_counts$n,
        names.arg = black_white_counts$variation,
        col = c("black", "white"),
        beside = TRUE,
        main = "Black and White Alexa Variants",
        ylab = "Frequency")
```