RWorksheet_Bernasol#4b

Shane Jhammielyn A. Bernasol

```
#1.Using the for loop Function

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] <-ifelse(j>=i, vectorA[abs(j-i)+1],0)
  }
}

matrixA
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    2    3    4    5
## [2,]    0    1    2    3    4
## [3,]    0    0    1    2    3
## [4,]    0    0    0    1    2
## [5,]    0    0    0    0    1
```

```
#2. using for() function.

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

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

```
#3. Get an input from the user

A <- as.integer(readline(prompt = "Enter starting position: "))
```

```
## Enter starting position:
```

```
fib1 <- 0
fib2 <- 1
count <- 1
fib_sequence <- c()

repeat {
  z <- fib1 + fib2

  if (!is.na(A) && count >= A && z <= 500) {
    fib_sequence <- c(fib_sequence, z)
  }

  fib1 <- fib2
  fib2 <- z
  count <- count + 1

  if (z > 500) break
}

print(fib_sequence)
```

```
## NULL
```

```
#4 A. Display first 6 rows
library(readr)

shoe_Sizes <- read_csv("/cloud/project/Worksheet 4/Worksheet 4B/Shoe-Sizes.csv")
```

```
## Rows: 28 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Gender
## dbl (2): Shoe size, 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.
```

```
head(shoe_Sizes)
```

```
## # A tibble: 6 x 3
##   `Shoe size` Height Gender
##         <dbl>  <dbl> <chr>
## 1         6.5   66   F
## 2         9     68   F
## 3         8.5   64.5 F
## 4         8.5   65   F
## 5        10.5   70   M
## 6         7     64   F
```

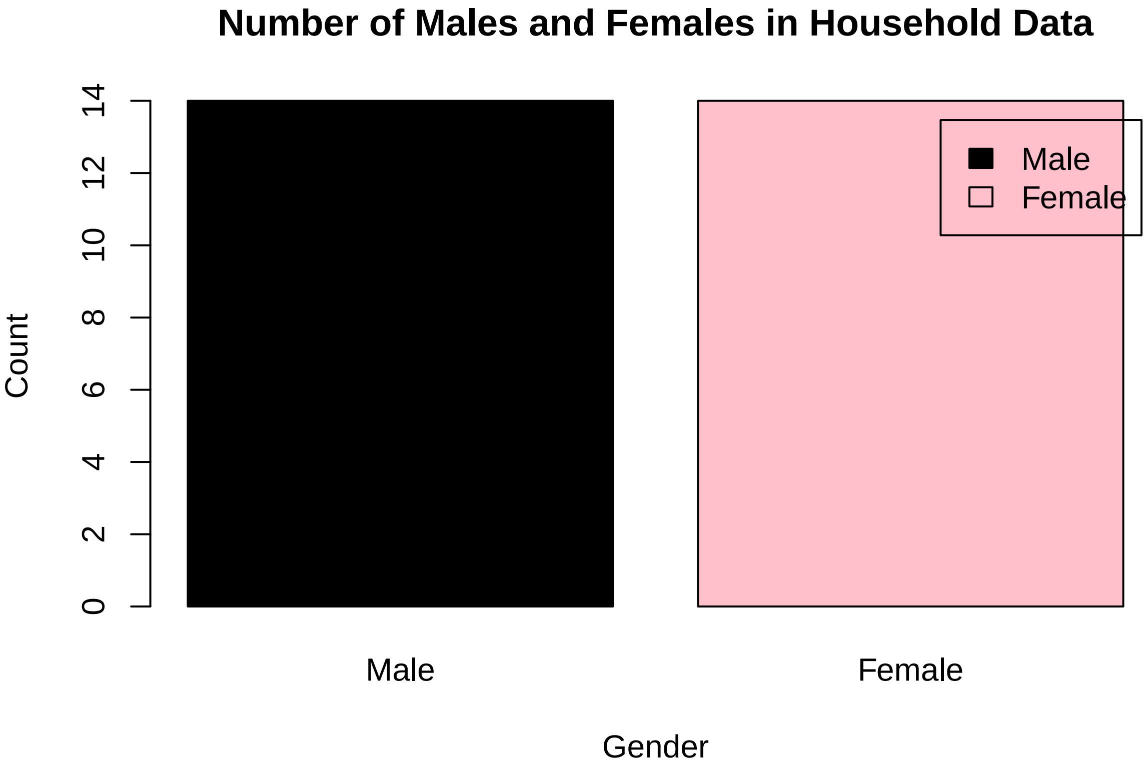
```
#4 B. Creating subset for gender (female and male)

male <- subset(shoe_Sizes, Gender == "M")
female <- subset(shoe_Sizes, Gender == "F")
num_males <- nrow(male)
num_females <- nrow(female)

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

```
## Number of Male observations: 14
```

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

```
## Number of Female observations: 14
```

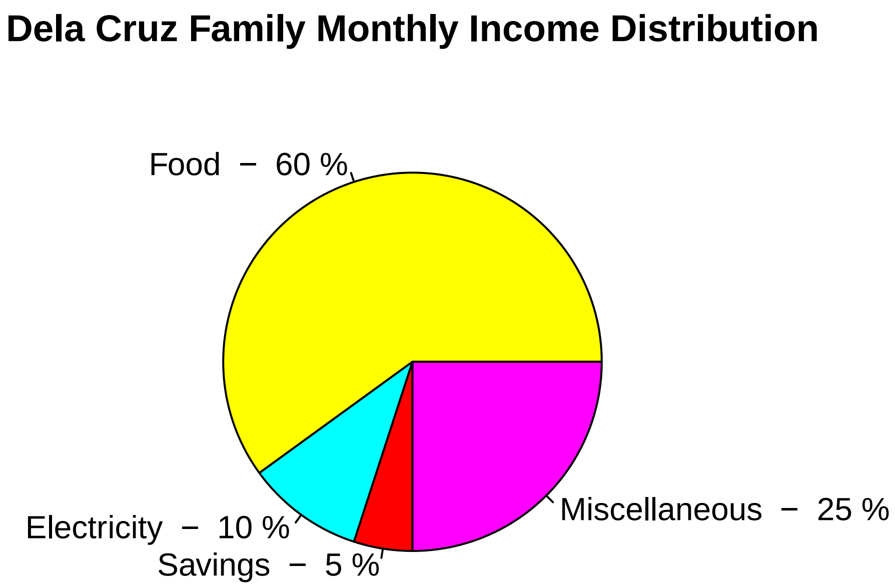
```
#4 C.Create a graph for the number of males and females for Household Data

gender_counts <- c(num_males, num_females)
names(gender_counts) <- c("Male", "Female")
barplot(gender_counts,
        main = "Number of Males and Females in Household Data",xlab = "Gender",ylab = "Count",col = c("
```



5a.

Monthly income of Dela Cruz family

```
#5 A. Monthly income of Dela Cruz family

expenses <- c(Food = 60, Electricity = 10, Savings = 5, Miscellaneous = 25)
colors <- c("yellow", "cyan", "red", "magenta")
pie(expenses,labels = paste(names(expenses), " - ", round((expenses / sum(expenses)) * 100, 1), "%"),co
```



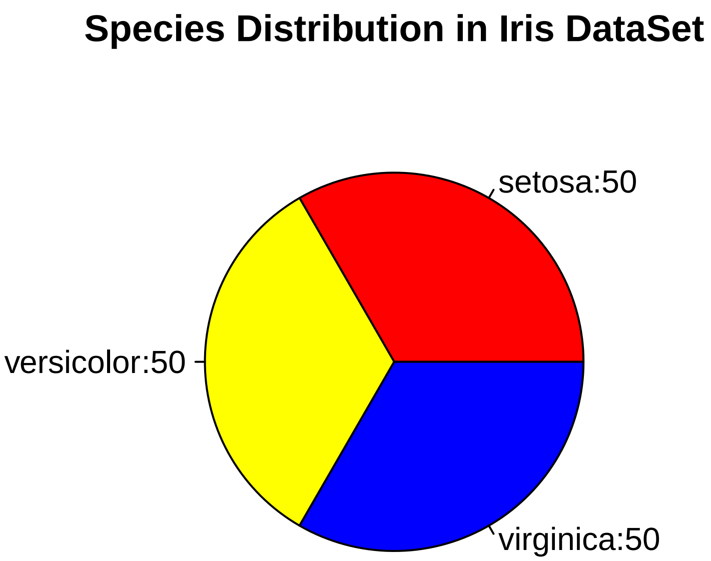
```
#6 A.
data(iris)
str(iris)
```

```
## '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 ...
```

```
#6 B.

means <-colMeans(iris[, 1:4])
means
```

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

```
#6 C.

species_count <- table(iris$Species)
species_labels <-paste(names(species_count), species_count, sep = ":")

color <-c("red", "yellow", "blue")

pie(species_count, labels= species_labels, col= color, main="Species Distribution in Iris DataSet")
```



```
#6 D.

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

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)
```

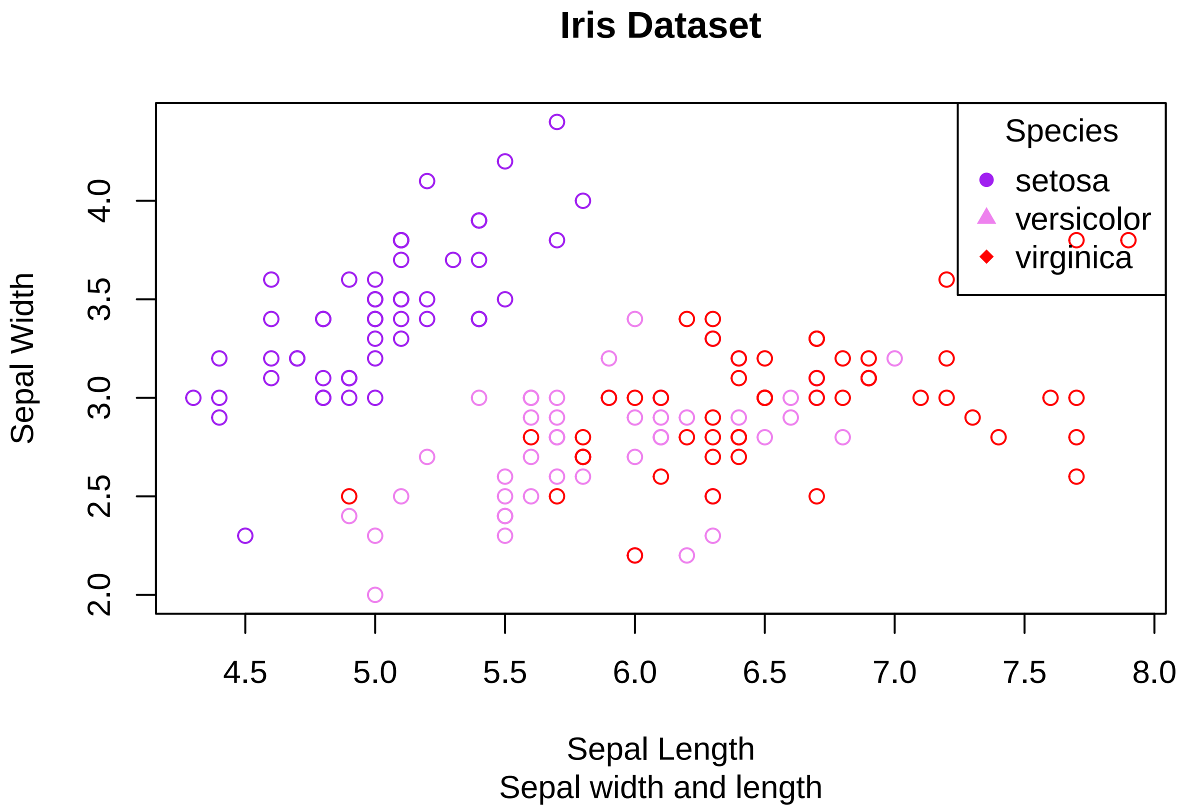
```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 95           5.6         2.7          4.2         1.3 versicolor
## 96           5.7         3.0          4.2         1.2 versicolor
## 97           5.7         2.9          4.2         1.3 versicolor
## 98           6.2         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
```

```
#6 E.

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

```
Species_color <- c("setosa"="purple","versicolor"="violet", "virginica"= "red")
species_pch <- c("setosa"= 16, "versicolor"= 17, "virginica"= 18)

plot(iris$Sepal.Length, iris$Sepal.Width, col= Species_color[iris$Species],
     xlab = "Sepal Length",
     ylab= "Sepal Width", main ="Iris Dataset", sub= "Sepal width and length")

legend ("topright", legend=levels(iris$Species), col= Species_color, pch = species_pch, title = "Specie
```



```
#6 F.Interpret the result.

# Interpretation of the Scatterplot Result
# The scatterplot shows the relationship between Sepal.Length and Sepal.Width for the three species.
# Setosa points (in orange) tend to have shorter sepal lengths and widths.
# Versicolor points (in pink) are more spread out and overlap slightly with virginica.
# Virginica points (in yellow) have larger sepal lengths and widths, distinguishing them from setosa an
```

```
#7.Import the alexa-file.xlsx

library(readxl)
data <- read_excel("/cloud/project/Worksheet 4/Worksheet 4B/alexa-file.xlsx")

data$variation <- gsub("Black  Dot", "Black Dot", data$variation)
data$variation <- gsub("White  Plus", "White Plus", data$variation)

head(data$variation)
```

```
## [1] "Black Dot" "Black Dot" "Black Dot" "Black Dot" "Black Dot"
```

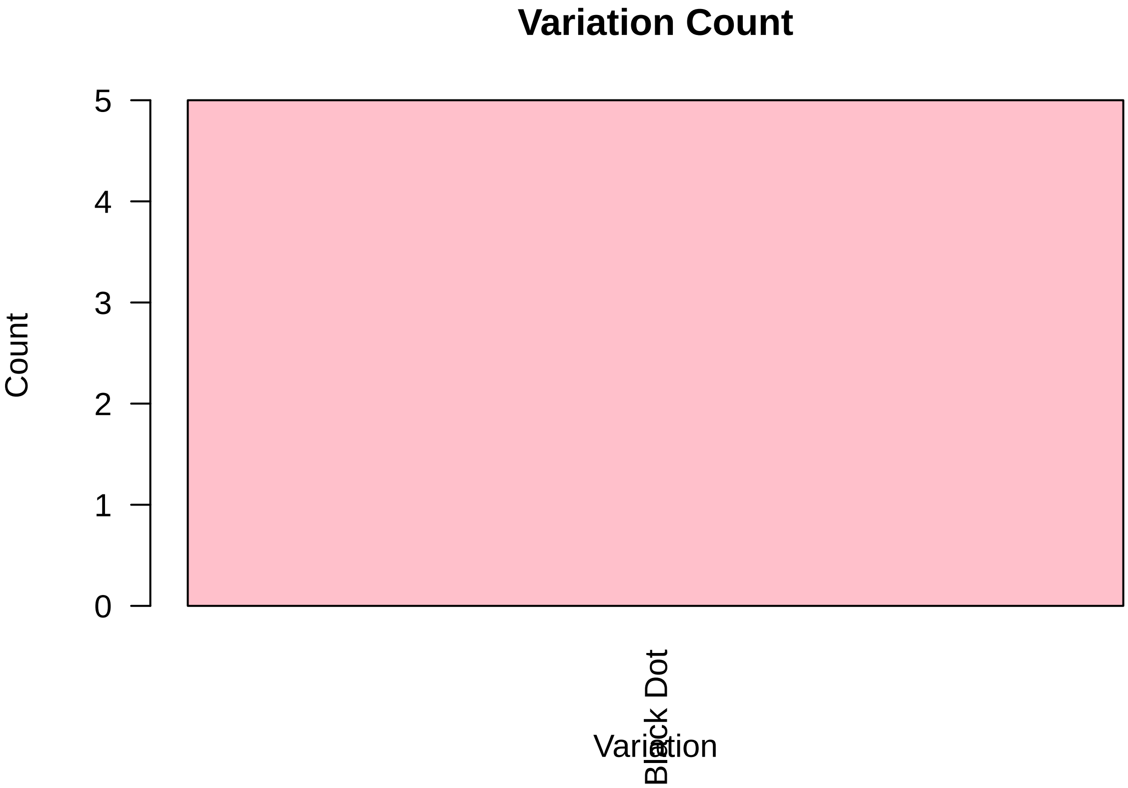
```
library(dplyr)
```

```
##
## 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
```

```
variation_count <- data %>%
  count(variation)
save(variation_count, file = "variations.RData")
print(variation_count)
```

```
## # A tibble: 1 x 2
##   variation     n
##   <chr>     <int>
## 1 Black Dot     5
```

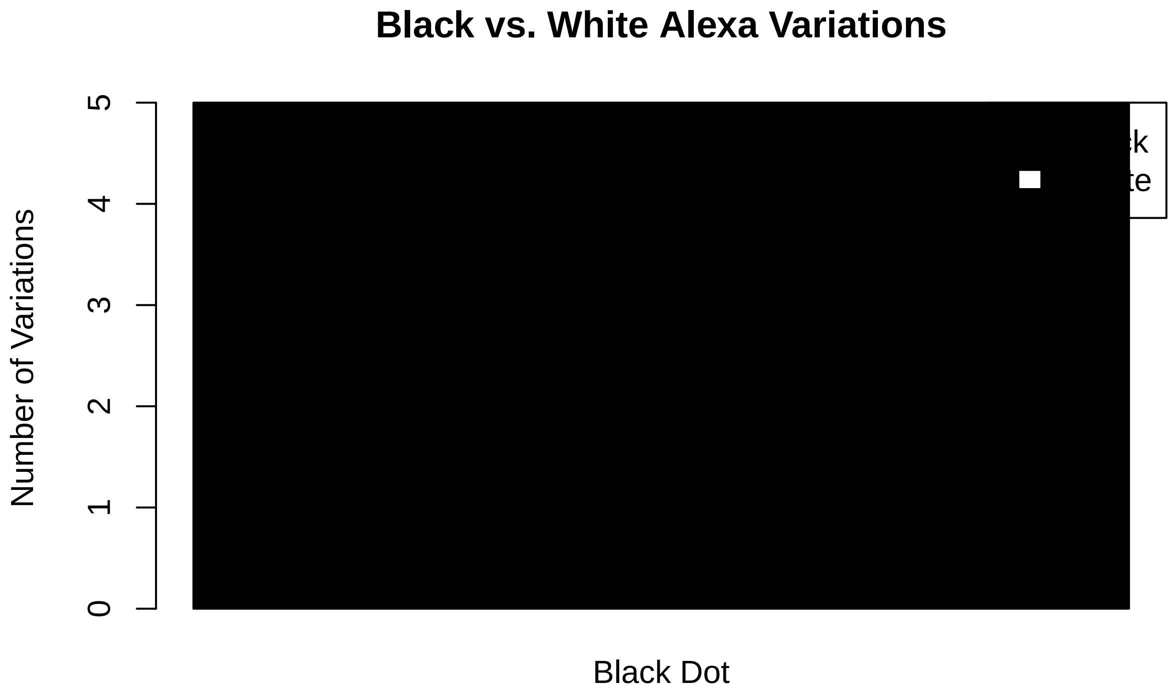
```
#7 B.

library(dplyr)
load("variations.RData")

barplot(variation_count$n, names.arg = variation_count$variation, col = "pink",
        main = "Variation Count", xlab = "Variation", ylab = "Count", las = 2)
```



```r
library(dplyr)

variation_count <- data %>%
  count(variation)

save(variation_count, file = "variations.RData")
print(variation_count)
```

```
## # A tibble: 1 x 2
##   variation     n
##   <chr>     <int>
## 1 Black Dot     5
```

```r
black_variations <- variation_count[grep("Black", variation_count$variation), ]
white_variations <- variation_count[grep("White", variation_count$variation), ]

barplot(cbind(black_variations$n, white_variations$n),
        beside = TRUE,
        names.arg = c(black_variations$variation, white_variations$variation),
        main = "Black vs. White Alexa Variations",
        ylab = "Number of Variations",
        col = c("black", "white"),
        legend.text = c("Black", "White"),
        args.legend = list(x = "topright"))
```


Black vs. White Alexa Variations
Number of Variations
ck
te
Black Dot