

RWork- sheet_Buenvenida#4b.Rmd

me

2024-11-11

1. Using the for loop, create an R script that will display a 5x5 matrix as shown in Figure 1. It must contain vectorA = [1,2,3,4,5] and a 5 x 5 zero matrix. Hint Use abs() function to get the absolute value

```
matrixA <- matrix(0, nrow = 5, ncol = 5)
for (i in 1:5) {
  for (j in 1:5) {
    matrixA[i, j] <- abs(i - j)
  }
}
matrixA
```

```
##      [,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 the string "*" using for() function. The output should be the same as shown in Figure

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

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

3. Get an input from the user to print the Fibonacci sequence starting from the 1st input up to 500. Use repeat and break statements. Write the R Scripts and its output.

```
start <- as.integer(readline(prompt = "Enter the starting number for the Fibonacci sequence: "))
```

```
## Enter the starting number for the Fibonacci sequence:
```

```
start <- 5
a <- 0
b <- 1
repeat {
  fib <- a + b
  a <- b
  b <- fib
  if (fib >= start) {
    cat(fib, " ")
  }
}
```

```

}
if (fib >= 500) {
  break
}
}

```

```
## 5 8 13 21 34 55 89 144 233 377 610
```

4.

- a. What is the R script for importing an excel or a csv file? Display the first 6 rows of the dataset? Show your codes and its result

```

shoes <- read.csv("shoe_sizes.csv")
shoes

```

```

##      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   65.0      F
## 6         7.0   64.0      F
## 7         9.5   70.0      F
## 8         9.0   71.0      M
## 9         7.5   64.0      M
## 10        10.0   75.0      F
## 11         8.5   74.5      F
## 12        12.0   71.0      M
## 13        10.5   71.0      M
## 14        13.0   77.0      M
## 15        11.5   72.0      M
## 16         5.0   59.0      F
## 17         5.0   64.0      F
## 18         7.5   72.0      M
## 19         6.5   66.0      M
## 20         8.5   72.0      M
## 21        11.0   72.0      M
## 22         8.5   69.0      M
## 23        10.0   72.0      M
## 24         9.0   69.0      M
## 25        13.0   70.0      M

```

- b. Create a subset for gender(female and male). How many observations are there in Male? How about in Female? Write the R scripts and its output.

```

data <- read.csv("shoe_sizes.csv")
male_data <- subset(data, Gender == "M")
female_data <- subset(data, Gender == "F")
male_count <- nrow(male_data)
female_count <- nrow(female_data)
cat("Number of observations in Male:", male_count, "\n")

```

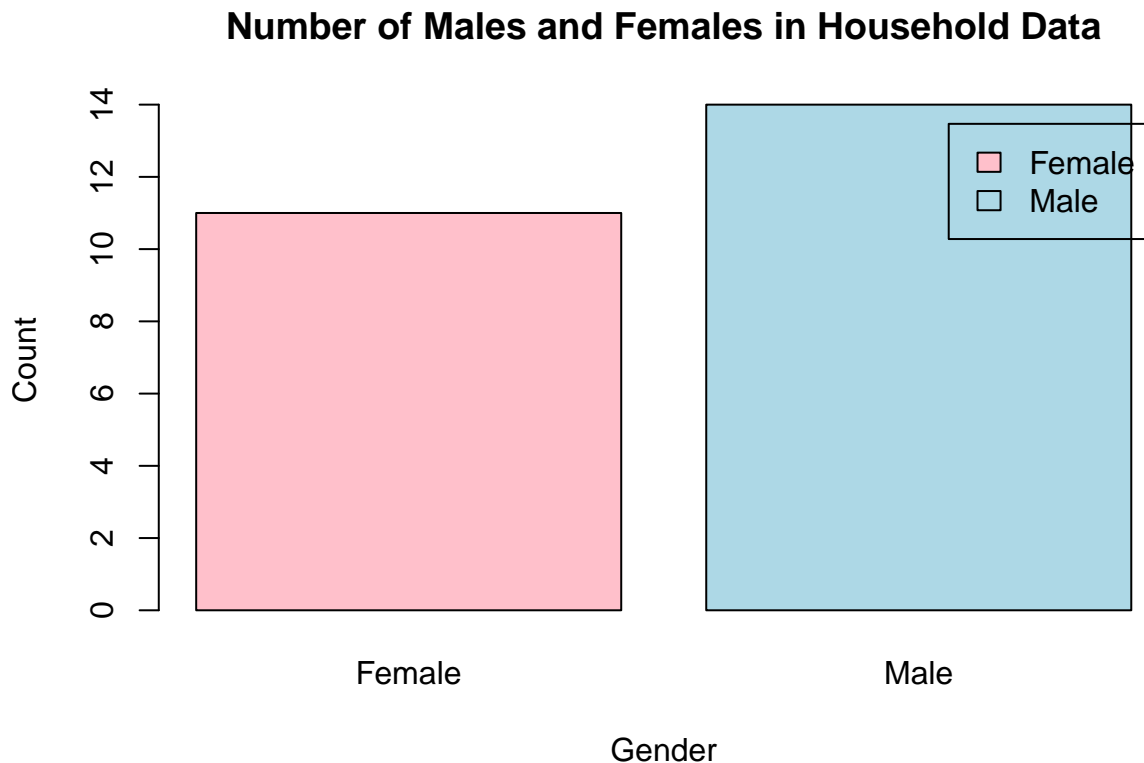
```
## Number of observations in Male: 14
```

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

```
## Number of observations in Female: 11
```

- c. Create a graph for the number of males and females for Household Data. Use `plot()`, chart type = `barplot`. Make sure to place title, legends, and colors. Write the R scripts and its result.

```
data <- read.csv("shoe_sizes.csv")
gender_counts <- table(data$Gender)
barplot(
  gender_counts,
  col = c("pink", "lightblue"),
  main = "Number of Males and Females in Household Data",
  xlab = "Gender",
  ylab = "Count",
  legend = c("Female", "Male"),
  names.arg = c("Female", "Male"))
```

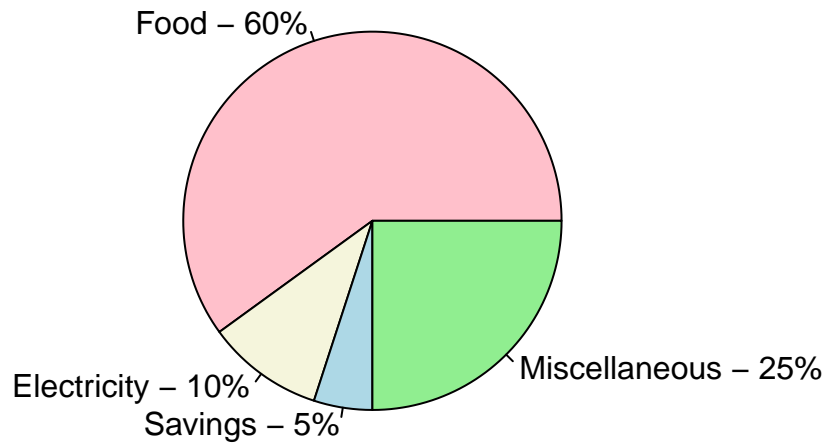


5.

- a. Create a piechart that will include labels in percentage. Add some colors and title of the chart. Write the R scripts and show its output.

```
expenses <- c(Food = 60, Electricity = 10, Savings = 5, Miscellaneous = 25)
percent <- paste0(names(expenses), " - ", round(expenses / sum(expenses) * 100), "%")
pie(expenses,
  labels = percent,
  col = c("pink", "beige", "lightblue", "lightgreen"),
  main = "Dela Cruz Family Expenses")
```

Dela Cruz Family Expenses



6. Use the iris dataset. `data(iris)`

a. Check for the structure of the dataset using the `str()` function. Describe what you have seen in the output.

```
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 ...
```

b. Create an R object that will contain the mean of the sepal.length, sepal.width, petal.length, and petal.width. What is the R script and its result?

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

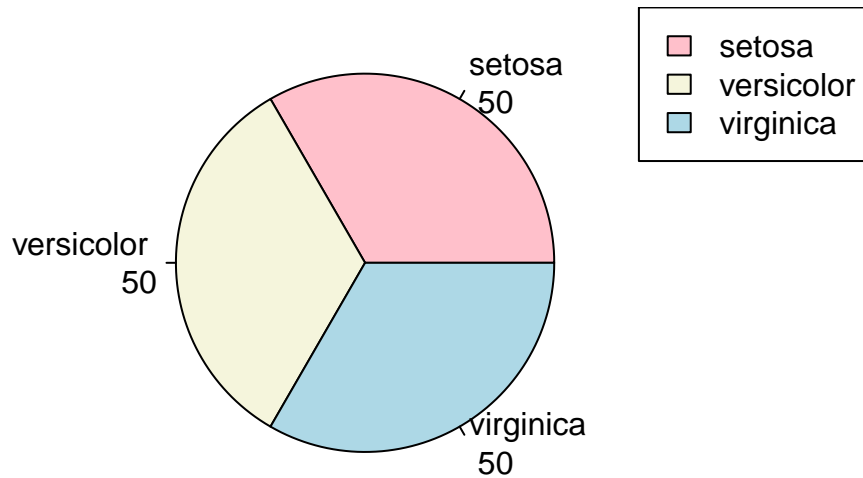
c. Create a pie chart for the Species distribution. Add title, legends, and colors. Write the R script and its result.

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

pie(species_distribution,
    main = "Species Distribution in Iris Dataset",
    col = c("pink", "beige", "lightblue"),
    labels = paste(names(species_distribution), "\n", species_distribution))

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

Species Distribution in Iris Dataset



- d. Subset the species into setosa, versicolor, and virginica. Write the R scripts and show the last six (6) rows of each species.

```
species_setosa <- subset(iris, Species == "setosa")
species_versicolor <- subset(iris, Species == "versicolor")
species_virginica <- subset(iris, Species == "virginica")
```

```
tail(species_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(species_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(species_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
```

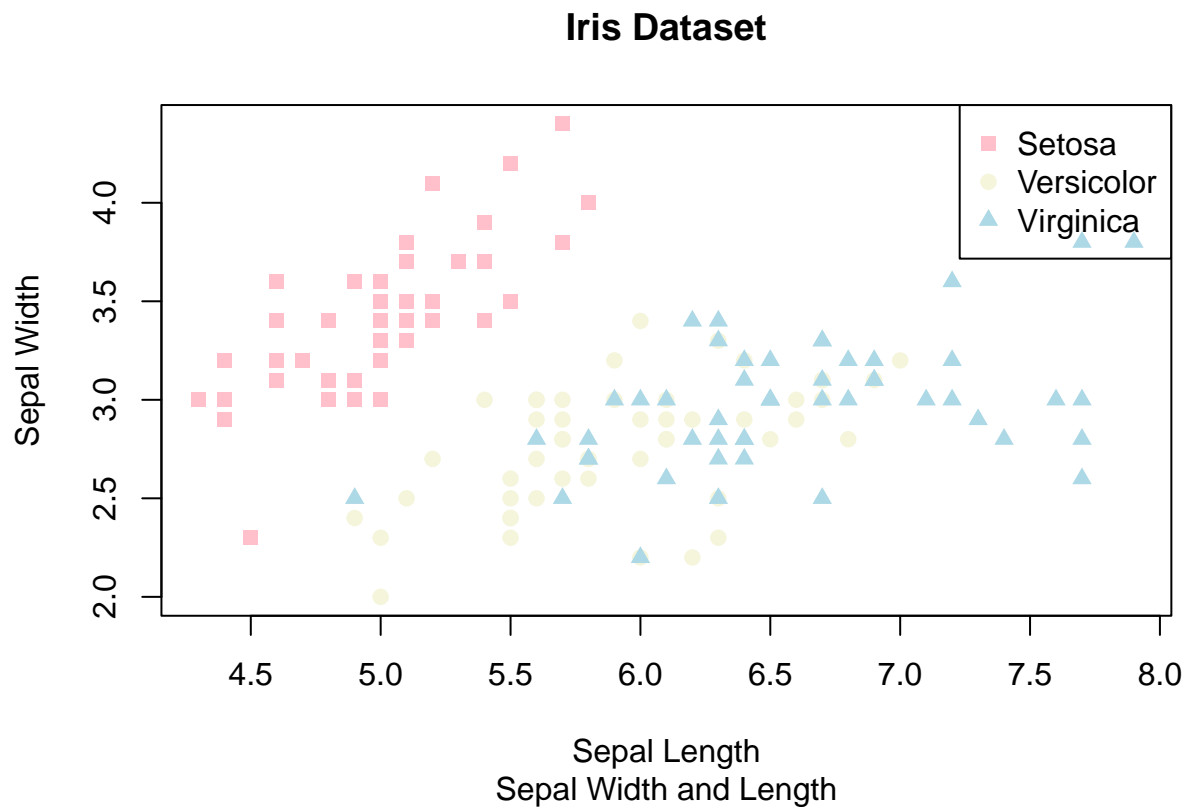
- e. Create a scatterplot of the sepal.length and sepal.width using the different species(setosa,versicolor,virginica).

Add a title = “Iris Dataset”, subtitle = “Sepal width and length, labels for the x and y axis, the pch symbol and colors should be based on the species.

```
species_colors <- c("setosa" = "pink", "versicolor" = "beige", "virginica" = "lightblue")
species_symbols <- c("setosa" = 15, "versicolor" = 19, "virginica" = 17)

plot(iris$Sepal.Length, iris$Sepal.Width,
     col = species_colors[iris$Species],
     pch = species_symbols[iris$Species],
     main = "Iris Dataset",
     sub = "Sepal Width and Length",
     xlab = "Sepal Length",
     ylab = "Sepal Width")

legend("topright", legend = c("Setosa", "Versicolor", "Virginica"),
     col = c("pink", "beige", "lightblue"),
     pch = c(15, 19, 17))
```



f. Interpret the result.

The *iris* dataset has 150 flowers, equally divided among three species: setosa, versicolor, and virginica. Each flower has measurements for sepal and petal length and width. The averages show that setosa flowers are generally smaller, especially in petal size, while versicolor and virginica are larger. The scatterplot shows clear groupings by species, with each occupying a different area based on sepal size, which highlights their distinct characteristics.

7.

a.

```
library(readxl)
alexa_data <- read_excel("alexa_file.xlsx")
unique(alexa_data$variation)

## [1] "Charcoal Fabric"          "Walnut Finish"
## [3] "Heather Gray Fabric"     "Sandstone Fabric"
## [5] "Oak Finish"              "Black"
## [7] "White"                   "Black Spot"
## [9] "White Spot"              "Black Show"
## [11] "White Show"              "Black Plus"
## [13] "White Plus"              "Configuration: Fire TV Stick"
## [15] "Black Dot"               "White Dot"

alexa_data$variation <- gsub("Black Dot", "BlackDot", alexa_data$variation)
alexa_data$variation <- gsub("Black Plus", "BlackPlus", alexa_data$variation)
alexa_data$variation <- gsub("Black Show", "BlackShow", alexa_data$variation)
alexa_data$variation <- gsub("Black Spot", "BlackSpot", alexa_data$variation)
alexa_data$variation <- gsub("White Dot", "WhiteDot", alexa_data$variation)
alexa_data$variation <- gsub("White Plus", "WhitePlus", alexa_data$variation)
alexa_data$variation <- gsub("White Show", "WhiteShow", alexa_data$variation)
alexa_data$variation <- gsub("White Spot", "WhiteSpot", alexa_data$variation)
unique(alexa_data$variation)

## [1] "Charcoal Fabric"          "Walnut Finish"
## [3] "Heather Gray Fabric"     "Sandstone Fabric"
## [5] "Oak Finish"              "Black"
## [7] "White"                   "Black Spot"
## [9] "White Spot"              "Black Show"
## [11] "White Show"              "Black Plus"
## [13] "White Plus"              "Configuration: Fire TV Stick"
## [15] "Black Dot"               "White Dot"
```

b.

```
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_counts <- alexa_data %>%
  count(variation)
print(variation_counts)

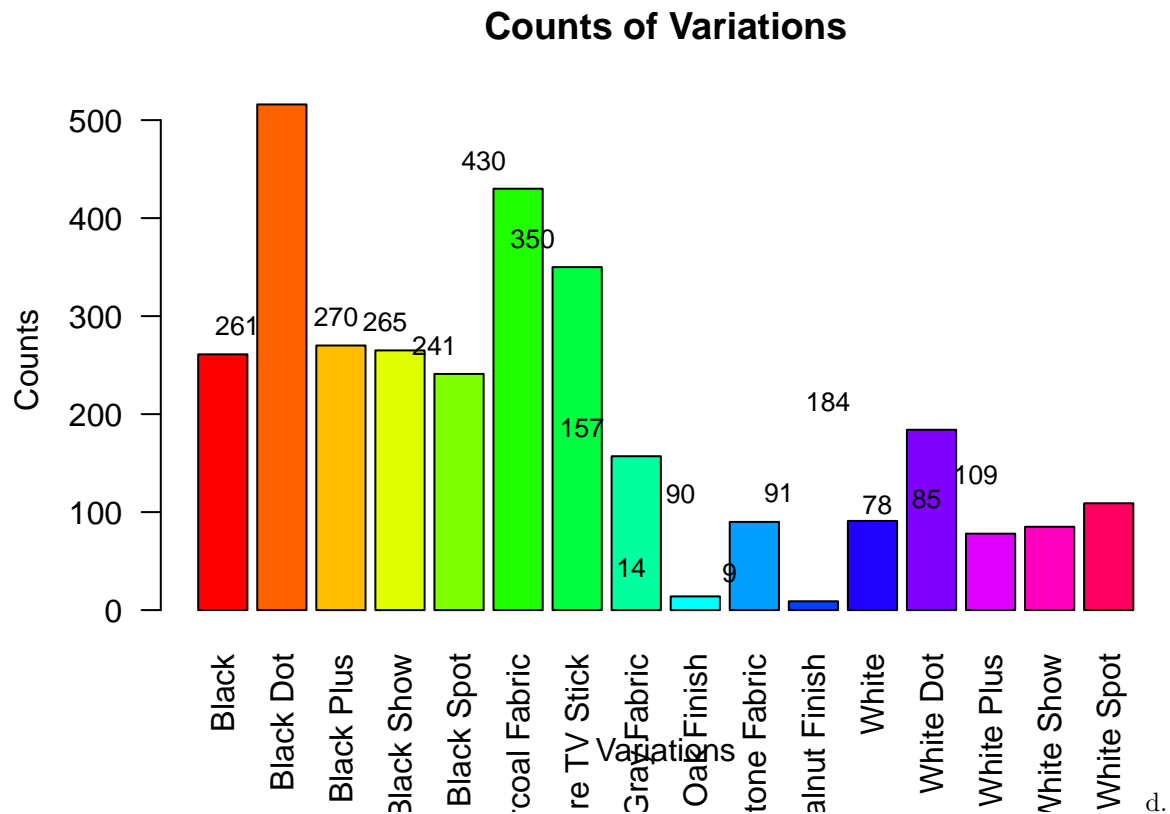
## # A tibble: 16 x 2
##   variation      n
##   <chr>      <int>
## 1 Black      261
## 2 Black Dot  516
## 3 Black Plus 270
```

```
## 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
```

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

c.

```
library(dplyr)
load("variations.RData")
variation_counts$variation <- gsub(" +", " ", variation_counts$variation)
variation_counts$variation <- trimws(variation_counts$variation)
bar_data <- variation_counts$n
bar_names <- variation_counts$variation
barplot(
  bar_data,
  main = "Counts of Variations",
  col = rainbow(length(bar_data)),
  names.arg = bar_names,
  xlab = "Variations",
  ylab = "Counts",
  las = 2,
  border = "black"
)
text(x = seq_along(bar_data), y = bar_data, labels = bar_data, pos = 3, cex = 0.8, col = "black")
```

```
library(ggplot2)
library(dplyr)
load("variations.RData")
variation_counts$variation <- gsub(" +", " ", variation_counts$variation)
variation_counts$variation <- trimws(variation_counts$variation)
bw_variations <- variation_counts %>%
  filter(grepl("Black|White", variation))
bar_data <- as.matrix(bw_variations$n)
bar_names <- bw_variations$variation
barplot(
  bar_data,
  beside = TRUE,
  main = "Counts of Black and White Variations",
  col = c("black", "gray", "lightgray", "white"),
  names.arg = bar_names,
  xlab = "Variations",
  ylab = "Counts",
  las = 2,
  border = "black"
)
text(x = seq_along(bar_data), y = bar_data, labels = bar_data, pos = 3, cex = 0.8, col = "black")
```

Counts of Black and White Variations

