Date: 19/08/2024
Name: Aakash Verma
Registration No.: 24-08-26
Lab: CS602-Lab
Machine Learning for Cyber Security

Assignment 1: To explore R for understanding programming fundamentals for data analysis applications.

Snapshot:

1.

```
> data(iris)
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
```

2.

```
> col = names(iris)
> col
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
> class(col)
[1] "character"
> typeof(col)
[1] "character"
```

3.

```
> 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 ...
>
```

4.

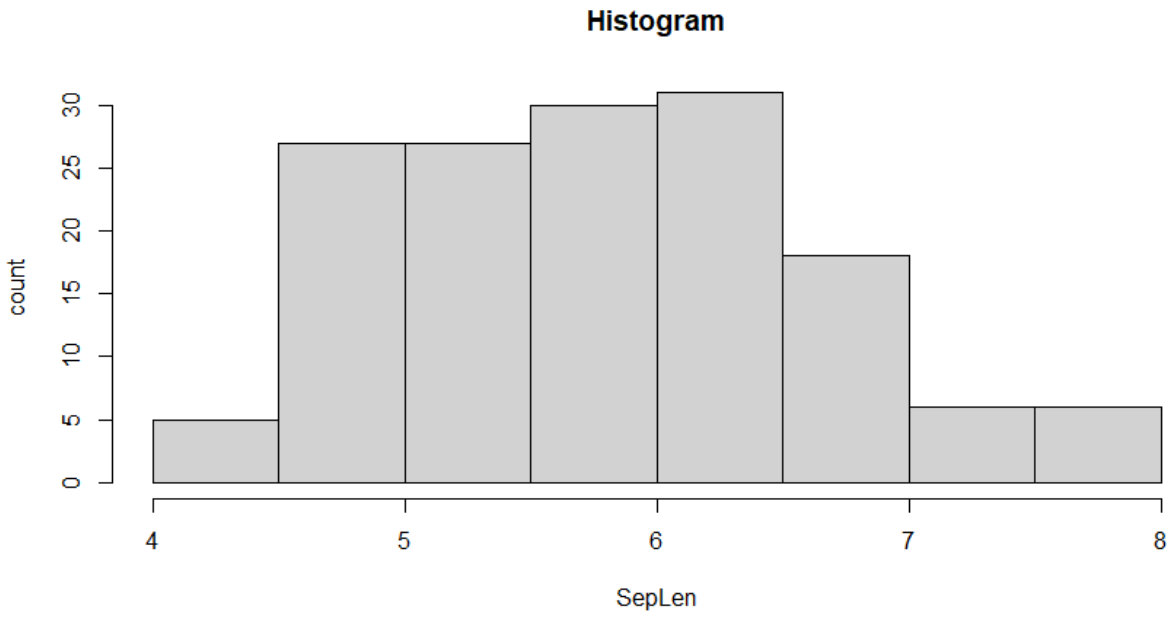
```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
>
```

5.

```
> unique(iris[, 5])
[1] setosa     versicolor virginica
Levels: setosa versicolor virginica
>
>
> colnames((iris))
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Species"
>
```

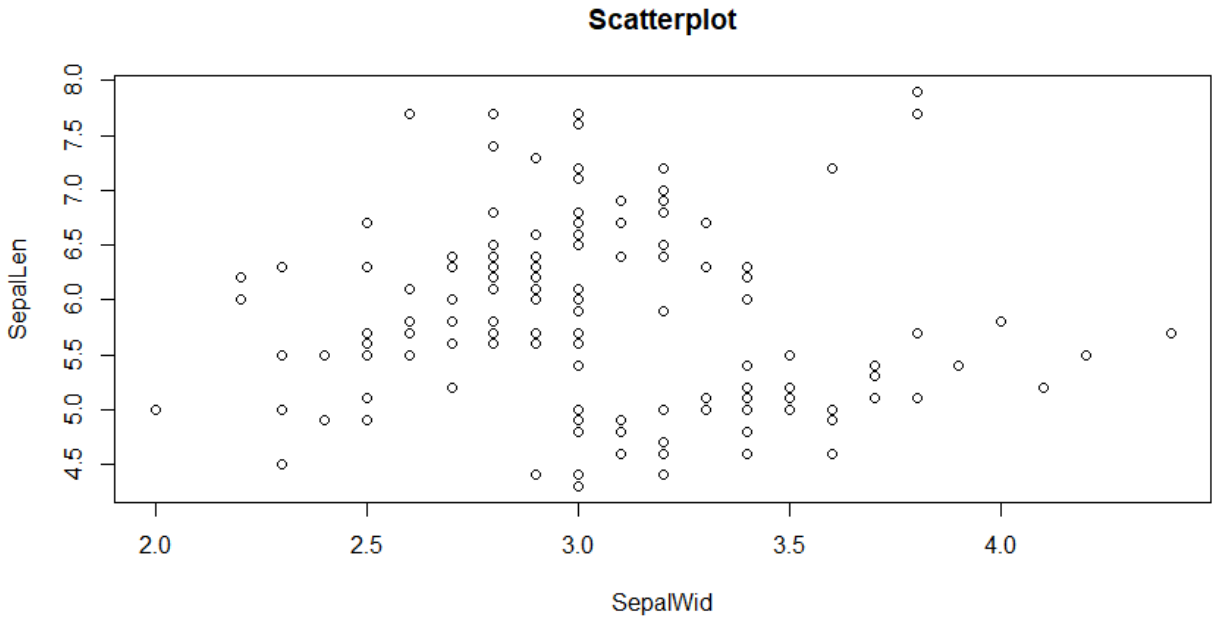
6.

```
> hist(iris$Sepal.Length,
+      main='Histogram',
+      xlab='SepLen',
+      ylab='count')
```



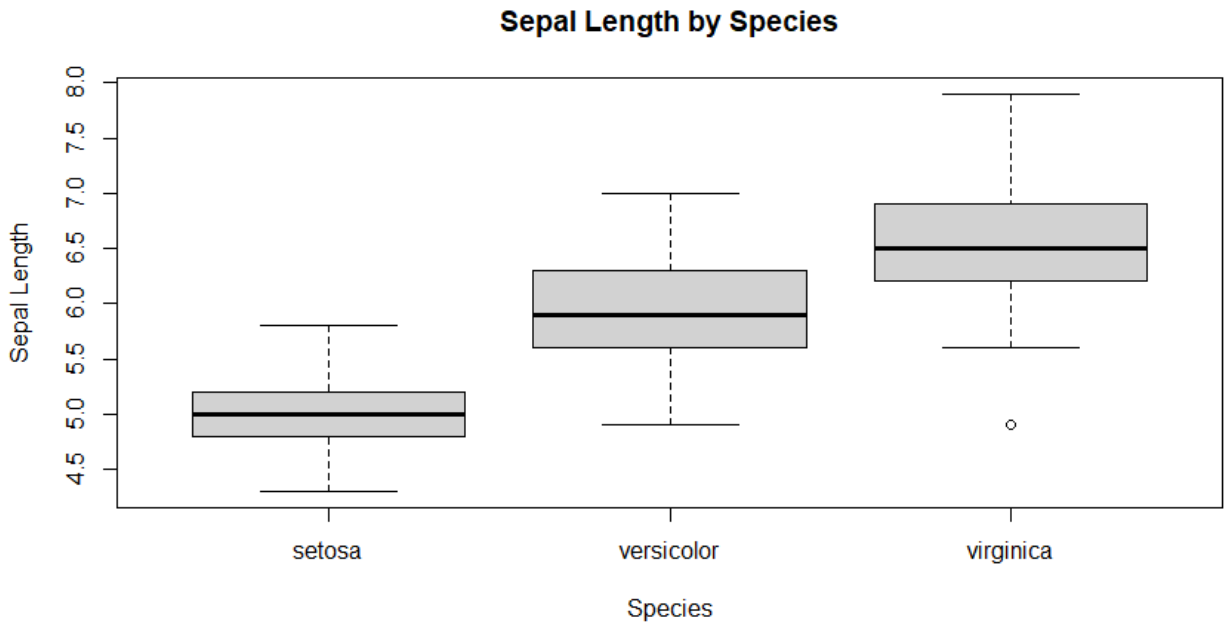
7.

```
> plot(iris$Sepal.Width, iris$Sepal.Length,
+      main='Scatterplot',
+      xlab='SepalWid',
+      ylab='SepalLen')
>
```



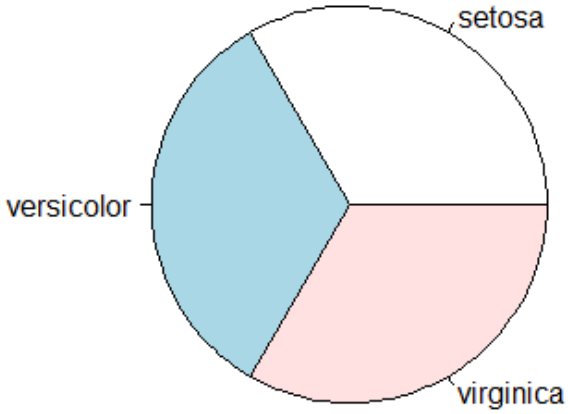
8.

```
> boxplot(Sepal.Length~Species,
+         data=iris,
+         main='Sepal Length -Species',
+         xlab='Species',
+         ylab='Sepal Length')
>
```



9.

```
> counts <- table(iris$Species)
> pie(counts)
>
```



10.

```
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
> |
```

11.

```
> A <- matrix(c(2, 4, 6, 8), nrow = 2, ncol = 2)
> B <- matrix(c(1, 3, 5, 7), nrow = 2, ncol = 2)
>
> dot_product <- A * B
```

```
> A
     [,1] [,2]
[1,]    2    6
[2,]    4    8
> B
     [,1] [,2]
[1,]    1    5
[2,]    3    7
```

```
> dot_product <- A * B
> dot_product
     [,1] [,2]
[1,]    2   30
[2,]   12   56
```

12.

```
> C <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, ncol = 3)
>
> D <- matrix(c(7, 8, 9, 10, 11, 12), nrow = 3, ncol = 2)
> C
function (...)  .Primitive("c")
> C
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> D
     [,1] [,2]
[1,]    7   10
[2,]    8   11
[3,]    9   12
```

```
> C %*% D
     [,1] [,2]
[1,]   76  103
[2,]  100  136
```

Conclusion:

1. Using R's built-in functions is more efficient than creating your functions and loops especially while working on some datasets.
2. Checking out the shape, dimension, and categorical data per column, as well as the min, max, and median per column, can give a better understanding of the data.
3. It is crucial to take care of nrow/ncol kind of params while dealing with matrices.