SHARDA UNIVERSITY
Beyond Boundaries

B.TECH HONS PROJECT

Exploratory analysis of dataset Iris

ADITYA KUMAR SINGH

EXPLORATORY ANALYSIS OF DATASET -IRIS

ADITYA KUMAR SINGH

Roll No-150101016

System Id-2015002511

Computer Science Engineering

2015-2019

BTech CSE-2nd year Cse-D

ACKNOWLEDEGMENT

I would like to thank **PROF. DR. ISHAN RANJAN** for providing me this golden opportunity to showcase my talent and think out of the box to learn and explore R-studio and various techniques involved while balancing my regular courses. This project encouraged me to do an in-depth study on how to analyse any given/random data set using R-studio.

Exploratory Analysis of Dataset-Iris

Introduction: Dataset- Iris Flower

Purpose of the analysis: To explore the dataset in R and share analysis about the data set.

Source: https://vincentarelbundock.github.io/Rdatasets/datasets.html

About Dataset:

The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis. The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres.

Libraries Used:

library(caret):Classification & regression training
library(car): Companion to applied regression
library(plyr):Tools for splitting, applying & combining data
library(dplyr): A Grammar of Data Manipulation
library(ggplot2):Create elegant data visualization using grammar of graphics
library(MASS): Support Functions and Datasets for Venables and Ripley's MASS

Other Libraries Used:

Lattice: Trellis graphics for r.

Kernel Lab: Kernel based machine learning lab.

randomForest: Breiman & Cutler's Random forest for classification and regression.

rPart: Recursive Partitioning & regression trees.

1. Arranging & Studying Dataset

1.1. **sapply:** List types for each attribute

```
> sapply(iris, class)

Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species 
   "numeric"    "numeric"    "numeric"    "numeric"  "character"
```

So, it is clear that there are 4 numeric and 1 character type variable in dataset.

1.1.1 dim(iris)

```
dim(iris)

[1] 150   5
```

dim(iris) gives the dimensions of the data set and it says it has 150 rows and 5 columns i.e. 150 observations and 5 variable

1.1.2 head(iris)

```
head(iris)
```

```
# A tibble: 6 × 5
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
         <dbl>       <dbl>        <dbl>       <dbl>   <chr>
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
```

While exploring the data set, the head command gives the first 6 rows of the data sets. It is observed that it has 5 columns with header names Sepal.Length, Sepal.Width, Petal.Length, Petal.Width and species. Species is one categorical type.

1.1.3. tail(iris)

```
tail(iris)

 # A tibble: 6 × 5
  Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
         <dbl>       <dbl>        <dbl>       <dbl>     <chr>
1          6.7         3.3          5.7         2.5 virginica
2          6.7         3.0          5.2         2.3 virginica
3          6.3         2.5          5.0         1.9 virginica
4          6.5         3.0          5.2         2.0 virginica
5          6.2         3.4          5.4         2.3 virginica
6          5.9         3.0          5.1         1.8 virginica
```

The tail command gives the last 6 rows of the data set.

1.1.4. str(iris)

```
str(iris)

Classes ‘tbl_df’, ‘tbl’ and '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     : chr  "setosa" "setosa" "setosa" "setosa" ...
- attr(*, "spec")=List of 2
 ..$ cols   :List of 5
 .. ..$ Sepal.Length: list()
 .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
 .. ..$ Sepal.Width : list()
 .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
 .. ..$ Petal.Length: list()
 .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
 .. ..$ Petal.Width : list()
 .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
 .. ..$ Species     : list()
 .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
 ..$ default: list()
 .. ..- attr(*, "class")= chr  "collector_guess" "collector"
 ..- attr(*, "class")= chr "col_spec"
```

From the structure of the iris data set we find that the data set consists of 4 numerical variable (Sepal.Length, Sepal.Width, Petal.Length and Petal.Width) and one Character type variable(Species).

1.1.5 summary(iris)

```
summary(iris)

  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width      Species
 Min.   :4.300   Min.   :2.000   Min.   : 1.000   Min.   : 0.100   Length:150
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   Class: chara
cter
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   Mode:charact
er
 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
```

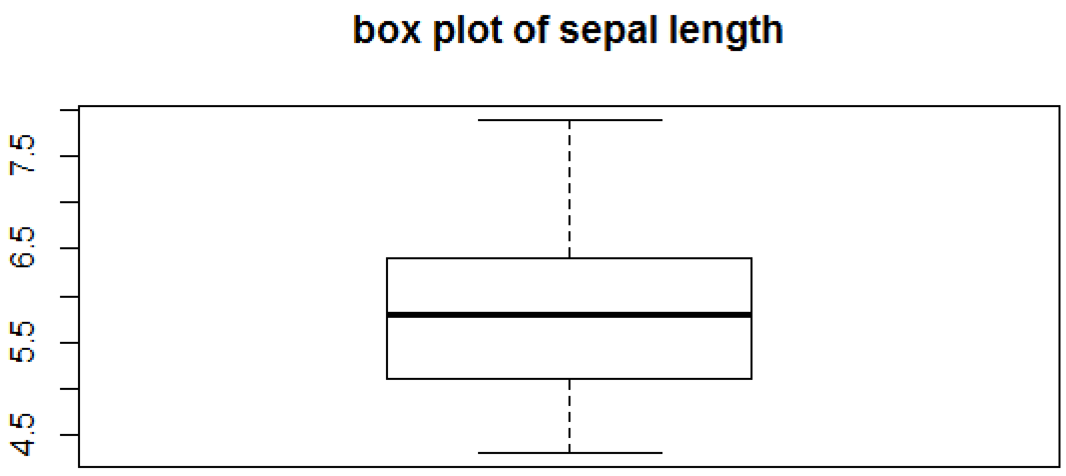
The data set has data of 150 flowers. The data is composed of 50 setosa flowers, 50 versicolor flowers and 50 virginica flowers.

Mean of Sepal Length, Sepal Width, Petal Length, Petal Width is 5.843, 3.057, 3.758, and 1.199 respectively.

2. Plotting

2.1 Boxplot for Sepal Length

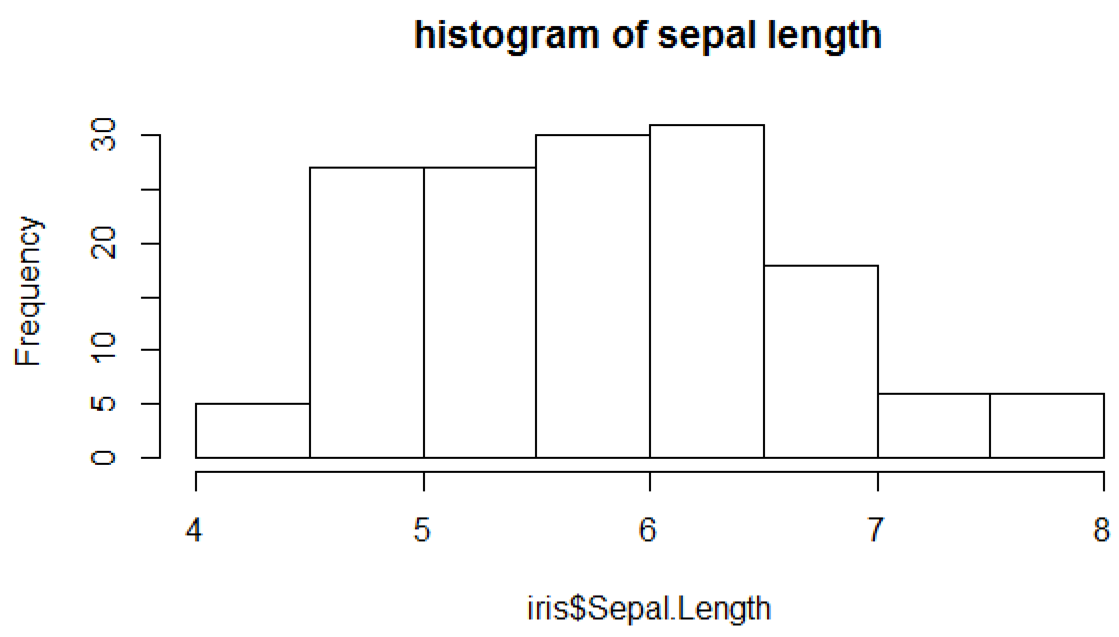
```
sepallength = iris[,1]
> boxplot(sepallength,main="box plot of sepal length")
```



For Sepal.Length the minimum length of the sepal is 4.3 and the maximum is 7.9 and median is at 5.8. 75% of the data has the sepal length below 6.4. 50% of the sepal length values lie above 5.8 and 50% lie between 5.1 and 5.8.

2.1.1 Histogram for Sepal Length

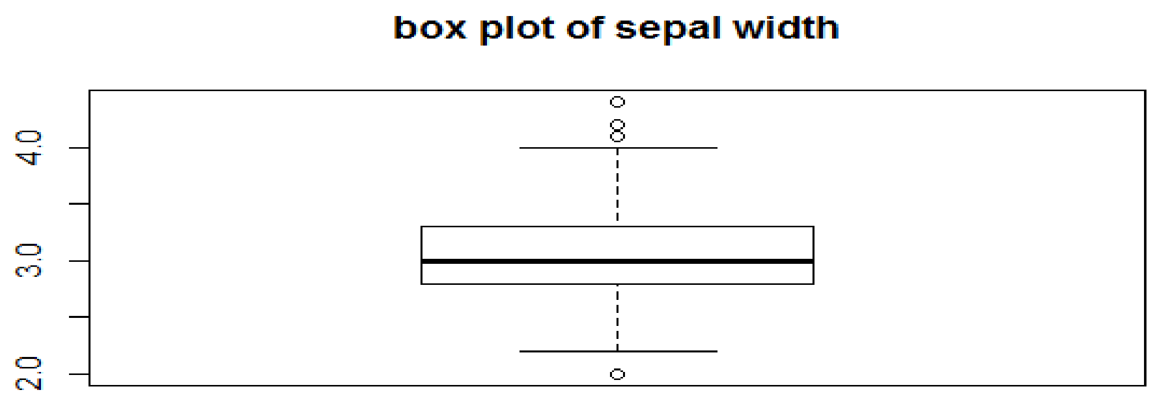
```
hist(iris$Sepal.Length,main="histogram of sepal length")
```



Here the mean is slightly greater than the median which says that the Sepal.Length value is bit positively skewed and also the histogram also says the same.

2.2. Boxplot for Sepal Width

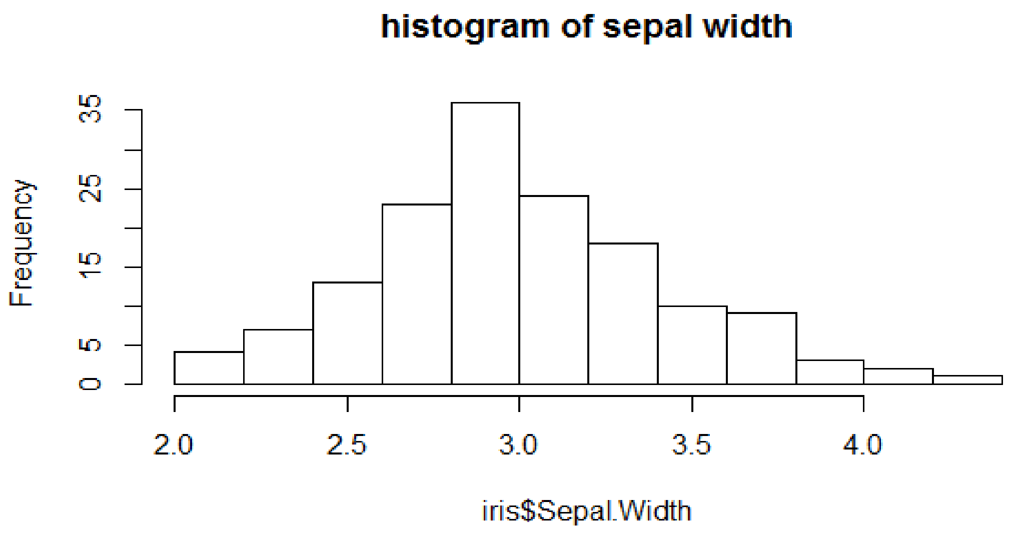
```
sepalwidth = iris[,2]
> boxplot(sepalwidth,main="box plot of sepal width")
```



From the summary plot Minimum sepal width is 2.0 and maximum is 4.4 while the median lies at 3.0. But as we can see in this case the box plot shows that there are some outliers so we cannot consider 4.4 and 2.0 as the maximum and minimum values respectively. In the data 75% of the values have width of the sepal below 3.3 and 50% of the data have sepal width between 2.8 and 3.0 and 50% of the sepal width lie above 3.0.

2.2.1 Histogram for Sepal Width

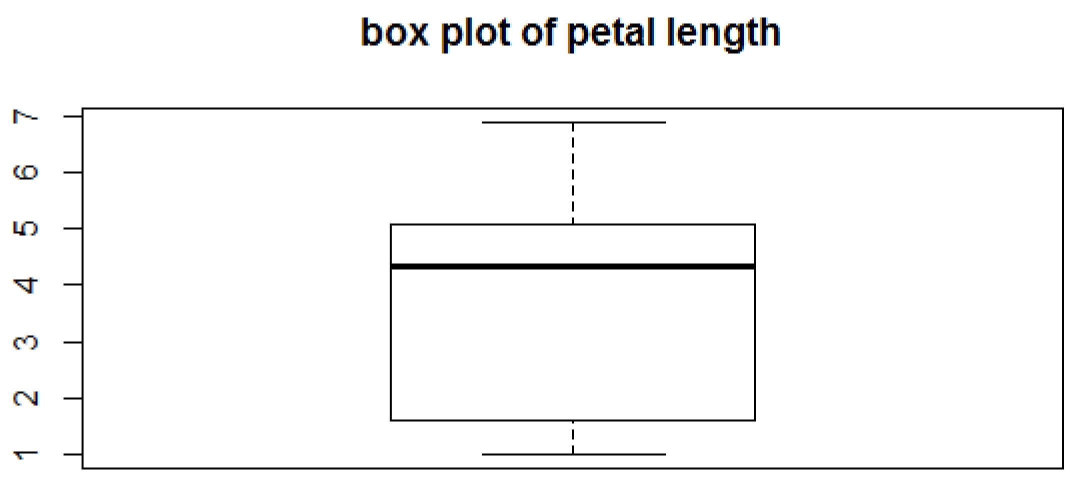
```
hist(iris$Sepal.Width,main="histogram of sepal width")
```



The histogram of the sepal width also says that the data is slightly positively skewed.

2.3 Boxplot for Petal Length

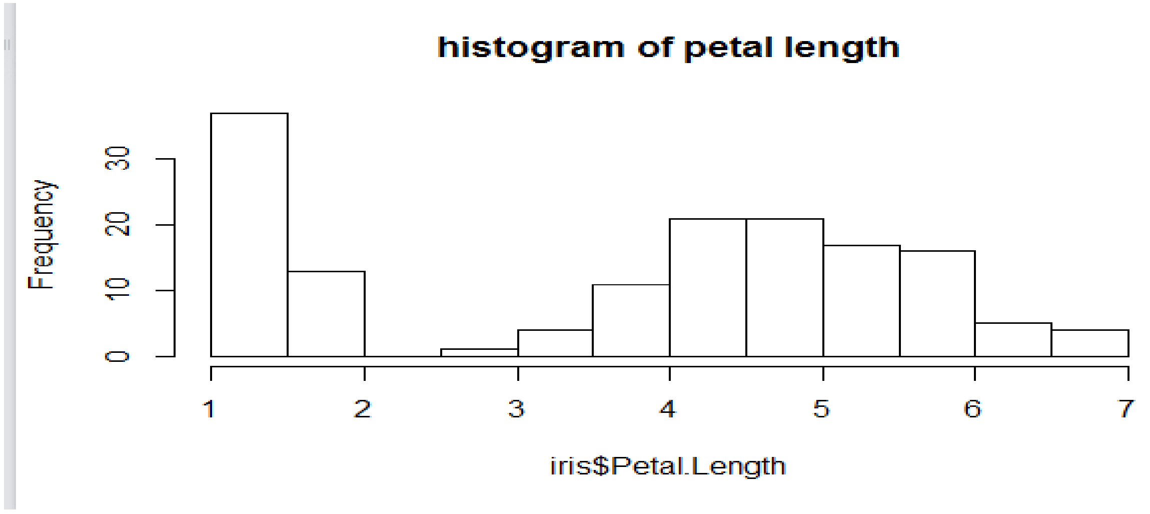
```
petallength = iris[,3]
> boxplot(petallength,main="box plot of petal length")
```



The minimum petal length is 1.0 and the maximum petal length is 6.9 and the median is at 4.35. 75% of the data has the petal length below 5.1. 50% of the data has the petal length value between 1.6 and 4.35.

2.3.1 Histogram for Petal Length

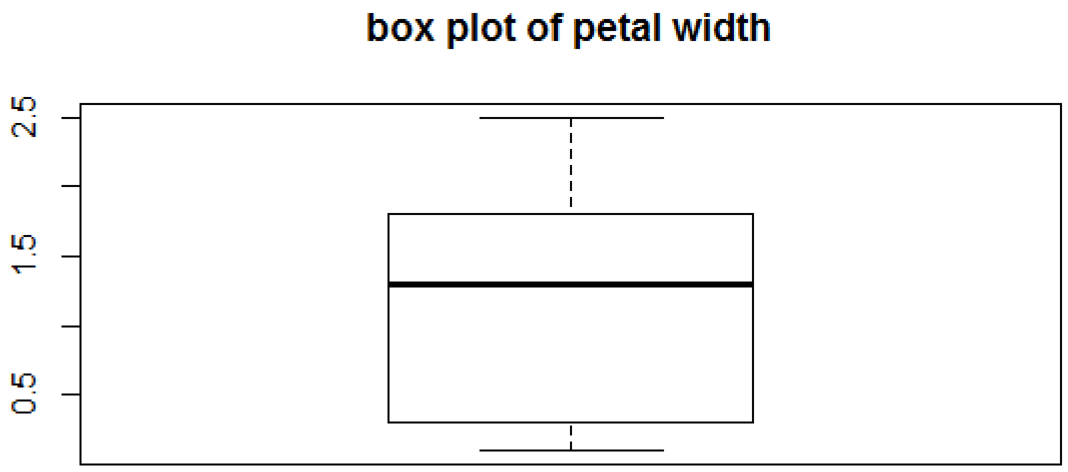
```
> hist(iris$Petal.Length,main="histogram of petal length")
```



From both the plots it is clear that the data is negatively skewed.

2.4. Boxplot for Petal Width

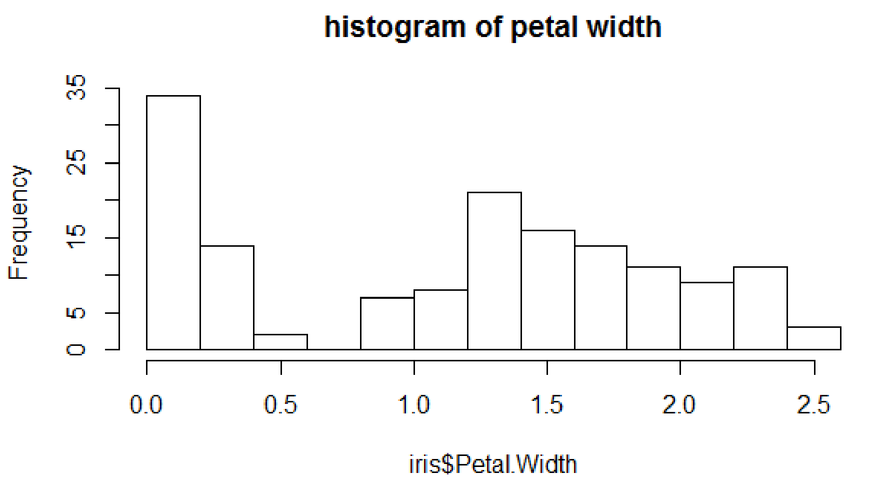
```
petalwidth = iris[,4]
> boxplot(petalwidth,main="box plot of petal width")
```



Similarly the petal width has a minimum value of 0.1 and maximum value of 2.5. The median of the petal width is at 1.3. 75% of the data has flowers whose petal width is below 1.8 and 50% of the data has petal width between 0.3 and 1.3 and 50% of the data has petal width below 1.3

2.4.1 Histogram for Petal Width

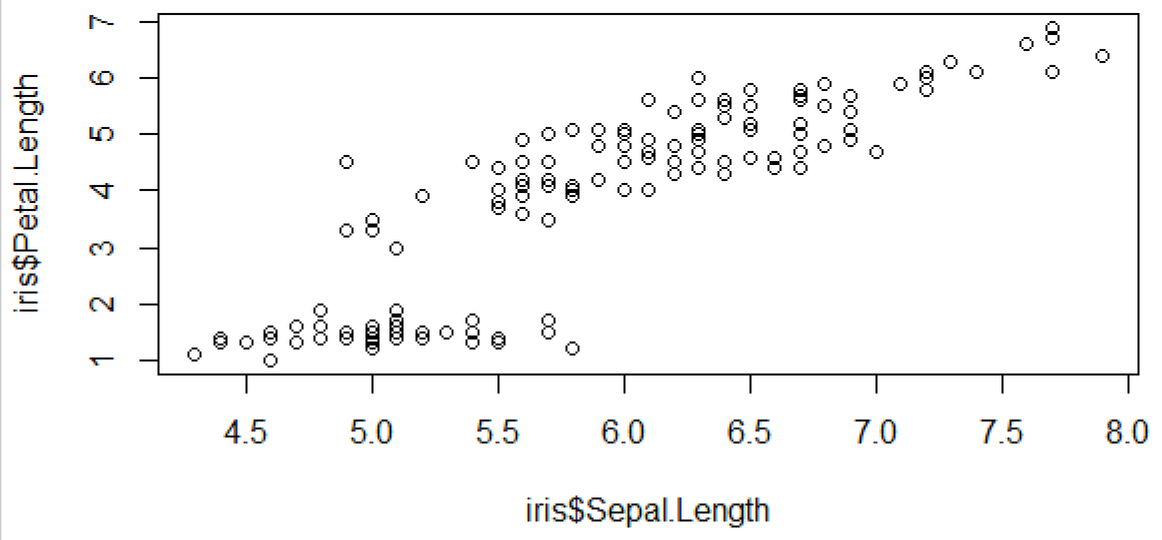
```
hist(iris$Petal.Width,main="histogram of petal width")
```



From the plots it is seen that petal width data is also slightly negatively skewed.

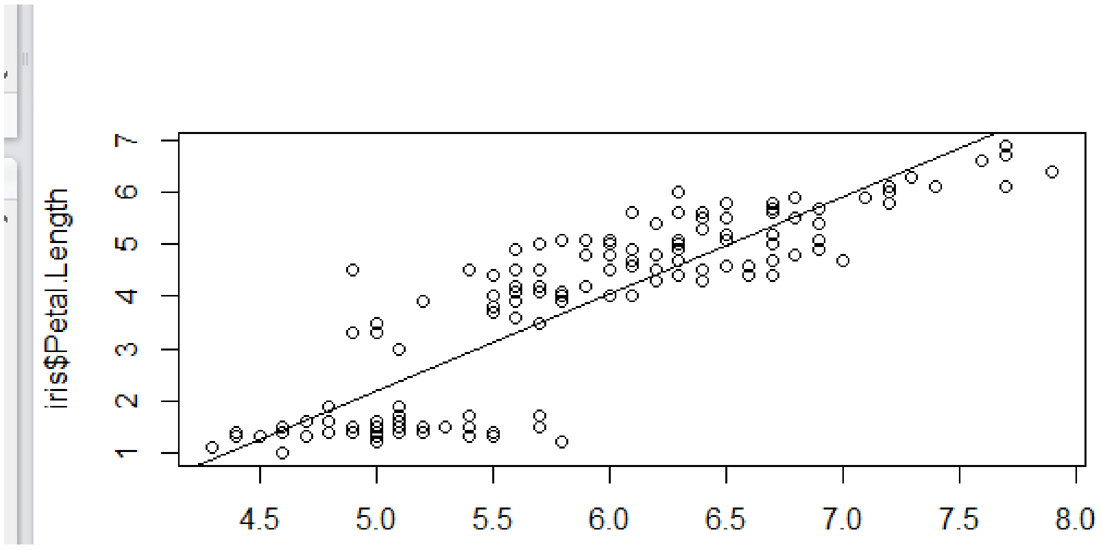
2.5 Scatter Plot against Sepal length & Petal Length

```
plot(iris $Sepal.Length, iris $Petal.Length)
```



2.5.1 Scatter Plot against Sepal length & Petal Length using Best Fit line

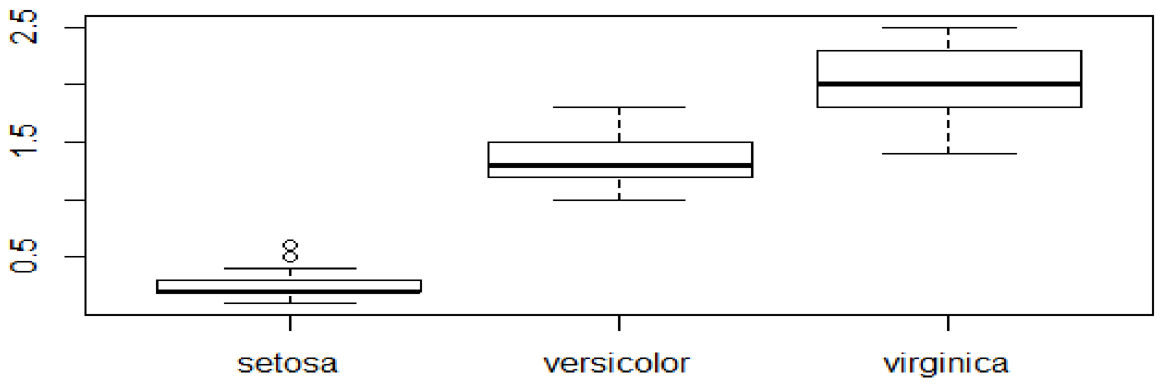
```
 plot(iris $Sepal.Length, iris $Petal.Length)
> abline(lm(iris $Petal.Length ~ iris $Sepal.Length))
```



It is seen that the petal length increases with the sepal length, for whatever be the species.

2.6. Boxplot representation of petal width of all three species

```
boxplot(iris $Petal.Width ~ iris $Species)
```



```
tapply(iris $Petal.Width, iris $Species, mean)

  setosa versicolor  virginica
   0.246      1.326      2.026
```

Based on their values and plot only, there means seem clearly different. So it is seen that petal width varies by species. It is evident that the petal width of virginica is more.

3. Covariance Analysis

```
cov(iris[,1:4])

             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    0.6856935  -0.0424340    1.2743154   0.5162707


Sepal.Width    -0.0424340   0.1899794   -0.3296564  -0.1216394
Petal.Length    1.2743154  -0.3296564    3.1162779   1.2956094
Petal.Width     0.5162707  -0.1216394    1.2956094   0.5810063
```

From the above it is clear that the Sepal length is positively related with petal length and width and negatively related with sepal width.Sepal width is negatively related to sepal length ,petal length and petal width which means sepal width decreased when these variables increases.

4. Correlation Analysis

```
  cor(iris[,1:4])

             Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000
```

And the correlation table says the degree to which these variables are correlated. From the table it is seen that there is a strong positive correlation of sepal length with petal length (87.17%) and also with petal width (81.7%). We don't see a strong correlation between sepal length and sepal width as the correlation percentage is just 11.75%. Petal length also has a strong correlation with petal width (96.3%).

5. Evaluate Some Algorithms

1. Set-up the test harness to use 10-fold cross validation.
2. Build 5 different models to predict species from flower measurements
3. Select the best model.

➔ Test Harness

```
> control <- trainControl(method="cv", number=10)
> metric <- "Accuracy"
```

We will 10-fold crossvalidation to estimate accuracy.

This will split our dataset into 10 parts, train in 9 and test on 1 and release for all combinations of train-test splits. We will also repeat the process 3 times for each algorithm with different splits of the data into 10 groups, in an effort to get a more accurate estimate.

We are using the metric of "Accuracy" to evaluate models. This is a ratio of the number of correctly predicted instances in divided by the total number of instances in the dataset multiplied by 100 to give a percentage.

➔ Build Models

We don't know which algorithms would be good on this problem or what configurations to use. Let's evaluate 5 different algorithms:

Linear Discriminant Analysis (LDA)

Classification and Regression Trees (CART).

k-Nearest Neighbors (kNN).

Support Vector Machines (SVM) with a linear kernel.

Random Forest (RF)

This is a good mixture of simple linear (LDA), nonlinear (CART, kNN) and complex nonlinear methods (SVM, RF). We reset the random number seed before reach run to ensure that the evaluation of each algorithm is performed using exactly the same data splits. It ensures the results are directly comparable.

Let's build our five models:

```
 set.seed(7)

  > fit.lda <- train(Species~., data=iris, method="lda", metric=
metric,     trControl=control)

 > set.seed(7)
  > fit.cart <- train(Species~., data=iris, method="rpart", metr
ic=metric, trControl=control)
 set.seed(7)
 > fit.knn <- train(Species~., data=iris, method="knn", metric=m
etric,    trControl=control)

 > set.seed(7)
 > fit.svm <- train(Species~., data=iris, method="svmRadial",met
ric=metric, trControl=control)

 > set.seed(7)
 > fit.rf <- train(Species~., data=iris, method="rf", metric=met
ric, trControl=control)
```

➔ **Select Best Model**

```
 > results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit
.knn,   svm=fit.svm, rf=fit.rf))
 > summary(results)
```

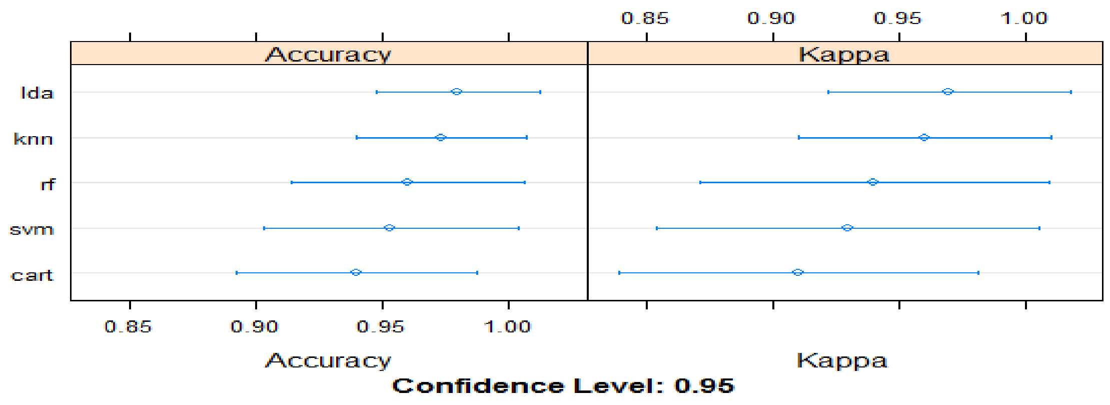
```
Call:
summary.resamples(object = results)

Models: lda, cart, knn, svm, rf
Number of resamples: 10

Accuracy
       Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
lda  0.8667  1.0000 1.0000 0.9800       1    1    0
cart 0.8000  0.9333 0.9333 0.9400       1    1    0
knn  0.8667  0.9500 1.0000 0.9733       1    1    0
svm  0.8000  0.9333 1.0000 0.9533       1    1    0
rf   0.8000  0.9333 1.0000 0.9600       1    1    0

 Kappa
     Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
 lda   0.8   1.000    1.0 0.97       1    1    0
 cart  0.7   0.900    0.9 0.91       1    1    0
 knn   0.8   0.925    1.0 0.96       1    1    0
 svm   0.7   0.900    1.0 0.93       1    1    0
 rf    0.7   0.900    1.0 0.94       1    1    0
```

We can see the accuracy of each classifier and also other metrics like Kappa:

LDA has highest accuracy & Kappa, so LDA is the best accurate model in this case.

We can also create a plot of the model evaluation results and compare the spread and the mean accuracy of each model

```
dotplot(results)
```



Summarization of best model (LDA):

```
print(fit.lda)

Linear Discriminant Analysis

150 samples
 4 predictor
 3 classes: 'setosa', 'versicolor', 'virginica'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
Resampling results:

 Accuracy  Kappa
 0.98      0.97
```

CONCLUSION

I have successfully Studied & analyzed Iris dataset using R-studio. The R-environment made it very easy to interpret the data and understand its attributes as well as its relationship efficiently.