DPA Assignment 1

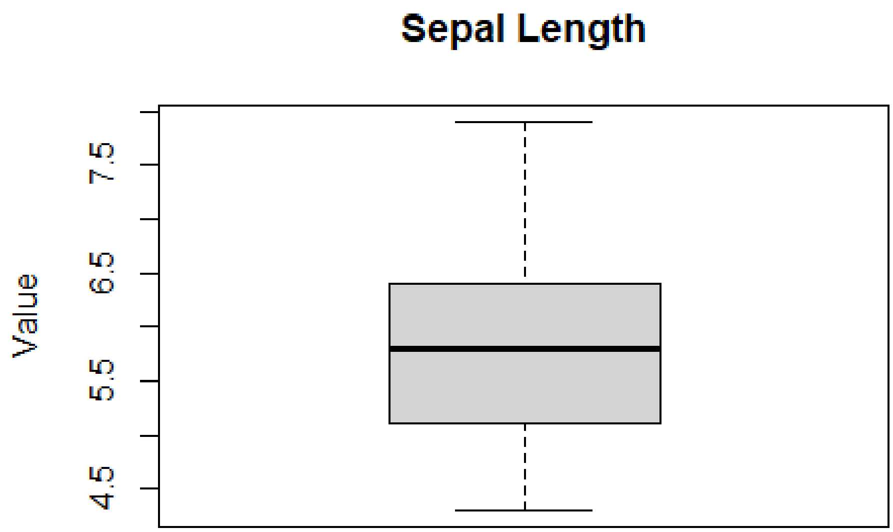
Girish

2023-01-17

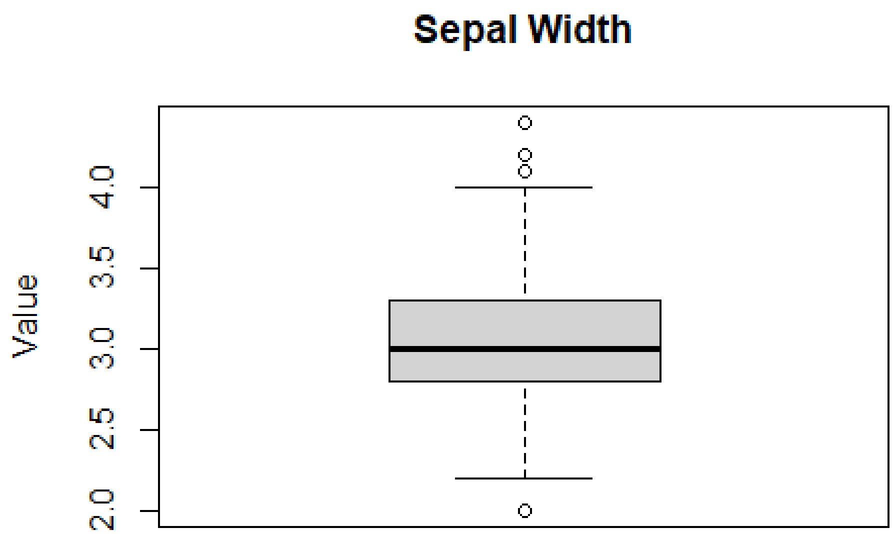
```
#Problem 1
setwd("D:/Users/giris/Documents/IIT/R Datasests/")

library (datasets)

#Load iris sample dataset into dataframe
iris_data <- data.frame(iris)

#Assigning a name for each feature
sep_len <- iris_data$Sepal.Length
sep_wid <- iris_data$Sepal.Width
pet_len <- iris_data$Petal.Length
pet_wid <- iris_data$Petal.Width
species <- iris_data$Species

#Showing summary and first 6 rows of iris dataset
summary(iris_data)

##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300  
##  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  
##        Species  
##  setosa    :50  
##  versicolor:50  
##  virginica :50  
##                 
##                 
##                 

head(iris_data)

##   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
```

#Creating box plot of each of the 4 features

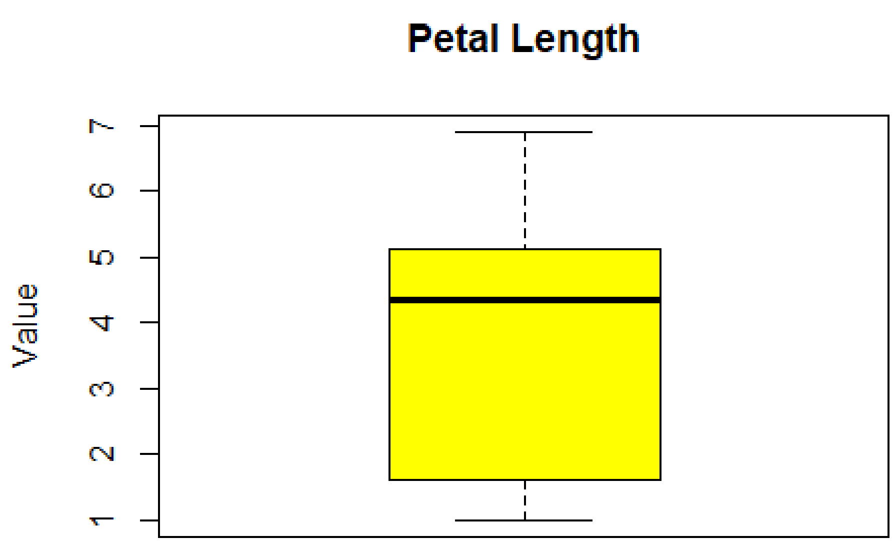
```
#Sepal Length Box Plot
boxplot(sep_len,data=iris_data, main="Sepal Length",
   ylab="Value")
```



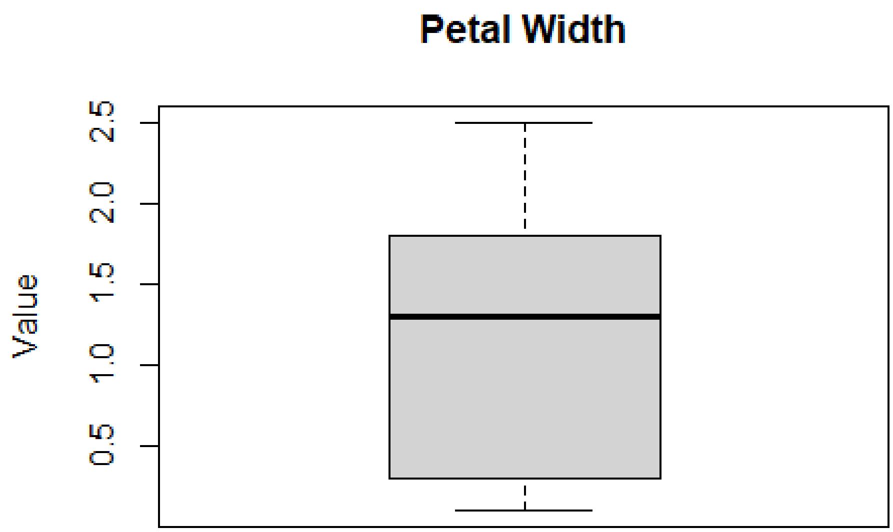

```
#Sepal Width Box Plot
boxplot(sep_wid,data=iris_data, main="Sepal Width",
   ylab="Value")
```




```r
#Petal Length Box Plot
boxplot(pet_len,data=iris_data, main="Petal Length",
   ylab="Value", col = "Yellow")
```



```r
#Petal Width Box Plot
boxplot(pet_wid,data=iris_data, main="Petal Width",
   ylab="Value")
```



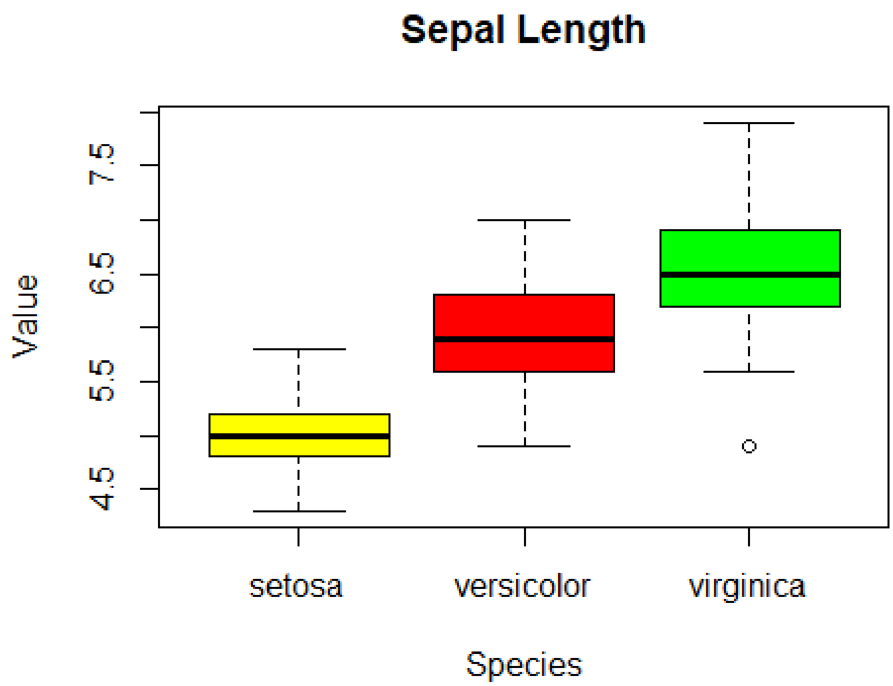

```r
#Compute Empirical IQR of each feature
sep_len_iqr = IQR(sep_len)
sep_wid_iqr = IQR(sep_wid)
pet_len_iqr = IQR(pet_len)
pet_wid_iqr = IQR(pet_wid)

sprintf("Sepal Length Empirical IQR - %f", sep_len_iqr)

## [1] "Sepal Length Empirical IQR - 1.300000"

sprintf("Sepal Width Empirical IQR - %f", sep_wid_iqr)

## [1] "Sepal Width Empirical IQR - 0.500000"

sprintf("Petal Length Empirical IQR - %f", pet_len_iqr)

## [1] "Petal Length Empirical IQR - 3.500000"

sprintf("Petal Width Empirical IQR - %f", pet_wid_iqr)

## [1] "Petal Width Empirical IQR - 1.500000"
```

#The petal length has the highest Empirical IQR so it is highlighted in Yellow in the plot above

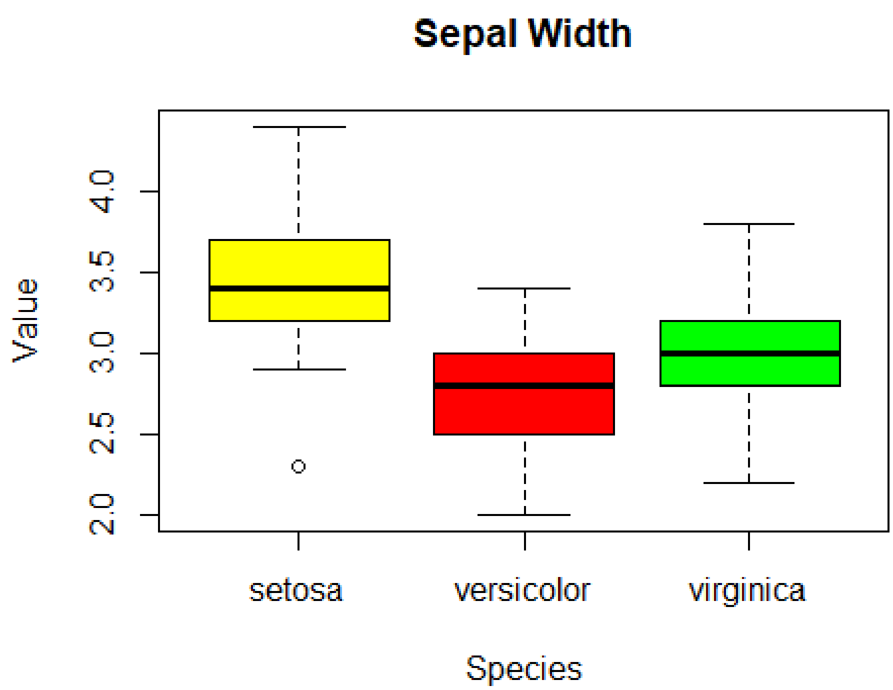
```r
#Calculate the parametric standard deviation for each feature

sep_len_sd = sd(sep_len)
sep_wid_sd = sd(sep_wid)
pet_len_sd = sd(pet_len)
pet_wid_sd = sd(pet_wid)

sprintf("Sepal Length SD - %f", sep_len_sd)

## [1] "Sepal Length SD - 0.828066"

sprintf("Sepal Width SD - %f", sep_wid_sd)

## [1] "Sepal Width SD - 0.435866"

sprintf("Petal Length SD - %f", pet_len_sd)

## [1] "Petal Length SD - 1.765298"

sprintf("Petal Width SD - %f", pet_wid_sd)

## [1] "Petal Width SD - 0.762238"
```

#The results of the standard deviations from above do agree with the empirical values because we can see that the Petal Length feature has both the highest standard deviation and empirical IQR while the Sepal Width feature has both the lowest standard deviation and empirical IQR.

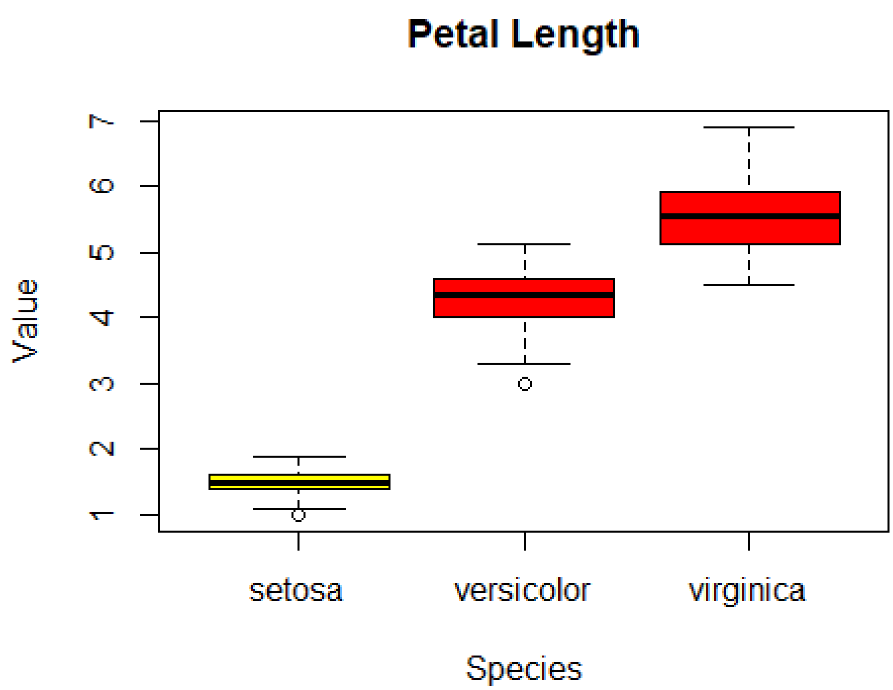
```r
#Sepal Length Colored Box Plot
boxplot(sep_len~species,data=iris_data, main="Sepal Length", xlab="Species",
   ylab="Value", col=c("Yellow","Red","Green"))
```



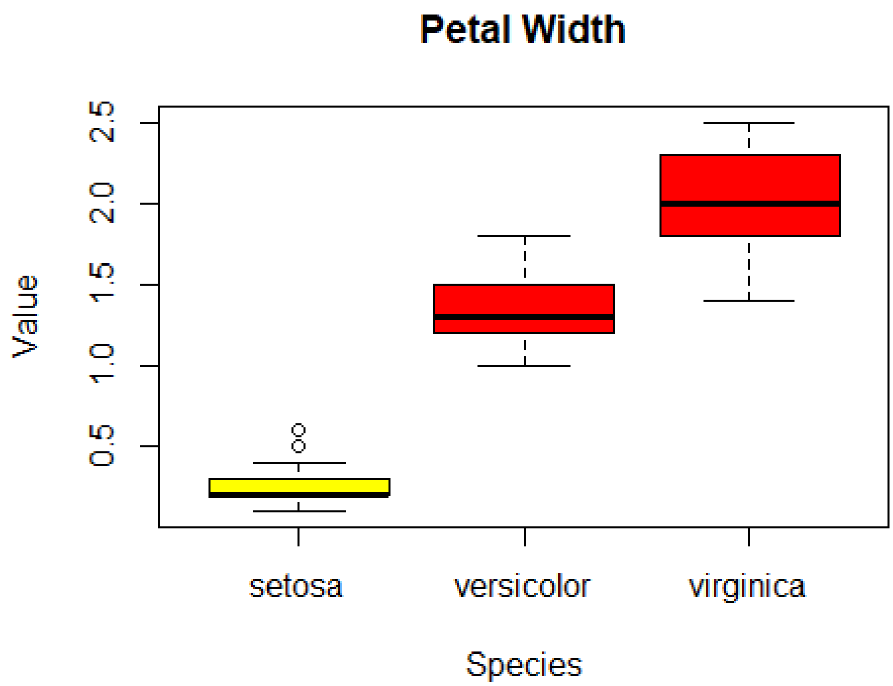

```r
#Sepal Width Colored Box Plot
boxplot(sep_wid~species,data=iris_data, main="Sepal Width", xlab="Species",
   ylab="Value", col=c("Yellow","Red","Green"))
```



```
#Petal Length Colored Box Plot
boxplot(pet_len~species,data=iris_data, main="Petal Length", xlab="Species",
   ylab="Value", col=c("Yellow","Red","Red"))
```

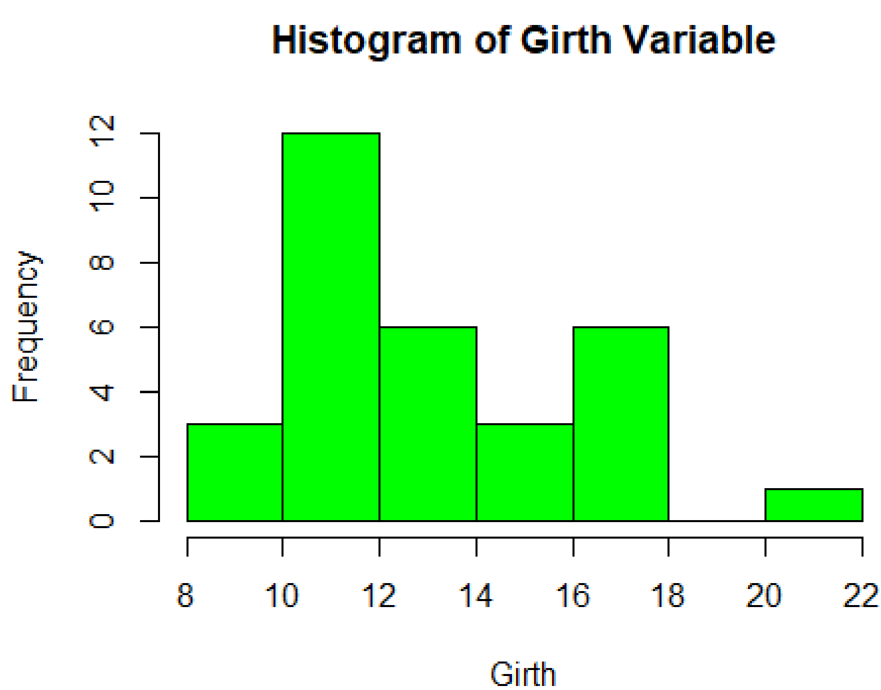



```
#Petal Width Colored Box Plot
boxplot(pet_wid~species,data=iris_data, main="Petal Width", xlab="Species",
   ylab="Value", col=c("Yellow","Red","Red"))
```

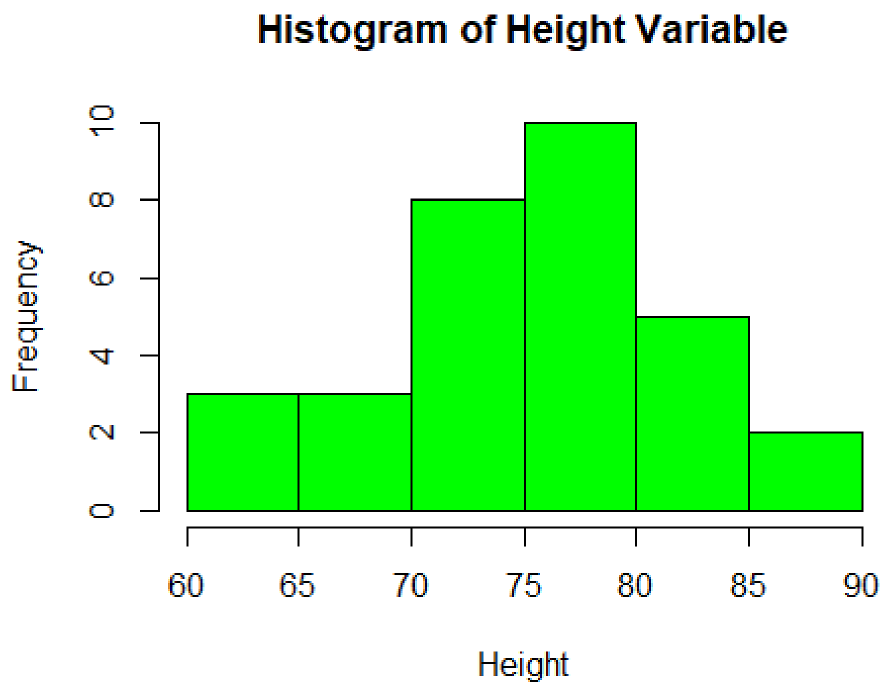



```
#Based on the plots below, the setosa flower type exhibits significantly
different Petal Length/Width when separated from the others. We can observe
that the Setosa has a much smaller Petal Length/Width when compared to the
other flower types.

#Problem 2
library(moments)

#Load trees sample dataset into dataframe
trees_data <- data.frame(trees)

#Assigning a name for each feature
girth <- trees_data$Girth
height <- trees_data$Height
volume <- trees_data$Volume

#Showing summary and first 6 rows of trees dataset
summary(trees_data)

##      Girth           Height       Volume
##  Min.   : 8.30   Min.   :63   Min.   :10.20
##  1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40
##  Median :12.90   Median :76   Median :24.20
##  Mean   :13.25   Mean   :76   Mean   :30.17
##  3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30
##  Max.   :20.60   Max.   :87   Max.   :77.00
```

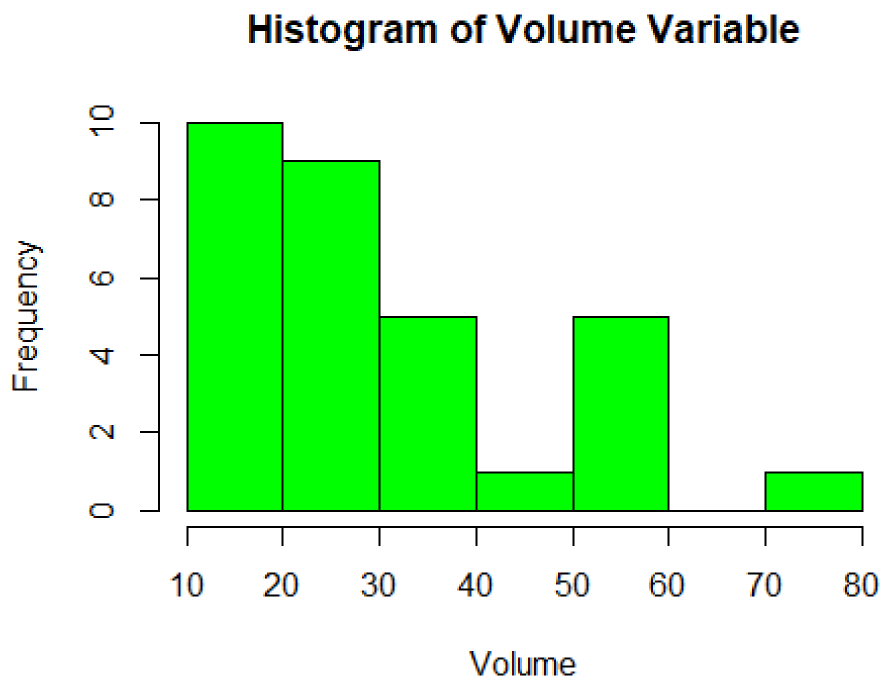
```r
head(trees_data)

##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7

#5-number summary of Girth
fivenum(girth)

## [1]  8.30 11.05 12.90 15.25 20.60

#5-number summary of Height
fivenum(height)

## [1] 63 72 76 80 87

#5-number summary of Volume
fivenum(volume)

## [1] 10.2 19.4 24.2 37.3 77.0

#Creating a histogram for Girth variable
hist(girth, main="Histogram of Girth Variable",
     xlab="Girth", col="green")
```



```r
#Creating a histogram for Height variable
hist(height, main="Histogram of Height Variable",
     xlab="Height", col="green")
```




```r
#Creating a histogram for Volume variable
hist(volume, main="Histogram of Volume Variable",
     xlab="Volume", col="green")
```



#A normal distribution histogram can be considered to be one that is bell-shaped. Based on the three variable histograms plots above, it can be seen that the Height histogram displays a normal distribution because of this bell-shaped form with only one peak.

#It can be seen that both Girth and Volume variables exhibit a positive skeweness when compared to the Height histogram because those 2 histograms exhibit some form of positive skeweness since on the left side, we can see larger values and on the right side, we can see smaller values. Height exhibits a slight negative skewness.

```
#Calculating the skewness of each variable
library(moments)
skewness(girth)

## [1] 0.5263163

skewness(height)

## [1] -0.374869

skewness(volume)

## [1] 1.064357
```

#Yes, when looking at the skewness and the visual inspection, we can see that they agree. The Girth and Volume both exhibit a positive skewness (0.53 and 1.06 respectively) while Height exhibits a negative skewness (-0.37).

```r
#Problem 3

#Load data from UCI repository
auto_mpg_data <- read.csv(file="https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data", header=F, sep="", as.is =4, col.names=c("mpg","cylinders","displacement","horsepower","weight","acceleration", "model year", "origin", "car name"))

#Showing summary and first 6 rows of dataset
summary(auto_mpg_data)

##       mpg          cylinders      displacement    horsepower
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Length:398
##  1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.2   Class :character
##  Median :23.00   Median :4.000   Median :148.5   Mode  :character
##  Mean   :23.51   Mean   :5.455   Mean   :193.4
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0
##  Max.   :46.60   Max.   :8.000   Max.   :455.0
##
##      weight      acceleration     model.year        origin
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000
##  1st Qu.:2224   1st Qu.:13.82   1st Qu.:73.00   1st Qu.:1.000
##  Median :2804   Median :15.50   Median :76.00   Median :1.000
##  Mean   :2970   Mean   :15.57   Mean   :76.01   Mean   :1.573
##  3rd Qu.:3608   3rd Qu.:17.18   3rd Qu.:79.00   3rd Qu.:2.000
##  Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000
##
##            car.name
##  ford pinto    :  6
##  amc matador   :  5
##  ford maverick :  5
##  toyota corolla:  5
##  amc gremlin   :  4
##  amc hornet    :  4
##  (Other)       :369

head(auto_mpg_data)

##   mpg cylinders displacement horsepower weight acceleration model.year
origin
## 1  18         8          307      130.0   3504         12.0         70
1
## 2  15         8          350      165.0   3693         11.5         70
1
## 3  18         8          318      150.0   3436         11.0         70
1
## 4  16         8          304      150.0   3433         12.0         70
1
## 5  17         8          302      140.0   3449         10.5         70
1
```

```
## 6  15         8           429      198.0   4341         10.0          70
1
##                      car.name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500
```

```
#Using as.numeric to obtain the column as a numeric vector but would receive
error since there is na present
auto_mpg_data$horsepower <- as.numeric(auto_mpg_data$horsepower)

## Warning: NAs introduced by coercion

#Original mean when the na records were ignored
original_mean <- mean(auto_mpg_data$horsepower, na.rm=TRUE)
print(original_mean)

## [1] 104.4694

#Calculating the median while ignoring na
median_auto_mpg <- median(auto_mpg_data$horsepower, na.rm=TRUE)
print(median_auto_mpg)

## [1] 93.5

#Replace na values with median using is.na()
auto_mpg_data$horsepower[is.na(auto_mpg_data$horsepower)] <- median_auto_mpg

#Using as.numeric to obtain the column as a numeric vector after replacing na
with median
auto_mpg_data$horsepower <- as.numeric(auto_mpg_data$horsepower)

#New mean when the na records were replaced with median
new_mean <- mean(auto_mpg_data$horsepower, na.rm=TRUE)
print(new_mean)

## [1] 104.304
```

#We can compare the original mean when NA were ignored which was 104.4694 and the new mean when the na records were replaced with the median which was 104.304. This shows that the mean has slightly decreased.

```
#Problem 4

#Load the Boston dataset
library(MASS)
boston_data <- data.frame(Boston)
```

```r
#Showing summary and first 6 rows of Boston dataset
summary(boston_data)
```

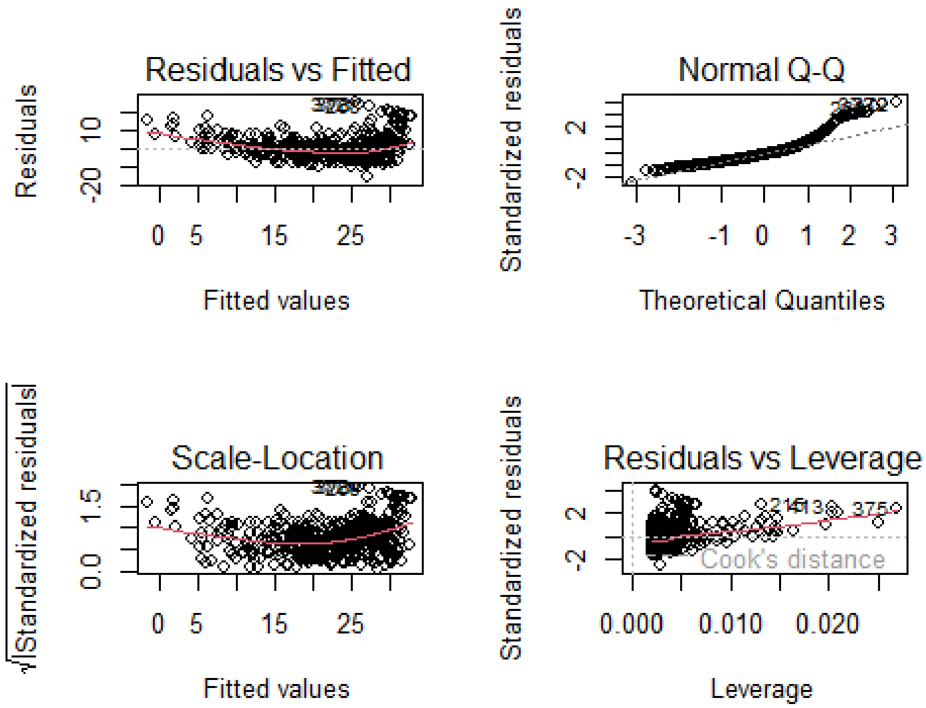
```
##       crim                zn             indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm             age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv      
##  Min.   : 1.73   Min.   : 5.00  
##  1st Qu.: 6.95   1st Qu.:17.02  
##  Median :11.36   Median :21.20  
##  Mean   :12.65   Mean   :22.53  
##  3rd Qu.:16.95   3rd Qu.:25.00  
##  Max.   :37.97   Max.   :50.00
```

```r
head(boston_data)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black
lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90
4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90
9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83
4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63
2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90
5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12
5.21
##   medv
## 1 24.0
```

```
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7

#Using lm to fit a regression between medv and lstat
fit <- lm(medv~lstat, data=boston_data)
summary(fit)

##
## Call:
## lm(formula = medv ~ lstat, data = boston_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16

#Plotting the resulting fit
plot(boston_data$lstat, boston_data$medv, col="blue", main = "Regression fit
between medv and lstat", xlab = "lstat", ylab = "medv")
abline(fit)
```


```
#Showing a plot of fitted values vs residuals (first plot/top left)
par (mfrow=c(2,2))
plot(fit)
```



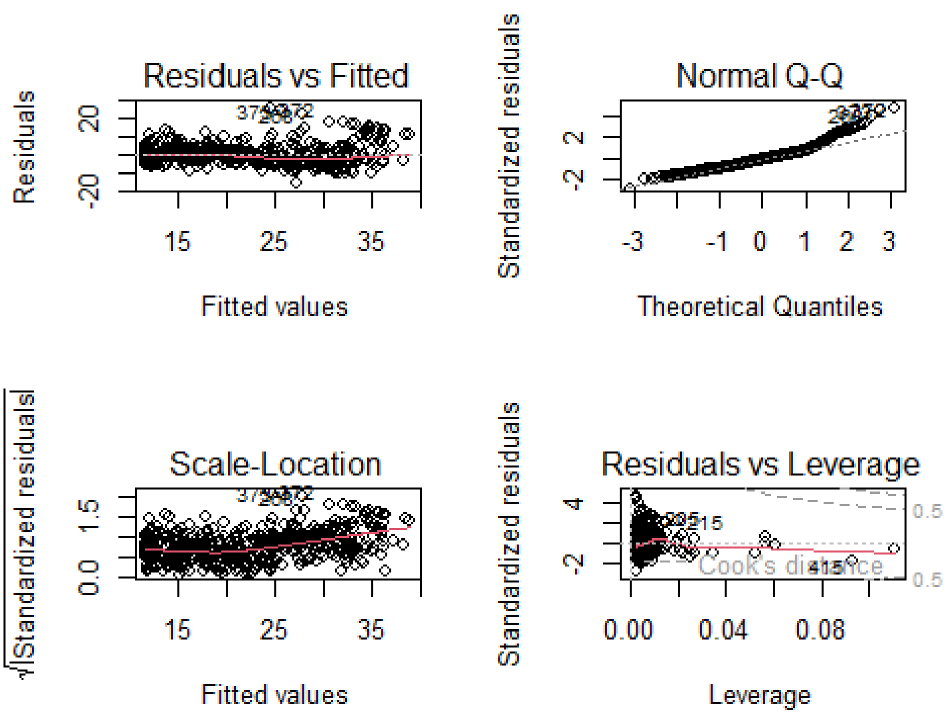
#If we look at the first plot above (fitted values vs residual with the resulting fit), there is a possibility that there is a non-linear relationship present. This can be because we can see a slope (curve) present within the fit which can represent a relationship that is not linear.

```
#Obtaining confidence interval after predicting response values for lstat
5,10 and 15
predict(fit, data.frame(lstat = (c(5, 10, 15))),interval = "confidence")

##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461

#Obtaining prediction interval after predicting response values for lstat
5,10 and 15
predict(fit, data.frame(lstat = (c(5, 10, 15))),interval = "prediction")

##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

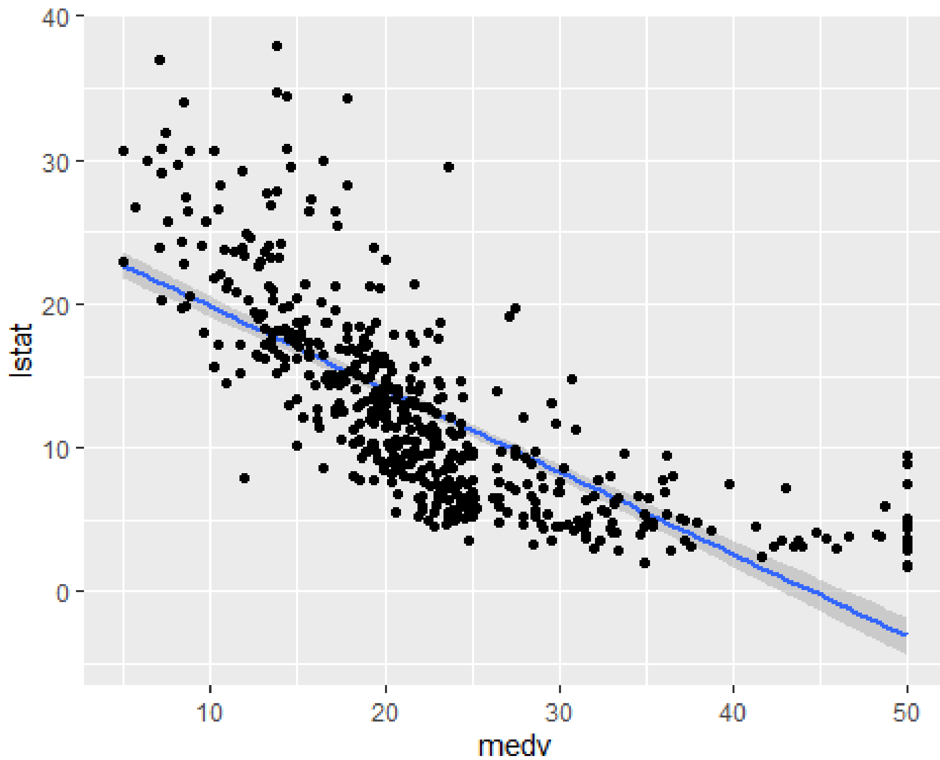
#It can be seen above that the results are not the same. A possible reason for this is because the prediction interval includes a wider range of values than the confidence interval since the prediction considers both reducible and irreducible error.

```
#Showing a new plot of fitted values vs residuals (first plot/top left) which
includes lstat^2
```

```
fit2 <- lm(boston_data$medv ~ boston_data$lstat + I(boston_data$lstat^2),
data=boston_data)
par(mfrow = c(2, 2))
plot(fit2)
```




```
#Plotting the relationship between the linear and non-linear fit for
comparison
library(ggplot2)
ggplot(boston_data,aes(medv,lstat,I(lstat^2)))+geom_point()+geom_smooth(metho
d="lm",se=TRUE)+
geom_point()

## `geom_smooth()` using formula = 'y ~ x'
```




```
#Showing summary
summary(fit2)

##
## Call:
## lm(formula = boston_data$medv ~ boston_data$lstat + 
I(boston_data$lstat^2),
##     data = boston_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -15.2834  -3.8313  -0.5295   2.3095  25.4148 
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             42.862007   0.872084   49.15   <2e-16 ***
## boston_data$lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(boston_data$lstat^2)  0.043547   0.003745   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.524 on 503 degrees of freedom
## Multiple R-squared:  0.6407, Adjusted R-squared:  0.6393 
## F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

#When looking at the multiple R-squared (0.6407) and the adjusted R-squared (0.6393), both are very closely related which means that our model fit the linear and non-linear fit properly and overfitting does not occur.