Assignment 1

Abhiram Ravipati

2024-02-05

Chapter 1 Recitation Exercises Chapter 2.2 Exercise 1

a) In the case where the sample size (n) is extremely large and the number of predictors (p) is small, a flexible statistical learning method is expected to perform better because a large sample size provides more data to estimate complex relationships between predictors and the response variables. The risk of overfitting is lower when there is sufficient data to support the complexity of the model.

b) In the case where there are large number of predictors (p) and the number of observations (n) is small, an inflexible statistical learning method is preferred generally. When the number of predictors is equal to or more than the number of data, flexible methods may have trouble with the overfitting. Inflexible methods may be reliable when there are fewer observations.

c) Generally, in this scenario a flexible statistical learning method is expected to perform better than an inflexible method. Because an inflexible method is able to model only linear relationships between the predictors and the response. Whereas flexible method is able to model non-linear relationships between the predictors and response. The flexible method is better at capturing the real relationship between the response and the predictors. Having more parameters for estimation makes the flexible methods less prone to overfit the data.

d) In this situation, a flexible statistical learning method would usually perform worse than an inflexible method. A flexible method will capture and tries to fit the noise in the data which leads to overfitting. Simpler models with lower variance may provide more accurate and stable predictions.

Exercise 2 a)This scenario is a regression problem because CEO salary (response variable) is a continuous variable. We want to understand the relationship between CEO salary and other variables so we are interested in inference. Our goal is to understand how various factors affect the salary of CEO. Here, the number of observations (n) is 500 and the number of predictors (p) is 3.

b)This scenario is a classification problem. We are interested to predict whether a new product will be a success or failure based on various features. Here the success or failure variable is a categorial variable with two classes. The number of observations (n) is 20 and the number of predictors (p) is 13.

c)We are interested in predicting the percent change in the USD/Euro exchange rate in relation to the weekly changes in the world stock markets. Percent change in USD/Euro (response variable) is a continuous variable. This scenario is a regression problem. Here, the number of observations (n) is 52 and the number of predictors (p) is 3.

Exercise 4

a)Classification:

Credit Scoring: Response: Whether an individual is likely to default on a loan Predictors: Credit History, income, debt-to-income ratio, employment status, etc., Goal: Prediction. The goal is to asses credit risk for new applicants.

Email Spam Filtering: Response: Whether an email is spam or not Predictors: Email content, sender, frequency of certain words, etc. Goal: Prediction. The goal is to filter out the spam emails automatically.

Medical Diagnosis: Response: Disease Classification (Cancer or Not) Predictors: Patient's medical history, test results, genetic markers, etc. Goal: The goal is to assist doctors in diagnosing diseases based on patient data.

b)Regression:

House Price Prediction: Response: Price of a house (continuous variable) Predictors: Number of bedrooms, square footage, location, amenities, etc. Goal: Prediction. The goal is to estimate the market value of a house.

Stock Price Forecasting: Response: Daily Stock Price Predictors: Historical stock prices, trading volumes, economic indicators, etc. Goal: Prediction. The goal is to forecast future stock prices for investment decisions.

Temperature Prediction: Response: Daily maximum temperature (continuous variable) Predictors: Previous weather conditions, geographical factors, time of the year, etc. Goal: Prediction. The goal is to forecast future temperatures for planning and decision-making.

c)Cluster Analysis:

Customer Segmentation: Goal: Grouping customers based on their purchasing behaviour. Variables: Purchase history, demographics, online behaviour. Use: Tailoring marketing strategies to different customer segments.

Image Segmentation in Medical Imaging: Goal: Identifying and grouping similar structures or tissues in medical images. Variables: Pixel intensity, texture features, spatial relationships. Use: Assisting radiologists in medical diagnosis and treatment planning.

Social Network Analysis: Goal: Identifying communities or groups of users with similar interests. Variables: Connections, interaction patterns, shared content. Uses: Enhancing targeted advertising, content recommendations or understanding social dynamics.

Exercise 6

In parametric statistical learning, a fixed set of parameters are calculated with the assumption that the relationship between the variables has a certain functional form. For inference and computing efficiency this simplicity and interpretability are important. Parametric models have trouble handling complex, non-linear patterns and they may produce biassed findings if the assumed form does not match the underlying relationship. Whereas non-parametric techniques are not restricted by a preset form they provide flexibility in capturing complex relationships but they may also be more computationally and interpretively difficult. The decision between parametric and non-parametric techniques is based on the objectives of the analysis and the properties of the data establishing a balance between interpretability and the capacity to reveal underlying complexity.

Exercise 7

a) Formula for computing Euclidean distance: d(p,q)=√(() ∑_i^n▒(q_i-p_i)^2) Observation X1 X2 X3 Y Euc. Dist 1 0 3 0 Red 3 2 2 0 0 Red 2 3 0 1 3 Red 3.16 4 0 1 2 Green 2.23 5 -1 0 1 Green 1.41 6 1 1 1 Red 1.73

b) For the Prediction with K = 1 we will be choosing the nearest neighbor which will be Observation 5, So Euclidean Distance will be 1.41. So, the prediction for K = 1 will be Green.

c) For the prediction with K = 3 we will be choosing the three nearest neighbors to the test point which will be Observation 5, 6 and 4 which are two Red and One is Green. So, prediction will be K = 3 which is Red.

d) If the Bayes decision boundary is highly nonlinear, it is better to choose a smaller value for K. A smaller K value allows the model to capture more intricate and local patterns which is important in situations with nonlinear decision boundaries. Hence, a smaller K value is expected to be best.

Chapter 3

Exercise 1

The equivalent p-values for “TV” and “radio” are quite significant but not for “newspaper.” So, we reject H_0 for “TV” and “radio,” indicating that their advertising expenditures have a major impact on revenue. We do not reject H_0 for “newspaper” indicating that there is no statistically significant effect of newspaper advertising budget on sales. Based on this it may be concluded that newspaper advertising budgets have no significant influence on sales.

Exercise 3

For the college graduate students, the regression equation can be represented as: y=50+20*GPA+0.07*IQ+0.01*GPA*IQ In the same way, high school graduates the regression equation can be represented as: y=85+10*GPA+0.07*IQ+0.01*GPA*IQ

a)

(iii) is correct based on the given equation which suggests that the starting salary for males is higher than for females on average when $50+20GPA+0.07$IQ+0.01GPAIQ. Simplifying this we will get GPA ≥

b) 137.1 thousand dollars. This is the predicted salary for a college graduate with an IQ of 110 and GPA of 4.0

c) False. The presence of a small coefficient for the GPA/IQ interaction term does not imply the absence of an interaction effect. The interaction effect is still evident; however, its magnitude is comparatively weaker when compared to the main effects of GPA and IQ.

Exercise 4

To get to a conclusion on the claim that there is no interaction effect, we need to conduct a hypothesis test for β_4=0 a)Compared to the cubic regression model, the linear regression model is anticipated to have a lower training RSS. This is as a result of the linear regression model's simplicity and decreased propensity to overfit the training set. Due to its increased complexity and propensity to fit the training set of data too closely the cubic regression model may perform worse when applied to fresh data.

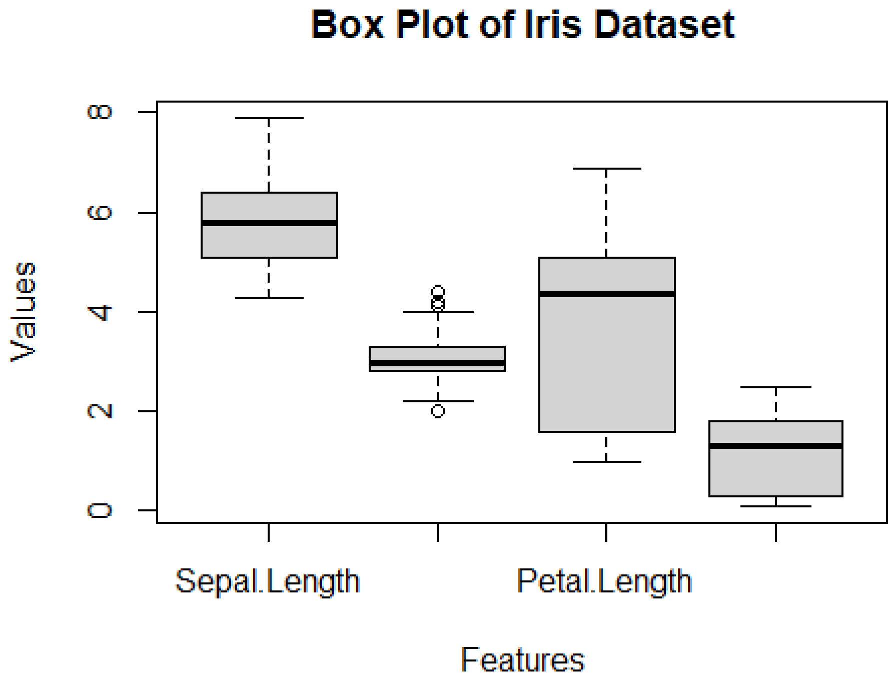
b)Regarding which model will have a lower test RSS we are unable to draw firm conclusions. This is because the training and test sets of data are different and it's probable that the cubic regression model won't overfit the training set as much as we had anticipated. It is also feasible though, that the cubic regression model will still have a higher test RSS than the linear regression model due to its continued overfitting of the training set.

c)Compared to the linear regression model the cubic regression model is anticipated to have a lower training RSS. This is due to the fact that the cubic regression model is a more flexible model that is better able to capture the underlying relationship in cases where the true relationship between X and Y is not linear. Because of its limited flexibility, the linear regression model has a higher propensity to underfit the training set.

d)We are unable to determine with certainty which model will have a lower test RSS because the data is inconclusive. This is due to the fact that the test and training sets of data differ, and it's probable that the cubic regression model won't overfit the training set to the desired extent. The cubic regression model may, nevertheless, continue to overfit the training set and have a higher test RSS than the linear regression model.

```
#2 Practicum Problems
#2.1 Problem 1
library(datasets)
library(ggplot2)
data_frame <- data.frame(iris)
head(data_frame)

##   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
```

```
boxplot(data_frame[,1:4], data = data_frame, main = "Box Plot of Iris
Dataset", xlab = "Features", ylab = "Values")
```



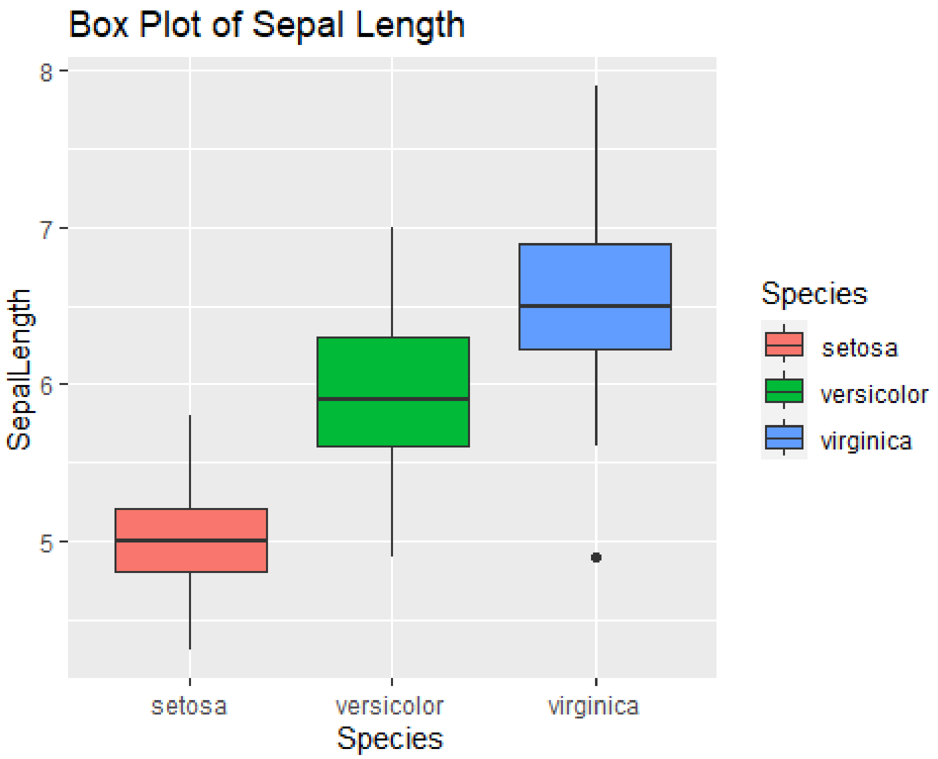
```
SepalLength <- data_frame$Sepal.Length
cat("Sepal Length's IQR :", IQR(SepalLength))

## Sepal Length's IQR : 1.3

SepalWidth <- data_frame$Sepal.Width
cat("Sepal Width's IQR:", IQR(SepalWidth))

## Sepal Width's IQR: 0.5

PetalLength <- data_frame$Petal.Length
cat("Petal Length's IQR:", IQR(PetalLength))

## Petal Length's IQR: 3.5

PetalWidth <- data_frame$Petal.Width
cat("Petal Width's IQR:",IQR(PetalWidth))

## Petal Width's IQR: 1.5
```

The Petal Length has the highest IQR among all which is 3.5

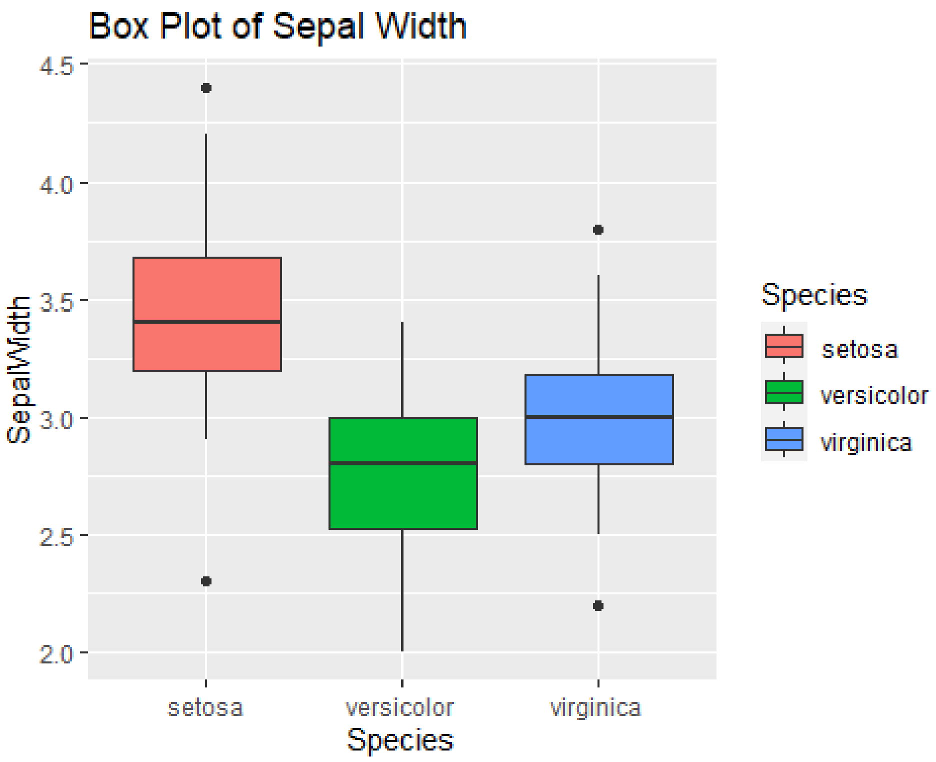
```
SepalLength <- data_frame$Sepal.Length
cat("Sepal Length's Standard Deviation :", sd(SepalLength))

## Sepal Length's Standard Deviation : 0.8280661

SepalWidth <- data_frame$Sepal.Width
cat("Sepal Width's Standard Deviation:", sd(SepalWidth))

## Sepal Width's Standard Deviation: 0.4358663

PetalLength <- data_frame$Petal.Length
cat("Petal Length's Standard Deviation:", sd(PetalLength))

## Petal Length's Standard Deviation: 1.765298

PetalWidth <- data_frame$Petal.Width
cat("Petal Width's Standard Deviation:",sd(PetalWidth))

## Petal Width's Standard Deviation: 0.7622377
```

The Petal Length has the highest Standard Deviation among all which is 1.76

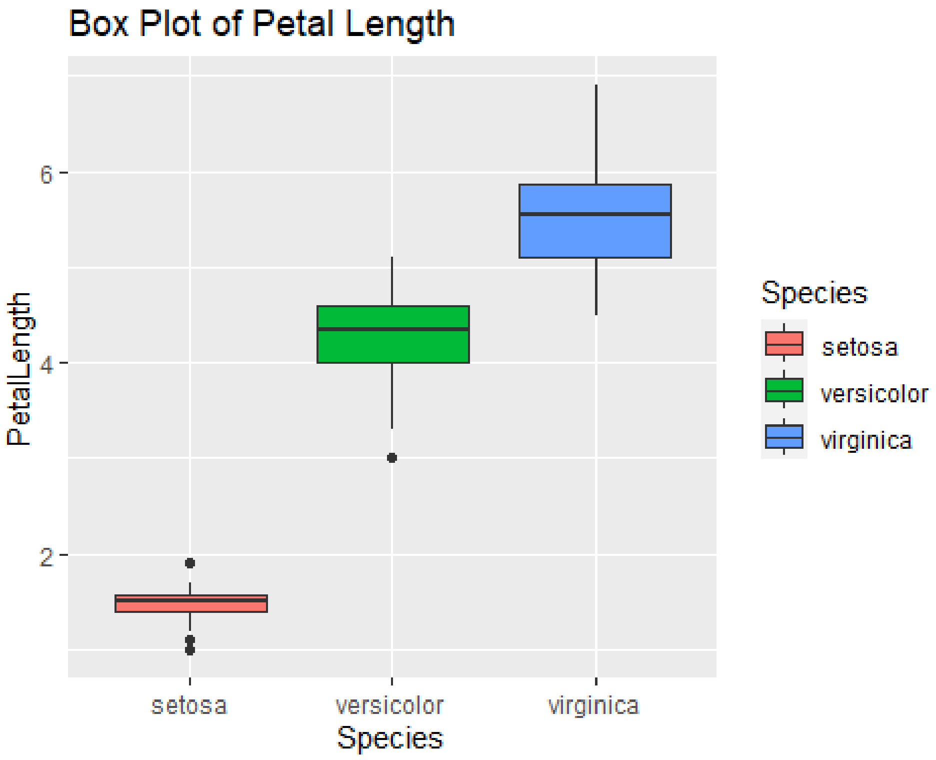
```
ggplot(data = data_frame, aes(x = Species, y = SepalLength, fill = Species))
+ geom_boxplot() + ggtitle("Box Plot of Sepal Length")
```



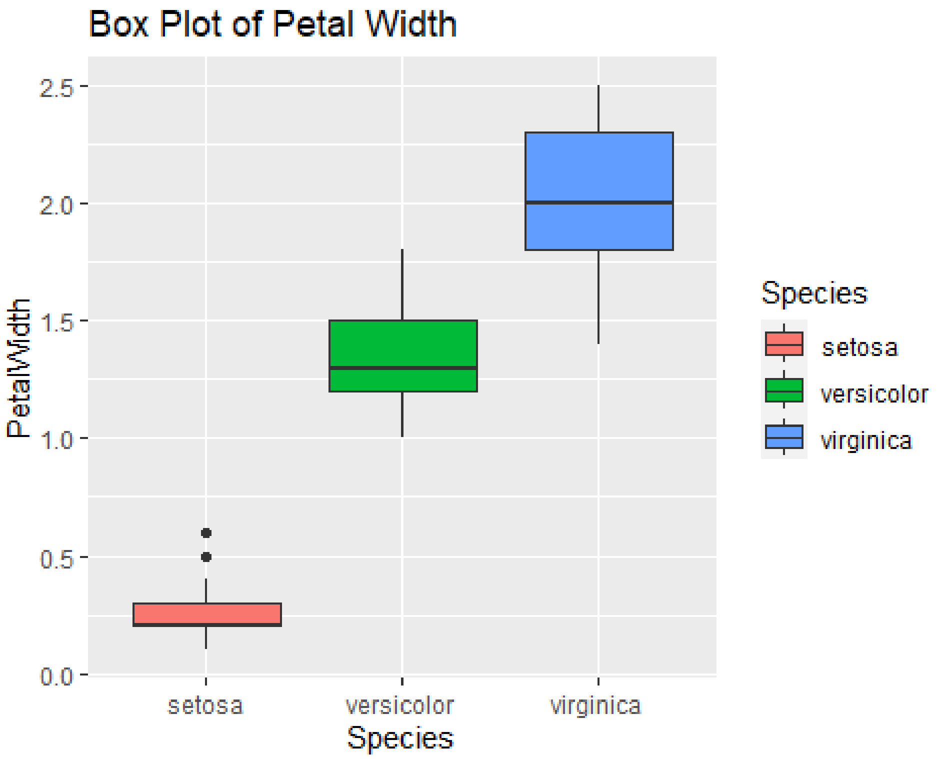
```
ggplot(data = data_frame, aes(x = Species, y = SepalWidth, fill = Species)) +
geom_boxplot() + ggtitle("Box Plot of Sepal Width")
```



```
ggplot(data = data_frame, aes(x = Species, y = PetalLength, fill = Species))
+ geom_boxplot() + ggtitle("Box Plot of Petal Length")
```



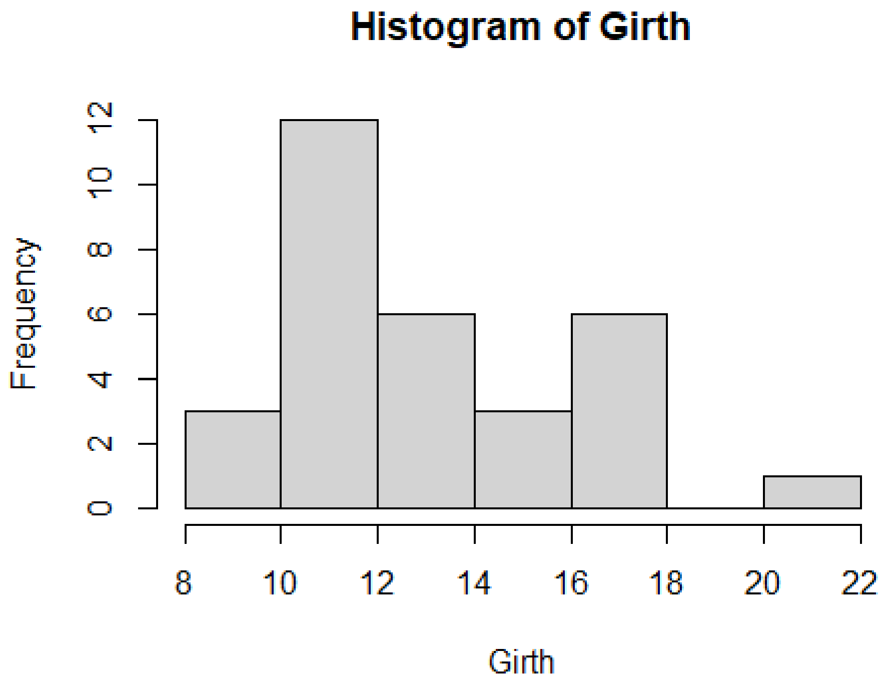
```r
ggplot(data = data_frame, aes(x = Species, y = PetalWidth, fill = Species)) +
geom_boxplot() + ggtitle("Box Plot of Petal Width")
```



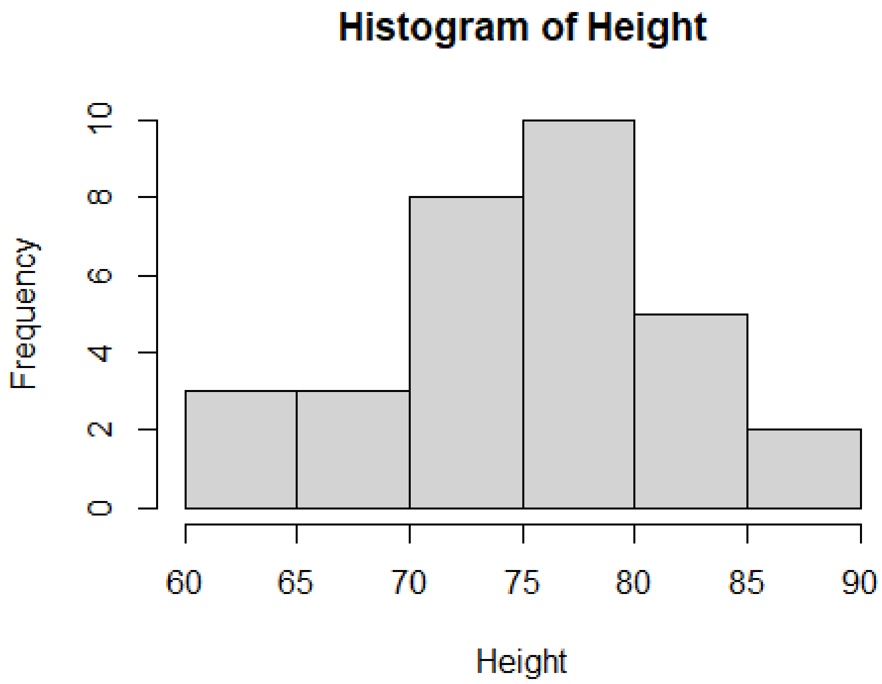
Based on the information, it is evident that the Setosa species stands apart from the other two species. So, it can be concluded that Setosa flowers have distinctly different characteristics particularly in the terms of Petal Length, Petal Width compared to other species.

```r
#2.2 Problem 2
library(moments)
dataframe_trees <- data.frame(trees)
head(trees)

##   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

Girth <- trees$Girth
Height <- trees$Height
Volume <- trees$Volume

summary(Girth)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    8.30   11.05   12.90   13.25   15.25   20.60
```

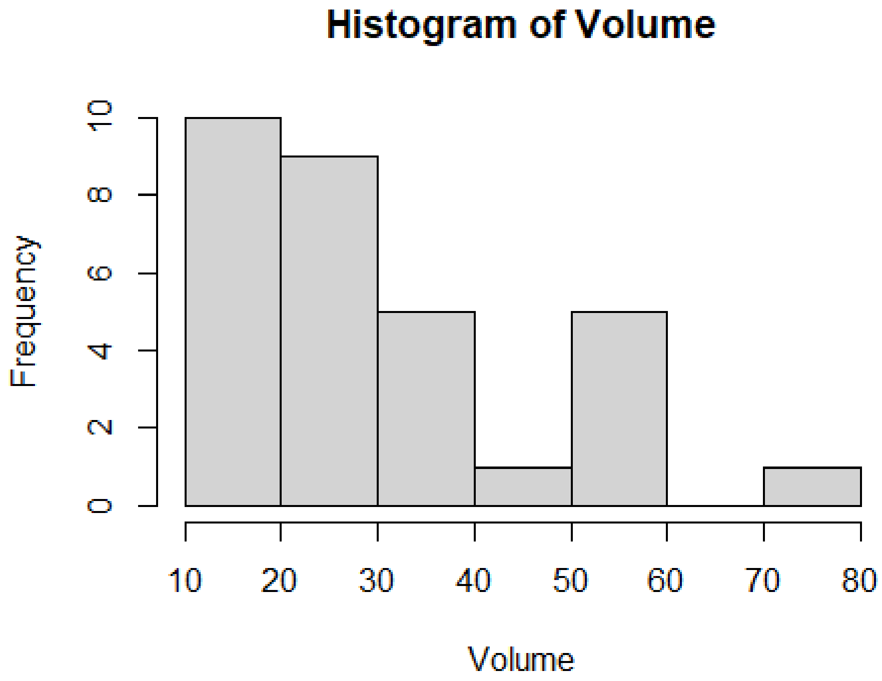
```
summary(Height)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##      63      72      76      76      80      87

summary(Volume)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   10.20   19.40   24.20   30.17   37.30   77.00

fivenum(Girth)

## [1]  8.30 11.05 12.90 15.25 20.60

fivenum(Height)

## [1] 63 72 76 80 87

fivenum(Volume)

## [1] 10.2 19.4 24.2 37.3 77.0

hist(Girth, main = "Histogram of Girth", xlab = "Girth")
```



```
hist(Height, main = "Histogram of Height", xlab = "Height")
```



```
hist(Volume, main = "Histogram of Volume", xlab = "Volume")
```



```
cat("Girth's Skewness:", skewness(Girth))
```

```
## Girth's Skewness: 0.5263163

cat("Height's Skewness:", skewness(Height))

## Height's Skewness: -0.374869

cat("Volume's Skewness:", skewness(Volume))

## Volume's Skewness: 1.064357
```

The height feature’s skewness appears to follow a normal distribution, as evident from the histograms and the skewness function result, which indicates a value of -0.37, the closest to zero among the features. This observation aligns with the histogram findings and validates the assertion. Also, it can be noted that the skewness values for the other two features, namely Girth and Volume, are positive, whereas Height exhibits a negative skewness.

```
#2.3 Problem 3
dataset_url <- "https://archive.ics.uci.edu/ml/machine-learning-
databases/auto-mpg/auto-mpg.data"
column_names <- c("mpg", "cylinders", "displacement", "horsepower", "weight",
"acceleration",
                 "model_year", "origin", "car_name")
auto_mpg_dataframe <- read.table(dataset_url, header = FALSE, col.names =
column_names, as.is = TRUE, stringsAsFactors = F, sep = "")

head(auto_mpg_dataframe)

##   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

auto_mpg_dataframe$horsepower <- as.numeric(auto_mpg_dataframe$horsepower)

## Warning: NAs introduced by coercion
```

```
is.numeric(auto_mpg_dataframe$horsepower)

## [1] TRUE

mean_before_replacement = mean(auto_mpg_dataframe$horsepower, na.rm = TRUE)
mean_before_replacement

## [1] 104.4694

median_before_replacement <- median(auto_mpg_dataframe$horsepower, na.rm =
TRUE)
median_before_replacement

## [1] 93.5

auto_mpg_dataframe$horsepower[is.na(auto_mpg_dataframe$horsepower)]  <-
median_before_replacement

mean_after_replacement = mean(auto_mpg_dataframe$horsepower, na.rm = TRUE)
mean_after_replacement

## [1] 104.304

median_after_replacement <- median(auto_mpg_dataframe$horsepower, na.rm =
TRUE)
median_after_replacement

## [1] 93.5
```

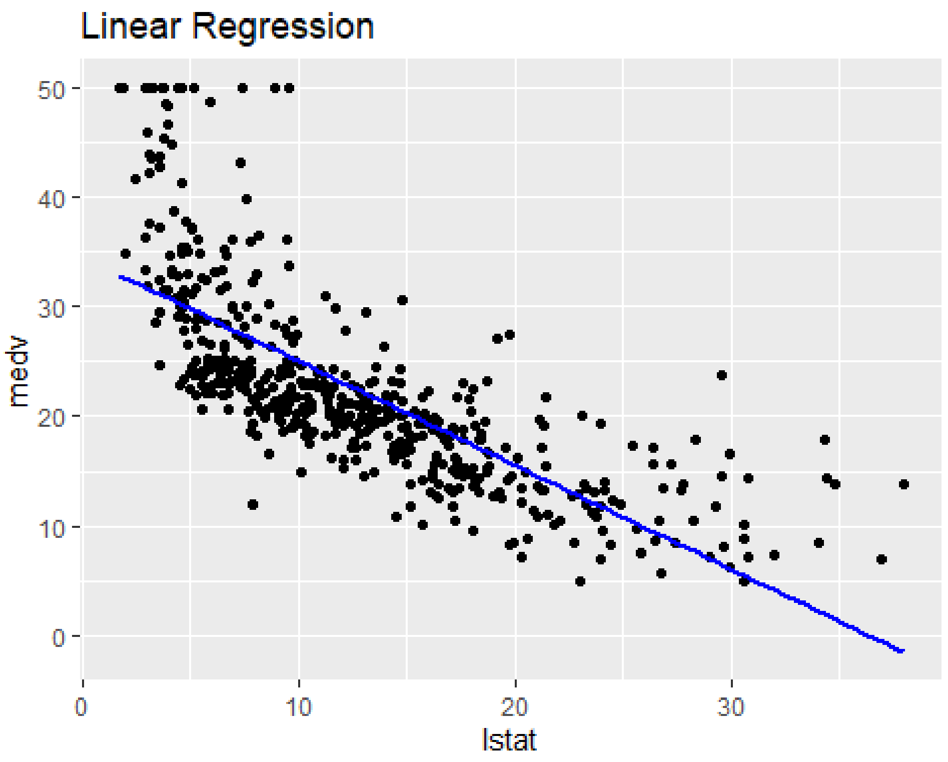
We can see above there is no much change in the mean after the median has been replaced in the place of NA's , the value of mean before replacing with median was 104.4694 after replacing was 104.304

```
#2.4 Problem 4
library(MASS)
data_boston <- data.frame(Boston)
head(data_boston)

##      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
```

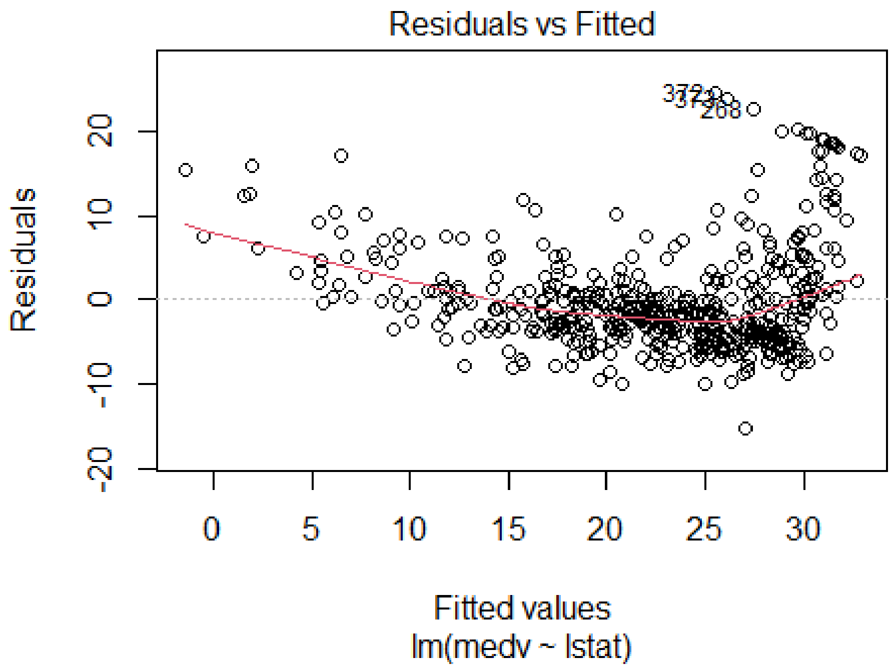
```
model <- lm(medv ~ lstat, data = data_boston)
summary(model)
```

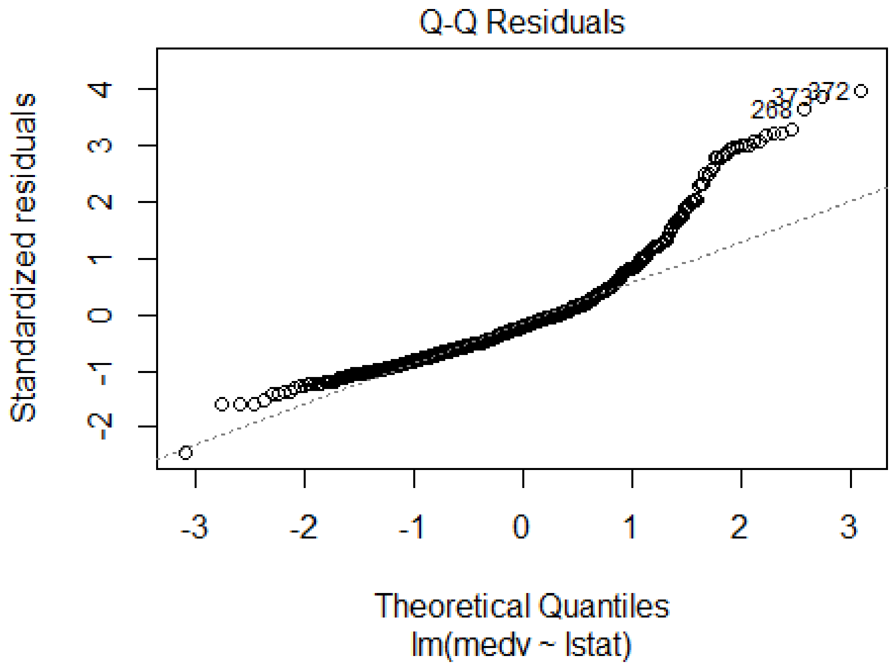
```
## 
## Call:
## lm(formula = medv ~ lstat, data = data_boston)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -15.168  -3.990  -1.318   2.034  24.500 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## 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
```

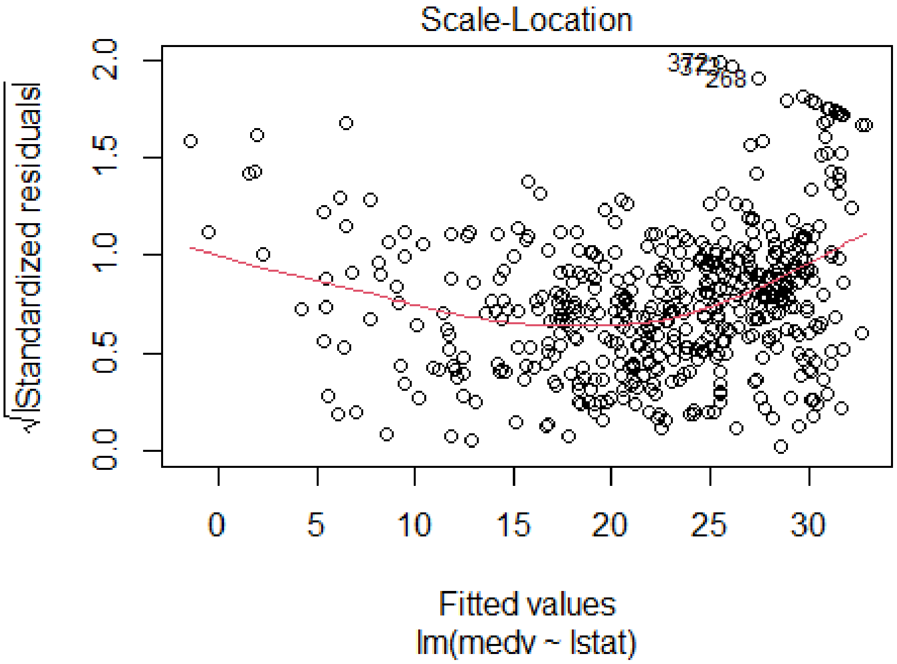
```
ggplot(data_boston, aes(x = lstat, y = medv)) + geom_point() +
geom_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "blue") +
labs(title = "Linear Regression", x = "lstat", y = "medv")
```

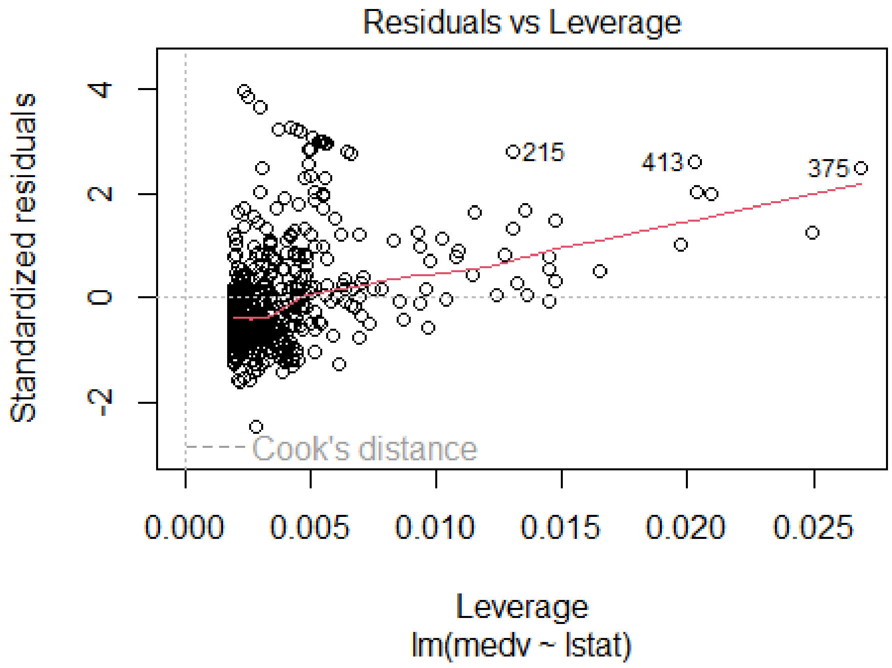


```r
plot(model)
```


Residuals vs Fitted
Residuals
-20
-10
0
10
20
0
5
10
15
20
25
30
372
373
268
Fitted values
lm(medv ~ lstat)


Q-Q Residuals
Standardized residuals
-2
-1
0
1
2
3
4
-3
-2
-1
0
1
2
3
372
373
268
Theoretical Quantiles
lm(medv ~ lstat)





```r
test <- data.frame(lstat = c(5, 10, 15))
predict(model, test, 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
```

```r
predict(model, test, 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
```

```r
nonlinear_model <- lm(medv ~ lstat + I(lstat^2), data = data_boston)
summary(nonlinear_model)
```

```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2), data = data_boston)
##
## 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 ***
## lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(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
```

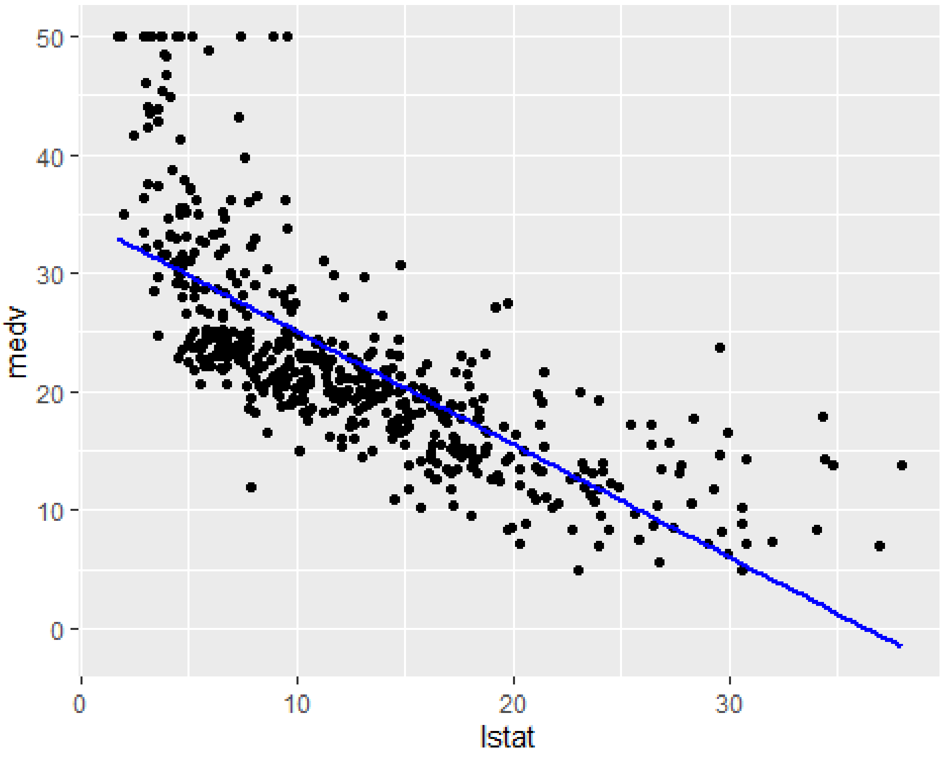
```r
r_squared_linear <- summary(model)$r.squared
r_squared_nonlinear <- summary(nonlinear_model)$r.squared

cat("R-squared for Linear Model:", r_squared_linear, "\n")
```

```
## R-squared for Linear Model: 0.5441463
```

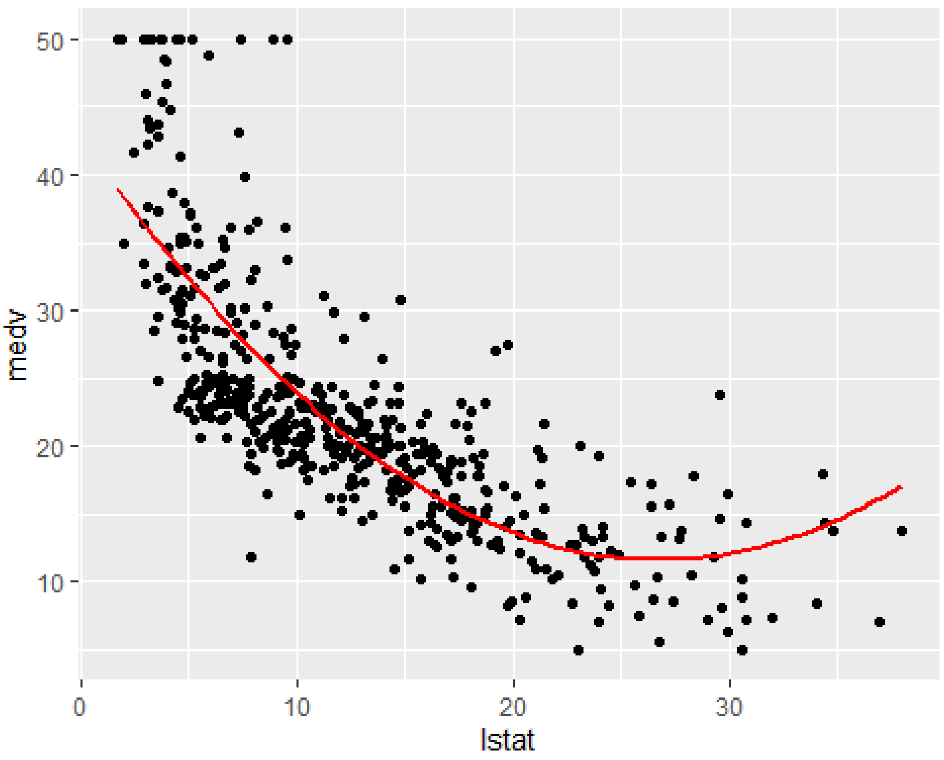
```r
cat("R-squared for Non-Linear Model:", r_squared_nonlinear, "\n")
```

```
## R-squared for Non-Linear Model: 0.6407169
```

```r
ggplot(data_boston, aes(x = lstat, y = medv)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x, se = FALSE, color = "blue") +
  labs(x = "lstat", y = "medv")
```



```r
ggplot(data_boston, aes(x = lstat, y = medv)) +
  geom_point() +
  stat_smooth(method = "lm", formula = y ~ x + I(x^2), se = FALSE, color =
"red") +
  labs(x = "lstat", y = "medv")
```


50
40
30
20
10
medv
0
10
20
30
lstat