Shiva Sankar Modala | A20517528 | CSP 571 Spring 2023

Data Preparation and Analysis Assignment 1

Recitation Exercises:

Chapter 2

Exercise 1 Answers:

a. **The sample size n is extremely large, and the number of predictors p is small.**

- Because of the large sample size, inflexible model will try to fit every data point, thus resulting in overfitting. However, **Flexible statistical learning model will** fit the data more closely and thus will **perform better** in this case.

b. **The number of predictors p is extremely large, and the number of observations n is small.**

- Since the sample size is already small in this situation, a flexible model will try to fit each data point, thus leading to overfitting. Hence **Inflexible model will perform better** in this case.

c. **The relationship between the predictors and response is highly non-linear.**

- In this case because of non-linearity, we have more degrees of freedom, thus **Flexible model will perform better.**

d. **The variance of the error terms, i.e., σ2 = Var(€), is extremely high.**

- High variance leads to overfitting and using a flexible model will further try to fit the noise in the error terms and would further increase the variance. Thus, **Inflexible model will perform better** here.

Exercise 2 Answers:

a. It is a **Regression** problem. Because CEO Salary is continuous.

We are interested in the **Inference**. Because we are interested in finding the factors that affect our target variable (CEO Salary)

N = 500, p = 3.

b. It is a **Classification** problem**.** Because the target variable (Result) is discrete i.e., either Success or Failure.

We are interested in the **Prediction.** Because we are only interested in the results (Success/Failure) and not concerned on how each feature contributes in getting a particular result

N = 20, p = 13

c. It is a **Regression** problem. Because % change in USD/Euro is continuous.

We are interested in the **Prediction.** Because we are only interested in the % change (a particular value) and not concerned on how each feature contributes to that % change.

N = 52 (52 weeks in 2012), p = 3

Exercise 4 Answers:

a. 3 Real-life applications of classification:

- Consider student's result data having variables like: result in sem1, result in sem2 and so on till 7th sem. The objective is to find out based on the students previous scores, what would be the result of a student in the final exam. This is an example of prediction as we are only concerned with the result.
- When releasing a new product and attempting to predict whether it will succeed or fail by compiling information on 50 previously released, comparable items. We have noted the price charged for the product, the marketing budget, the price of the competitors, and eleven additional characteristics for each product, including whether it was successful or unsuccessful. This is an example based on inference.
- Which university to pick. Predictors: University Rankings, Program Ranking, Location, retention rate, average salary, cost of attendance, scholarships. Response: List of universities. This is an example of Prediction.

b. 3 Real-life applications of regression:
- Percentage (%) of precipitation. Predictors: Temperature, Air Pressure, Moisture, Humidity, etc. Response: Percentage of rainfall (Precipitation). Goal: Inference
- Mileage of a vehicle. Predictors: Weight, Number of cylinders, Engine type, Fuel type, Power, etc.
Based on these parameters we get a numeric value for mileage. Response: Mileage. Goal: Inference.
- Per capita income of an economy. Predictors: GDP, Population, Literacy Rate, Tax Revenue, Inflation rate. Response: Per capita income. Goal: Inference

c. 3 Real-life applications of cluster-analysis:
- Recognizing communities within large groups of people. Predictors: Country, 1st Language, 2nd Language, economy.
- Cluster analysis can be used to identify areas where there are greater incidences of types of crime. By identifying these distinct areas, hot spots can be created based on where a similar crime has happened in the past
- Cluster analysis can be used to identify customer buying patterns. Frequent items bought together can be identified. Predictors: Price, Quantity, Product type.
- Cluster analysis can be used to pinpoint regions with higher rates of specific forms of accidents prone zones. Cluster analysis can be used to determine accidents based on the cluster analysis.

Exercise 6 Answers:

Parametric Learning	**Non-Parametric Learning**
Reduces the problem of estimating *f* down to one of estimating set of parameters because it assumes a form for *f.*	Does not assume a particular form of *f*, so requires a very large sample to accurately estimate *f.*
$y_i=\beta_0+\beta_1 x_i+e_i$	$y_i=f(x_i)+e_i$
In parametric model, the model to fit the data is known in prior.	In non-parametric model, the data tells, what the 'regression' should look like.

Advantage: The main advantage of a parametric model to either a classifier or a regressor is the simplification of model *f* to a few parameters as not many observations are required as compared to the non-parametric model.

Disadvantage: The disadvantages of a parametric model to either a classifier or a regressor are potentially inaccurate estimate *f*, if the form of *f* assumed is wrong or to overfit the observations if more flexible models are used.

Exercise 7 Answers:

a. $Euclidian\ Distance = \sqrt{(x_2 - x_1)^2+(y_2 - y_1)^2+(z_2 - z_1)^2}$

Obs.	***X1***	***X2***	***X3***	***Y***	**Distance**
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. Prediction with *K* = 1:
For K = 1, we choose single nearest point, which in this case is the 5th observation (Distance 1.41). Hence class associated with that observation is Green. Hence for K = 1 our prediction is **Green**.

c. Prediction with *K* = 3:
For K = 3, we choose three nearest points, which in this case are Obs. 5 (Distance 1.41), Obs.6 (Distance 1.73) and Obs. 2 (Distance 2). Majority of these observation is class Red. Hence class associated with that observation is Red. Hence for K = 3 our prediction is **Red**.

d. If the Bayes decision boundary in this problem is highly nonlinear

The value of K will be **small** for Bayes decision boundaries that are significantly non-linear.

Chapter 3

Exercise 1 Answers:

If p-values < 0.05 then reject the null hypothesis.
In this case, we reject the null hypothesis for TV and Radio but not for Newspaper because the associated p-values are extremely significant for "TV" and "radio" but not for "newspaper". We can infer that newspaper advertising budget has little impact on sales.

Exercise 3 Answers:

a. High School: $Y = 50 + 20(GPA) + 0.07(IQ) + 35(0) + 0.01(GPA \times IQ) - 10(GPA \times 0)$
Y= 50+20(GPA)+0.07(IQ)+0.01(GPA × IQ)
College Graduates: $Y = 50 + 20(GPA) + 0.07(IQ) + 35(1) + 0.01(GPA \times IQ) - 10(GPA \times 1)$
Y= 85+10(GPA)+0.07(IQ)+0.01(GPA × IQ)
By subtracting these two equations we get **Y = 35-10(GPA)**
On solving above two equations we GPA = 3.5,
Hence, for a fixed value of IQ and GPA, high school graduates earn more on average than college graduates. **Hence answer is (iii**)

b. College graduate: Y = 85+10(4.0) + 0.07(110) + 0.01(4.0 × 110) = 137.1
We got 137.1. Hence the salary is **$ 137,100**

c. We must test the null hypothesis H0: = 0 and use the p-value associated with the t-statistic to determine whether the GPA/IQ has an effect on the quality of the model. **Hence False.**

Exercise 4 Answers:

a. Since X and Y have a linear connection, it is easy to assume that the cubic regression line and the least square fitted line are both close to the true regression line. Since RSS is the residuals of squared sum, it is reasonable to predict that the liner's RSS will be smaller when compared to the cubic regression line. As there would be little separation between the fitted and actual lines, RSS would likewise be small. Cubic regression's RSS would be quite high.

b. Again, no test data is given. We must employ lateral thinking and deal with preconceptions. For cubic regression, the training RSS would be larger as well, therefore the test RSS would be higher. (As implied by point a) Given that the relationship between X and Y is linear, linear regression still stands to benefit by having a lower test RSS.

c. Because of the nonlinear nature of the relationship, a regression line that closely resembles the distributed points in the 2D space would have a lower RSS. Since cubic regression would more closely track the points, its RSS would be lower than that of linear regression.

d. Since "how far" is not specified, we would be unable to respond to this with the given information at hand. When compared to cubic regression, it could happen that linear regression is close to points. The RSS linear would be low in this scenario.

2. Practicum Problems

(I'm attaching the pdf markdown files I created from r script.)

Problem 1

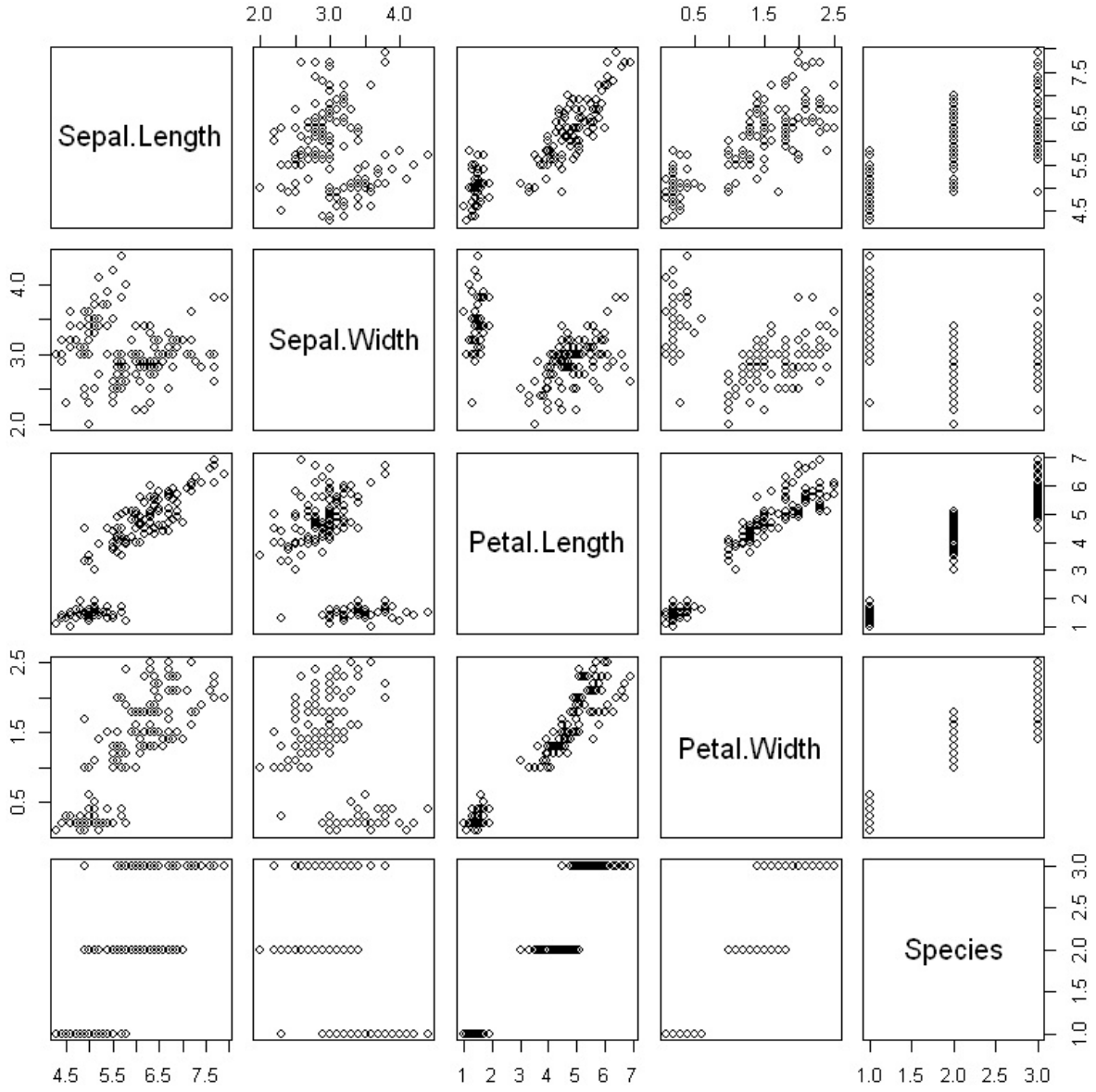
In [82]:
```
## I'm including the results and explanation as the comments
library(datasets)
iris_data = data.frame(iris) # Loading the Iris Data set
head(iris_data) # First six values in the Iris Data set
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa

In [83]:
```
str(iris)
```

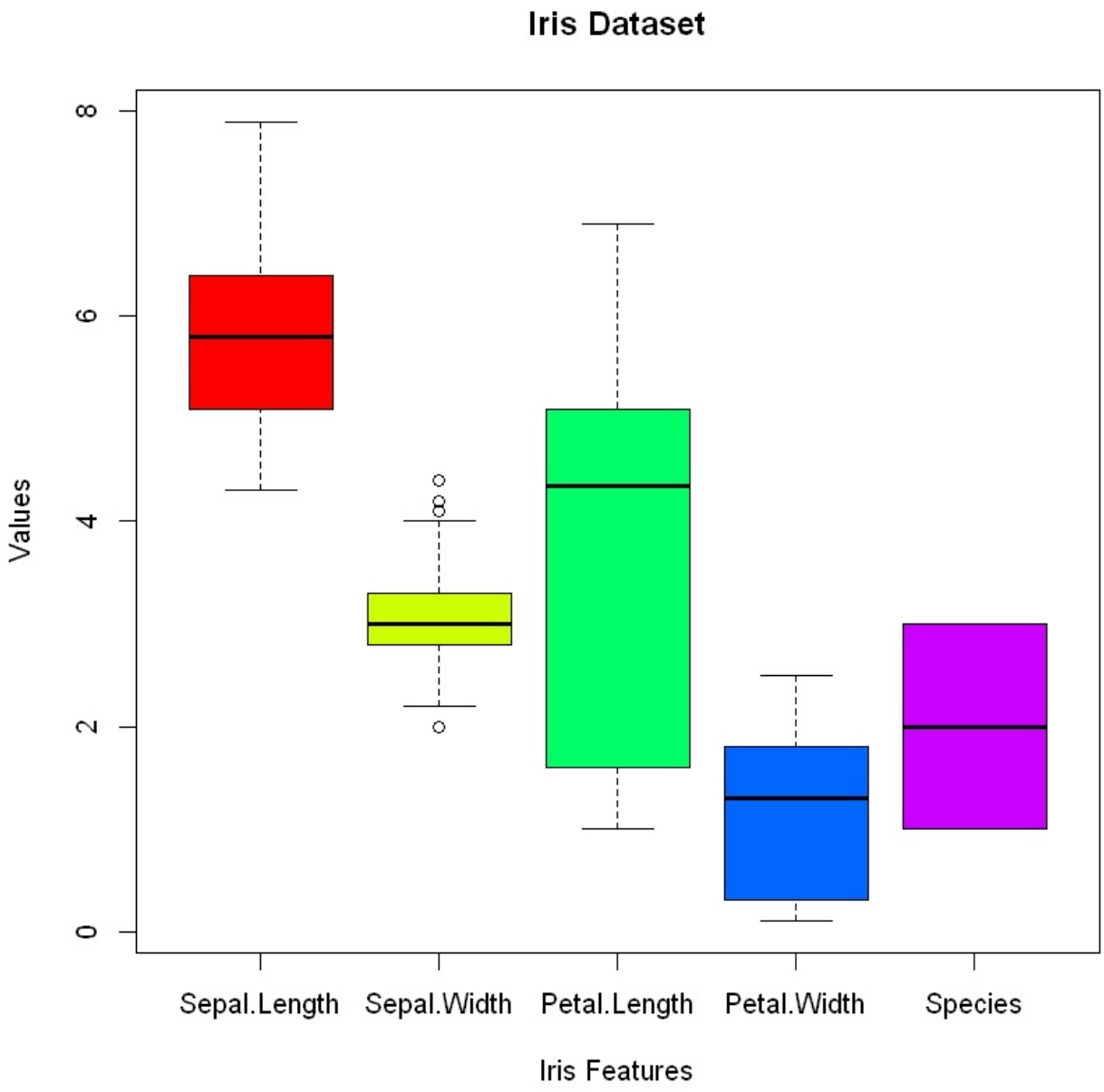
```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

In [84]:
```
pairs(iris)
```



In [85]:
```
# Box plot of each of the 4 features
boxplot(iris_data,
main = "Iris Dataset",
```

```
col = rainbow(5),
xlab = "Iris Features",
ylab = "Values")
```



In [86]:
```
IQR(iris_data$Sepal.Length)
```

1.3

In [87]:
```
IQR(iris_data$Sepal.Width)
```

0.5

In [88]:
```
IQR(iris_data$Petal.Length)
```

3.5

In [89]:
```
IQR(iris_data$Petal.Width)
```

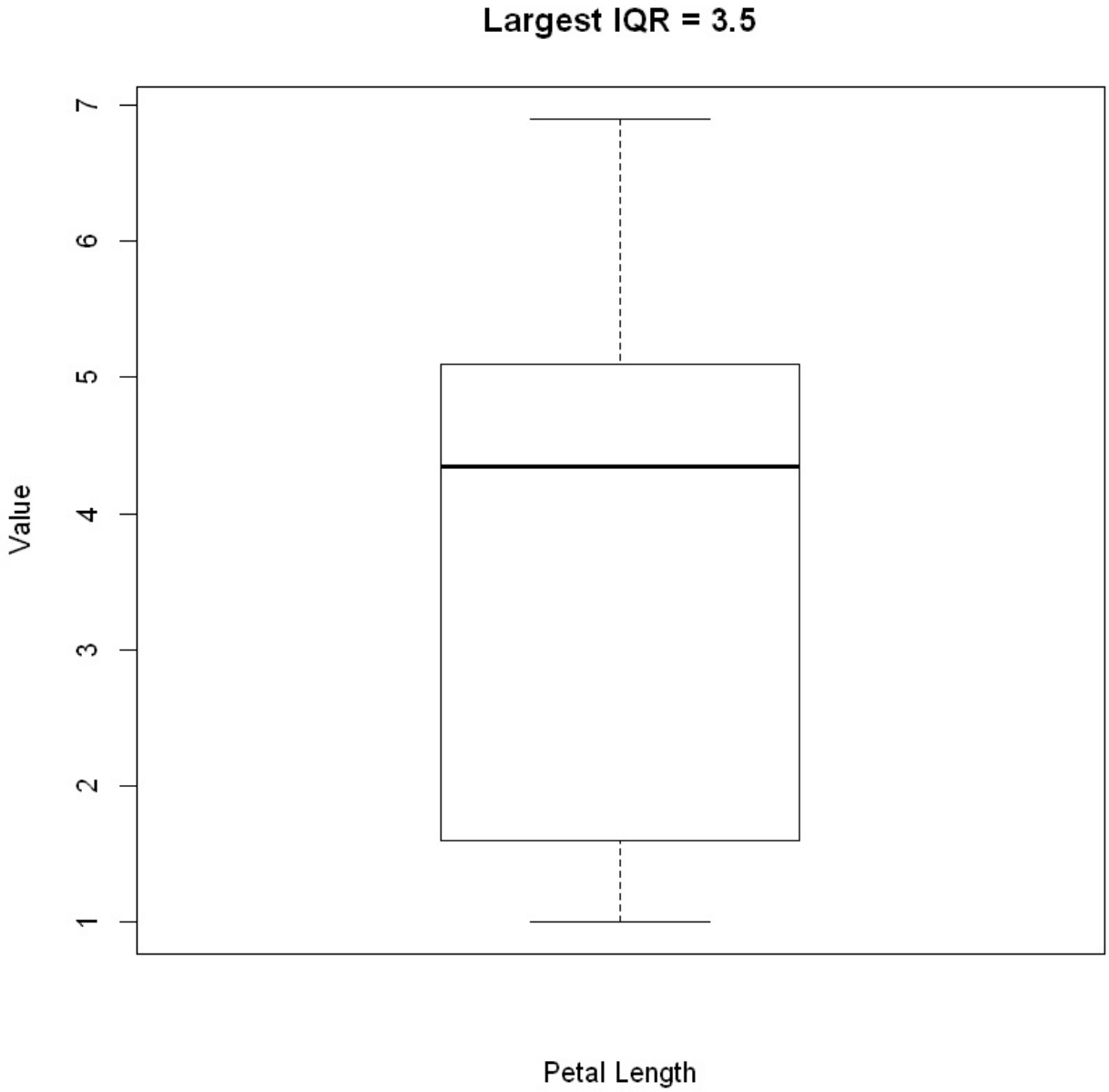
1.5

In [90]:
```
cat("Petal Length has the highest empiral IQR at 3.5.")
```

```
Petal Length has the highest empiral IQR at 3.5.
```

In [91]:
```
# Box plot of the petal length
boxplot(iris_data$Petal.Length, main="Largest IQR = 3.5", xlab="Petal Length",ylab ="Value" )
# Standard Deviation of each features
cat("Standard Deviation of Sepal Length: ",sd(iris_data$Sepal.Length))
```

```
Standard Deviation of Sepal Length:  0.8280661
```



In [92]:
```
cat("Standard Deviation of Sepal Width: ",sd(iris_data$Sepal.Width))
```

```
Standard Deviation of Sepal Width:  0.4358663
```

In [93]:
```
cat("Standard Deviation of Petal Length: ",sd(iris_data$Petal.Length))
```

```
Standard Deviation of Petal Length:  1.765298
```

In [94]:
```
cat("Standard Deviation of Petal Width: ",sd(iris_data$Petal.Width))
```

```
Standard Deviation of Petal Width:  0.7622377
```

In [95]:
```
# Mean of each features
cat("Mean of Sepal Length: ",mean(iris_data$Sepal.Length))
```

```
Mean of Sepal Length:  5.843333
```

In [96]:
```
cat("Mean of Sepal Width: ",mean(iris_data$Sepal.Width))
```

```
Mean of Sepal Width:  3.057333
```

In [97]:
```
cat("Mean of Petal Length: ",mean(iris_data$Petal.Length))
```

```
Mean of Petal Length:  3.758
```

In [98]:
```
cat("Mean of Petal Width: ",mean(iris_data$Petal.Width))
```

```
Mean of Petal Width:  1.199333
```

In [99]:
```
# Median of each features
cat("Median of Sepal Length: ",median(iris_data$Sepal.Length))
```

```
Median of Sepal Length:  5.8
```

In [100…
```
cat("Median of Sepal Width: ",median(iris_data$Sepal.Width))
```

```
Median of Sepal Width:  3
```

In [101…
```
cat("Median of Petal Length: ",median(iris_data$Petal.Length))
```

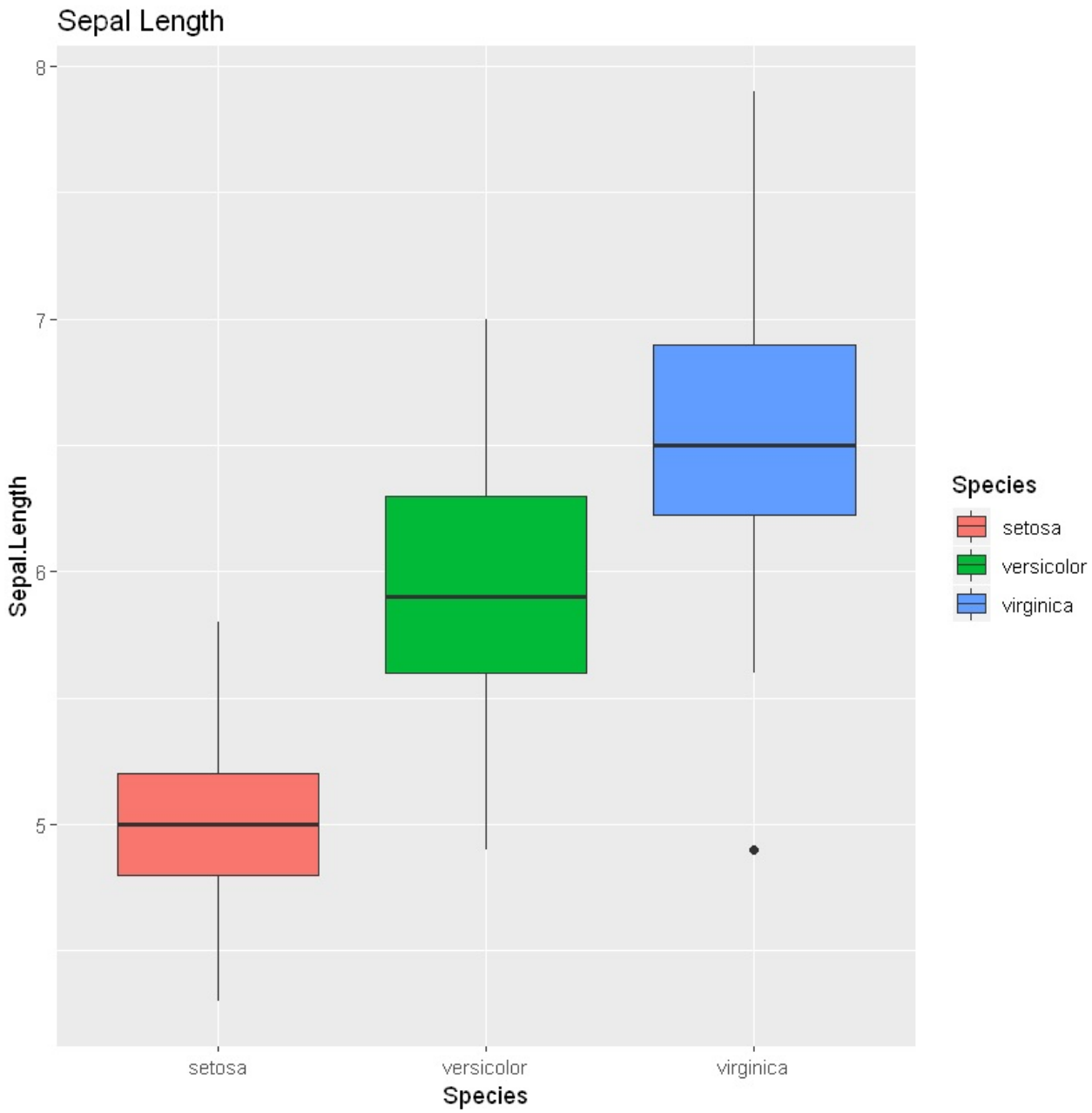
```
Median of Petal Length:  4.35
```

In [102…
```
cat("Median of Petal Width: ",median(iris_data$Petal.Width))
```

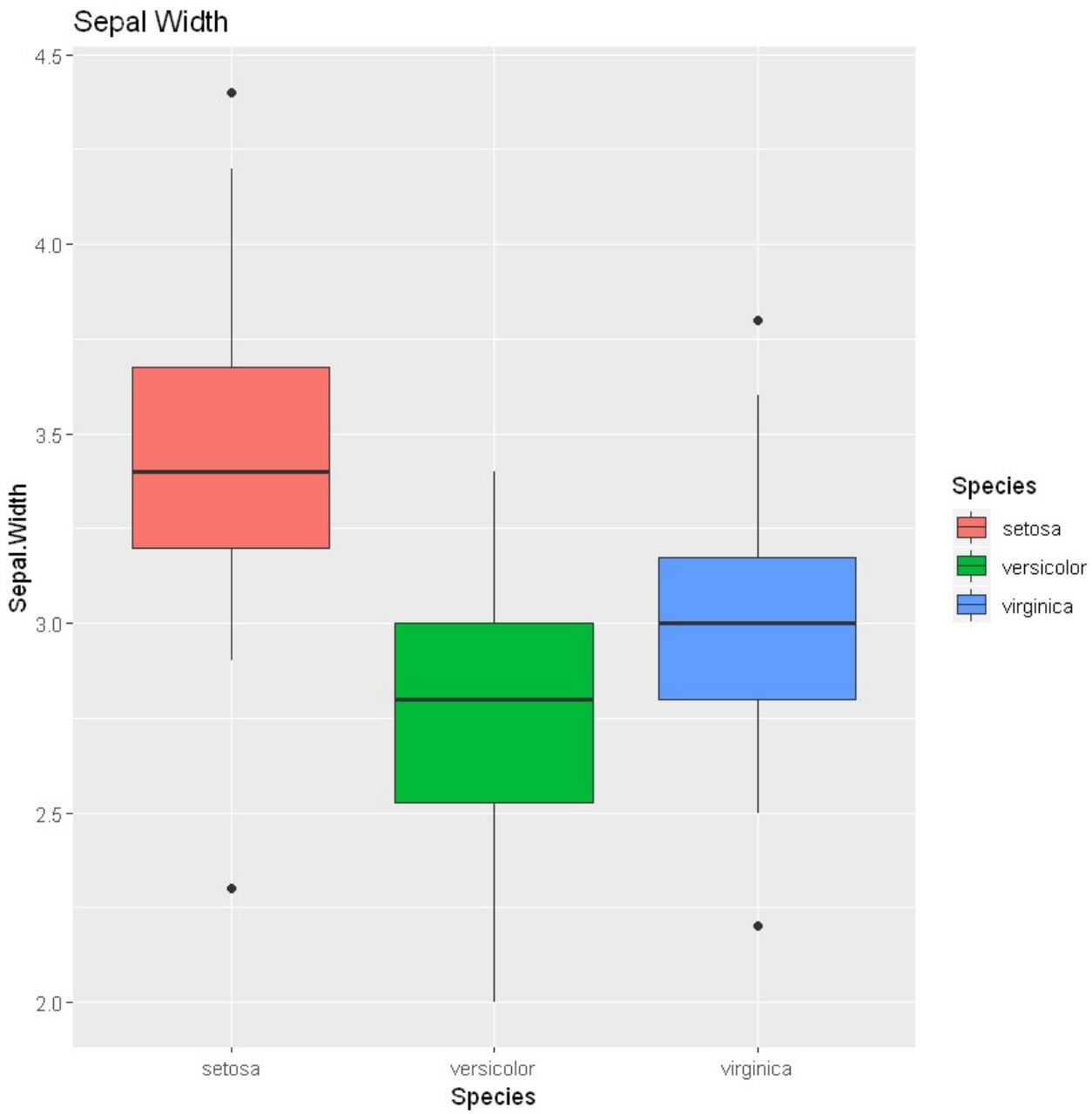
```
Median of Petal Width:  1.3
```

In [103…
```
# Installed ggplot2 from the packages
library(ggplot2) # Loading the ggplot2 library
```

In [104…
```
# Plotting the box whisker per flower species
# Sepal Length
ggplot(data = iris_data,
aes(x = Species,
y = Sepal.Length,
fill = Species)) +
geom_boxplot() + ggtitle("Sepal Length")
```

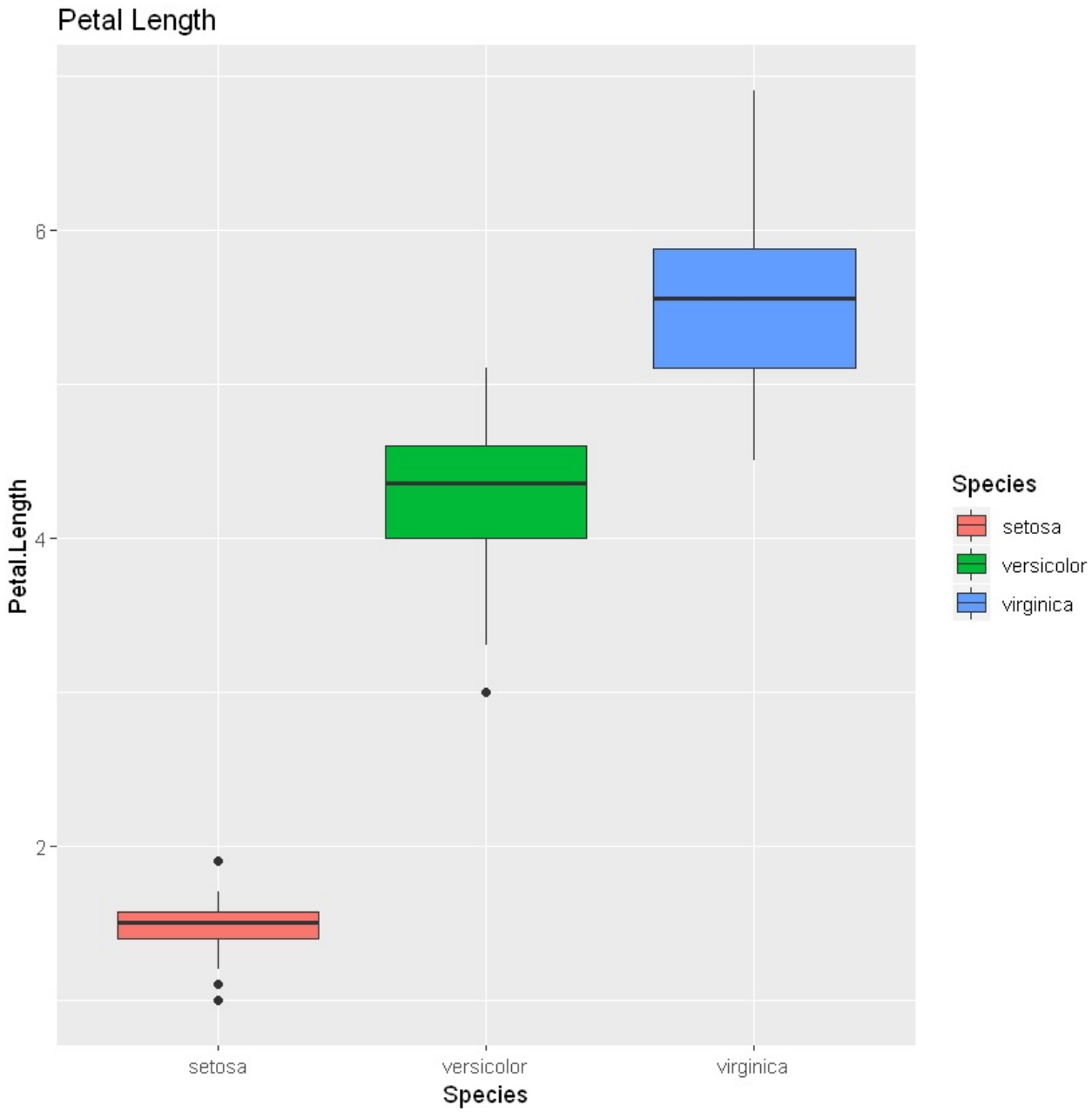


In [105…
```
# Sepal Width
ggplot(data = iris_data,
aes(x = Species,
y = Sepal.Width,
fill = Species)) +
geom_boxplot() + ggtitle("Sepal Width")
```



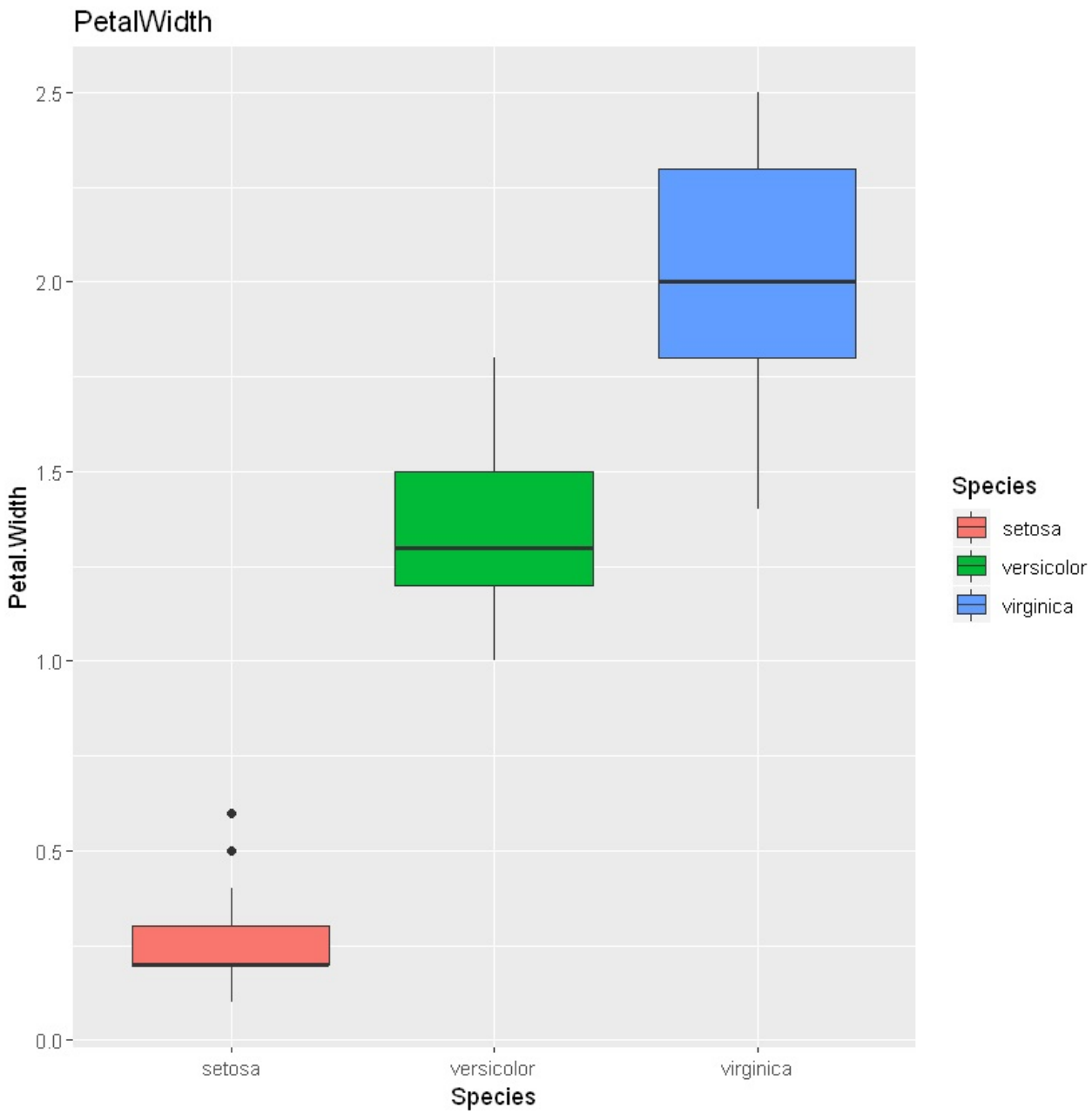
In [106…

```
# Petal Length
ggplot(data = iris_data,
aes(x = Species,
y = Petal.Length,
fill = Species)) +
geom_boxplot() + ggtitle("Petal Length")
```



In [107…

```
# Petal Width
ggplot(data = data_iris,
aes(x = Species,
y = Petal.Width,
fill = Species)) + geom_boxplot() + ggtitle("PetalWidth")
```



In [108…

```
# Results

#Feature with largest empirical IQR = Petallength (3.5)
#Feature with largest Standard Deviation = Petallength (1.765298)
#From the above plots of Petallength and Petalwidth, Setosa Species comes
#out to be different from the other classes.
```

Problem 2

In [9]:
```
## I'm including the results and explanation as the comments
treeData = data.frame(trees) # Loading the tree data frame
# Summary of each tree features
summary(trees$Girth)
```

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

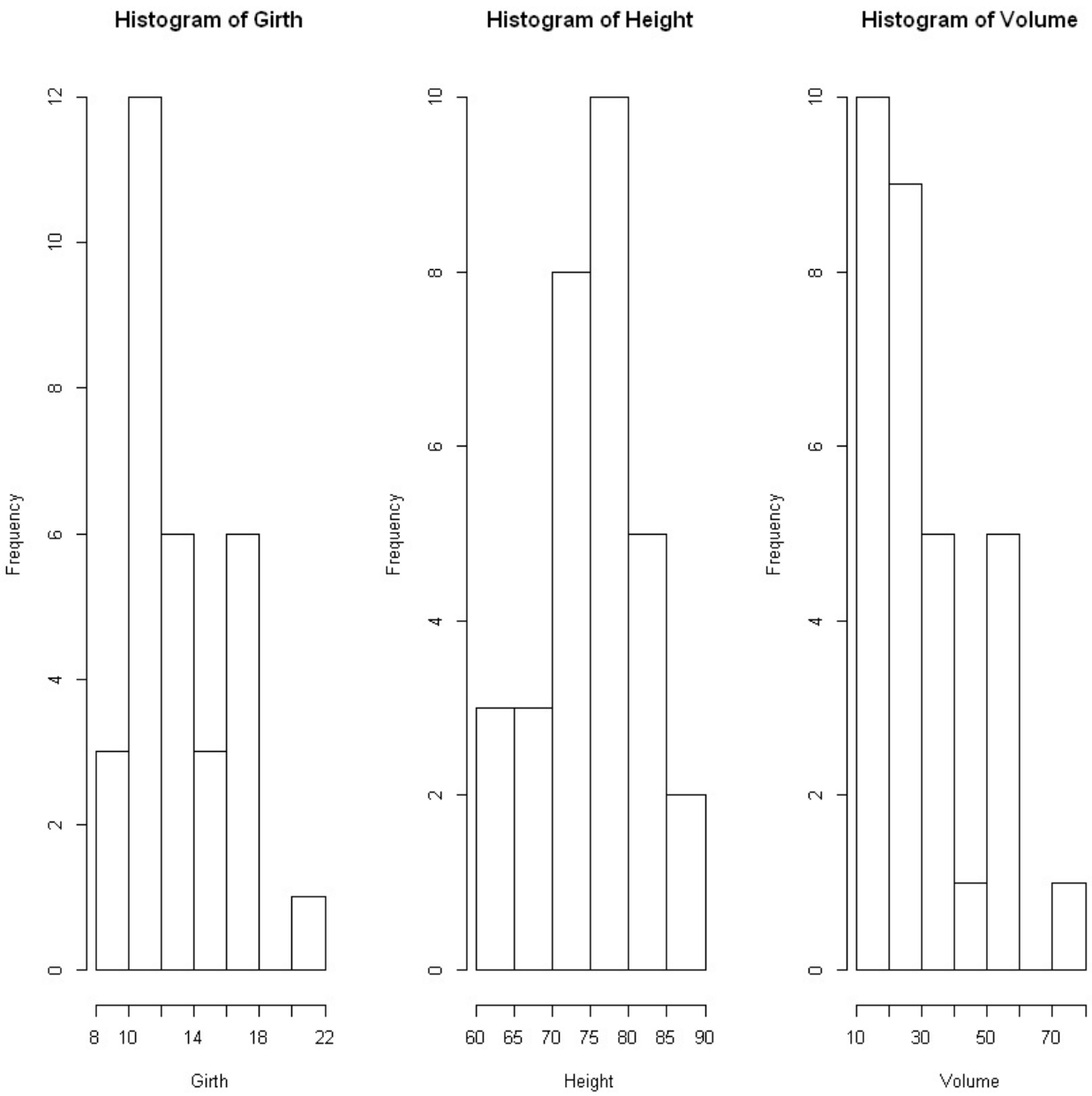
In [10]:
```
summary(trees$Height)
```

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

In [11]:
```
summary(trees$Volume)
```

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

In [12]:
```
# Histogram of each variable
par(mfrow=c(1,3))
hist(trees$Girth,xlab = "Girth", main="Histogram of Girth")
hist(trees$Height,xlab = "Height", main="Histogram of Height")
hist(trees$Volume,xlab = "Volume", main="Histogram of Volume")
```



In [13]:
```
# Installed moments from the packages
library(moments) # Loading the moments
# Skewness of each tree variable
cat("\nSkewness of Girth:",skewness(treeData$Girth))
```

```
Skewness of Girth: 0.5263163
```

In [15]:
```
cat("\nSkewness of Height:",skewness(treeData$Height))
```

```
Skewness of Height: -0.374869
```

In [16]:
```
cat("\nSkewness of Volume:",skewness(treeData$Volume))
```

```
Skewness of Volume: 1.064357
```

In [ ]:
```
## Results
# On visual inspection the Height variable seems to look Normally Distributed.
# Further, after calculating the skewness of each variable below,
# the skewness of Height is closest to 0, which is in line
# with earlier inspection for the height variable.
# Variables Girth and Volume has positive skewness
# while variable Height has negative skewness.
```

Problem 3

In [1]:
```
## I'm including the results and explanation as the comments

# Loading the auto-mpg.data from UCI repository
auto <- read.table("https://archive.ics.uci.edu/ml/machine-learning-databases
/auto-mpg/auto-mpg.data-original")
names(auto) <- c("mpg", "cyl", "disp", "hp", "weight", "acc", "model.yr",
                 "origin", "name")
cat("\nMean before replacment:",mean(auto$hp, na.rm = T))
```

```
Mean before replacment: 105.0825
```

In [2]:
```
# TO replace the null is Horsepower, we used median to fill null values
mpg_median = median(auto$hp, na.rm = T)
auto$hp[is.na(auto$hp)] = mpg_median

cat("\nMean after replacment:",mean(auto$hp, na.rm = T))
```

```
Mean after replacment: 104.9335
```

In [3]:
```
## Results

#After replacing the NA's with median values, we can see that the
#mean has decreased a little bit from 105.0825 to 104.9335
```

Problem 4

In [112…
```
## I'm including the results and explanation as the comments
library(ggplot2)
library (MASS) # Installed the Mass package
library (ISLR) # Installed ISLR for Statistical analysis
names(Boston)
```

1. 'crim'
2. 'zn'
3. 'indus'
4. 'chas'
5. 'nox'
6. 'rm'
7. 'age'
8. 'dis'
9. 'rad'
10. 'tax'
11. 'ptratio'
12. 'black'
13. 'lstat'
14. 'medv'

In [113…
```
# Fitting the linear regression model for the
# Use lm to fit a regression between medv and lstat
lm.fit =lm(medv~lstat ,data=Boston )
attach(Boston)
lm.fit =lm(medv~lstat)
lm.fit
```

```
Call:
lm(formula = medv ~ lstat)

Coefficients:
(Intercept)        lstat
      34.55        -0.95
```

In [114…
```
# Checking the data in linear model
summary(lm.fit)
```

```
Call:
lm(formula = medv ~ lstat)

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
```

In [115…
```
names(lm.fit )
```

1. 'coefficients'
2. 'residuals'
3. 'effects'
4. 'rank'
5. 'fitted.values'
6. 'assign'
7. 'qr'
8. 'df.residual'
9. 'xlevels'
10. 'call'
11. 'terms'
12. 'model'

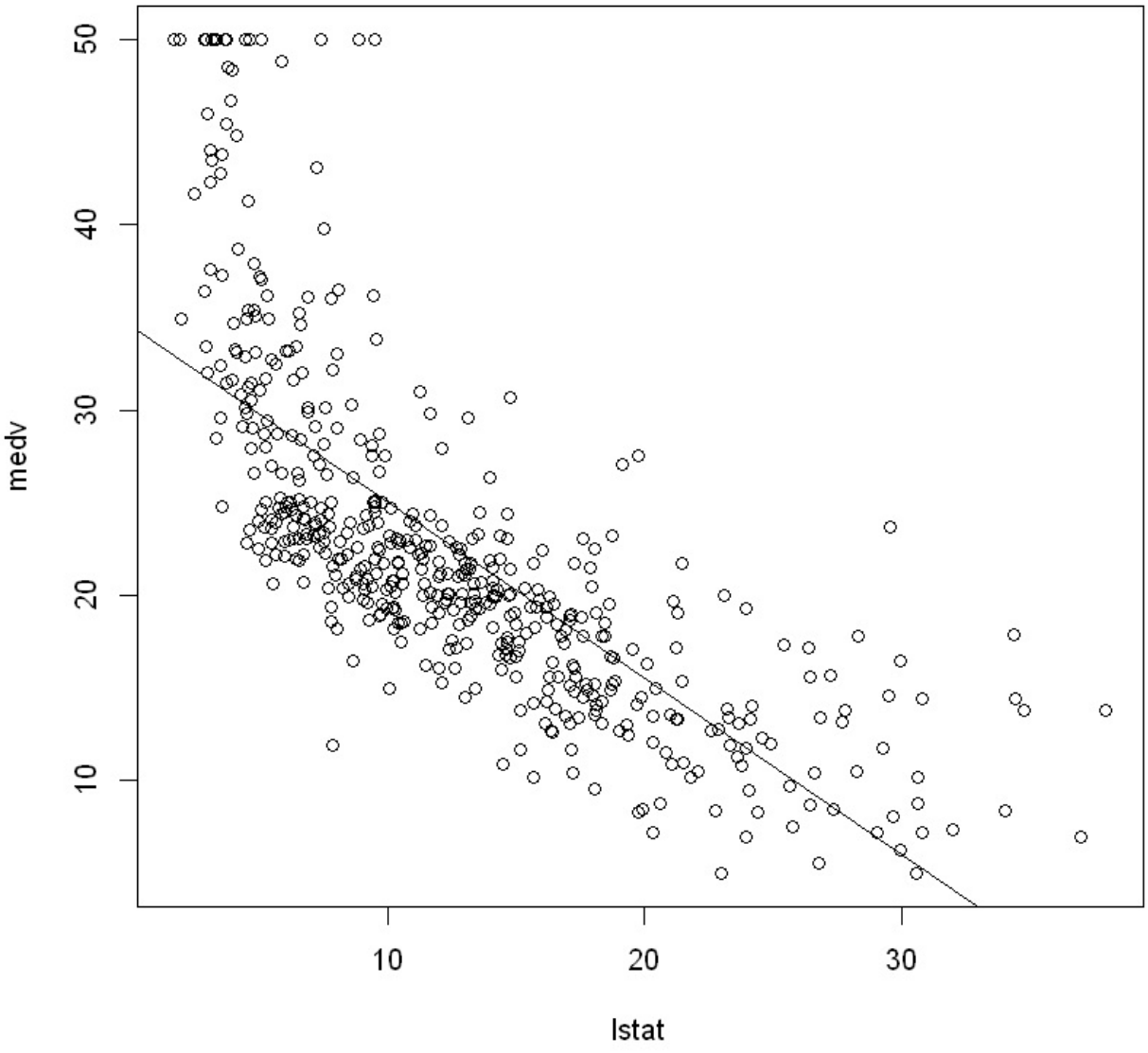
In [116…
```
coef(lm.fit)
```

(Intercept)	34.5538408793831
lstat	-0.950049353757991

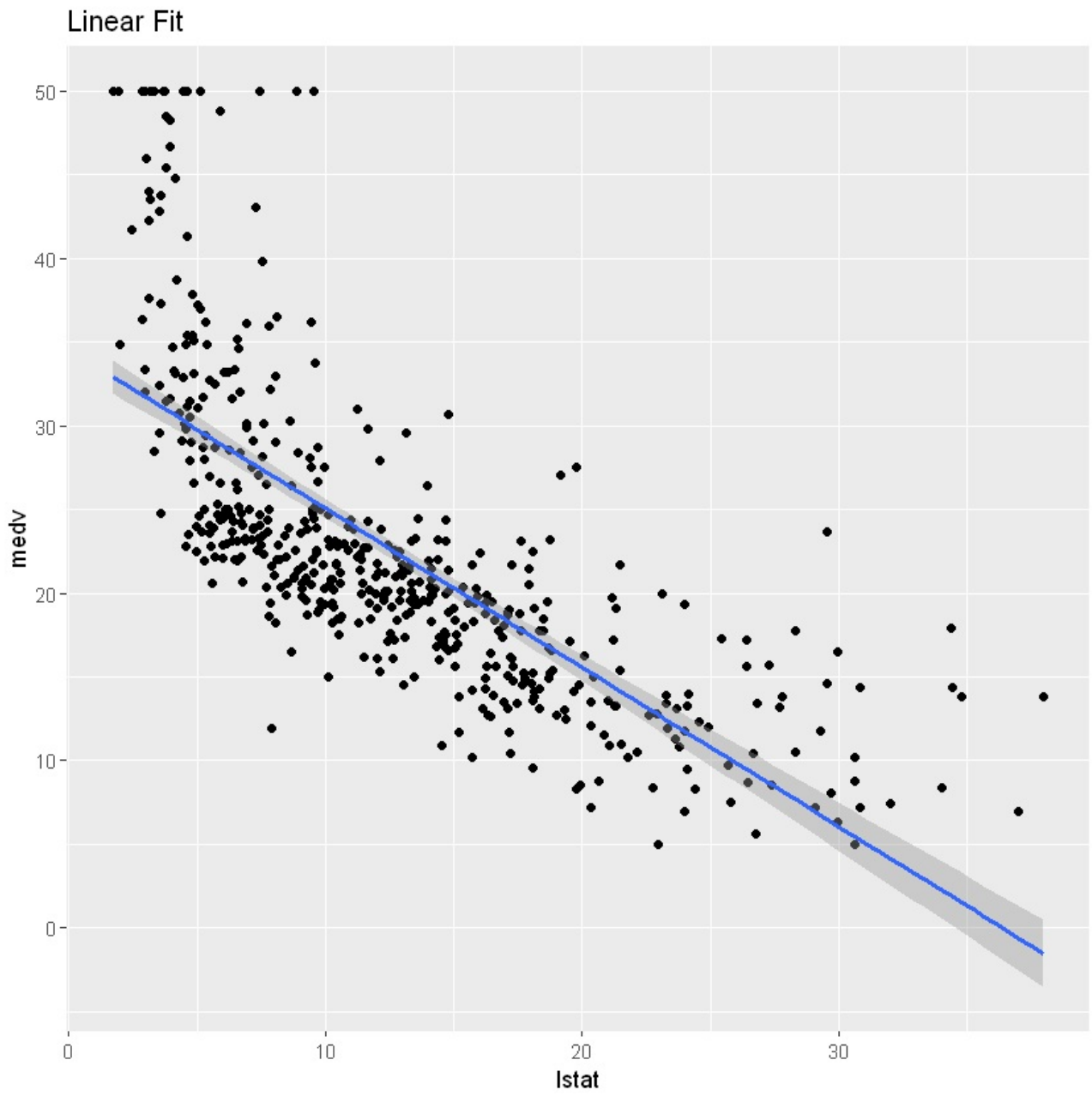
In [117…
```
confint (lm.fit) # Confidence interval for the model
```

	2.5 %	97.5 %
(Intercept)	33.448457	35.6592247
lstat	-1.026148	-0.8739505

In [118…
```
# plot the resulting fit and show a plot of fitted values vs. residuals.
plot(lstat ,medv)
abline (lm.fit)
```



In [120…
```
# Same plotting using ggplot
ggplot(Boston, aes(lstat, medv)) +
geom_point() + stat_smooth(method = lm, se = TRUE) +
ggtitle("Linear Fit")
```



In [121…
```
# predict function to calculate values response values for lstat of 5, 10, and 15
predict (lm.fit ,data.frame(lstat=c(5 ,10 ,15) ), interval ="confidence")
```

fit	lwr	upr
29.80359	29.00741	30.59978
25.05335	24.47413	25.63256
20.30310	19.73159	20.87461

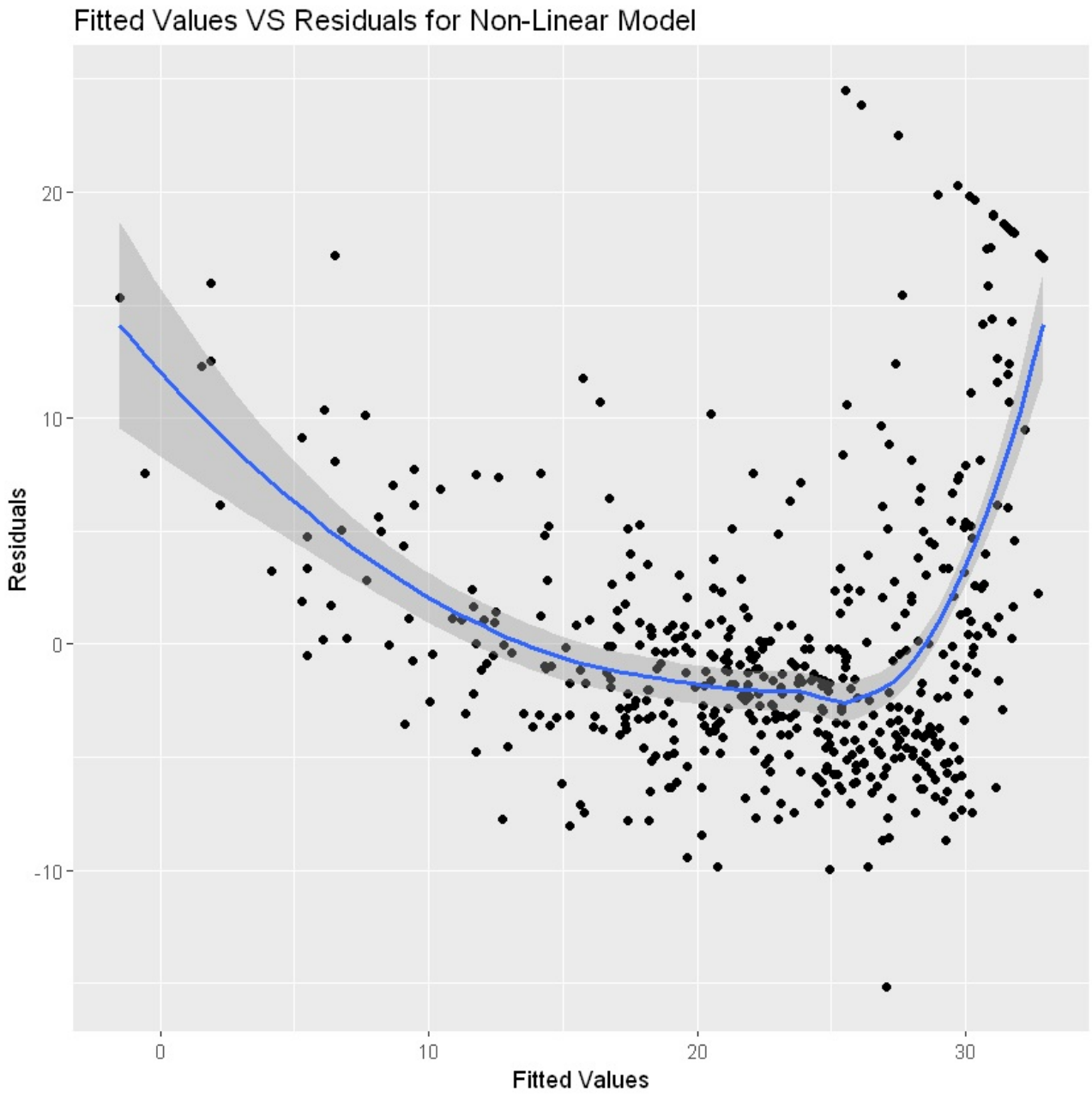
In [122…
```
predict (lm.fit ,data.frame(lstat=c(5 ,10 ,15) ), interval ="prediction")
```

fit	lwr	upr
29.80359	17.565675	42.04151
25.05335	12.827626	37.27907
20.30310	8.077742	32.52846

In [123…
```
#The results have different confidence and prediction levels.
# For example, the 95% confidence interval for a lstat value of 10 is (24.47, 25.63),
#while the 95% prediction interval is (12.828, 37.28).
#As a result, the prediction interval is much wider than the confidence interval,
#which is centered around the same point (a predicted value of 25.05 for medv when
#lstat is equal to 10).
```

In [124…
```
# Plotting the fit and residuals using ggplot
ggplot(lm.fit, aes(x =lm.fit$fitted.values, y = lm.fit$residuals)) +
geom_point() + stat_smooth(se = TRUE) + labs(x = "Fitted Values", y = "Residuals") +
ggtitle("Fitted Values VS Residuals for Non-Linear Model")
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



In [125…
```
cat("Modify the regression to include lstat^2")
```

```
Modify the regression to include lstat^2
```

In [126…
```
# Modify the regression to include lstat2
linearModel2=lm(medv~lstat + I(lstat^2))
summary(linearModel2)
```

```
Call:
lm(formula = medv ~ lstat + I(lstat^2))

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
```

In [127…
```
coef(linearModel2)
```

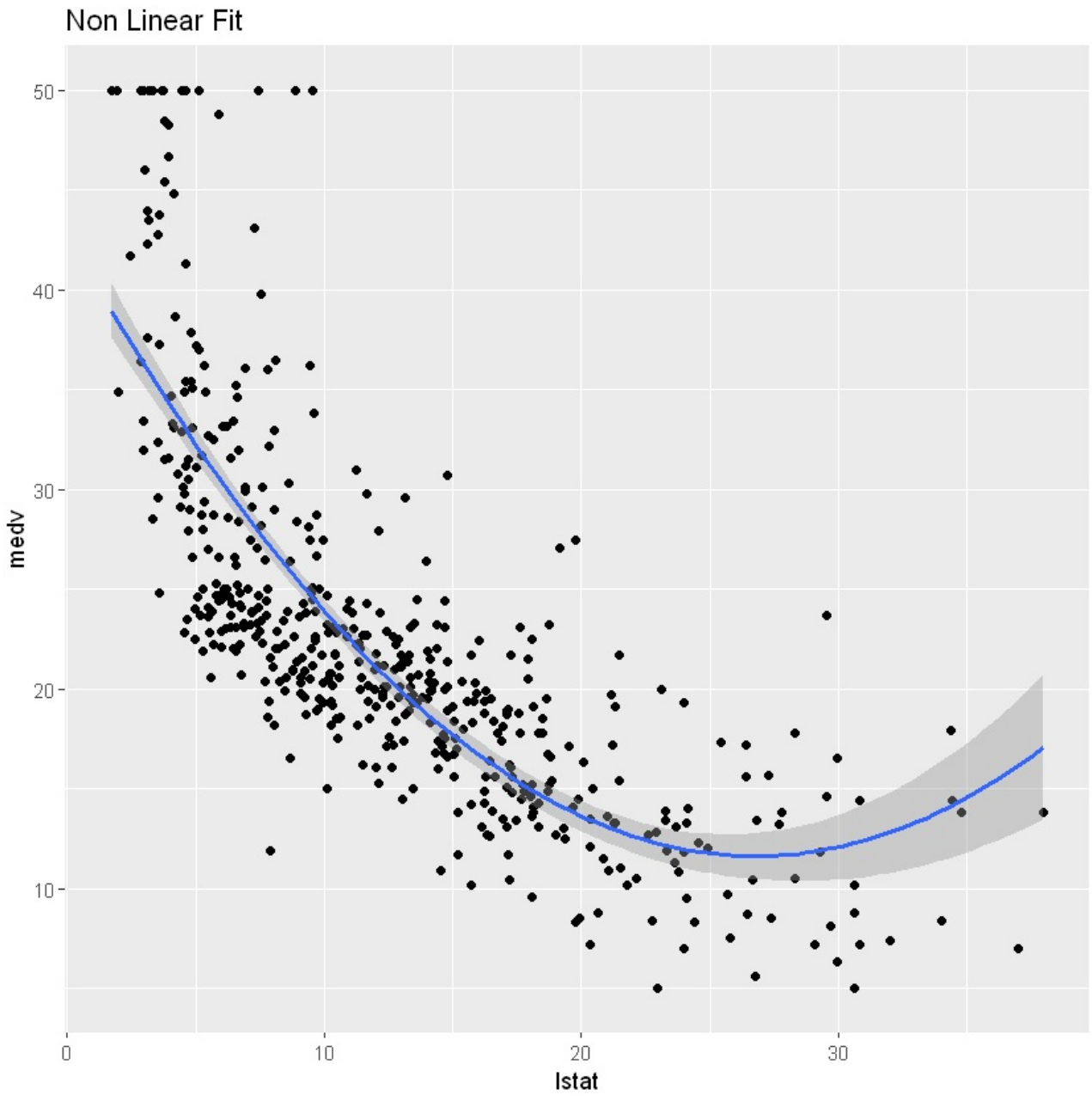
(Intercept)	42.8620073281693
lstat	-2.33282109828273
I(lstat^2)	0.0435468893582221

In [128…
```
anova(lm.fit,linearModel2)
```

Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
504	19472.38	NA	NA	NA	NA
503	15347.24	1	4125.138	135.1998	7.630116e-28

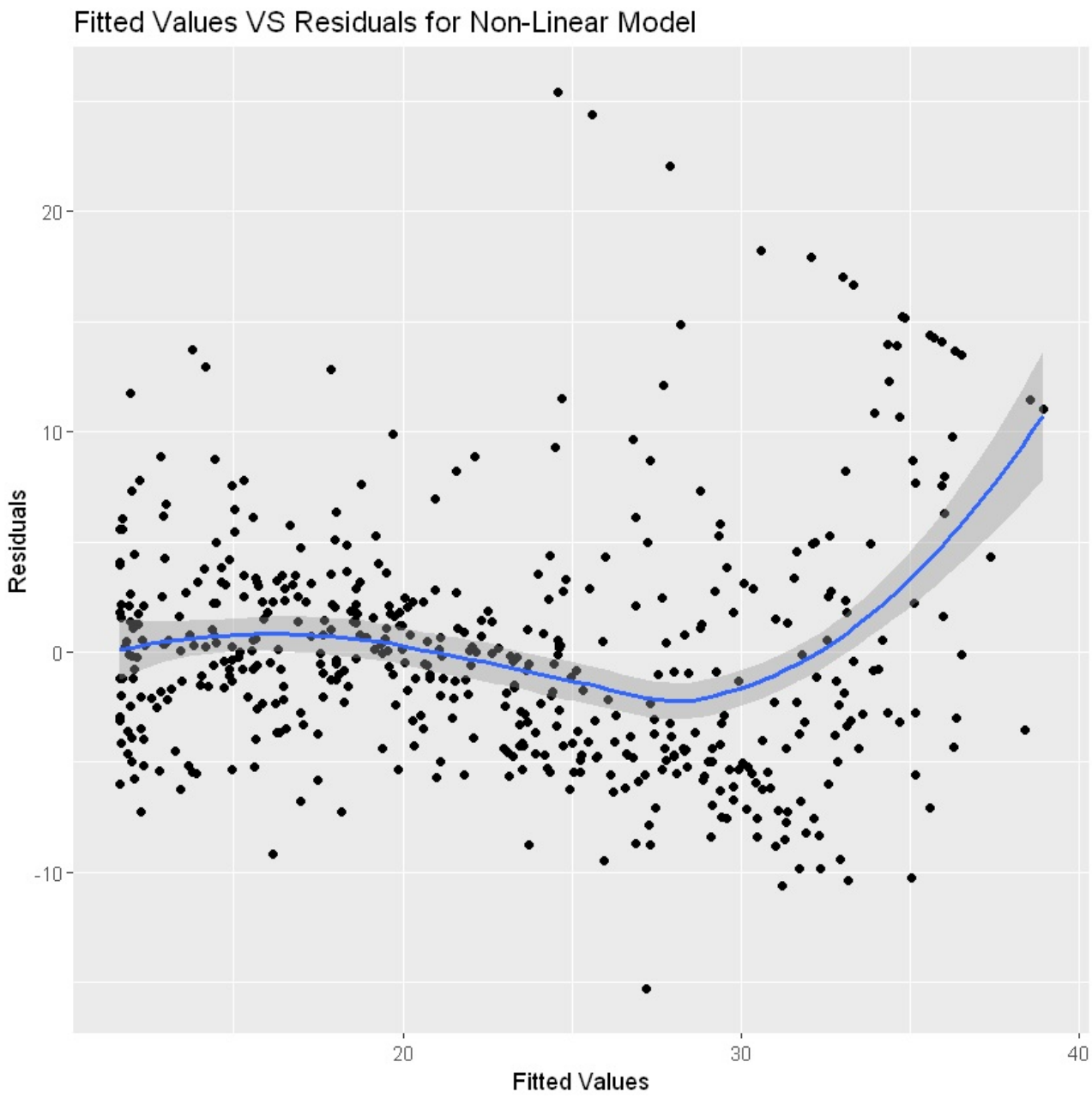
In [129…
```
#The R2 value for the nonlinear model is 0.6407.The quadratic term's close to zero
#p-value shows that it results in a better model.
#We further quantify how much the quadratic fit outperforms the linear fit using
#the anova() method.
#The anova() function does a test of comparison between the two models.
#The alternative hypothesis is that the full model is preferable,
#with the null hypothesis being that the two models fit the data equally well.
#Here, the p-value is almost 0 and the F-statistic is 135.
#This shows unequivocally that the model that includes both lstat and lstat2 is
#considerably superior to the model that only includes lstat.
```

In [130…
```
# use ggplot2and stat smooth to plot the relationship.
ggplot(Boston,aes(x= lstat,y = medv)) +
geom_point() +
stat_smooth(method = "lm",
formula = y ~ x + I(x^2),
se = TRUE) +
ggtitle("Non Linear Fit")
```



In [131…
```
# use ggplot2 and stat smooth to plot the relationship.
ggplot(linearModel2, aes(x =linearModel2$fitted.values, y = linearModel2$residuals)) +
geom_point() + stat_smooth(se = TRUE) + labs(x = "Fitted Values", y = "Residuals") +
ggtitle("Fitted Values VS Residuals for Non-Linear Model")
```

```
`geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



In [ ]: