CSP 571 Assignment 1

Name: Prabhu Avula
CWID: A20522815

Recitation Exercises

1.1 - Chapter 2:

Problem 1:

a) Due to the enormous sample size, the flexible technique would perform better as it can extract more information. The risk of overfitting is reduced because n is quite large. Hence, the flexible technique would be better for this scenario than the inflexible technique.
b) In the reverse scenario, because we have many predictors and a smaller number of observations, there is a massive risk of overfitting. So, to minimize instances where the model picks interference due to overfitting, it's best to employ the inflexible technique rather than the flexible one.
c) This scenario seems to be a highly non-linear one. Hence, it's best to use the flexible method here.
d) From the given variance error terms, it's safe to assume that it is high. That generally means, there would be a lot of interference due to the higher chance of overfitting. Hence, to minimize it, it's best to use the inflexible method.

Problem 2:

a) In this case, we are interested in the CEO's salary. It's a variable that's continuous. Therefore, it is an inference problem. Hence, best to call it a regression problem. Now, from the given data, we can say that n = 500 observations (Number of firms), and p = 3 variables (factors).
b) In this circumstance, we are simply interested if the product is a success or a failure. A simple, yes, or no, type of scenario. We are invested in the prediction. Hence, classification. Also, n = 20 observations (for the number of products they researched), and p = 13 variables (such as marketing budget, price, etc.).
c) This is a regression problem, primarily because, it is common knowledge that currency exchange rates are continuous, constantly changing throughout the year. Our interest here is in the prediction of weekly exchange rates. So, given the data, n = 52 observations (number of weeks), and p = 3 variables (each country's exchange rate's percentage change).

Problem 4:

A. Scenarios where classification is used:

   Invest in a vehicle. The response would either be a yes or a no. The dependent factors would be things such as commute distance, the proximity of both locations to public transport, the subject's income stream, credit history, state of the weather, and travel time. The goal is to predict if, based on the factors, should the subject invest in a vehicle or not.

   Figuring out if a transaction is valid. The response would be a yes or no. The dependent factors would be things such as date, time, amount, location, sender details, receiver details, and account holder's transaction history. The goal is to predict if the transaction is valid or not to minimize fraud.

   Recognizing images in autonomous vehicles such as a Tesla. The response would be a multi-variable. It could be a truck, a dumpster, a pedestrian, or a car. The dependent factors include image depth, pixel count, lidar scans, etc. The goal is to make sure the model processes the image correctly in real time and gives the appropriate variable class as a response so the driver can understand.

B. Real-time scenarios of regressions:

   Credit check in finance. The response is to find out the creditworthiness of an individual or a corporate entity. The dependent factors would include income, basic financial information, levels of debt, recent loans, etc. The main objective encompasses both prediction and inference. Prediction is vital for evaluating the potential risk associated with giving credit to an individual or corporate entity. Additionally, inference plays a role in deciding which financial variables carry the highest significance in determining creditworthiness.

   Percentage of alcohol in each liquor. The response is to find out the percentage of alcohol in a spirit or liquor. Factors include things such as fermentation time, chlorides and acids added, the type of grape used, and the barrel type used. The goal is prediction because we are trying to figure out how good the liquor is going to be. Cheers.

   Stock variation prediction. The response is to find out how much variation there is in a stock over a day's time. Factors include things such as the stock's performance so far, the company's financial position, and social setting-related things such as the management in charge of the company, its growth, etc.

C. Real-time scenarios using clustering.

Working in Account Receivables, I used clustering techniques to group faulty payments from the client. A set pattern that I could recognize immediately and train the application to flag as a faulty payment.

Through the process of clustering individuals according to their genetic markers or profiles, biologists and researchers often get valuable insights into evolutionary mechanisms, gene trends, and the behavior of inherited traits within certain population groups or demographics.

Cluster analysis can be used to group transactions showing similarities in attributes such as transaction amount, frequency, and location. This helps in the detection of potential fraudulent patterns, enhancing security measures.

Problem 6:

The parametric approach presupposes a specific form for a function f, simplifying the estimation process to determine a set of parameters. On the other hand, the non-parametric approach needs a substantial dataset for an accurate estimation of the function, as it refrains from assuming a specific form. Parametric methods offer the advantage of requiring fewer observations compared to non-parametric methods. However, they are prone to inaccurate estimations if the assumed form of the function is incorrect.

Problem 7:

a) Euclidean Distance = √ ((x2 - x1) ^2 + (y2 - y1) ^2 + (z2 - z1) ^2)
In this case, we are calculating the Euclidean distance between each observation and the test point where X1 = X2 = X3 = 0:

Observation 1: √ (02 +32 +02) = 3 | Red
Observation 2: √ (22 +02 +02) = 2 | Red
Observation 3: √ (02 +12 +32) = 3.16 | Red
Observation 4: √ (02 +12 +22) = 2.23 | Green
Observation 5: √ (-12 +02 +12) = 1.41 | Green
Observation 6: √ (12 +12 +12) = 1.73 | Red

b) For K=1, we choose the single nearest neighbor for making a prediction. In this instance, the fifth observation exhibits the closest Euclidean distance of 1.41 to our test point. Likewise, the prediction would be derived from the response variable value of that observation, which is green.

c) When K=3, we look at the three closest neighbors. In this situation, those neighbors are Observation 5 (which is 1.41 units away), Observation 6 (1.73 units away), and Observation 2 (2 units away). Out of these three, Observations 5 and 6 are red, while Observation 2 is green. Since red appears more often, the spot at (0, 0, 0) will be red.

d) When the Bayes decision boundary exhibits high nonlinearity, it implies that the relationship between predictors and response variables is likely not a linear one. In situations where such nonlinearities exist, it is generally known that smaller values of K will yield superior performance. This is because a smaller K value would offer greater flexibility in capturing patterns within the data, in contrast to larger K values which tend to smooth out these patterns by considering other, far neighbors.

1.2 Chapter 3:

Problem 1:

The p-values provided in the table are associated with null hypotheses stating that the advertising budgets for Radio, TV, Intercept, and newspaper have no influence on sales. The negative p-values for Interceptor, TV, and Radio lead us to reject these null hypotheses, indicating that these three variables indeed have an impact on sales. However, the p-value for Newspaper is 0.8599, suggesting that Newspaper expense does not significantly affect sales.

Problem 3:

A. The least square can be calculated with the following equation:

   Salary = 50+20*GPA+0.07*IQ+35*Graduates+0.01*GPA*IQ-10*GPA* Graduates

   So, the value for High School Graduates is:
   50+20*GPA+0.07*IQ+0.01*GPA*IQ
   And, for College Graduates, it would be:
   85+10*GPA+0.07*IQ+0.01*GPA*IQ

   So, the starting salary for High School Graduates is greater than College Graduates' average if 50+20*GPA >= 85+10*GPA -> GPA>=3.5

   Therefore, option iii is right.

B. Plugging into the least squares regression model, we have

   $Y = 50 + 20(4.0) + 0.07(110) + 35(1) + 0.01(4.0)(110) − 10(4.0)(1) = 137.1.$

This gives us a predicted salary of $137,000 for a college grad with an IQ of 110 and a GPA of 4.0.

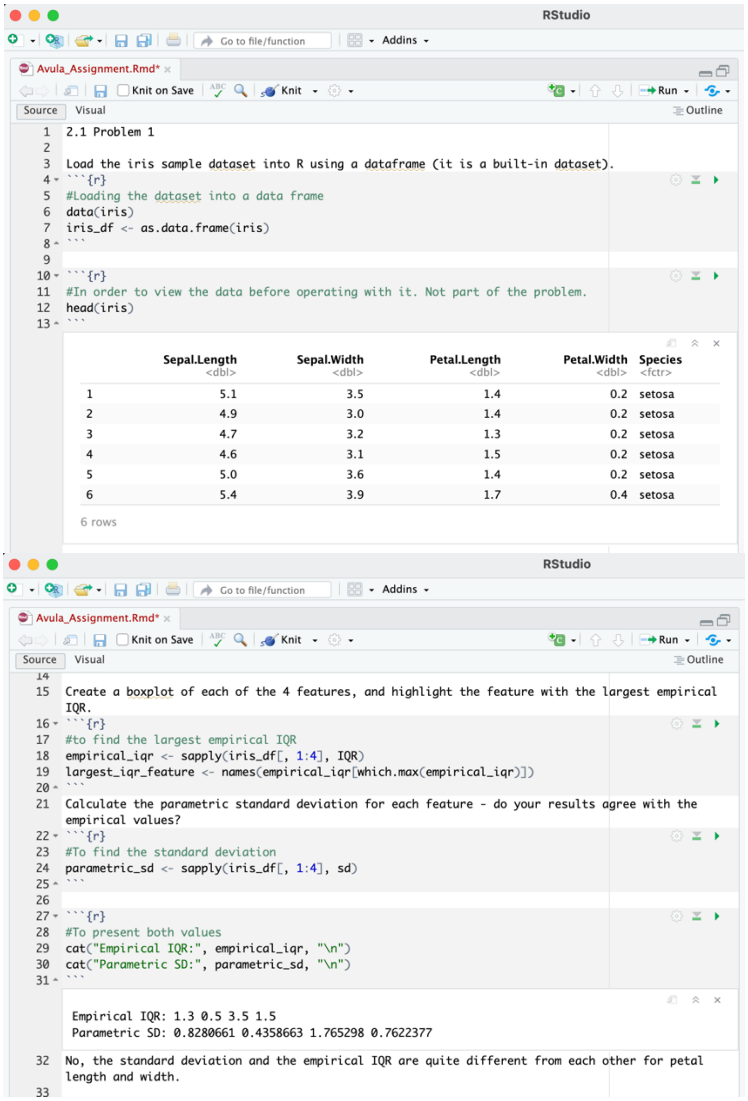
C. False. Although the interaction might be small, it would still have an effect because the magnitude of the interaction and the statistical significance have no relation.

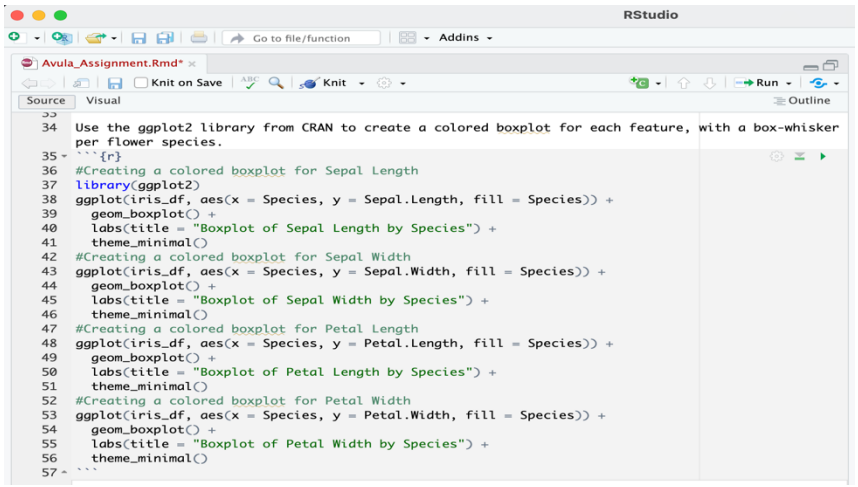
Problem 4:

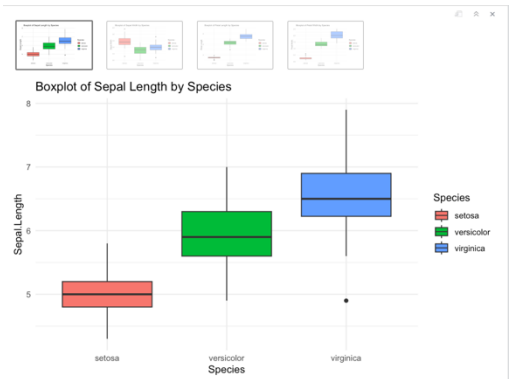
A. If the underlying relationship between the predictors and the response variable is linear, the use of cubic regression might introduce unnecessary interference and lead to overfitting. Likewise, the training RSS for the cubic regression model may be higher than that of the simpler linear regression model.
B. It is not easy to draw an objective conclusion because the test RSS is reliant on the test data. However, it can be expected that cubic regression has a higher test RSS than linear regression as there will be a higher chance for errors due to overfitting.
C. It would be uncertain if a cubic relationship can adequately account for the non-linearity as there is a lack of sufficient information to confirm.
D. Not enough information to draw a conclusion on the matter.

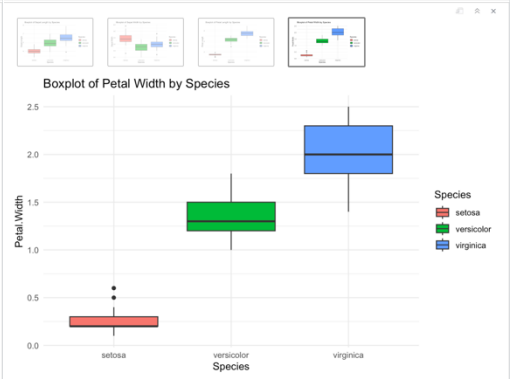
Practicum Problems

2.1 Problem 1:




RStudio
Go to file/function
Addins
Avula_Assignment.Rmd*
Knit on Save
Knit
Run
Source
Visual
Outline
Use the ggplot2 library from CRAN to create a colored boxplot for each feature, with a box-whisker per flower species.
```{r}
#Creating a colored boxplot for Sepal Length
library(ggplot2)
ggplot(iris_df, aes(x = Species, y = Sepal.Length, fill = Species)) +
geom_boxplot() +
labs(title = "Boxplot of Sepal Length by Species") +
theme_minimal()
#Creating a colored boxplot for Sepal Width
ggplot(iris_df, aes(x = Species, y = Sepal.Width, fill = Species)) +
geom_boxplot() +
labs(title = "Boxplot of Sepal Width by Species") +
theme_minimal()
#Creating a colored boxplot for Petal Length
ggplot(iris_df, aes(x = Species, y = Petal.Length, fill = Species)) +
geom_boxplot() +
labs(title = "Boxplot of Petal Length by Species") +
theme_minimal()
#Creating a colored boxplot for Petal Width
ggplot(iris_df, aes(x = Species, y = Petal.Width, fill = Species)) +
geom_boxplot() +
labs(title = "Boxplot of Petal Width by Species") +
theme_minimal()
```





RStudio

Go to file/function | Addins

Avula_Assignment.Rmd*

Knit on Save | Knit | Run

Source | Visual | Outline

Species

```
58
59  Which flower type exhibits a significantly different Petal Length/Width once it is separated from
    the other classes?
60  ```{r}
61  #To perform t-tests for Petal Length and Width
62  t_test_length <- t.test(setosa_data$Petal.Length, other_species_data$Petal.Length)
63  t_test_width <- t.test(setosa_data$Petal.Width, other_species_data$Petal.Width)
64
65  #To extract the p-values from the t-test results
66  p_value_length <- t_test_length$p.value
67  p_value_width <- t_test_width$p.value
68
69  #To print the p-values
70  cat("T-test for Petal Length - p-value:", p_value_length, "\n")
71  cat("T-test for Petal Width - p-value:", p_value_width, "\n")
72  ```

    T-test for Petal Length - p-value: 1.746188e-69
    T-test for Petal Width - p-value: 1.347804e-60

73  Virginica exhibits a different petal length/width once separated.
74
```

2.2 Problem 2:

2.2 Problem 2

Load the trees sample dataset into R using a dataframe (it is a built-in dataset), and produce a 5-number summary of each feature.

```r
#Load the dataset into a frame
data(trees)
trees_df <- as.data.frame(trees)
```

```r
#For my observation only
head(trees)
```

Description: df [6 × 3]

	Girth <dbl>	Height <dbl>	Volume <dbl>
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

6 rows

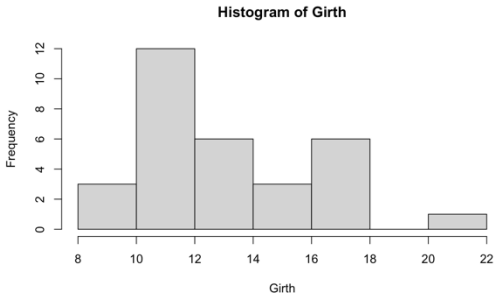
Produce a 5-number summary of each feature

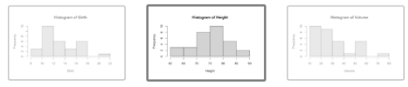
```r
#Simple line to get the summary.
summary(trees_df)
```

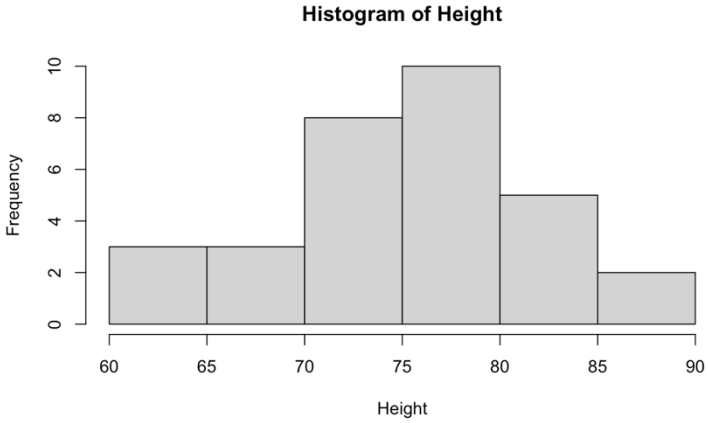
```
     Girth           Height       Volume     
 Min.   : 8.30   Min.   :63   Min.   :10.20  
 1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40  
 Median :12.90   Median :76   Median :24.20  
 Mean   :13.25   Mean   :76   Mean   :30.17  
 3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30  
 Max.   :20.60   Max.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

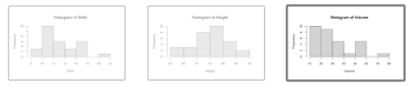
Create a histogram of each variable - which variables appear to be normally distributed based on visual inspection?

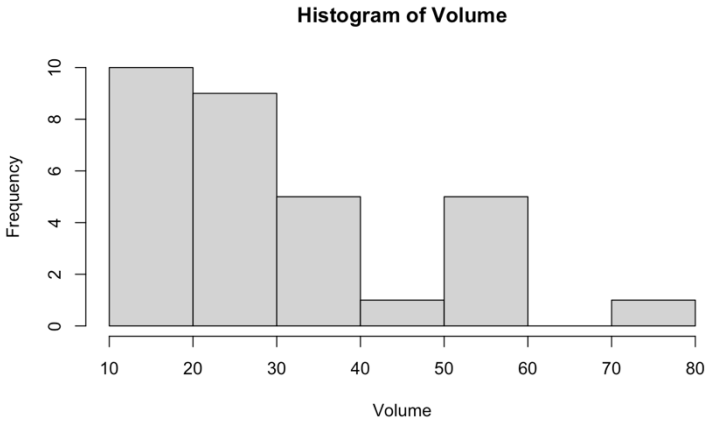
```r
#to get the histograms for girth, volume and height
hist(trees_df$Girth, main="Histogram of Girth", xlab="Girth")
hist(trees_df$Height, main="Histogram of Height", xlab="Height")
hist(trees_df$Volume, main="Histogram of Volume", xlab="Volume")
```






Histogram of Height
Frequency
0
2
4
6
8
10
60
65
70
75
80
85
90
Height




Histogram of Volume
Frequency
0
2
4
6
8
10
10
20
30
40
50
60
70
80
Volume

Looking at the histograms, all of them are skewed.

Do any variables exhibit positive or negative skewness? Install the moments library from CRAN use the skewness function to calculate the skewness of each variable. Do the values agree with the visual inspection?

```{r}
#get the necessary packages
install.packages("moments")
library(moments)
```

```
 trying URL 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.3/moments_0.14.1.tgz'
 Content type 'application/x-gzip' length 54871 bytes (53 KB)
 ==================================================
 downloaded 53 KB


 The downloaded binary packages are in
        /var/folders/2q/9m45whf90zjg8hs1lzn3hm140000gn/T//RtmpxCuAUc/downloaded_packages
```

```{r}
#Calculate the skewness for each
skewnessgirth <- skewness(trees_df$Girth)
skewnessheight <- skewness(trees_df$Height)
skewnessvolume <- skewness(trees_df$Volume)
```

```{r}
#present the skewness for each
cat("Skewness of Girth:", skewnessgirth, "\n")
cat("Skewness of Height:", skewnessheight, "\n")
cat("Skewness of Volume:", skewnessvolume, "\n")
```

```
 Skewness of Girth: 0.5263163
 Skewness of Height: -0.374869
 Skewness of Volume: 1.064357
```

Based on the calculations as well as the histograms from the previous cells, height has a slight negative skew whereas girth has a slightly positive skew and volume has a positive skew. Hence, the values match the visual inspection.

2.3 Problem 3

2.3 Problem 3

Load the auto-mpg sample dataset from the UCI Machine Learning Reposi- tory (auto-mpg.data) into R using a dataframe (Hint: You will need to use read.csv with url, and set the appropriate values for header,as.is, and sep).

```r
#Loading the dataset from the repo
url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data"

#Reading the dataset. Did not need to specify a separator
auto_mpg <- read.table(url, header = FALSE, sep = "", na.strings = "?")

#Defining column names
col_names <- c("mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration", "model_year", "origin",
"car_name")

#Setting the column names
colnames(auto_mpg) <- col_names
```

```r
#Just to view the table before operating on it.
head(auto_mpg)
```

Description: df [6 × 9]

	mpg <dbl>	cylinders <int>	displacement <dbl>	horsepower <dbl>	weight <dbl>	acceleration <dbl>	model_year <int>	origin <int>
1	18	8	307	130	3504	12.0	70	1
2	15	8	350	165	3693	11.5	70	1
3	18	8	318	150	3436	11.0	70	1
4	16	8	304	150	3433	12.0	70	1
5	17	8	302	140	3449	10.5	70	1
6	15	8	429	198	4341	10.0	70	1

6 rows | 1–9 of 9 columns

The horsepower feature has a few missing values with a ? - and will be treated as a string. Use the as.numeric casting function to obtain the column as a nu- meric vector, and replace all NA values with the median.

```{r}
#Converting the hp column to numeric
auto_mpg$horsepower <- as.numeric(auto_mpg$horsepower)
#Precautionary check to make sure its numeric
is.numeric(auto_mpg$horsepower)
#to check the median
medianhp<-median(auto_mpg$horsepower,na.rm =TRUE)
medianhp
#To check the mean
meanhp<-mean(auto_mpg$horsepower,na.rm =TRUE)
meanhp
#To check null rows
nullrows<-sum(is.na(auto_mpg$horsepower))
nullrows
```

```
[1] TRUE
[1] 93.5
[1] 104.4694
[1] 6
```

```{r}
#Calculating the mean of the hp column before and after replacing missing values
original_mean <- mean(auto_mpg$horsepower, na.rm = TRUE)
#to replace any nulls with median
auto_mpg$horsepower[is.na(auto_mpg$horsepower)] <- median.default(auto_mpg$horsepower, na.rm = TRUE)
```

How does this affect the value obtained for the mean vs the original mean when the records were ignored?

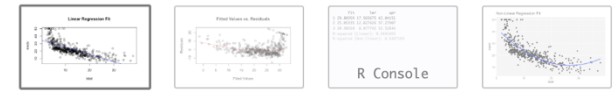
```{r}
# Print the original and updated means
cat("Original Mean:", original_mean, "\n")
cat("Updated Mean:", mean(auto_mpg$horsepower), "\n")
```

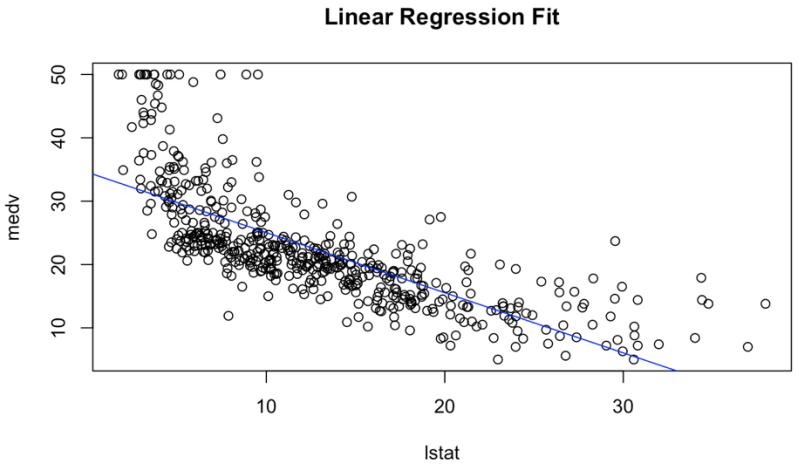
```
Original Mean: 104.4694
Updated Mean: 104.304
```

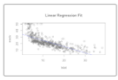
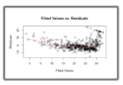
Since replacing null rows with the median and calculating the mean again, we see a slight decrease from the original mean. This makes sense as the median is 93.5, well below the original mean. Hence, the overall updated mean decreased.

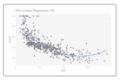
2.4 Problem 4

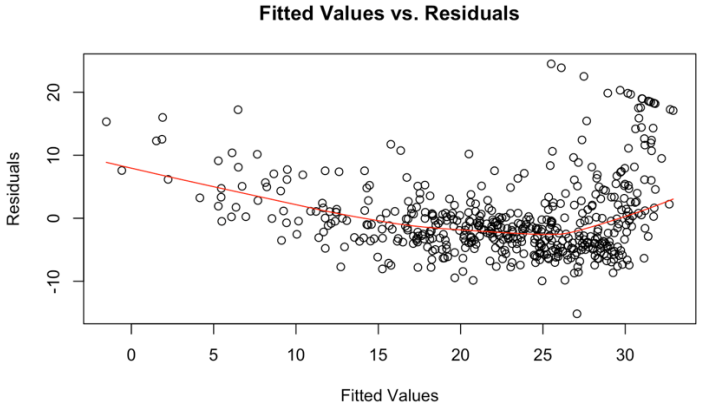
```r
#Loading the MASS package and the Boston dataset
library(MASS)
data(Boston)

#Fitting a linear regression model
model <- lm(medv ~ lstat, data = Boston)

#Plotting the regression fit
plot(Boston$lstat, Boston$medv, main = "Linear Regression Fit", xlab = "lstat", ylab = "medv")
abline(model, col = "blue")

#Plotting fitted values vs. residuals
residuals <- residuals(model)
fitted_values <- fitted(model)
plot(fitted_values, residuals, main = "Fitted Values vs. Residuals", xlab = "Fitted Values", ylab = "Residuals")

#Checking for non-linearity by adding a smoother
lines(lowess(fitted_values, residuals), col = "red")

#Calculating predictions for lstat of 5, 10, and 15
new_data <- data.frame(lstat = c(5, 10, 15))
predictions <- predict(model, newdata = new_data, interval = "prediction", level = 0.95)
print(predictions)

#Fitting a non-linear regression model including lstat^2
model_nonlinear <- lm(medv ~ lstat + I(lstat^2), data = Boston)

#Calculating R-squared values for both linear and non-linear models
r_squared_linear <- summary(model)$r.squared
r_squared_nonlinear <- summary(model_nonlinear)$r.squared

#Comparing R-squared values
cat("R-squared (Linear):", r_squared_linear, "\n")
cat("R-squared (Non-linear):", r_squared_nonlinear, "\n")

#Plotting the non-linear fit using ggplot2
library(ggplot2)
ggplot(Boston, aes(x = lstat, y = medv)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x + I(x^2), se = FALSE, color = "blue") +
  labs(title = "Non-Linear Regression Fit", x = "lstat", y = "medv")
```

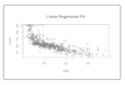
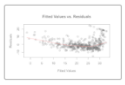


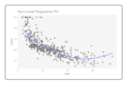


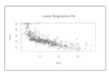
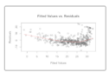


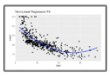


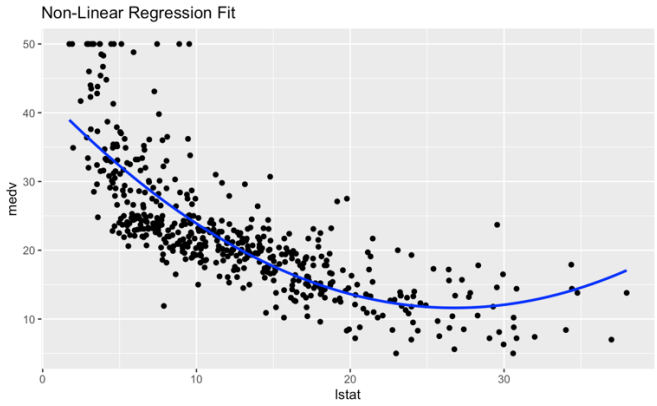



```
       fit       lwr      upr
1 29.80359 17.565675 42.04151
2 25.05335 12.827626 37.27907
3 20.30310  8.077742 32.52846
R-squared (Linear): 0.5441463
R-squared (Non-linear): 0.6407169
```





As per the graphs, there is an indication of non-linearity in the relationship between lstat and medv. Furthermore, the confidence interval and prediction interval differ in their interpretation of response values. A wider prediction interval indicates greater uncertainty around a specific value, while a narrower confidence interval reflects uncertainty around the estimated mean.