HW1

Sohaib Syed

2023-01-24

Rectitation Exercises

1.1 Chapter 2

Exercise 1

a) With a large n and small p, we can expect that a more flexible method will perform better because it has more data to extract information from possibly doing a better job at avoiding underfitting

b) A flexible model would be worse because the small number of samples (n) will cause the learning method to overfit and use the small amount of data to (assumingly) generalize that on a larger test data set.

c) in a non-linear case higher flexibility learning methods will perform better since they're less restricted.linear regression is considered to be a more inflexible method, and using that approach on a data set with non-linear relationship, first of all doesn't make sense, and then also will not allow a lot proper learning of relationship between response and predictors.

d) when variance of error is high using flexible methods can cause fitting to the error, so the flexible method would be worse.

Excercise 2

a) Regression, inference; n=500, p=3

b) Classification, prediction; n=20, p=13

c) regression, prediction; n=52,p=3

Exercise 4

a) First: A bank loan application, where a bank wants to see if an applicant for a loan should receive a loan or not. Response should be yes or no, where the predictors are income range, married or not married, current debt range, homeowner. This case is merely a prediction application as there is no need for looking for insight on predictors.

Second: a doctor trying to diagnose a patient. In an extremely simple case a doctor is classifying whether a patient has the flu or not. Predictors are body temperature, blood pressure, oxygen levels. This is a prediction task as we aren't looking into how the predictors have an effect on if someone has a flu, rather we simply want an answer.

Third: An airline company wants to see which factors contribute most to cancelled flights. The response would be whether a flight is cancelled or not, and predictors are weather at departure city, weather at arrival city, experience of pilots, number of passengers, airplane model. This is an inference application because the airline would like to know which predictors are most likely to be the ones contributing to cancelled flights.

b) First: Avg Price of eggs in US next year. Response is price of eggs in dollars, predictors are number of farms in US, number of chickens in US, number of truck drivers in US, global temperature. This would be a prediction application to see the average price of eggs.

Second: number of people infected by a disease. Response is number of people with disease, predictors are city population, contagiousness of disease, length that an infected person stays contagious, number of people a person meets a week.

Third: mass of a person. Response would be weight in pounds, predictors are caloric intake, time spent exercising, height, age.

c) First: grouping something as fungus, bacteria or virus

Second: grouping carnivore, herbivore, omnivore

Third: grouping countries based on climate. Tropical, Desert, Cold

Exercise 6

Parametric approaches assume the form of a function (linear, exponential, etc.) and try to estimate a known number parameters of that assumed function, meanwhile non-parametric approaches do not start off with an assumed form of f, and use an arbitrary number of parameters. The advantages of using a parametric approach are that we do not need a large number of observations since we know the exact parameters we are trying to estimate for least error. The disadvantages can come from estimating an incorrect function f which can cause a large error.

Exercise 7

```r
training<-data.frame(x1=c(0,2,0,0,-1,1),x2=c(3,0,1,1,0,1),
                     x3=c(0,0,3,2,1,1),y=c('r','r','r','g','g','r'))

# formula d=sqrt((x12-x11)**2)+(x22-x21)**2)+(x32-x31)**2))

# since new X1,X2,X3 all equal 0, simply the square root of sum of squares of
# observed points
euc_dis<-c(sqrt(training[1,1:3][1]**2+
                  training[1,1:3][2]**2+training[1,1:3][3]**2),
           sqrt(training[2,1:3][1]**2+training[2,1:3][2]**2+
                  training[2,1:3][3]**2),
           sqrt(training[3,1:3][1]**2+training[3,1:3][2]**2+
                  training[3,1:3][3]**2),
           sqrt(training[4,1:3][1]**2+
                  training[4,1:3][2]**2+training[4,1:3][3]**2),
           sqrt(training[5,1:3][1]**2+training[5,1:3][2]**2+
                  training[5,1:3][3]**2),
           sqrt(training[6,1:3][1]**2+training[6,1:3][2]**2+
                  training[6,1:3][3]**2))

euc_dis<-c(3,2,3.162278,2,236068,1.414214,1.732051)
# These are the final euclidean distances
euc_dis
```

a

```
## [1] 3.000000e+00 2.000000e+00 3.162278e+00 2.000000e+00 2.360680e+05
## [6] 1.414214e+00 1.732051e+00
```

```


k=1
NO=sort(euc_dis)[1] # the k-closest observations
Prob_Red<-1/k*sum(training[5,'y']=='r') #5th observation was closest
prob_green<-1/k*sum(training[5,'y']=='g') # 5th observation was closest
prob_green
```

b

```
## [1] 1
```

```
Prob_Red
```

```
## [1] 0
```

```
# Since green is greater than red, that means the prediction is
# that the new point is green
```

```


k=3
NO=c(5,2,6)
Prob_Red<-1/k*sum(training[NO,'y']=='r')
prob_green<-1/k*sum(training[NO,'y']=='g')
prob_green
```

c

```
## [1] 0.3333333
```

```
Prob_Red
```

```
## [1] 0.6666667
```

```
# Since red is greater than green, that means the prediction is that
# the new point is red
```

d referencing Figure 2.16 from ISLR2, it seems that a highly non-linear boundary, or a very flexible boundary, will result from having a lower K. So for this problem, K is best if smaller since it will allow for the boundary to be less smooth and incorporate the non-linearity by being highly flexible.

1.2 Chapter 3

Exercise 1

If we are to approach each p-value separately then the null hypothesis are H0: TV=0 sales, H0: radio =0 sales, H0: newspaper =0 sales. In this case, we would reject H0 for TV has no effect on predicting sales and H0 for radio has no effect on predicting sales, but fail to reject H0 for newspaper has no effect on predicting sales. essentially, Tv and Radio are useful in predictors.

If we approach as single null hypotheses then we have H0: Tv=radio=newspaper =0. looking at the p-values we would reject H0 and conclude that at least one of Tv, radio, or newspaper is useful in predicting sales

Exercise 3

a) y_hat=50+20GPA+.07IQ+35(Level)+.01(GPA x IQ)-10(GPA x Level)

for college: 50+20GPA+.07IQ+35+.01(GPA x IQ)-10GPA

y_hat= 85+10GPA+.07IQ+.01(GPA x IQ) 136

for highschool: 50+20GPA+.07IQ+0+.01(GPA x IQ)-0

y_hat=50+20GPA+.07IQ+.01(GPA x IQ)

the answer is iii because when GPA is greater or equal to 3.5 the highschoolers earn more since that fitted model has a greater coefficient for the GPA predictor

```

y_hat=85+10*(4)+.07*(110)+.01*(4*110)
y_hat
```

b)

```
## [1] 137.1
```

salary of a college graduate is predicted to be $137,100

c) False, because I don't think the coefficient is enough to indicate whether an interaction is taking place or not. Rather, a hypothesis test would be a better indicator by using F scores or p-values.

Exercise 4

a) while the true relationship is linear, because we have added more predictors to our functions the cubic regression will have a lower RSS since it will fit the data better

b) again assuming true relationship was linear, the test RSS for linear will be lower than the cubic regression RSS because the cubic regression overfit to training data.

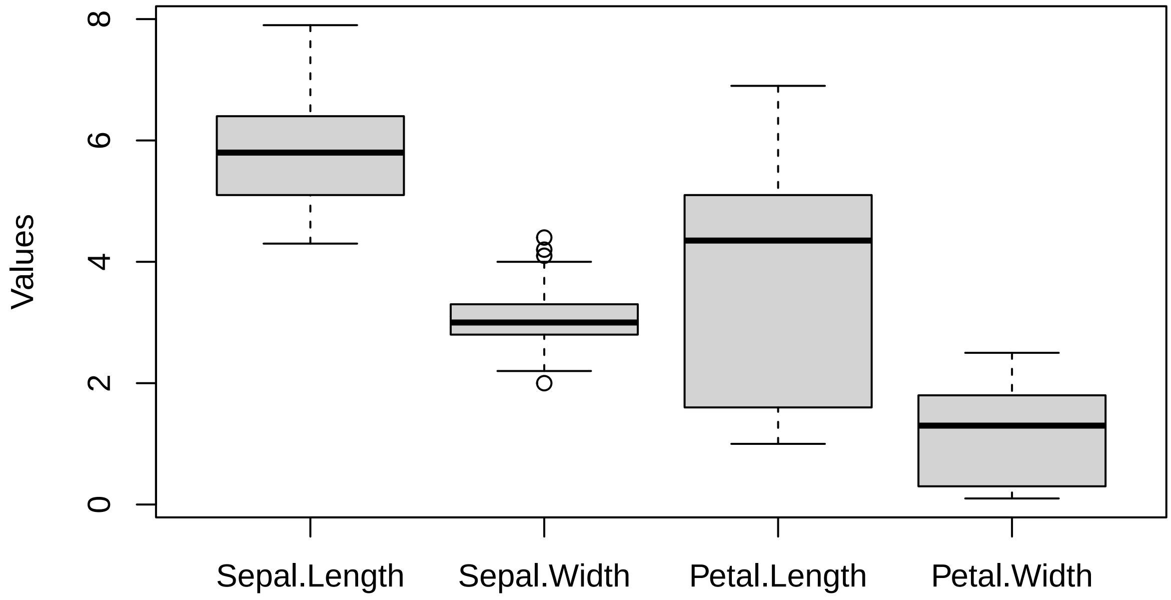
c) If it is certain that true relationship isn't linear, it is certain that a more flexible fit will have a lower RSS. In this case the cubic regression should have lower RSS as it fill fit the non-linear relationship closer than a linear regression would

d) I am unable to tell because we do not know the exact relationship. depending on 'how far from linear' cubic regression might still be an overfit, and linear regression will still perform better. Maybe the relationship is way beyond linear and even past cubic, in this case cubic will perform better as it is more flexible.

Practicum Problems

2.1 Problem 1

```
data(iris)
boxplot(iris[1:4],ylab='Values')
```



```
IQR(iris$Sepal.Length)
```

```
## [1] 1.3
```

```
IQR(iris$Sepal.Width)
```

```
## [1] 0.5
```

```
IQR(iris$Petal.Length)
```

```
## [1] 3.5
```

```
IQR(iris$Petal.Width)
```

```
## [1] 1.5
```

Petal.Length has the highest IQR

```
sd(iris$Sepal.Length)
```

```
## [1] 0.8280661
```

```
sd(iris$Sepal.Width)
```

```
## [1] 0.4358663
```

```
sd(iris$Petal.Length)
```

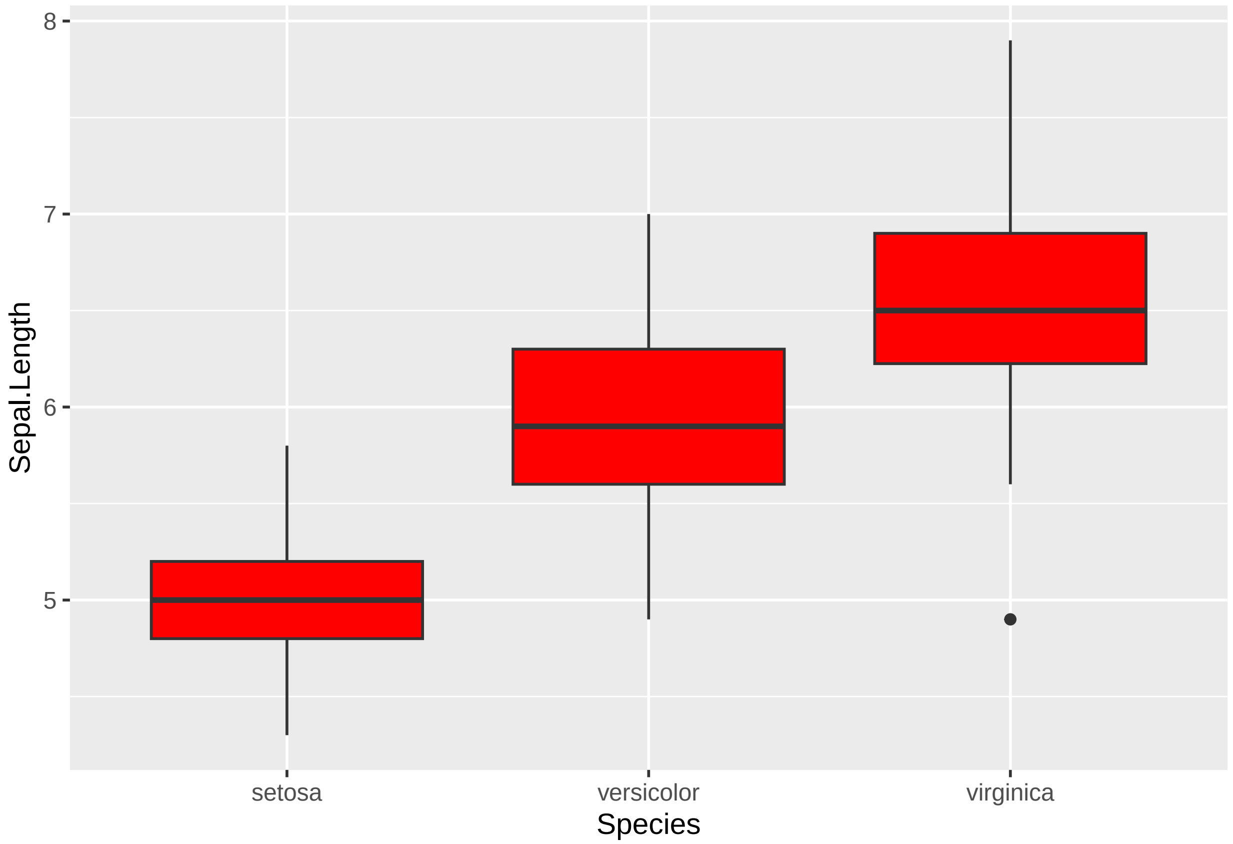
```
## [1] 1.765298
```

```
sd(iris$Petal.Width)
```

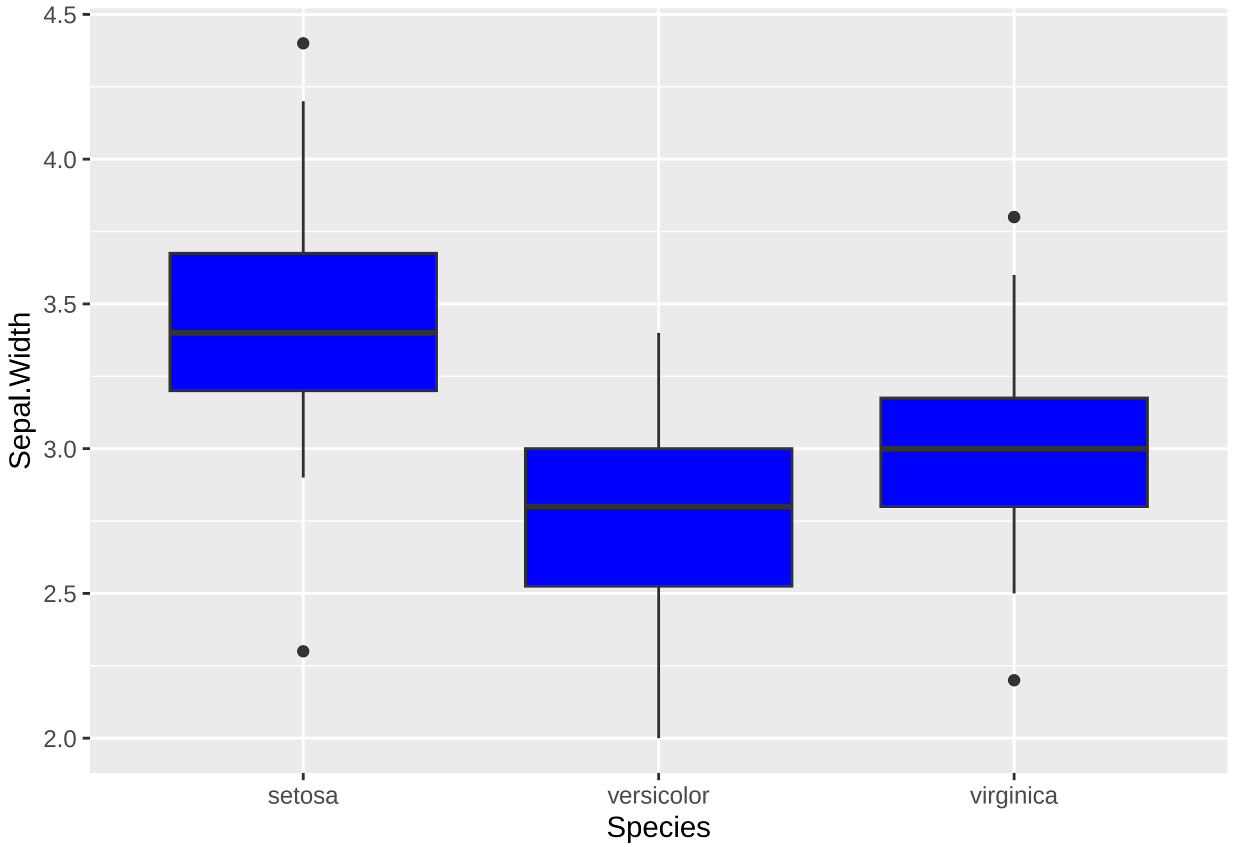
```
## [1] 0.7622377
```

If by agree it means that the Petal.Length also has the largest parametric standard deviation then yes.

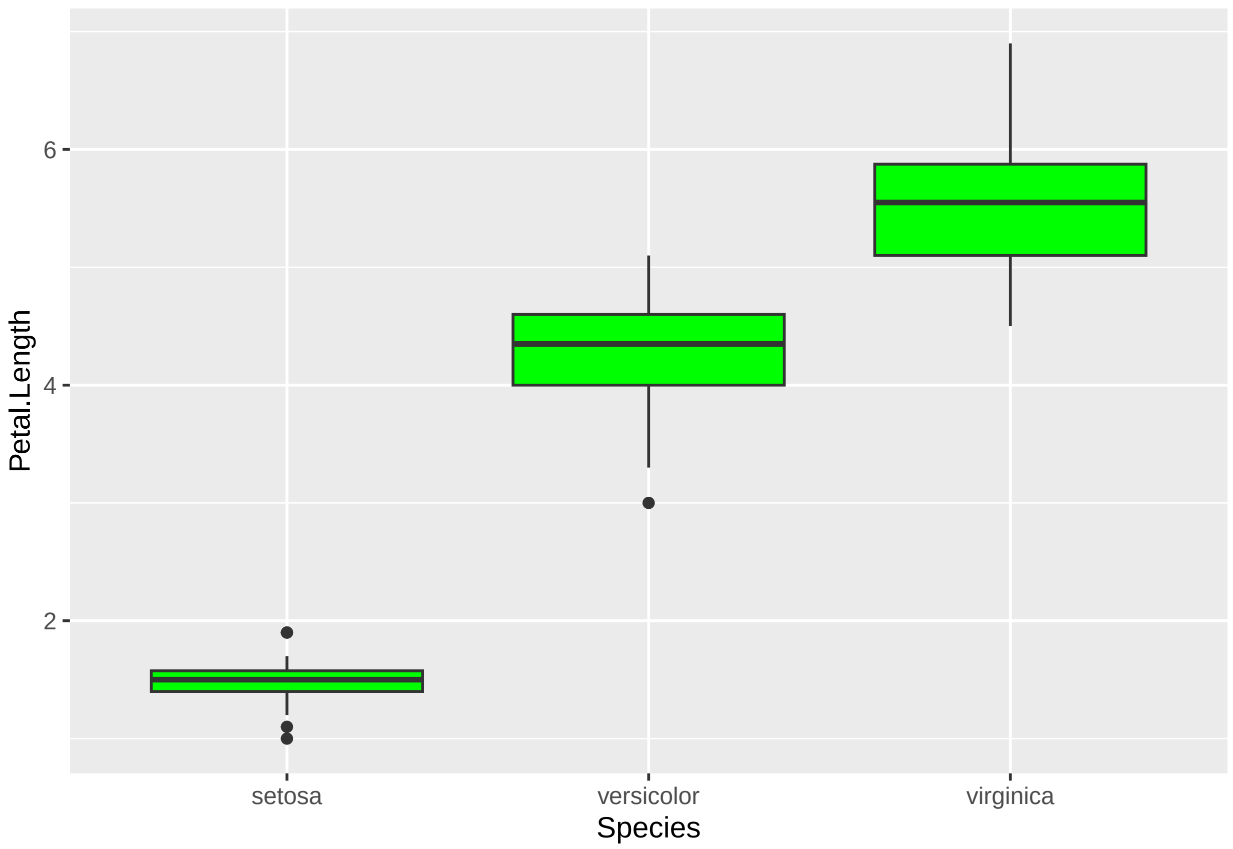
```
library(ggplot2)
ggplot(iris)+geom_boxplot(aes(x=Species,y=Sepal.Length),fill='red')
```



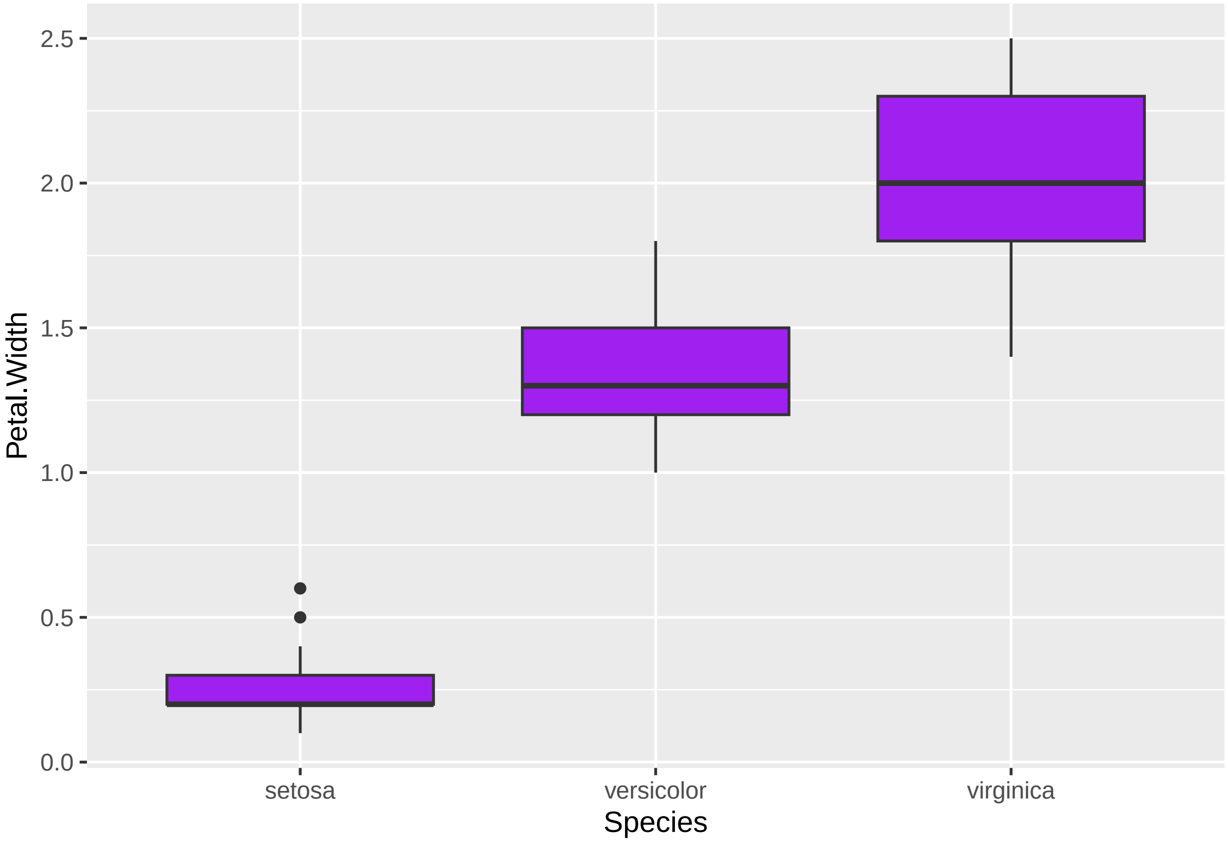
```
ggplot(iris)+geom_boxplot(aes(x=Species,y=Sepal.Width),fill='blue')
```



```
ggplot(iris)+geom_boxplot(aes(x=Species,y=Petal.Length),fill='green')
```



```
ggplot(iris)+geom_boxplot(aes(x=Species,y=Petal.Width),fill='purple')
```



The virginica species exhibits the biggest difference in petal width, and also for petal length

2.2 Problem 2

```
data(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
```

```
fivenum(trees$Girth)
```

```
## [1]  8.30 11.05 12.90 15.25 20.60
```

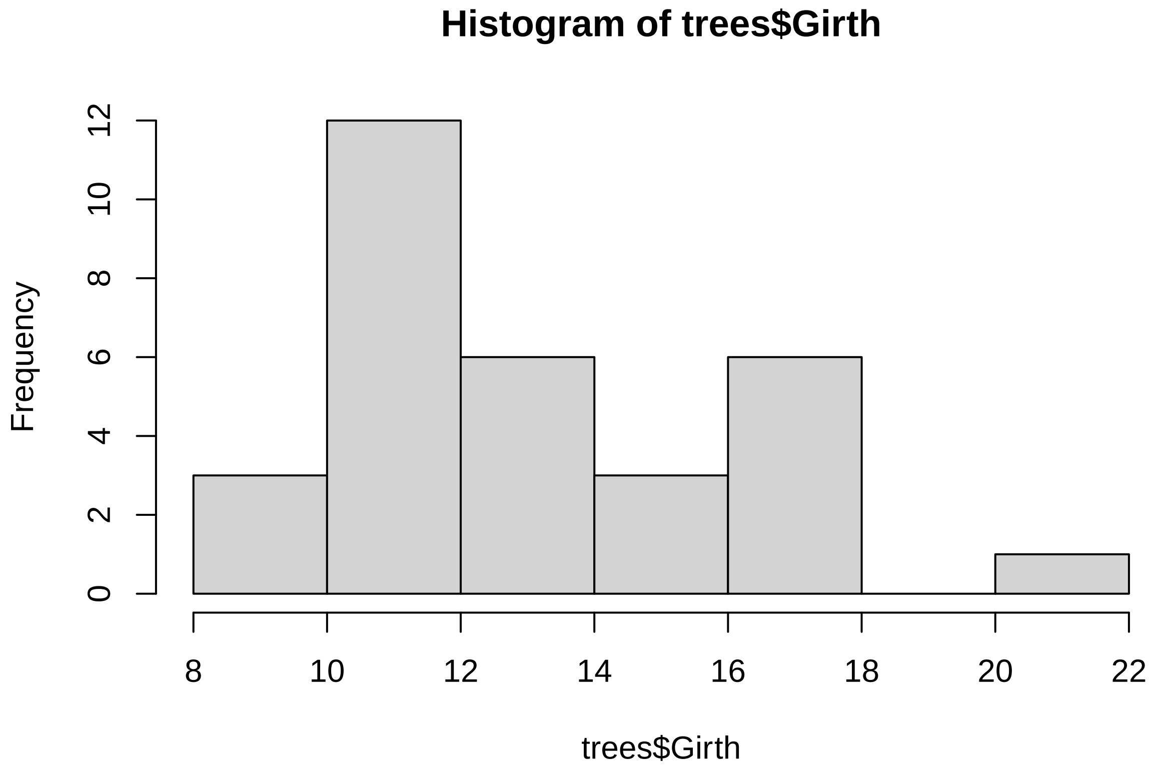
```
fivenum(trees$Height)
```

```
## [1] 63 72 76 80 87
```

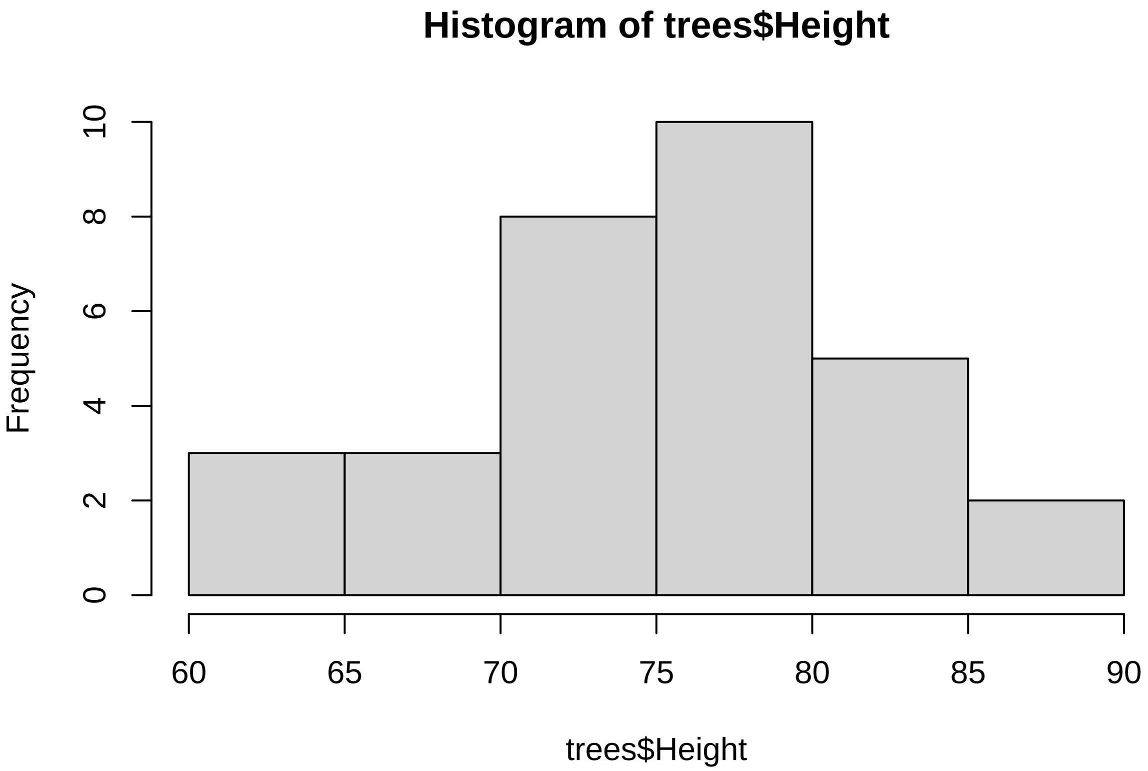
```
fivenum(trees$Volume)
```

```
## [1] 10.2 19.4 24.2 37.3 77.0
```

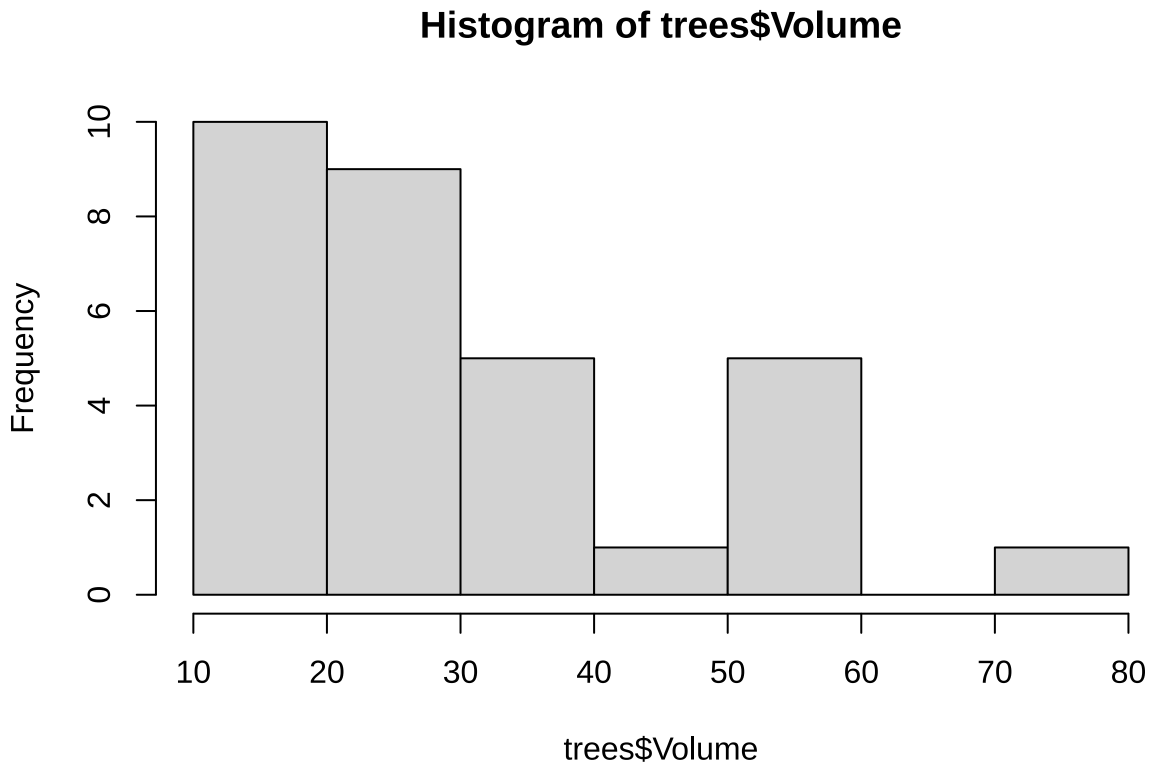
```
hist(trees$Girth)
```



```
hist(trees$Height)
```



```
hist(trees$Volume)
```



The tree heights appears to be the normally distributed variable, while girth and volume appear to be positively skewed

```
library(moments)
skewness(trees$Girth)
```

```
## [1] 0.5263163
```

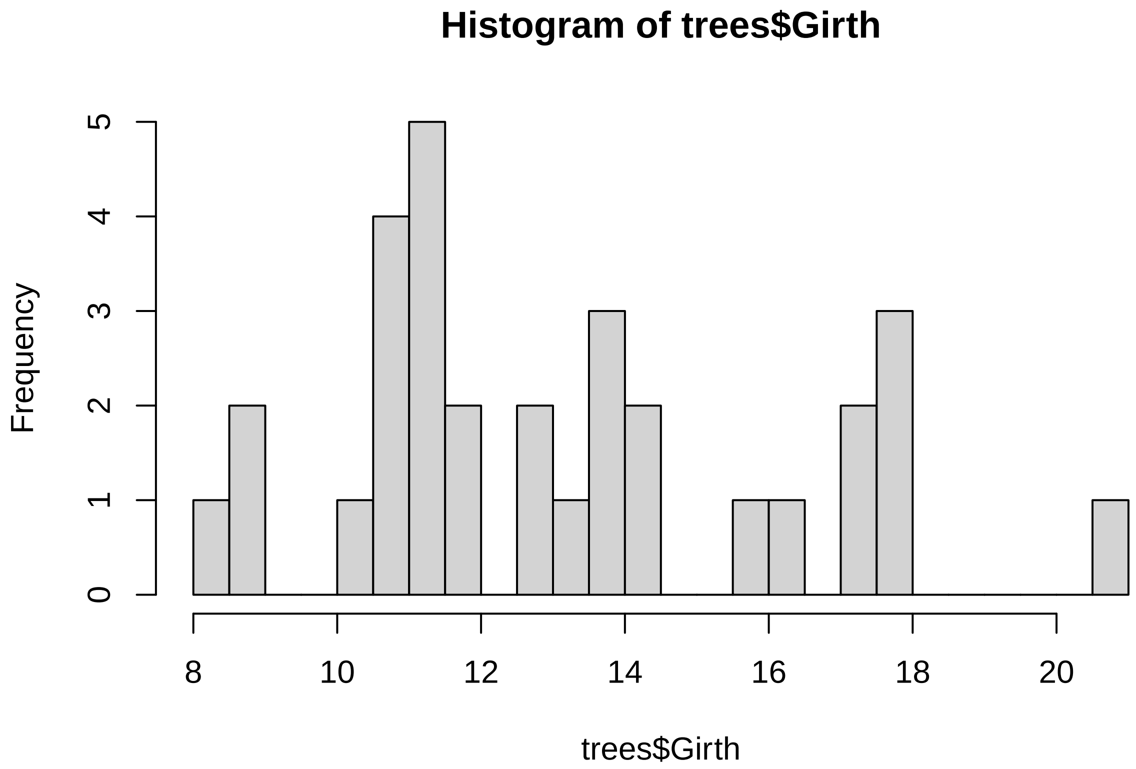
```
skewness(trees$Height)
```

```
## [1] -0.374869
```

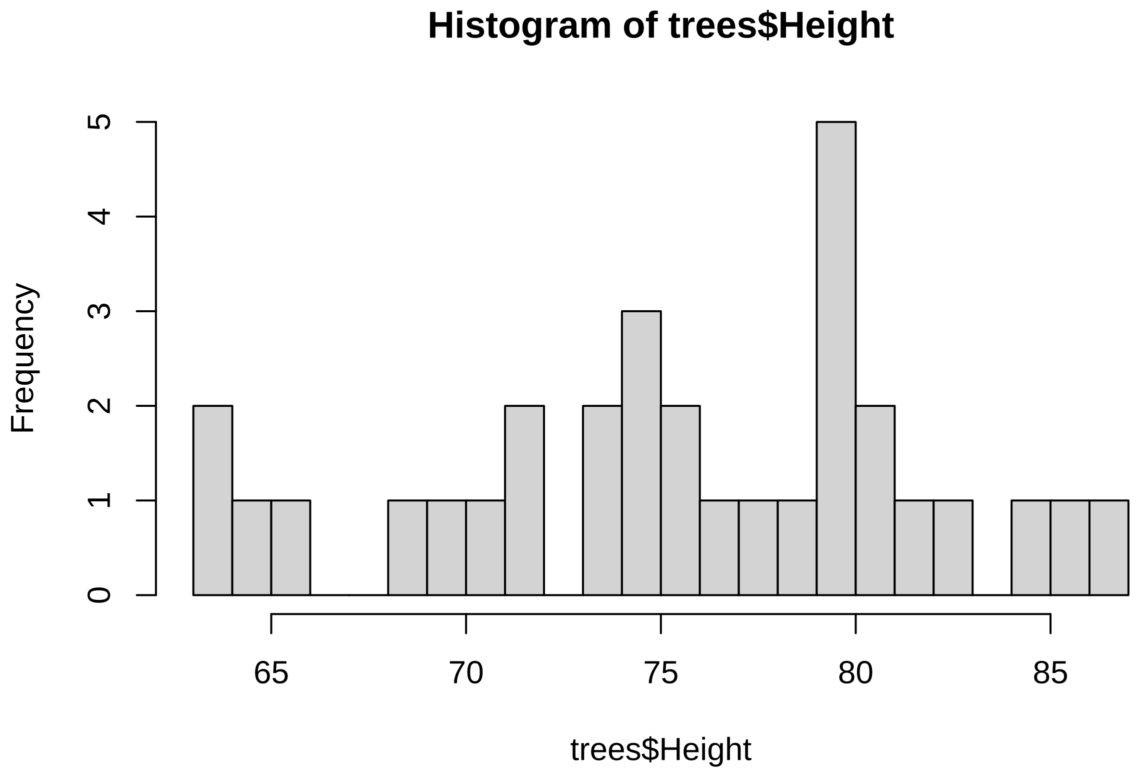
```
skewness(trees$Volume)
```

```
## [1] 1.064357
```

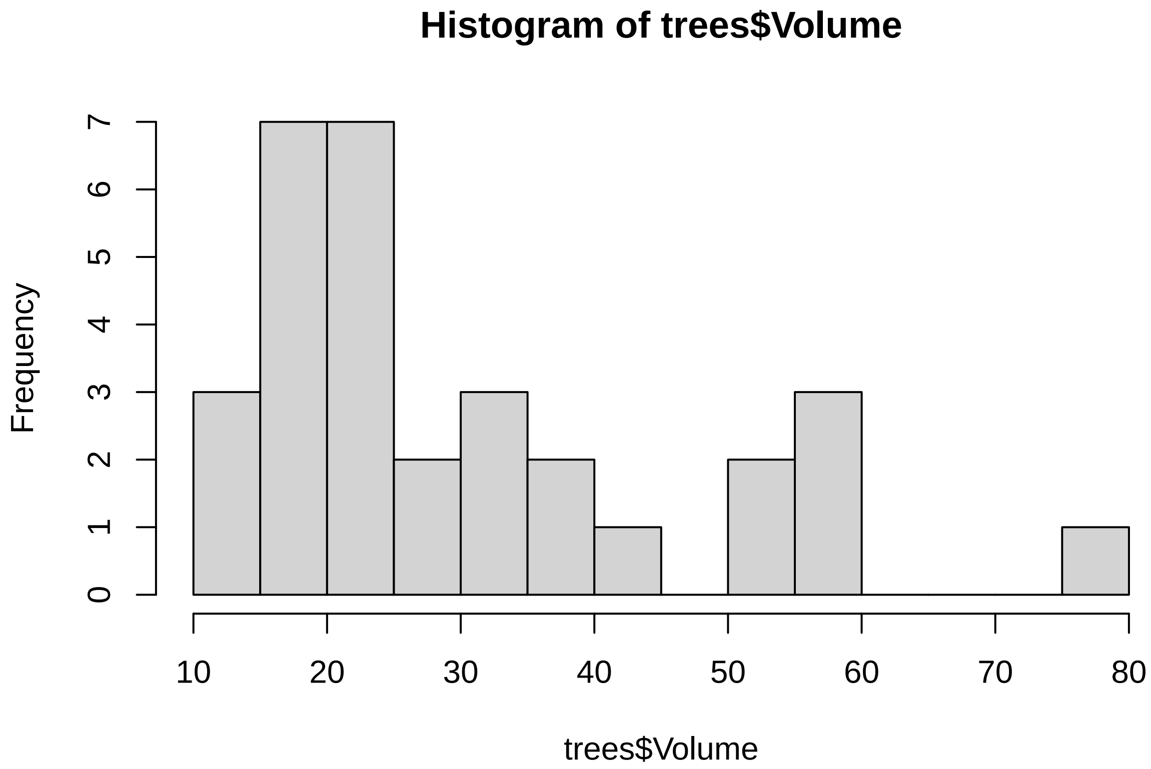
```
hist(trees$Girth,breaks = 20)
```



```
hist(trees$Height,breaks = 20)
```



```
hist(trees$Volume,breaks = 20)
```



In terms of the girth and volume the calculations align with the visual inspection and the calculations show a positive skew. The height however is calculated to have a slight negative skew which did not align with the original histogram with default values, but when increasing the breaks we see that indeed the skew is negative.

2.3 Problem 3

```
auto_df<-read.csv(url('https://archive.ics.uci.edu/ml/machine-learning-databases/auto-mpg/auto-mpg.data

# https://www.tutorialspoint.com/how-to-replace-na-values-in-columns-of-an-r-data-frame-form-the-mean-o
# ^^^ where I had to look to see how to calculate median and mean with NA values, and how to replace NA


# convert to numeric
auto_df$horsepower<-as.numeric(auto_df$horsepower)
```

```
## Warning: NAs introduced by coercion
```

```
# display current mean ignoring NA
mean(auto_df$horsepower,na.rm = T)
```

```
## [1] 104.4694
```

```
#display current median ignoring NA
median(auto_df$horsepower,na.rm=T)
```

```
## [1] 93.5
```

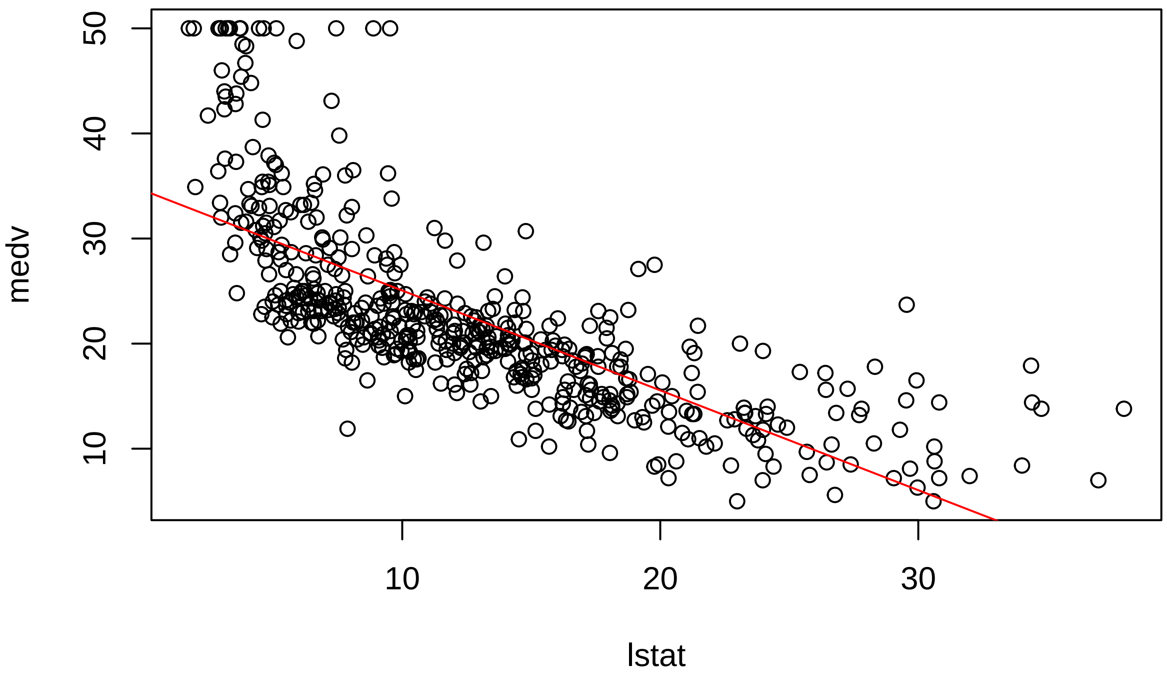
```
# replace NA in horsepower with current median
auto_df$horsepower[is.na(auto_df$horsepower)]<-median(auto_df$horsepower,na.rm=TRUE)

#display new mean of horsepower now that NA's are replaced with median
mean(auto_df$horsepower)
```

```
## [1] 104.304
```

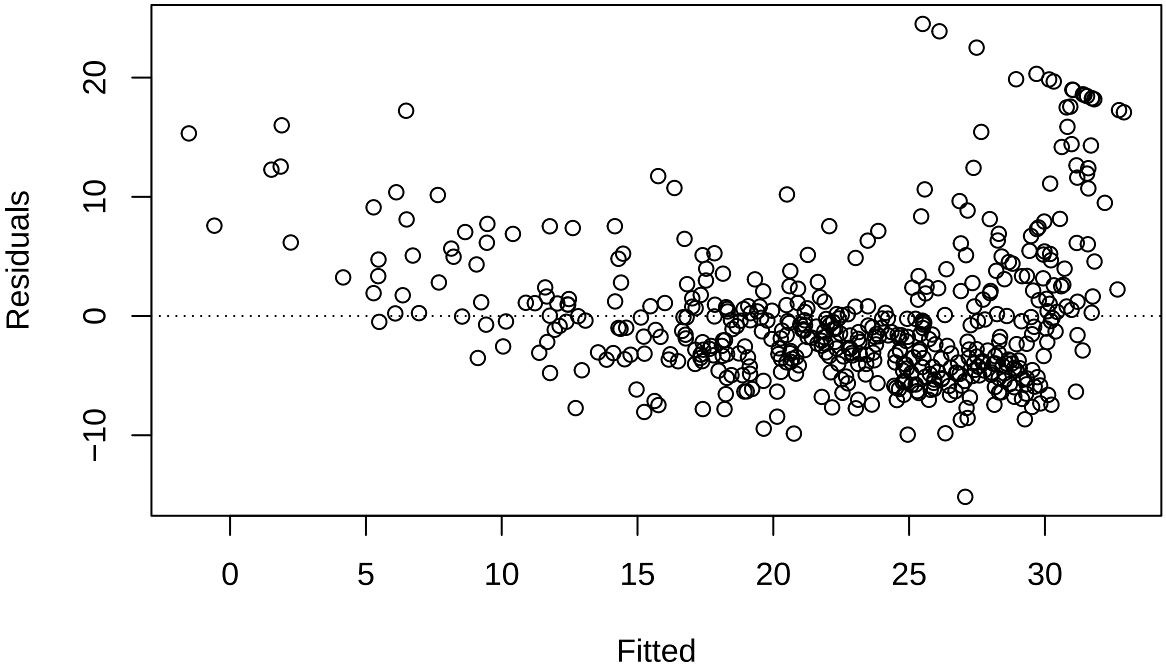
As shown the new mean is smaller but by a very small difference. This is likely due to the addition of a mere 6 points that were smaller than the previous mean which caused the mean to drop slightly.

2.4 Problem 4

```
library(MASS)
boston_df<-Boston
boston_fit<-lm(medv~lstat,data=boston_df)
plot(boston_df$lstat,boston_df$medv, xlab='lstat',ylab='medv')
abline(boston_fit,col='red')
```



```
plot(fitted(boston_fit),residuals(boston_fit),xlab='Fitted',ylab='Residuals')
abline(0,0,lty=3)
```



Yes, there is a possible non-linear relationship

```
boston_conf_int<-predict(boston_fit,data.frame(lstat=c(5,10,15)),interval = 'c')
boston_pred_int<-predict(boston_fit,data.frame(lstat=c(5,10,15)),interval = 'p')
boston_conf_int
```

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

```
boston_pred_int
```

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

The prediction intervals are much wider at 95% interval, and it makes sense because the confidence interval is giving a window where for the given points there is a 95% chance you will find the output in the current set of data, on the other hand the prediction interval is giving a 95% chance of where new response points would be given the new input points.

```
boston_fit_nonline<-lm(medv~lstat+I(lstat*lstat),data=boston_df)
summary(boston_fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat, data = boston_df)
##
## 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
```

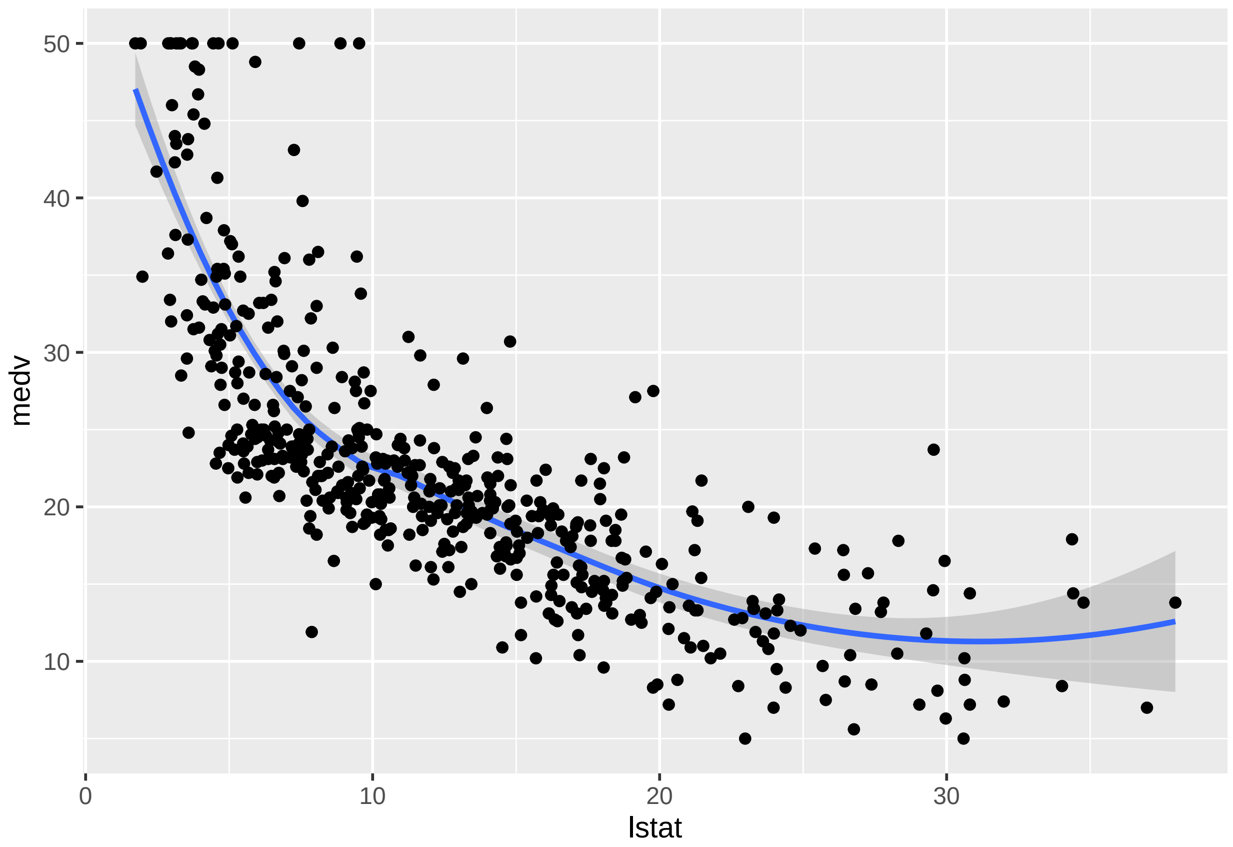
```
summary(boston_fit_nonline)
```

```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat * lstat), data = boston_df)
##
## 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 * lstat)  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
```

The adjusted R^2 increased with the addition of the squared term. The first fit had R^2 = 0.5432, and the fit with the squared term had R^2=.6393.

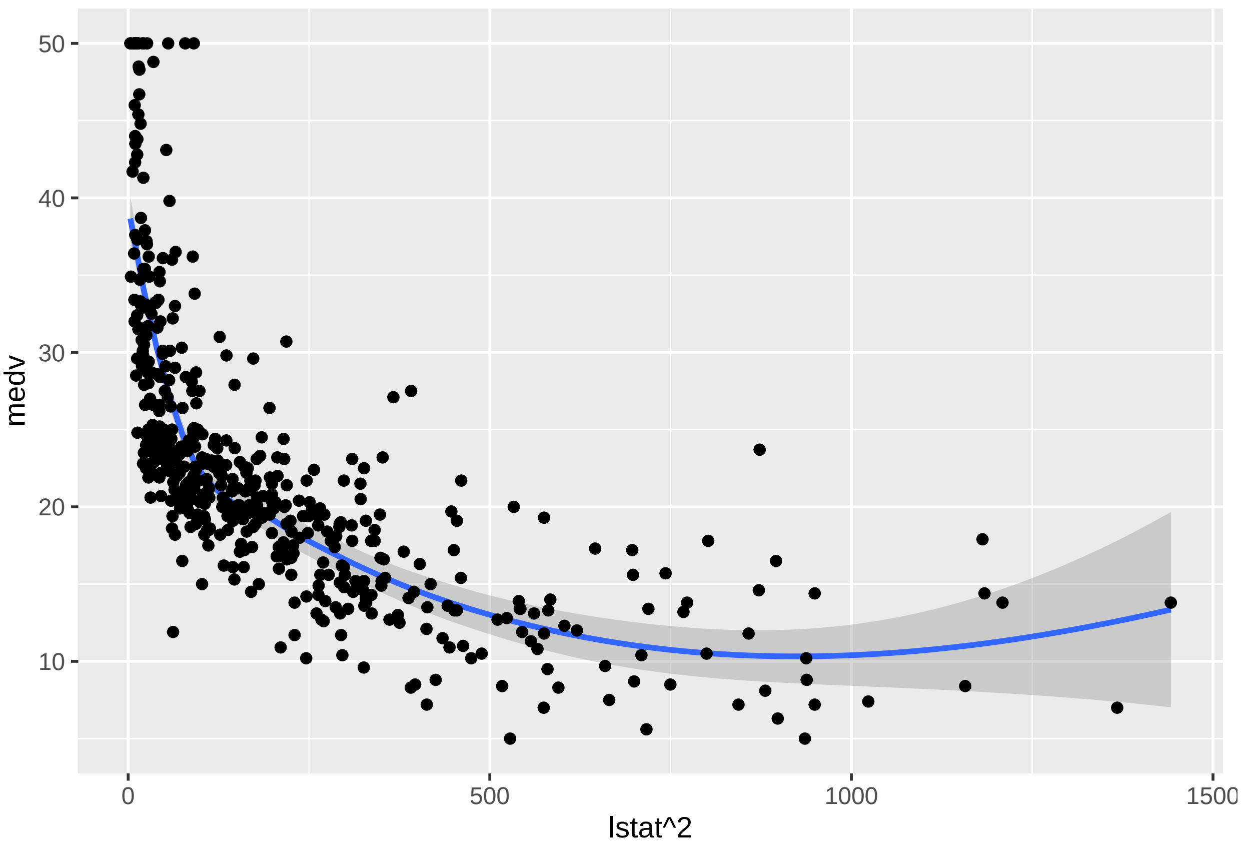
```
ggplot(data = boston_df,aes(x=lstat,y=medv))+stat_smooth()+geom_point()
```

```
## ‘geom_smooth()‘ using method = ’loess’ and formula = ’y ~ x’
```



```
ggplot(data = boston_df,aes(x=lstat**2,y=medv))+stat_smooth()+geom_point()
```

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



I guess this is what I was asked to do. I see that the predictor and response certainly have some non-linear relationship, but I guess all it asked to do was plot relationship so here it is.