Tom Meagher
DS 502 Homework 1
September 20, 2016

1. Section 2.4, page 52, question 1

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

When the sample size is extremely large, and the number of predictors is small, a flexible method would generally perform the best. This is because a flexible method can handle variance between data points a lot better than an inflexible method. In contrast, an inflexible method would likely produce very high bias—unless the points are truly lineage. Furthermore, if the number of predictors are small, an inflexible method, like multiple linear regression, does not have many features to use in it's model.

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

If the number of predictors is extremely large, an inflexible method would generally perform better than a flexible one. Since an inflexible method has lower bias when there are a few number of observations, its accuracy will better. In contrast, a flexible method would have high variation if the number of observations is small. For example, if another point was added to the data set (or an entirely different training set was used), the flexible method would likely not be able to fit the point very well without being at risk of overfitting.

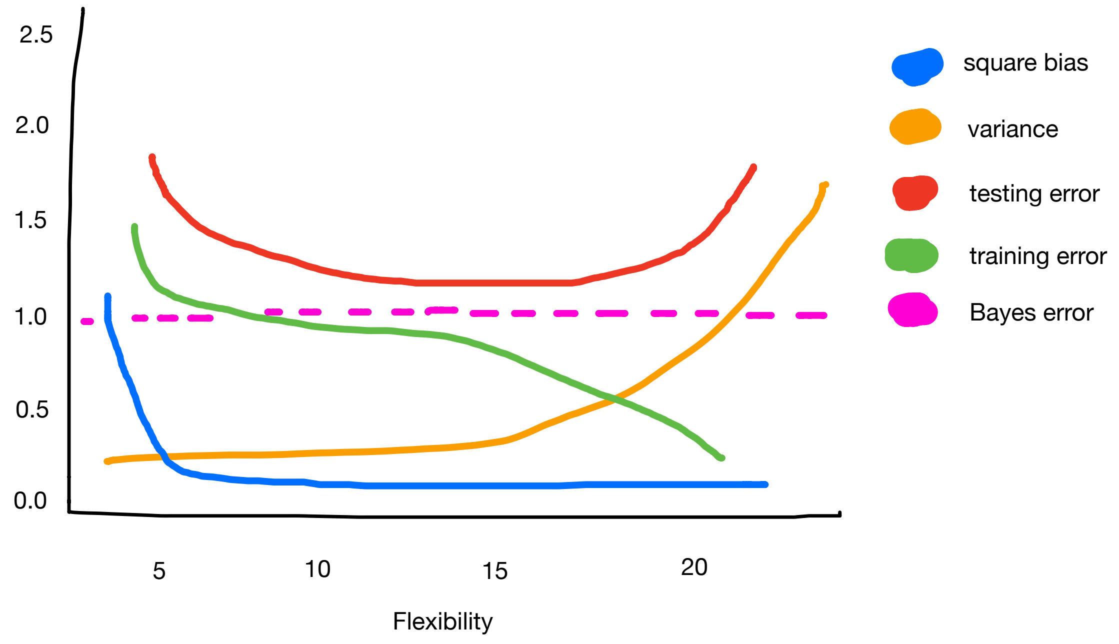
c. The relationship between the predictors and response is highly non-linear.

Since the relationship between the predictors and response is highly non-linear, a flexible method will be better than an inflexible one. This is because inflexible methods, like linear regression, would produce results that are inaccurate due to high bias (high error) by forcing the data to fit a linearity. In contrast, a flexible method is well-suited for highly non-linear data because it can fit the non-linearities—minimize the error to all the points in the data set—quite well.

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

In this case, we would expect the performance of an inflexible statistical learning method to be better than an flexible method because more flexible methods have high variance. For example, since the the data points are spread out, producing high variance, a flexible method would likely track error between the points, which leads to poor performance due to overfitting.

2. Section 2.4, page 52-53, question 3



b. Explain why each of the five curves has the shape displayed in part a.

The training error decreases at the flexibility increases because the model is fit to the training data. In contrast the testing error is a *U* shape because as flexibility initially increases the test data fits quite well, but as flexibility continues to increase there is a danger of overfitting so the error increases. Bates error remains constant for the entire model and is independent, thus the straight horizontal line. As the bias-variance trade-off suggests bias, as flexibility increases, the square bias decreases. In contrast to square bias, variance increases when the flexibility of the model increases.

3. Section 2.4, page 53, question 6

Parametric methods make assumptions about the functional form of *f*, while non-parametric methods do not make any explicit assumptions about *f*. The most common assumption for the parametric method is a linear model, but it could be another type. Next, training data is used to fit the model (estimate *f* or its unknown parameters) with least squares or another approach. Since non-parametric methods do not make any explicit assumptions about *f*, they accurately fit a wider range of possible shapes. For example, if the model is not assumed to be linear, then it is more flexible if the data is non-linear. The disadvantage is because there is not an assumption about the model, a lot of data is needed to accurately estimate *f*. Intuitively, this also makes sense because with less data points and no assumption, the model will overfit and track the random noise.

4. Section 2.4, page 54-55, question 8

a. read College data set

```
college_data <- read.csv(file="data/College.csv", header=TRUE, sep=",")
```

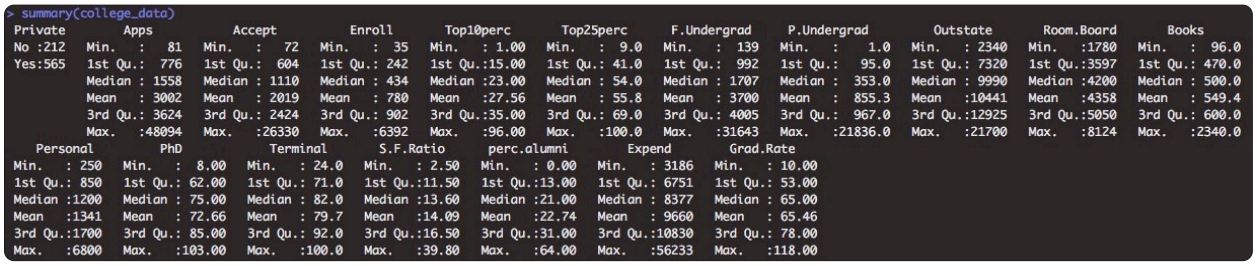
b.

```
rownames(college_data)=college_data[,1]
fix(college_data)

college_data=college_data[,-1]
fix(college_data)
```

c.

i.

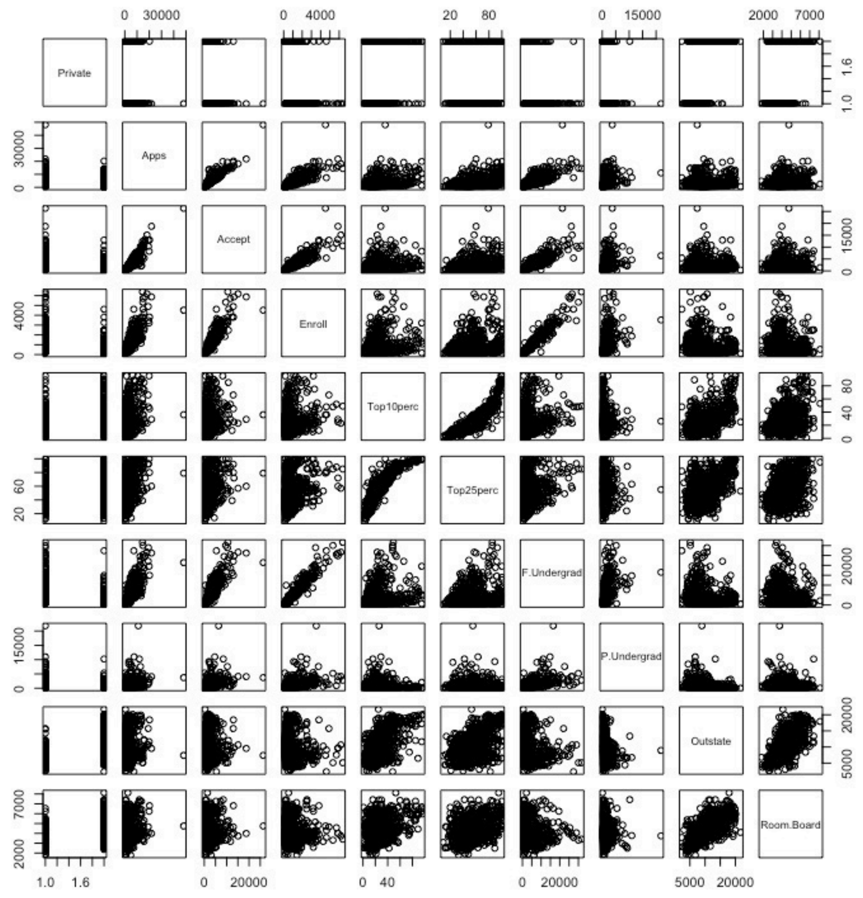
```
summary(college_data)
```



```
> summary(college_data)
 Private        Apps           Accept          Enroll       Top10perc       Top25perc      F.Undergrad     P.Undergrad        Outstate       Room.Board       Books       
 No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00   Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340   Min.   :1780   Min.   :  96.0  
 Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320   1st Qu.:3597   1st Qu.: 470.0  
           Median : 1558   Median : 1110   Median : 434   Median :23.00   Median : 54.0   Median : 1707   Median :  353.0   Median : 9990   Median :4200   Median : 500.0  
           Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56   Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441   Mean   :4358   Mean   : 549.4  
           3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925   3rd Qu.:5050   3rd Qu.: 600.0  
           Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00   Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700   Max.   :8124   Max.   :2340.0  
    Personal         PhD            Terminal       S.F.Ratio      perc.alumni        Expend         Grad.Rate     
 Min.   : 250   Min.   :  8.00   Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186   Min.   : 10.00  
 1st Qu.: 850   1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751   1st Qu.: 53.00  
 Median :1200   Median : 75.00   Median : 82.0   Median :13.60   Median :21.00   Median : 8377   Median : 65.00  
 Mean   :1341   Mean   : 72.66   Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660   Mean   : 65.46  
 3rd Qu.:1700   3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830   3rd Qu.: 78.00  
 Max.   :6800   Max.   :103.00   Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233   Max.   :118.00  
```

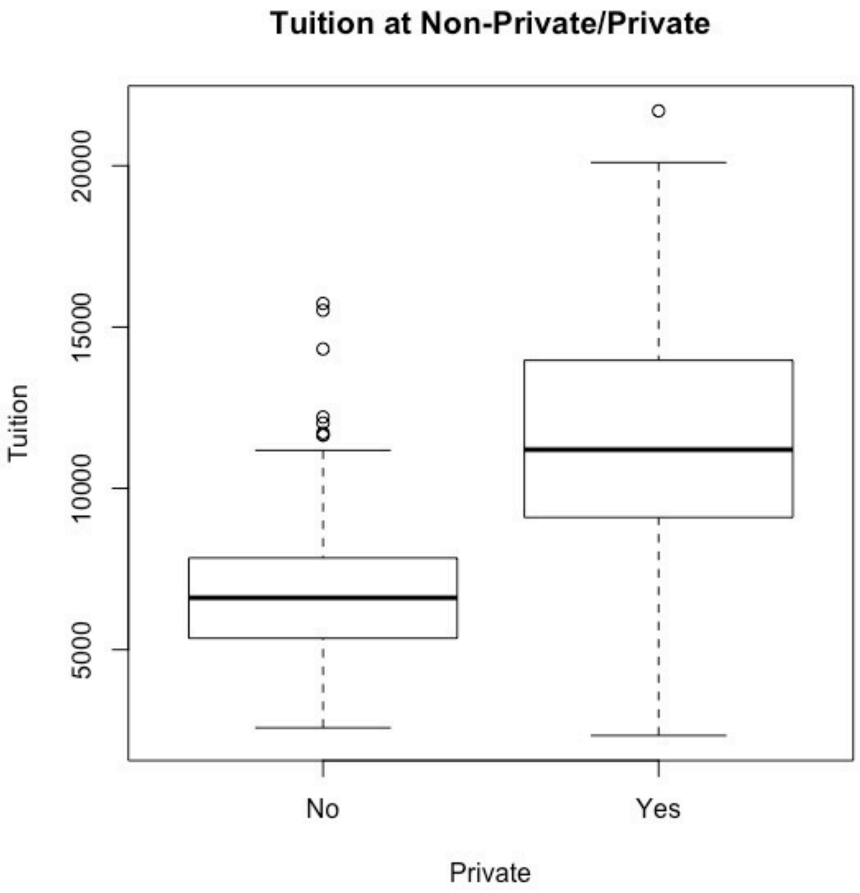
ii.

```
pairs(college_data[
,1:10])
```



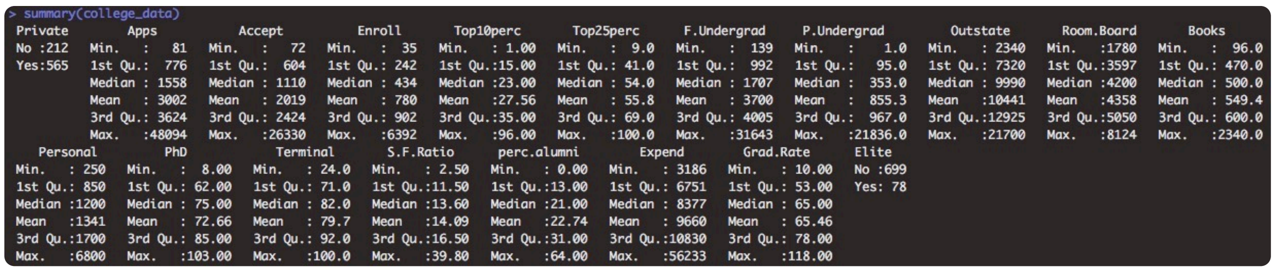
iii.

```
attach(college_data)
Private=as.factor(Private)
plot(Private, Outstate, xlab="Private", ylab="Tuition",
    main="Tuition at Non-Private/Private")
```

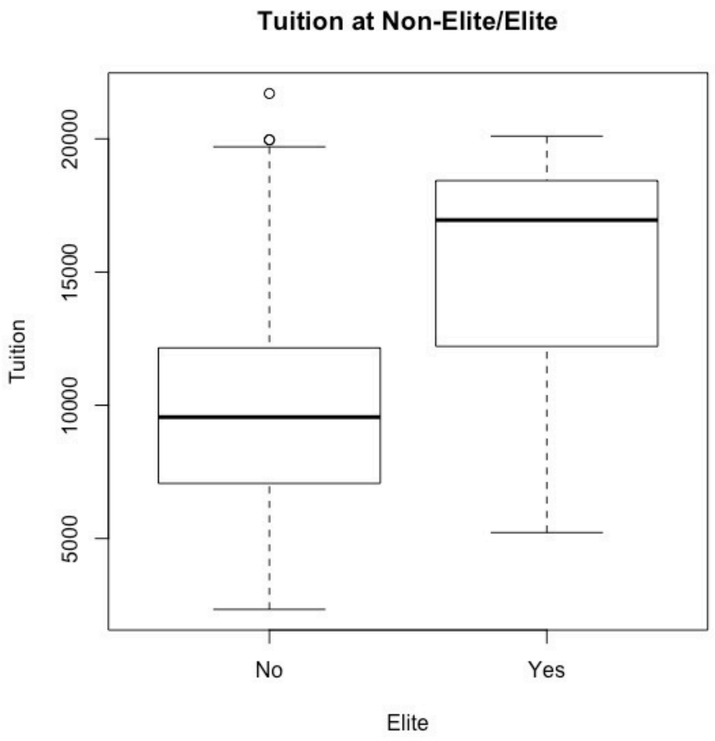


iv.

```
Elite=rep("No",nrow(college_data))
Elite[Top10perc>50]="Yes"
Elite=as.factor(Elite)
college_data=data.frame(college_data, Elite)
summary(college_data)
plot(Elite, Outstate, xlab="Elite", ylab="Tuition",
    main="Tuition at Non-Elite/Elite")
```

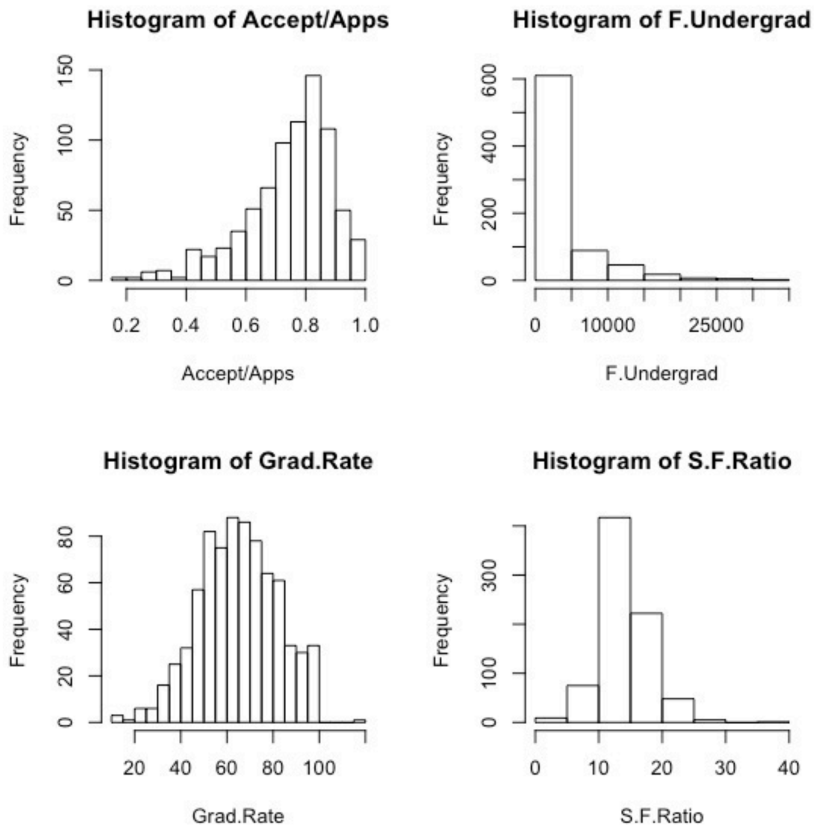


```
> summary(college_data)
 Private        Apps           Accept          Enroll       Top10perc       Top25perc      F.Undergrad     P.Undergrad          Outstate       Room.Board       Books       
 No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00   Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340   Min.   :1780   Min.   :  96.0  
 Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00   1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320   1st Qu.:3597   1st Qu.: 470.0  
           Median : 1558   Median : 1110   Median : 434   Median :23.00   Median : 54.0   Median : 1707   Median :  353.0   Median : 9990   Median :4200   Median : 500.0  
           Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56   Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441   Mean   :4358   Mean   : 549.4  
           3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00   3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925   3rd Qu.:5050   3rd Qu.: 600.0  
           Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00   Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700   Max.   :8124   Max.   :2340.0  
    Personal         PhD             Terminal        S.F.Ratio      perc.alumni        Expend         Grad.Rate      Elite    
 Min.   : 250   Min.   :  8.00   Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186   Min.   : 10.00   No :699  
 1st Qu.: 850   1st Qu.: 62.00   1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751   1st Qu.: 53.00   Yes: 78  
 Median :1200   Median : 75.00   Median : 82.0   Median :13.60   Median :21.00   Median : 8377   Median : 65.00            
 Mean   :1341   Mean   : 72.66   Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660   Mean   : 65.46            
 3rd Qu.:1700   3rd Qu.: 85.00   3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830   3rd Qu.: 78.00            
 Max.   :6800   Max.   :103.00   Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233   Max.   :118.00            
```

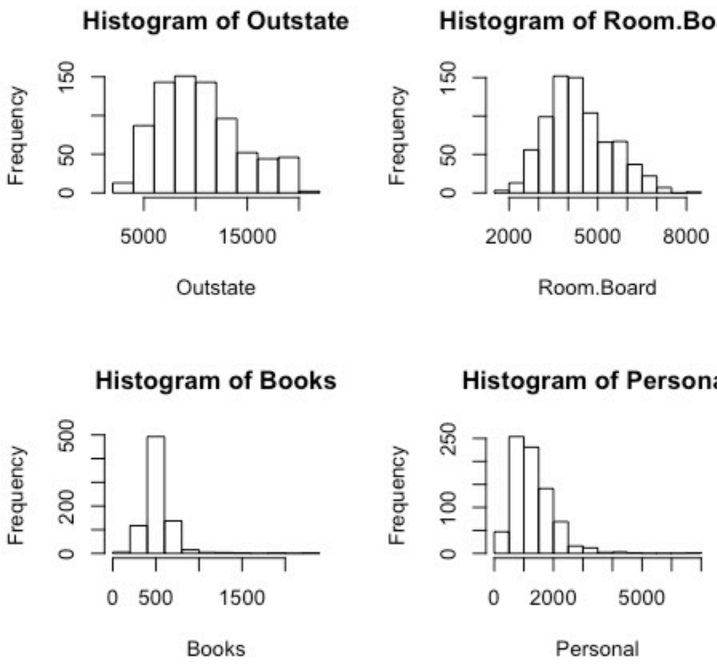


v.

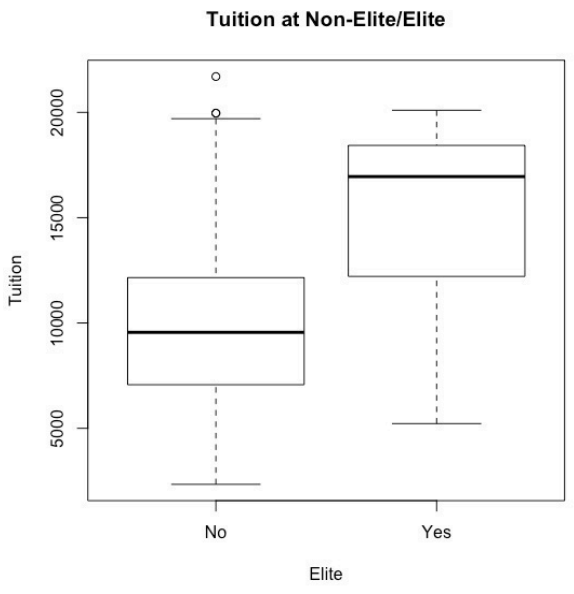
```
par(mfrow=c(2,2))
hist(Accept/Apps,
breaks=15)
hist(F.Undergrad,
breaks=5)
hist(Grad.Rate,
breaks=30)
hist(S.F.Ratio,
breaks=10)
```



vi.
Costs across outstate, room and board, books, and personal expenses look to be fairly normally distributed. If anything though, they are a bit skewed to the right with a longer tail on the right side than left.



Looking into the outstate cost chart a little more (because this seems to be the highest average cost), we divide it into elite and non-elite. While there are a couple outliers in the non-elite data, the five-number summary comparison provided by the box plot is very interesting. For example, the upper adjacent values for the elite and non-elite are nearly the same, but the first quartile, median, and third quartile are very different. In fact, the first quartile of the elite schools is roughly equal to the third quartile of the non-elite data. Perhaps the non-elite data was very sensitive to the to outliers. Brining this back to a real-world perspective, we can attempt to hypothesize about why outstate tuition in costs more at elite versus non-elite schools. It could be that elite schools can charge a premium to attend because the quality justifies the price. If we look at the number of acceptances at elite schools, it is comparatively much lower than that of non-elite. One last thing to consider, when we are doing comparisons between both groups is the sample size, or how many data points you have to work with. In this case there are only 78 elite schools versus 699 non-elite schools. With a larger number of elite schools the chart could change drastically.



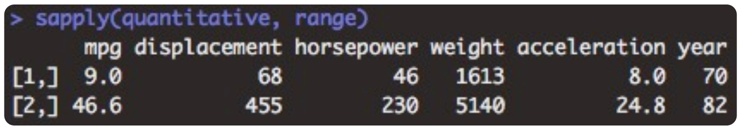
5. Section 2.4, page 56, question 9

a.

Mpg, displacement, horsepower, weight, acceleration, and year are contain numeric values, therefore they are quantitative predictors. In contrast, cylinders, origin, and name are qualitative. Even though cylinders and origin contain numeric values, there are only a small number of possible values for each so I am choosing to represent them as such.

b.

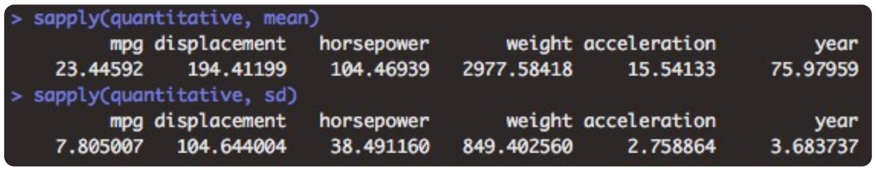
```
# put all quantitative predictors into data frame
quantitative <- subset(auto_data, select = c(mpg, displacement, horsepower, weight,
acceleration, year))
sapply(quantitative, range)
```



```
> sapply(quantitative, range)
      mpg displacement horsepower weight acceleration year
[1,]  9.0           68         46   1613          8.0   70
[2,] 46.6          455        230   5140         24.8   82
```

c.

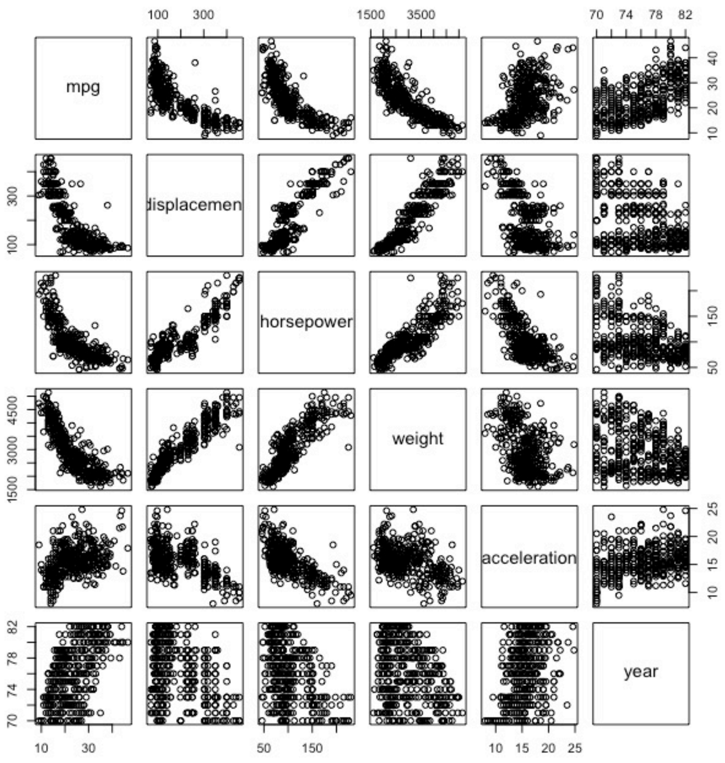
```
# use sapply on quantitative predictors data frame
sapply(quantitative, mean)
sapply(quantitative, sd)
```



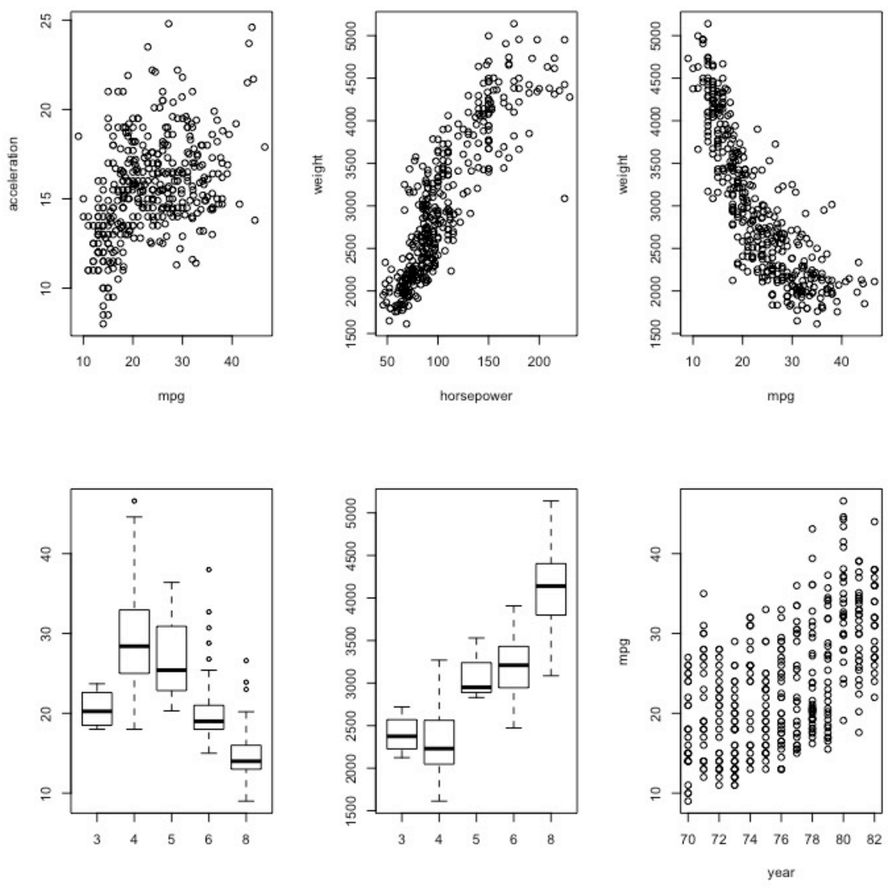
```
> sapply(quantitative, mean)
       mpg displacement   horsepower       weight acceleration         year
  23.44592    194.41199    104.46939   2977.58418     15.54133     75.97959
> sapply(quantitative, sd)
       mpg displacement   horsepower       weight acceleration         year
  7.805007   104.644004    38.491160   849.402560     2.758864     3.683737
```

d.

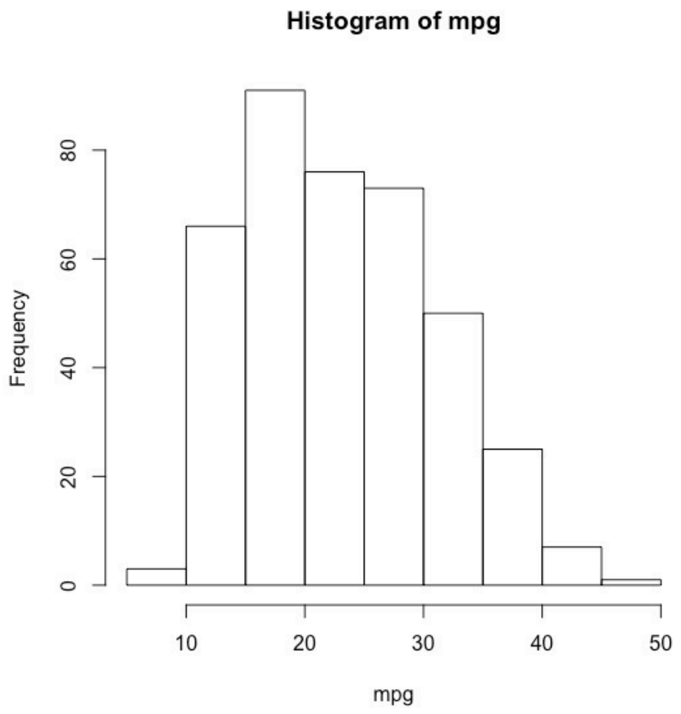
```
pairs(quantitative)
```



```
par(mfrow=c(2,3))
plot(mpg,
acceleration)
plot(horsepower,
weight)
plot(mpg, weight)
plot(cylinders,
mpg)
plot(cylinders,
weight)
plot(year, mpg)
```



```
hist(mpg)
```



f.
Yes, the scatterplot matrix does a good job of suggesting that other variables might be useful in predicting mpg because you can see the strength and directionality of the relationship between mpg and the predictors. Based on the matrix, it looks like the quantitative variables of displacement, horsepower, and weight might be useful as they all look like they have strong negative correlation to mpg.

6. Section 3.7, page 120, question 1

The null hypothesis, to which the p-values in Table 3.4 correspond, is checking whether all of the predictors' coefficients in the regression are zero. Based on the p-values, and a 95% confidence level (alpha = 0.05), TV (<0.0001) and radio (<0.0001) are both significant, while newspaper (0.8599) is not because it is larger than alpha. Since TV and radio are significant, this means that isolating one of them and holding all the other predictors constant results in a $46 and $189 increase in sales respectively for every $1,000 spent (on average). In contrast, since newspaper is not significant in this model, it could be removed, but then the model must be re-run.

7. Section 3.7, page 121, question 5

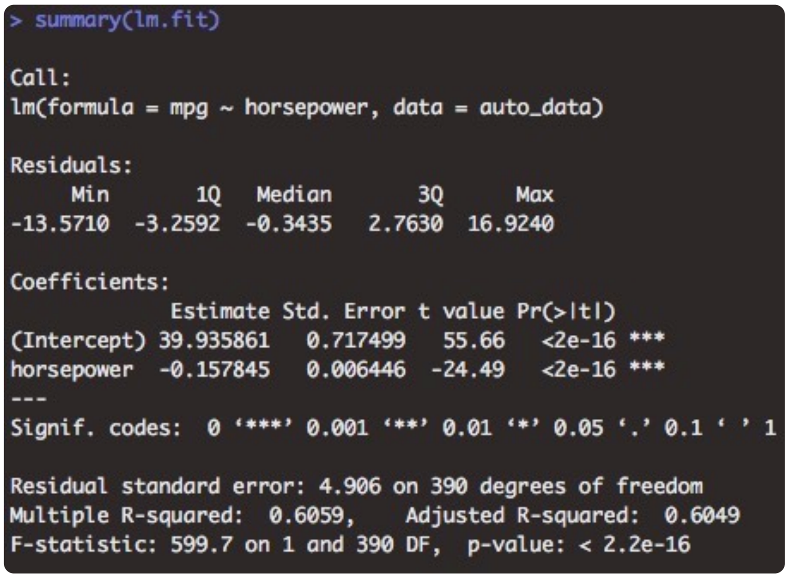
βˆ = (sum i=1..n xi yi)/(sum i'=1..n x^2i')
Start: yˆ i = x i βˆ
Substitute βˆ: yˆ i = x i (sum i=1..n xi yi)/(sum i'=1..n x^2i')
Move x i into the sum: yˆ i = (sum i=1..n xi yi)/(sum i'=1..n x^2i')
Further reduce: aj = (xi xj)/sum xk^2
End: yˆ i = sum aj yj

8. Section 3.7, page 121, question 6

Using (3.4), the least squares line always passes the through point (x bar, y bar) because the resultant line minimizes the sum of the vertical distances for each observation. Since the fit averages the sum of minimizing the square errors, it will always pass through the average of the x and y's (x bar, y bar) for the data points.

9. Section 3.7, page 121-122, question 8

a.
```
auto_data <-
read.csv(file="data/
Auto.csv", header=TRUE,
sep=",", na.strings="?")
auto_data=na.omit(auto_data)
lm.fit=lm(mpg~horsepower,
data=auto_data)
summary(lm.fit)
```

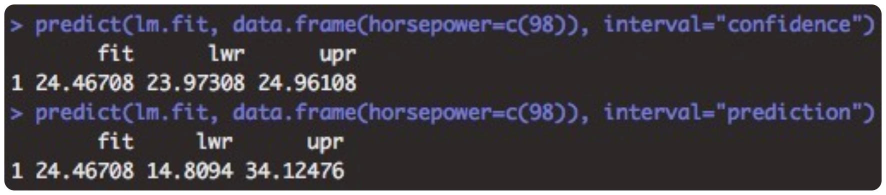

```
> summary(lm.fit)

Call:
lm(formula = mpg ~ horsepower, data = auto_data)

Residuals:
     Min       1Q   Median       3Q      Max
-13.5710  -3.2592  -0.3435   2.7630  16.9240

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 39.935861   0.717499   55.66   <2e-16 ***
horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.906 on 390 degrees of freedom
Multiple R-squared:  0.6059,    Adjusted R-squared:  0.6049
F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

i.
Yes, there is a strong relationship between the predictor and response because the residual standard error (RSE), 4.906, is less than 10%. Also, the coefficient of determination is 0.6059 – fairly close to one, meaning a decent amount of the variation in mpg is explained by horsepower.

ii.
Based on the class RSE guidelines, the RSE (4.906) is less than 10% so it's a very strong relationship. If the RSE is small, it means that the regression line fit the data well enough so the difference between the observed and predicted values (a.k.a. residual) were minimized, resulting in less error for the model.

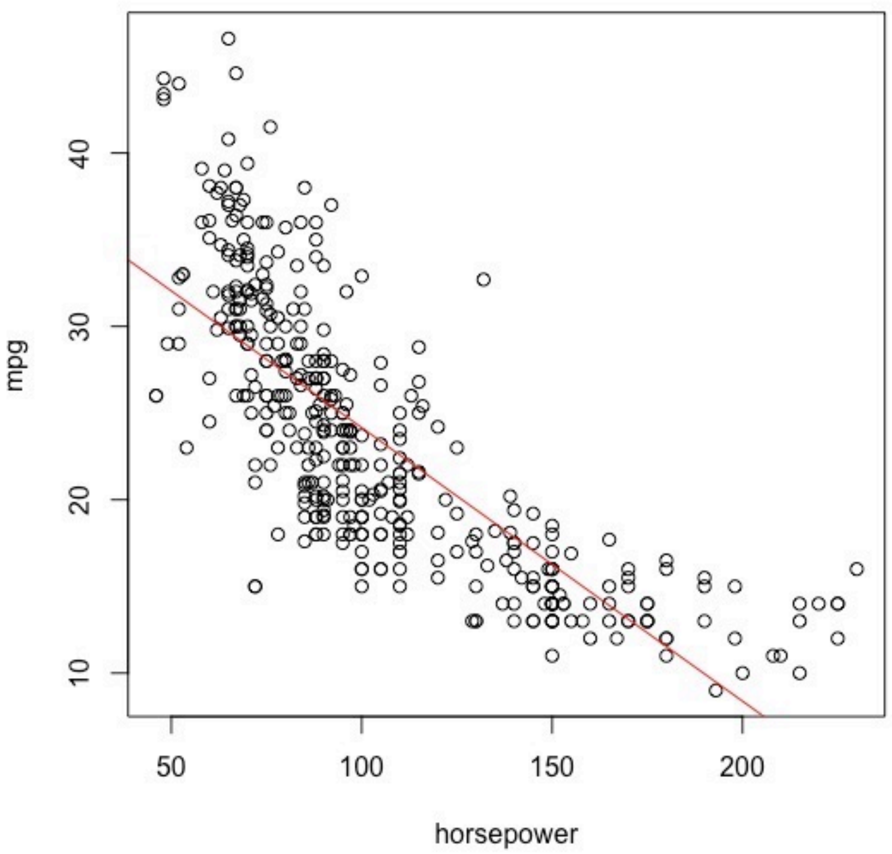
iii.
The relationship between the predictor and response is negative because the coefficient of the predictor's estimate is negative (-0.1578). This means that for every unit of horsepower that is added, mpg decreases by an average of -0.1578 miles per gallon.

iv.
```
predict(lm.fit, data.frame(horsepower=c(98)), interval="confidence")
predict(lm.fit, data.frame(horsepower=c(98)), interval="prediction")
```



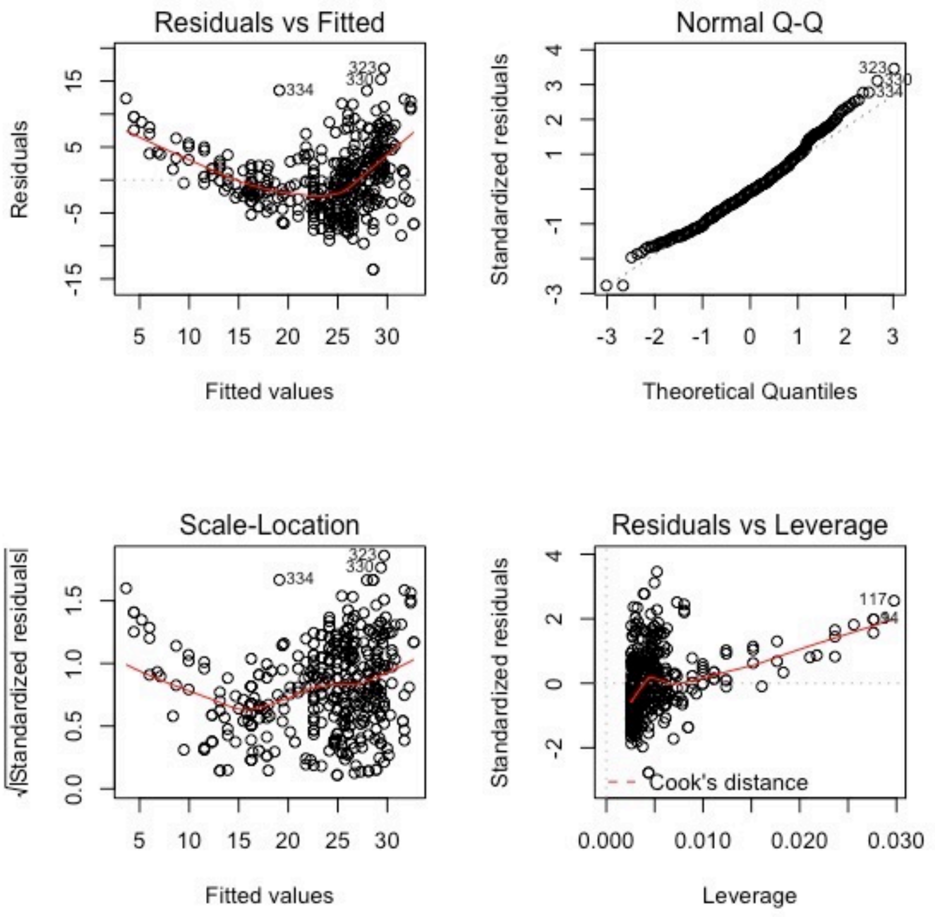
```
> predict(lm.fit, data.frame(horsepower=c(98)), interval="confidence")
       fit      lwr      upr
1 24.46708 23.97308 24.96108
> predict(lm.fit, data.frame(horsepower=c(98)), interval="prediction")
       fit     lwr      upr
1 24.46708 14.8094 34.12476
```

b.
```
plot(horsepower, mpg)
abline(lm.fit, col="red")
```



c.
```
par(mfrow=c(2, 2))
plot(lm.fit)
```

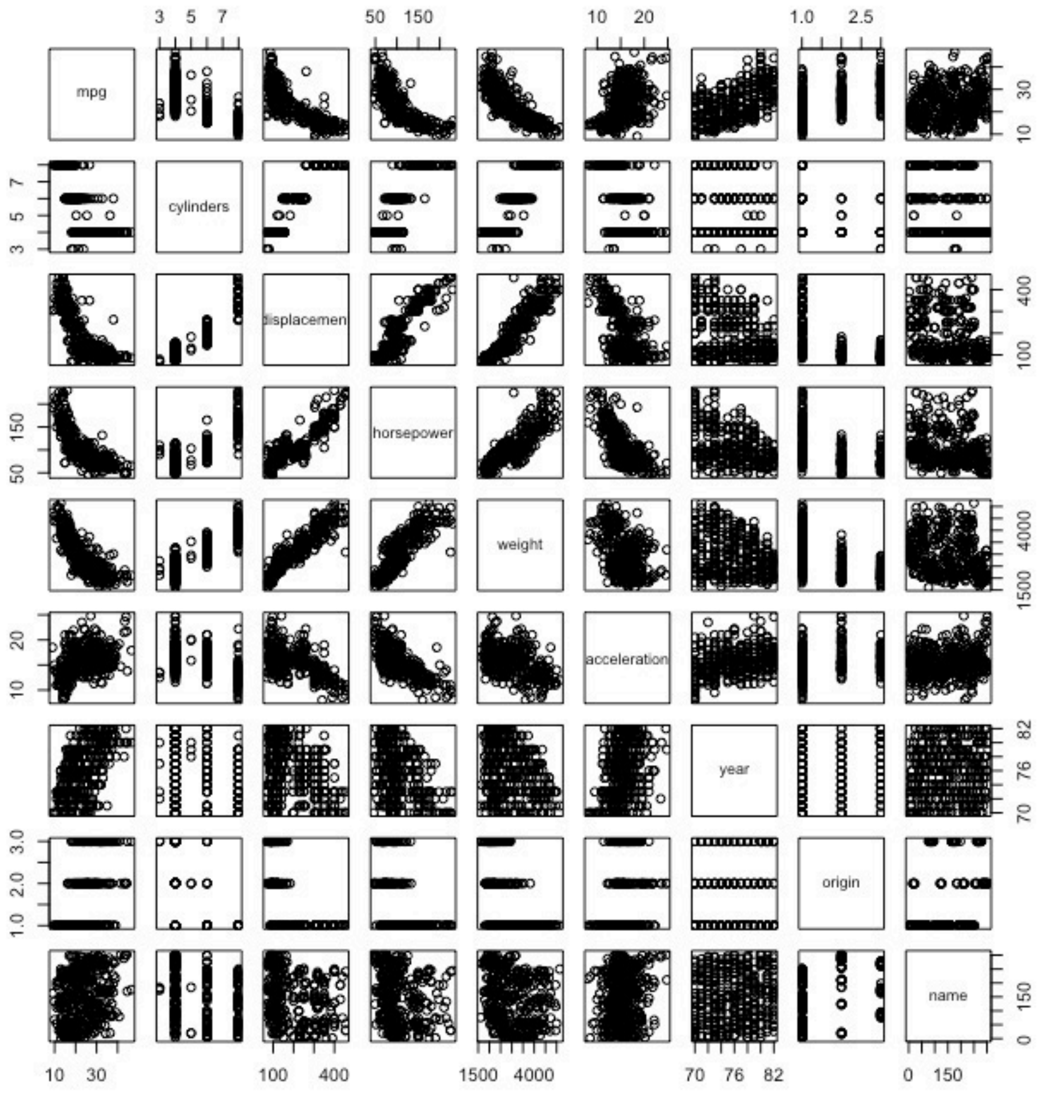
Based on the Residuals vs Fitted chart, the fit appears to be good, but also non-linear since the shape of the residuals line (in red) is roughly an *U*. Furthermore, looking at the Normal Q-Q plot confirms that the fit looks relatively normal. When assessing fit using the Scale-Location chart (with studentized residuals), all the residuals are less than 2.0, even a few points that look like outliers. Lastly, assessing leverage using the Residuals vs Leverage chart, there appear to be multiple observations that have high level that could have a sizable impact on the fit.



10. Section 3.7, page 122, question 9

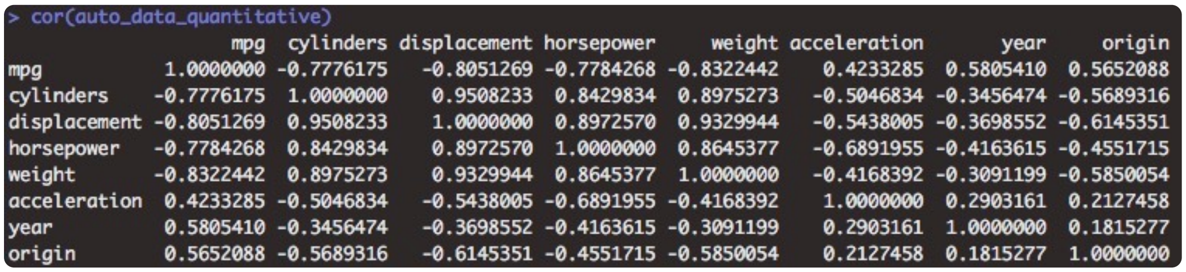
a.

```
auto_data <- read.csv(file="data/Auto.csv", header=TRUE, sep=",", na.strings="?")
auto_data=na.omit(auto_data)
pairs(auto_data)
```



b.

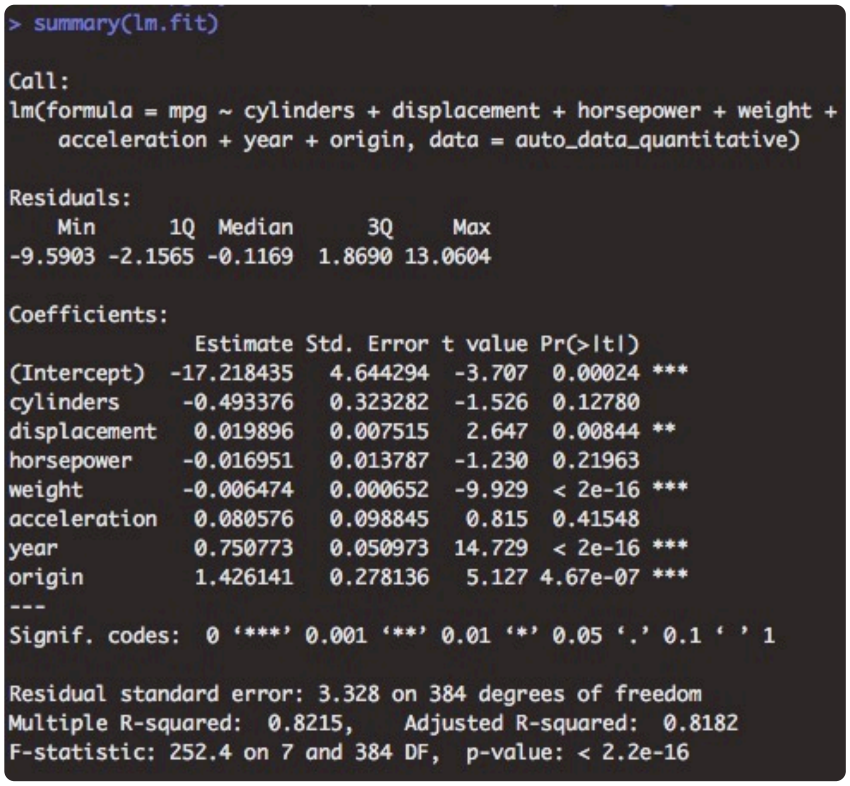
```
# treat all variables as quantitative, exclude name
auto_data_quantitative <- subset(auto_data, select = -c(name))
cor(auto_data_quantitative)
```



```
> cor(auto_data_quantitative)
                    mpg  cylinders displacement horsepower     weight acceleration       year     origin
mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442    0.4233285  0.5805410  0.5652088
cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273   -0.5046834 -0.3456474 -0.5689316
displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944   -0.5438005 -0.3698552 -0.6145351
horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377   -0.6891955 -0.4163615 -0.4551715
weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000   -0.4168392 -0.3091199 -0.5850054
acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392    1.0000000  0.2903161  0.2127458
year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199    0.2903161  1.0000000  0.1815277
origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054    0.2127458  0.1815277  1.0000000
```

c.

```
lm.fit=lm(mpg~cylinders+displacement+horsepower+weight+acceleration+year+origin,
    data=auto_data_quantitative)
summary(lm.fit)
```



```
> summary(lm.fit)

Call:
lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
    acceleration + year + origin, data = auto_data_quantitative)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5903 -2.1565 -0.1169  1.8690 13.0604

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
cylinders     -0.493376   0.323282  -1.526  0.12780
displacement   0.019896   0.007515   2.647  0.00844 **
horsepower    -0.016951   0.013787  -1.230  0.21963
weight        -0.006474   0.000652  -9.929  < 2e-16 ***
acceleration   0.080576   0.098845   0.815  0.41548
year           0.750773   0.050973  14.729  < 2e-16 ***
origin         1.426141   0.278136   5.127 4.67e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.328 on 384 degrees of freedom
Multiple R-squared:  0.8215,    Adjusted R-squared:  0.8182
F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

i.
Yes, there is a relationship between the predictors and response. If you isolate one predictor and hold the others constant, then for an unit increase in that variable, mpg will likely change.

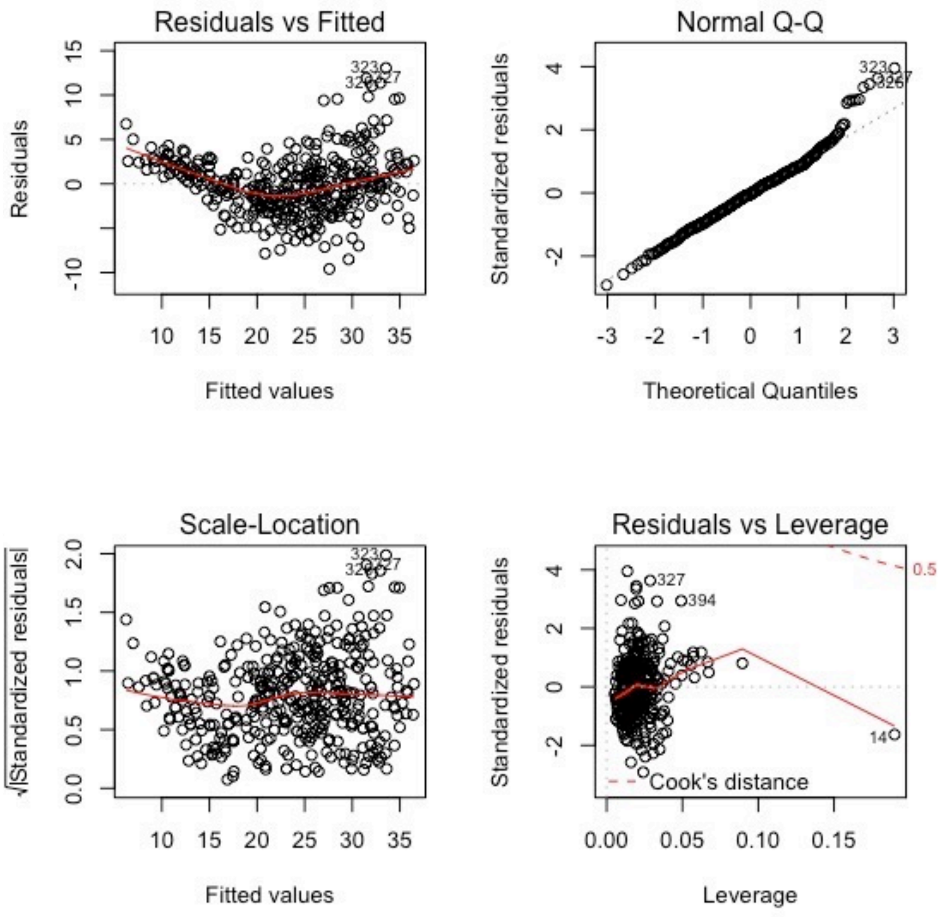
ii.
Origin, year, weight, and displacement all appear to have a statistically significant relationship to the response. This is because each has a p-value less than 0.05, meaning they are significant in the model. While origin, year, and weight's p-values are very small, displacement's (0.008) is quite a bit larger than the others.

iii.
The coefficient for the year variable suggests (if holding all the other predictors constant) an increase in one year will lead to an average increase of 0.75 miles per gallon. We can say this with 95% confidence.

d.

```
par(mfrow=c(2, 2))
plot(lm.fit)
```

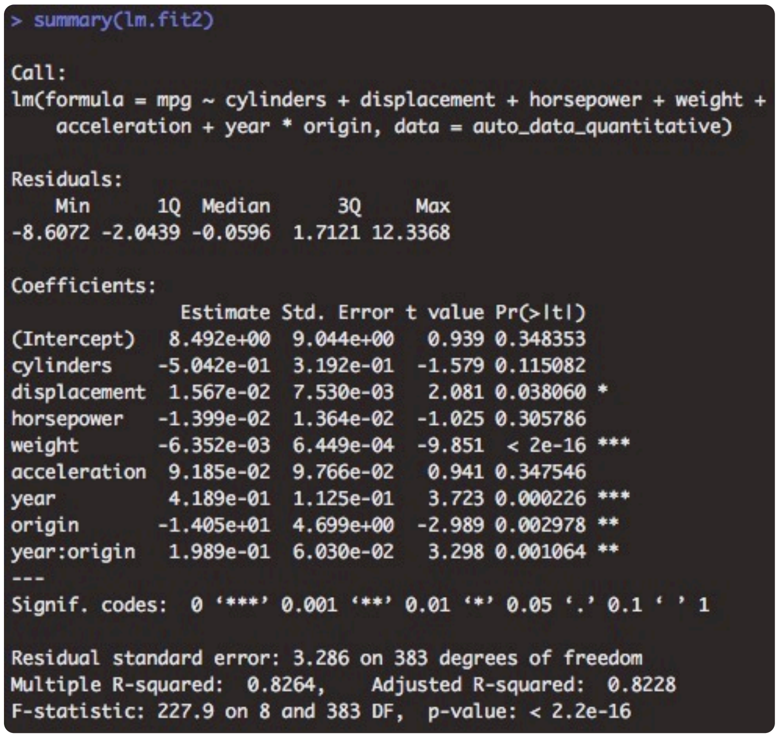
The fit appears to be good. Based on the Residuals vs Fitted chart (top left), the residual line (in red) exhibits somewhat of an *U* shape, meaning the data data could be non-linear. Looking at the Scale-Location chart, there appear to be a few outliers, but they are not unusually high (residual value of 2.0) because it is not larger than 3.0 and most of the residuals are below 1.75. The Residuals vs Leverage plot identifies observation 14 as having unusually high leverage. The least squares line would likely be quite different if observation 14 was removed from the data.



e.

```
lm.fit2=lm(mpg~cylinders
+displacement+horsepower
+weight+acceleration
+year*origin,

data=auto_data_quantitative)
summary(lm.fit2)
```

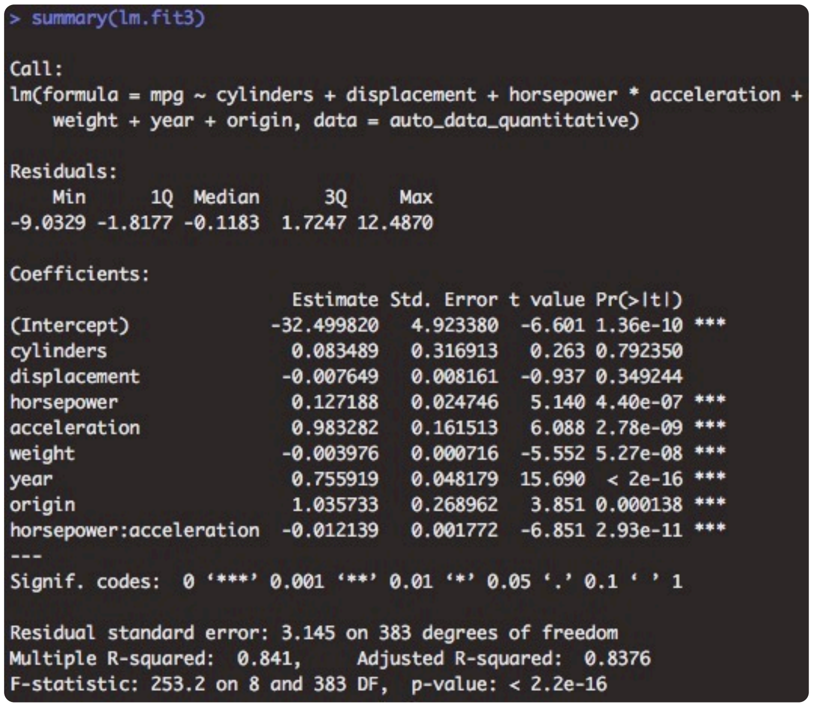
year * origin is statistically significant



```
> summary(lm.fit2)

Call:
lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
    acceleration + year * origin, data = auto_data_quantitative)

Residuals:
    Min      1Q  Median      3Q     Max
-8.6072 -2.0439 -0.0596  1.7121 12.3368

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.492e+00  9.044e+00   0.939 0.348353
cylinders    -5.042e-01  3.192e-01  -1.579 0.115082
displacement  1.567e-02  7.530e-03   2.081 0.038060 *
horsepower   -1.399e-02  1.364e-02  -1.025 0.305786
weight       -6.352e-03  6.449e-04  -9.851  < 2e-16 ***
acceleration  9.185e-02  9.766e-02   0.941 0.347546
year          4.189e-01  1.125e-01   3.723 0.000226 ***
origin       -1.405e+01  4.699e+00  -2.989 0.002978 **
year:origin   1.989e-01  6.030e-02   3.298 0.001064 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.286 on 383 degrees of freedom
Multiple R-squared:  0.8264,    Adjusted R-squared:  0.8228
F-statistic: 227.9 on 8 and 383 DF,  p-value: < 2.2e-16
```

```
lm.fit3=lm(mpg~cylinders
+displacement
+horsepower*acceleration
+weight+year+origin,

data=auto_data_quantitative
)
summary(lm.fit3)
```

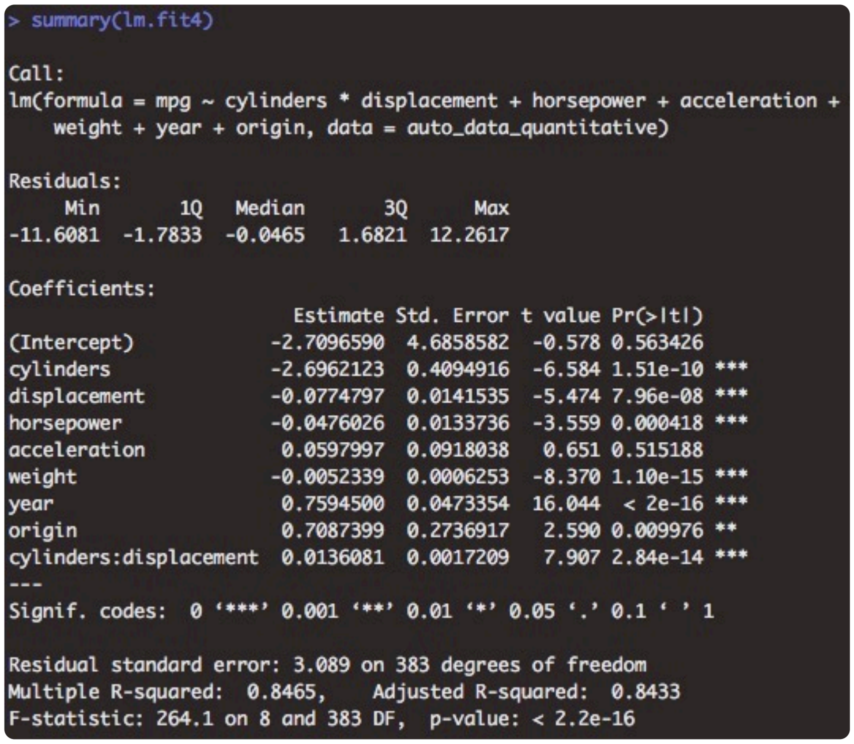
horsepower * acceleration is statistically significant, but not significant in the original model



```
> summary(lm.fit3)

Call:
lm(formula = mpg ~ cylinders + displacement + horsepower * acceleration +
    weight + year + origin, data = auto_data_quantitative)

Residuals:
    Min      1Q  Median      3Q     Max
-9.0329 -1.8177 -0.1183  1.7247 12.4870

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)            -32.499820   4.923380  -6.601 1.36e-10 ***
cylinders                0.083489   0.316913   0.263 0.792350
displacement            -0.007649   0.008161  -0.937 0.349244
horsepower               0.127188   0.024746   5.140 4.40e-07 ***
acceleration             0.983282   0.161513   6.088 2.78e-09 ***
weight                  -0.003976   0.000716  -5.552 5.27e-08 ***
year                     0.755919   0.048179  15.690  < 2e-16 ***
origin                   1.035733   0.268962   3.851 0.000138 ***
horsepower:acceleration -0.012139   0.001772  -6.851 2.93e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.145 on 383 degrees of freedom
Multiple R-squared:  0.841,     Adjusted R-squared:  0.8376
F-statistic: 253.2 on 8 and 383 DF,  p-value: < 2.2e-16
```

```
lm.fit4=lm(mpg~cylinders*
displacement+horsepower
+acceleration+weight+year
+origin,
data=auto_data_quantitati
ve)
summary(lm.fit4)
```

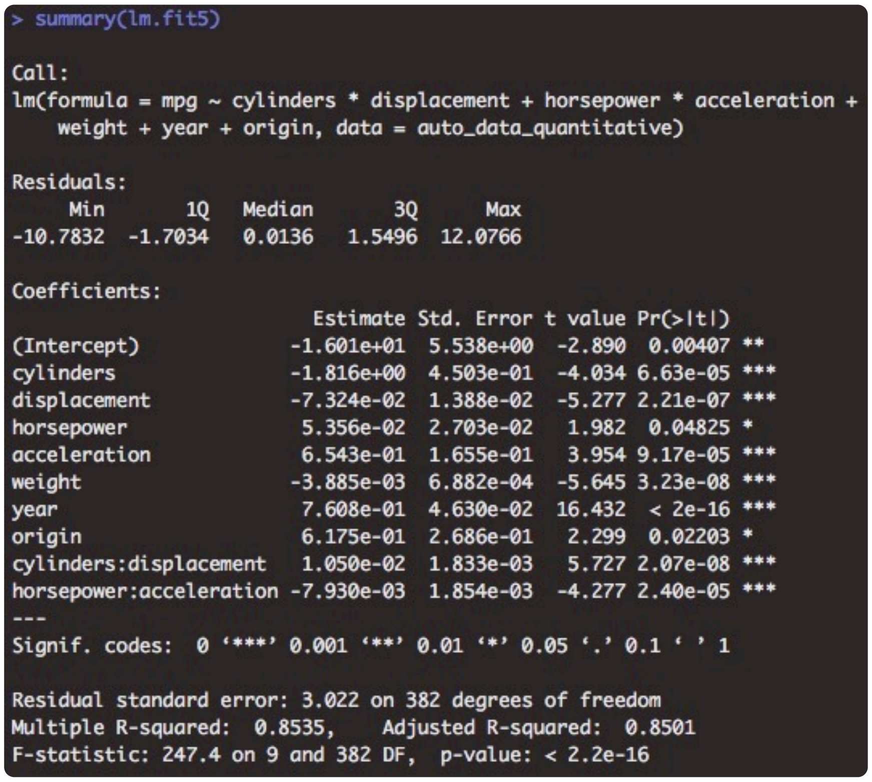
cylinders * displacement is statistically significant, but not significant in the original model



```
> summary(lm.fit4)

Call:
lm(formula = mpg ~ cylinders * displacement + horsepower + acceleration +
    weight + year + origin, data = auto_data_quantitative)

Residuals:
     Min       1Q   Median       3Q      Max
-11.6081  -1.7833  -0.0465   1.6821  12.2617

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           -2.7096590  4.6858582  -0.578 0.563426
cylinders             -2.6962123  0.4094916  -6.584 1.51e-10 ***
displacement          -0.0774797  0.0141535  -5.474 7.96e-08 ***
horsepower            -0.0476026  0.0133736  -3.559 0.000418 ***
acceleration           0.0597997  0.0918038   0.651 0.515188
weight                -0.0052339  0.0006253  -8.370 1.10e-15 ***
year                   0.7594500  0.0473354  16.044  < 2e-16 ***
origin                 0.7087399  0.2736917   2.590 0.009976 **
cylinders:displacement  0.0136081  0.0017209   7.907 2.84e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.089 on 383 degrees of freedom
Multiple R-squared:  0.8465,	Adjusted R-squared:  0.8433
F-statistic: 264.1 on 8 and 383 DF,  p-value: < 2.2e-16
```

```
lm.fit5=lm(mpg~cylinder
s*displacement
+horsepower*acceleratio
n+weight+year+origin,
data=auto_data_quantita
tive)
summary(lm.fit5)
```

Using the interaction terms: cylinders * displacement and horsepower * acceleration, all of the predictors become significant in the model (although to varying degrees). Horsepower (p-value: 0.04825) is still significant because it is less than 0.05, but is not as significant as horsepower * acceleration (2.40e-05).

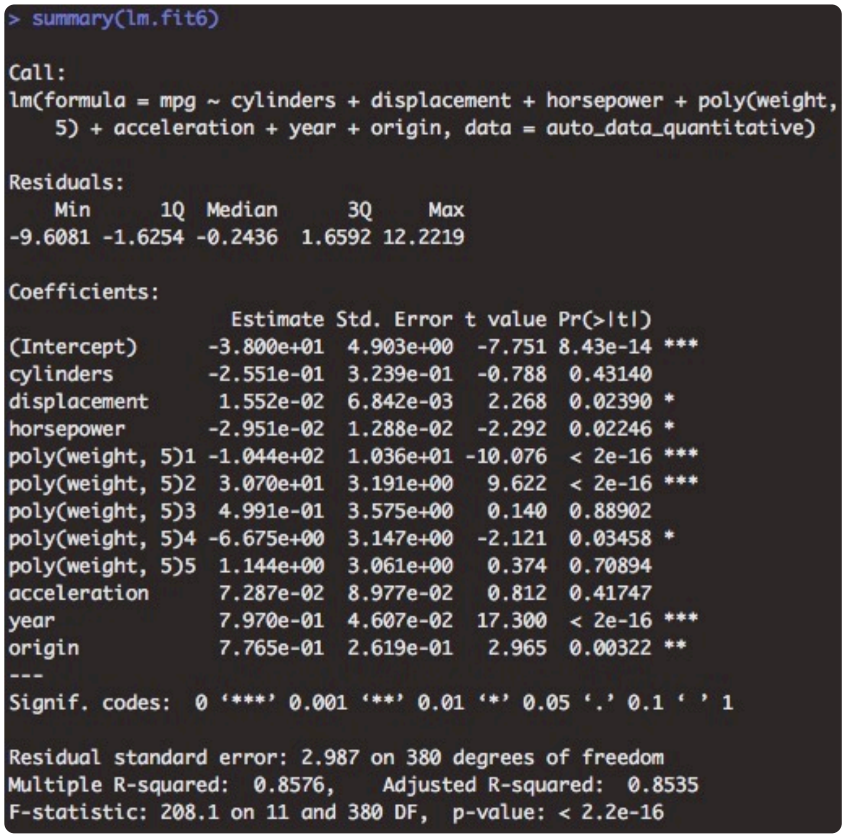


```
> summary(lm.fit5)

Call:
lm(formula = mpg ~ cylinders * displacement + horsepower * acceleration +
    weight + year + origin, data = auto_data_quantitative)

Residuals:
     Min       1Q   Median       3Q      Max
-10.7832  -1.7034   0.0136   1.5496  12.0766

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)            -1.601e+01  5.538e+00  -2.890  0.00407 **
cylinders              -1.816e+00  4.503e-01  -4.034 6.63e-05 ***
displacement           -7.324e-02  1.388e-02  -5.277 2.21e-07 ***
horsepower              5.356e-02  2.703e-02   1.982  0.04825 *
acceleration            6.543e-01  1.655e-01   3.954 9.17e-05 ***
weight                 -3.885e-03  6.882e-04  -5.645 3.23e-08 ***
year                    7.608e-01  4.630e-02  16.432  < 2e-16 ***
origin                  6.175e-01  2.686e-01   2.299  0.02203 *
cylinders:displacement  1.050e-02  1.833e-03   5.727 2.07e-08 ***
horsepower:acceleration -7.930e-03  1.854e-03  -4.277 2.40e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.022 on 382 degrees of freedom
Multiple R-squared:  0.8535,	Adjusted R-squared:  0.8501
F-statistic: 247.4 on 9 and 382 DF,  p-value: < 2.2e-16
```

f.

```
lm.fit6=lm(mpg~cylinders
+displacement+horsepower
+poly(weight,
5)+acceleration+year
+origin,
data=auto_data_quantitati
ve)
summary(lm.fit6)
```

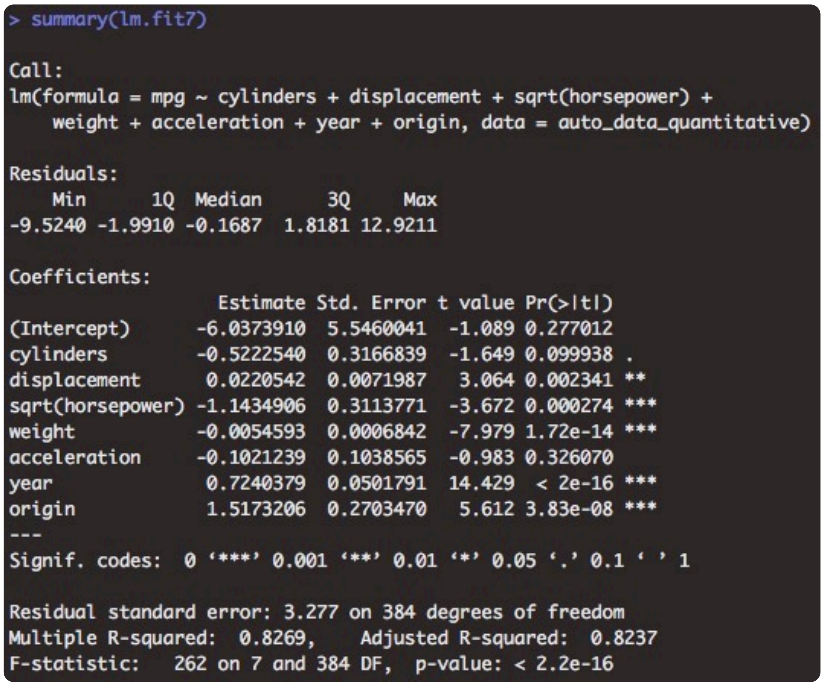
Interestingly, transforming weight up to a fourth order has a significant effect for each polynomial, except the third and fifth transformations.



```
> summary(lm.fit6)

Call:
lm(formula = mpg ~ cylinders + displacement + horsepower + poly(weight,
    5) + acceleration + year + origin, data = auto_data_quantitative)

Residuals:
    Min      1Q  Median      3Q     Max
-9.6081 -1.6254 -0.2436  1.6592 12.2219

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -3.800e+01  4.903e+00  -7.751 8.43e-14 ***
cylinders        -2.551e-01  3.239e-01  -0.788  0.43140
displacement      1.552e-02  6.842e-03   2.268  0.02390 *
horsepower       -2.951e-02  1.288e-02  -2.292  0.02246 *
poly(weight, 5)1 -1.044e+02  1.036e+01 -10.076  < 2e-16 ***
poly(weight, 5)2  3.070e+01  3.191e+00   9.622  < 2e-16 ***
poly(weight, 5)3  4.991e-01  3.575e+00   0.140  0.88902
poly(weight, 5)4 -6.675e+00  3.147e+00  -2.121  0.03458 *
poly(weight, 5)5  1.144e+00  3.061e+00   0.374  0.70894
acceleration      7.287e-02  8.977e-02   0.812  0.41747
year              7.970e-01  4.607e-02  17.300  < 2e-16 ***
origin            7.765e-01  2.619e-01   2.965  0.00322 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.987 on 380 degrees of freedom
Multiple R-squared:  0.8576,	Adjusted R-squared:  0.8535
F-statistic: 208.1 on 11 and 380 DF,  p-value: < 2.2e-16
```

```
lm.fit7=lm(mpg~cylinders
+displacement
+sqrt(horsepower)+weight
+acceleration+year
+origin,
data=auto_data_quantitati
ve)
summary(lm.fit7)
```

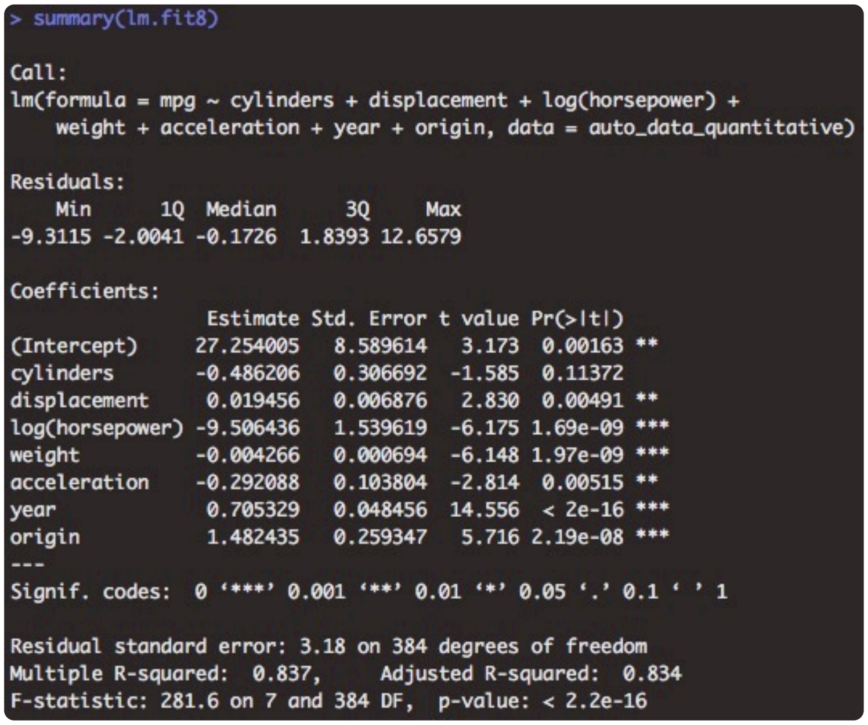
Horsepower (not significant in the original model) is significant with a square root transformation



```
> summary(lm.fit7)

Call:
lm(formula = mpg ~ cylinders + displacement + sqrt(horsepower) +
    weight + acceleration + year + origin, data = auto_data_quantitative)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5240 -1.9910 -0.1687  1.8181 12.9211

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)      -6.0373910  5.5460041  -1.089 0.277012
cylinders        -0.5222540  0.3166839  -1.649 0.099938 .
displacement      0.0220542  0.0071987   3.064 0.002341 **
sqrt(horsepower) -1.1434906  0.3113771  -3.672 0.000274 ***
weight           -0.0054593  0.0006842  -7.979 1.72e-14 ***
acceleration     -0.1021239  0.1038565  -0.983 0.326070
year              0.7240379  0.0501791  14.429  < 2e-16 ***
origin            1.5173206  0.2703470   5.612 3.83e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.277 on 384 degrees of freedom
Multiple R-squared:  0.8269,	Adjusted R-squared:  0.8237
F-statistic:   262 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
lm.fit8=lm(mpg~cylinders+displacement+log(horsepower)+weight+acceleration+year+origin, data=auto_data_quantitative)
summary(lm.fit8)
```

A logarithmic transformation of horsepower makes log(horsepower) and acceleration significant, when neither were in the original model



```
> summary(lm.fit8)

Call:
lm(formula = mpg ~ cylinders + displacement + log(horsepower) +
    weight + acceleration + year + origin, data = auto_data_quantitative)

Residuals:
    Min      1Q  Median      3Q     Max 
-9.3115 -2.0041 -0.1726  1.8393 12.6579 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)     27.254005   8.589614   3.173  0.00163 ** 
cylinders       -0.486206   0.306692  -1.585  0.11372    
displacement     0.019456   0.006876   2.830  0.00491 ** 
log(horsepower) -9.506436   1.539619  -6.175 1.69e-09 ***
weight          -0.004266   0.000694  -6.148 1.97e-09 ***
acceleration    -0.292088   0.103804  -2.814  0.00515 ** 
year             0.705329   0.048456  14.556  < 2e-16 ***
origin           1.482435   0.259347   5.716 2.19e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.18 on 384 degrees of freedom
Multiple R-squared:  0.837,     Adjusted R-squared:  0.834 
F-statistic: 281.6 on 7 and 384 DF,  p-value: < 2.2e-16
```

Overall, applying different transformations of the predictors appears to make them fit the data better. We must be careful though that as new features are created/added, the model is still optimized for minimizing the mean squared error (MSE) on the test, not training data. Otherwise, we will overfit and the model will describe random noise in the test data; adding more data or running on a new testing set will result in poor performance. Furthermore, if we had domain knowledge about cars, mpg, etc. we might be able to do better at feature engineering. For example, if we know that two predictors have an interaction effect or benefit from a certain transformation (in the context of the domain), we could test the model with those features straight away instead of trying various combinations first.