Analytics 512 Homework 2

Arif Ali

02/09/2016

Exercise 3

Part A

The equation is set up as the following:

$$\hat{y} = 50 + GPA * \beta_1 + IQ * \beta_2 + Gender * \beta_3 + (GPA * IQ) * \beta_4 + (GPA * Gender) * \beta_5$$

By putting in the beta values, we get:

$$\hat{y} = 50 + GPA * 20 + IQ * 0.07 + Gender * 35 + (GPA * IQ) * 0.01 + (GPA * Gender) * -10$$

From the updated $\hat{y}$, we know that i and ii are wrong because depending on the value of the GPA, females could make more.

Part B

```
In [3]: 50+20*4+110*0.07+1*35+110*4*0.01+4*1*-10
```

```
Out[3]: 137.1
```

Part C

This isn't true, LASSO regression incorporates variable selection by adding a coefficient of zero for predictors that are not statistically significiant. The p-value needs to be computed for each of the predictors first.

Exercise 4

Part A

Based on the equations, I would expect that cubic regression model would have a lower RSS compared to the simple linear regression. This could be because the cubic regression would have a closer fit compared to the linear regression model.

Part B

The linear regression would probably have a smaller RSS compared the a cubic regression model with respect to the test data. This is because both models would have been trained on the training data set, so the closer fit could result in a model that is too closely fitted to the training dataset.

Part C

I would basically follow the same logic behind part A. The cubic regression will still allow for a closer fit. This is more compounded by the fact we know that the relationship is not linear.

Part D

Unlike A or B, we don't know the relation between Y and X except for the fact that the relationship is not linear. However, we don't even know if the relationship is cubic or any type of polynomial. Since we are not able to ascertain what the relationship between Y and X is, as opposed to what it isn't, it can't be determined.

Exercise 8

```
In [4]: library(ISLR)
        data(Auto)
```

Part A

In [5]:

```
horsepower.lm = lm(mpg~horsepower, data = Auto)
summary(horsepower.lm)
confint(horsepower.lm, level = 0.95)
predict(horsepower.lm, interval = "confidence")[Auto$horsepower==98,]
```

Out[5]:

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

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
```

Out[5]:

	2.5 %	**97.5 %**
(Intercept)	38.52521	41.34651
horsepower	-0.1705170	-0.1451725

Out[5]:

	fit	**lwr**	**upr**
180	24.46708	23.97308	24.96108
229	24.46708	23.97308	24.96108

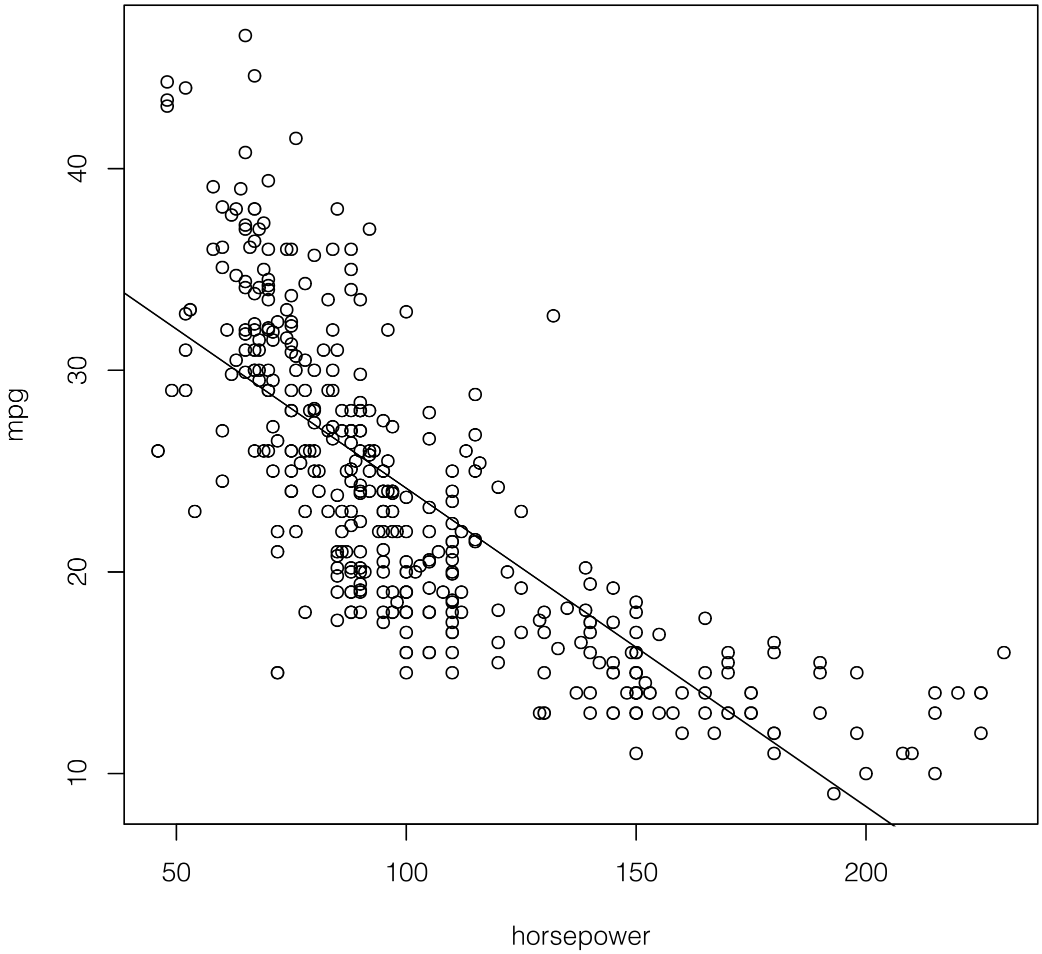
i/ii: Based on the F-statistic and the p-value, there is a strong relationship between the predictor (horsepower) and the response variable (mpg)

iii: The Coefficient is negative which indicates a negative relationship between the predictor and response

Part B

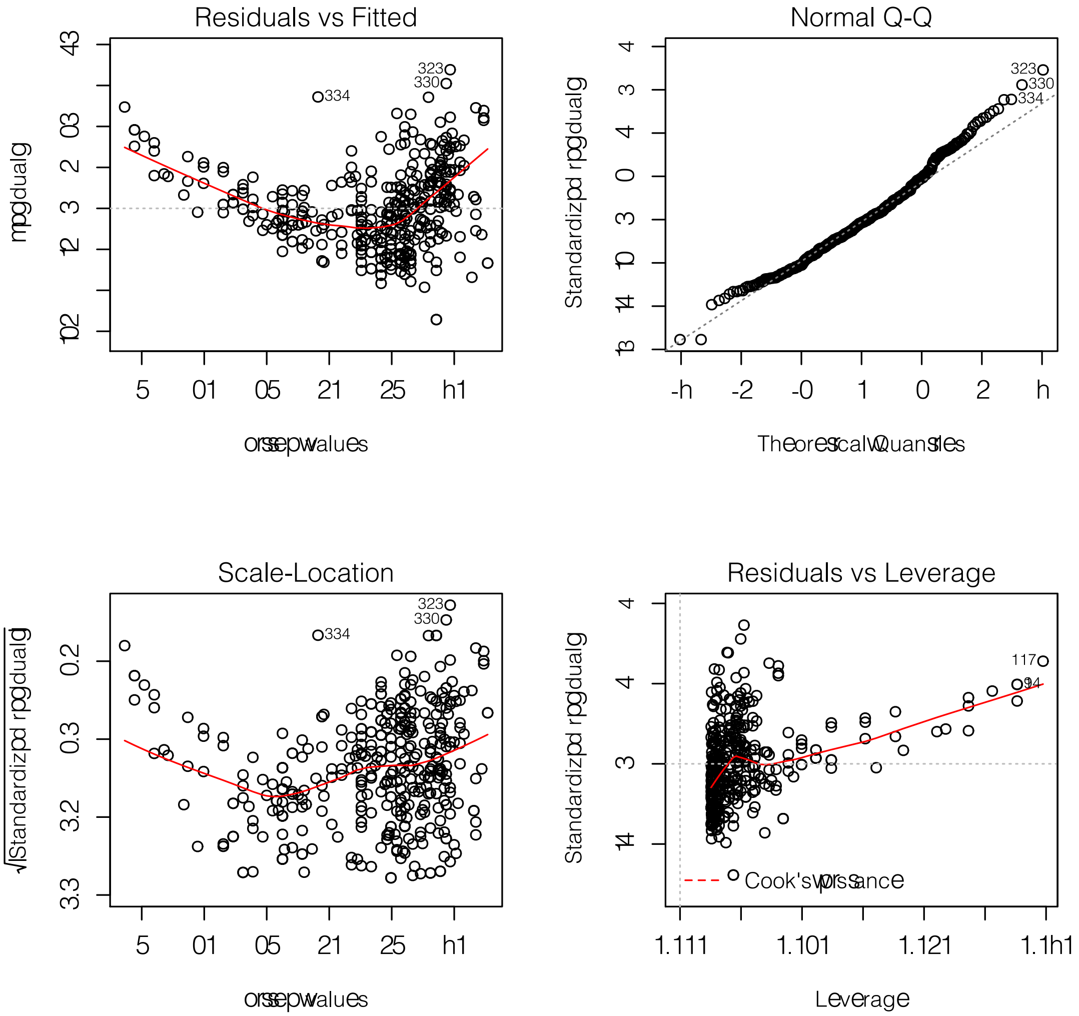
In [6]:

```
plot(mpg~horsepower, data = Auto)
abline(horsepower.lm)
```



Part C

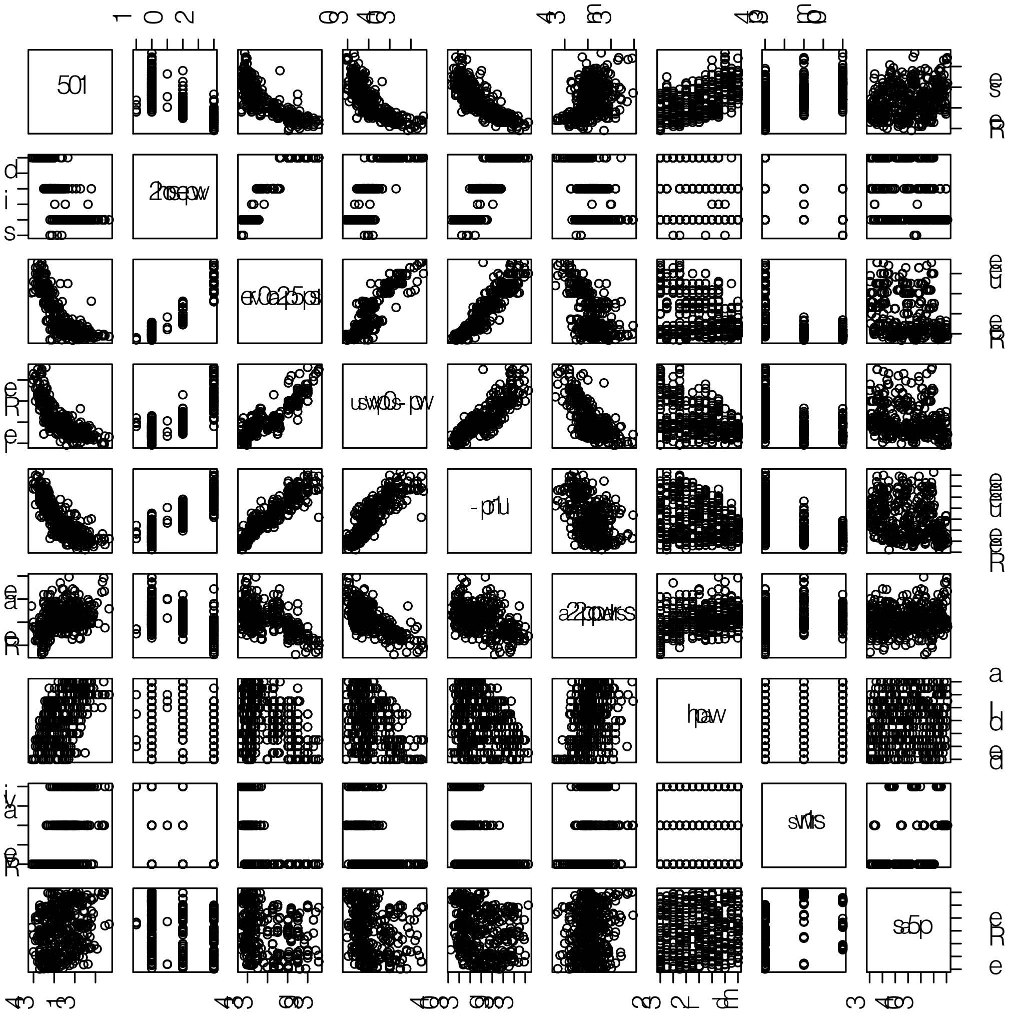
```
In [7]: par(mfrow = c(2,2))
        plot(horsepower.lm)
```



Exercise 9

Part A

In [8]:
```
pairs(Auto)
```



Part B

In [9]:
```
cor(Auto[,-ncol(Auto)])
```

Out[9]:

	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

Part C

In [10]:

```
auto.lm = lm(mpg~.,data=Auto[,-ncol(Auto)])
summary(auto.lm)
```

Out[10]:

```
Call:
lm(formula = mpg ~ ., data = Auto[, -ncol(Auto)])

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
```

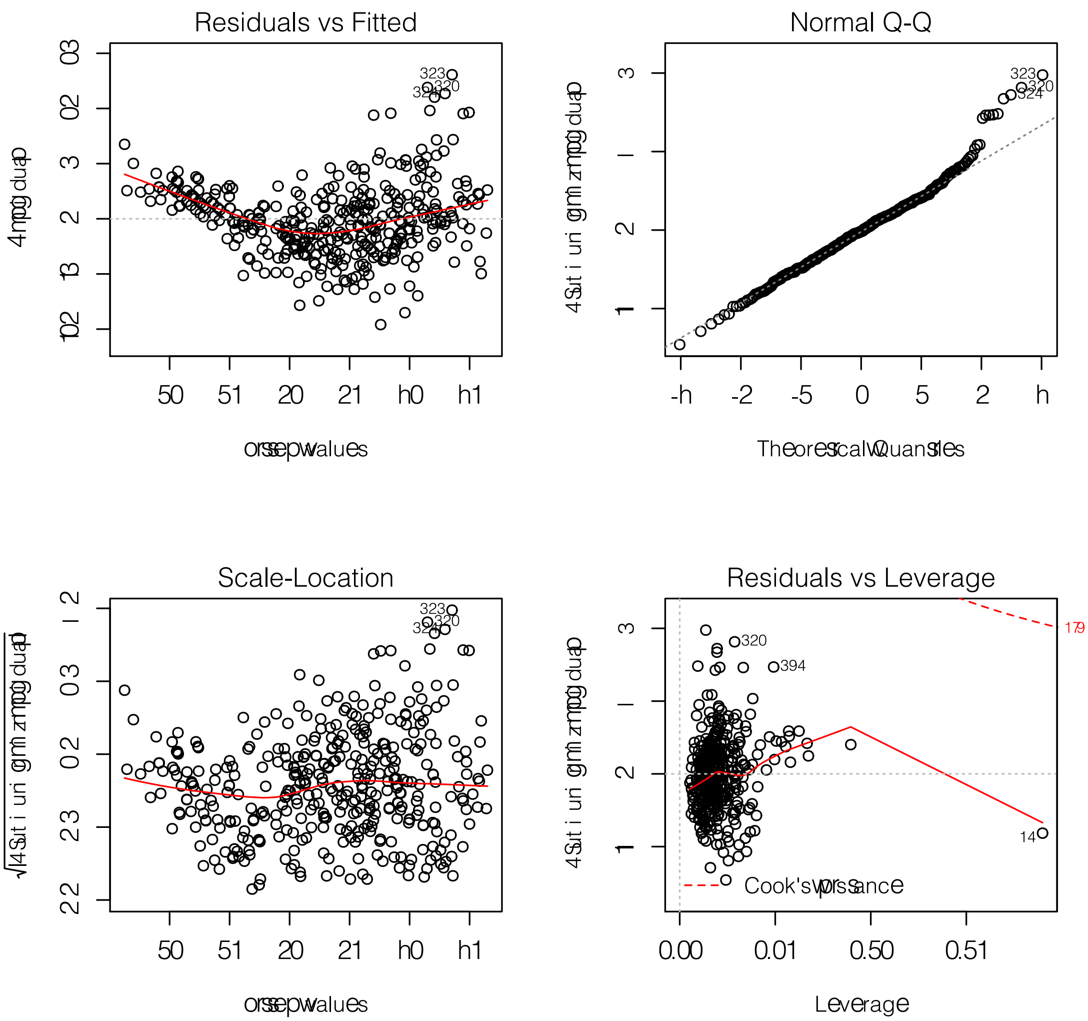
i: The F-statistic is very high and the p-value associated with it is very low, so there is an overall strong relationship between the predictors and the response variable (mpg)

ii: The Predictors with regards to displacement, weight, year, and orgin are statistically significant with respect to mpg. Cylinders and acceleration are not considered Statistically significiant due to the p-values being ≥ 0.1. I'm not surprised by acceleration being statistically less significiant because of the Tesla Model S P85.

iii: The coefficient is positive, so the newer the car, the better the mpg.

Part D

```
In [11]: par(mfrow = c(2,2))
         plot(auto.lm)
```



Point 14 seems to have some high leverage as opposed to 327 and 394 which which noted are not that far out as 14. From the normal Q-Q plot indicates that the standardized residuals do not follow a normal distribution.

Part E

For part E and F, I got rid of the non-significant predictors (Cylinders and acceleration)

```
In [12]: auto.lm.interaction = lm(mpg~(displacement:weight)+(year:origin),data=Auto[,-ncol(Auto)])
         summary(auto.lm.interaction)

         auto.lm.interaction = lm(mpg~(year:weight)+(displacement:origin),data=Auto[,-ncol(Auto)])
         summary(auto.lm.interaction)
```

```
Out[12]: Call:
         lm(formula = mpg ~ (displacement:weight) + (year:origin), data = Auto[,
             -ncol(Auto)])

         Residuals:
             Min      1Q  Median      3Q     Max
         -13.198  -2.832  -0.279   2.193  16.860

         Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
         (Intercept)          2.679e+01  8.319e-01  32.200  < 2e-16 ***
         displacement:weight -9.940e-06  5.398e-07 -18.416  < 2e-16 ***
         year:origin          2.690e-02  4.471e-03   6.016 4.14e-09 ***
         ---
         Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

         Residual standard error: 4.562 on 389 degrees of freedom
         Multiple R-squared:  0.6601,    Adjusted R-squared:  0.6584
         F-statistic: 377.8 on 2 and 389 DF,  p-value: < 2.2e-16
```

```
Out[12]: Call:
         lm(formula = mpg ~ (year:weight) + (displacement:origin), data = Auto[,
             -ncol(Auto)])

         Residuals:
              Min       1Q   Median       3Q      Max
         -12.3555  -3.3328  -0.5134   2.5797  17.6789

         Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
         (Intercept)          4.584e+01  9.841e-01  46.587   <2e-16 ***
         year:weight         -9.707e-05  4.987e-06 -19.466   <2e-16 ***
         displacement:origin -2.081e-03  3.366e-03  -0.618    0.537
         ---
         Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

         Residual standard error: 4.961 on 389 degrees of freedom
         Multiple R-squared:  0.5981,    Adjusted R-squared:  0.596
         F-statistic: 289.4 on 2 and 389 DF,  p-value: < 2.2e-16
```

In both models, with the exception of displacement:origin, the other interactions are statistically significant based on the p-values of the interactions.

Part F

```
In [13]: auto.lm.transformation = lm(mpg~log(displacement)+weight+
                                     sqrt(year)+I(origin)^2,data=Auto[,-ncol(Auto)])
         summary(auto.lm.transformation)

         auto.lm.transformation = lm(mpg~log(weight)+displacement+
                                     sqrt(origin)+I(year)^2,data=Auto[,-ncol(Auto)])
         summary(auto.lm.transformation)
```

```
Out[13]: Call:
         lm(formula = mpg ~ log(displacement) + weight + sqrt(year) +
             I(origin)^2, data = Auto[, -ncol(Auto)])

         Residuals:
              Min       1Q   Median       3Q      Max
         -10.8260  -1.9314  -0.0845   1.7774  13.2013

         Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
         (Intercept)       -6.069e+01  8.953e+00  -6.779 4.54e-11 ***
         log(displacement) -2.982e+00  1.006e+00  -2.964  0.00322 **
         weight            -4.483e-03  5.712e-04  -7.849 4.17e-14 ***
         sqrt(year)         1.280e+01  8.433e-01  15.181  < 2e-16 ***
         I(origin)          7.782e-01  2.860e-01   2.721  0.00681 **
         ---
         Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

         Residual standard error: 3.322 on 387 degrees of freedom
         Multiple R-squared:  0.8207,    Adjusted R-squared:  0.8188
         F-statistic: 442.7 on 4 and 387 DF,  p-value: < 2.2e-16
```

```
Out[13]: Call:
         lm(formula = mpg ~ log(weight) + displacement + sqrt(origin) +
             I(year)^2, data = Auto[, -ncol(Auto)])

         Residuals:
            Min     1Q Median     3Q    Max
         -9.694 -1.898 -0.006  1.582 12.978

         Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
         (Intercept)  129.580139  11.146538  11.625  < 2e-16 ***
         log(weight)  -21.612576   1.447356 -14.932  < 2e-16 ***
         displacement   0.008163   0.004064   2.009 0.045274 *
         sqrt(origin)   2.393215   0.680243   3.518 0.000486 ***
         I(year)        0.807836   0.046499  17.373  < 2e-16 ***
         ---
         Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

         Residual standard error: 3.108 on 387 degrees of freedom
         Multiple R-squared:  0.843,     Adjusted R-squared:  0.8414
         F-statistic: 519.6 on 4 and 387 DF,  p-value: < 2.2e-16
```

Based on the p-values for each of the transformations for the first transformation regression model, it appears that each of the transformations is statistically significiant as evident by the p-values. Interesting, it appears that $origin^2$ is less statistically significant compared to log(displacement) or square root of year.

For the second tranfromation model, displacement seems to still not be as statistically significant compared to the transformations of the other predictors. As in the case of the first transformation model, this does not mean displacement is not statistically significiant. Under a basica variable selection method (backwards elimination), I wouldn't eliminate it.

Exercise 12

Part A

From the book, we know that $\hat{\beta} = (\sum_{i=1}^{n} x_i y_i)/(\sum_{i=1}^{n} x_i^2)$. In order for the coefficients for Y onto X and X onto Y to be same:

$$(\sum_{i=1}^{n} x_i y_i)/(\sum_{i=1}^{n} x_i^2) = (\sum_{i=1}^{n} y_i x_i)/(\sum_{i=1}^{n} y_i^2) \implies \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} y_i^2 \implies \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

Part B

```
In [11]: X = rnorm(100)
         Y = X^2
         train = data.frame(X,Y)
```

By setting $Y = X^2$, $\sum_{i=1}^{n} x_i \not= \sum_{i=1}^{n} y_i$

```
In [9]: summary(lm(Y~X, data = train))
```

```
Out[9]: Call:
        lm(formula = Y ~ X, data = train)

        Residuals:
            Min      1Q  Median      3Q     Max 
        -0.9081 -0.8383 -0.3504  0.4035  3.4092 

        Coefficients:
                    Estimate Std. Error t value Pr(>|t|)    
        (Intercept)  0.90803    0.10166   8.932 2.51e-14 ***
        X           -0.02816    0.10654  -0.264    0.792    
        ---
        Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

        Residual standard error: 1.012 on 98 degrees of freedom
        Multiple R-squared:  0.0007121, Adjusted R-squared:  -0.009485 
        F-statistic: 0.06984 on 1 and 98 DF,  p-value: 0.7921
```

```
In [10]: summary(lm(X~Y, data = train))
```

```
Out[10]: Call:
         lm(formula = X ~ Y, data = train)

         Residuals:
              Min       1Q   Median       3Q      Max 
         -1.91826 -0.77557  0.06857  0.77050  2.01446 

         Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
         (Intercept) -0.06297    0.12962  -0.486    0.628
         Y           -0.02529    0.09571  -0.264    0.792

         Residual standard error: 0.9596 on 98 degrees of freedom
         Multiple R-squared:  0.0007121, Adjusted R-squared:  -0.009485 
         F-statistic: 0.06984 on 1 and 98 DF,  p-value: 0.7921
```

Part C

```
In [15]: X = rnorm(100)
         Y = sample(X,size = 100, replace = F)
         train = data.frame(X,Y)
```

When attempting $Y = X$ in was given the following warning:

Warning message: In summary.lm(lm(Y ~ X, data = train)): essentially perfect fit: summary may be unreliable

so I shook up Y in order for $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$ to hold.

```
In [16]: summary(lm(Y~X, data = train))
```

```
Out[16]: Call:
         lm(formula = Y ~ X, data = train)

         Residuals:
              Min       1Q   Median       3Q      Max 
         -2.88896 -0.73102  0.05766  0.74532  2.19650 

         Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
         (Intercept) -0.05348    0.10346  -0.517    0.606
         X            0.07185    0.10075   0.713    0.477

         Residual standard error: 1.033 on 98 degrees of freedom
         Multiple R-squared:  0.005162,  Adjusted R-squared:  -0.004989 
         F-statistic: 0.5085 on 1 and 98 DF,  p-value: 0.4775
```

```
In [17]: summary(lm(X~Y, data = train))
```

```
Out[17]: Call:
         lm(formula = X ~ Y, data = train)

         Residuals:
              Min       1Q   Median       3Q      Max 
         -2.78098 -0.74697  0.04643  0.67993  2.09618 

         Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
         (Intercept) -0.05348    0.10346  -0.517    0.606
         Y            0.07185    0.10075   0.713    0.477

         Residual standard error: 1.033 on 98 degrees of freedom
         Multiple R-squared:  0.005162,  Adjusted R-squared:  -0.004989 
         F-statistic: 0.5085 on 1 and 98 DF,  p-value: 0.4775
```