Factors affecting fuel consumption

Matthew Heym, Thi Van Nguyen, Matthew Pressimone

January 5, 2021

Contents

List of Figures

1 Introduction

One aspect that plays an incredible role in the daily lives of people around the world is transportation. The importance of transportation and its effect on economics, international trade, monetary policy, and the fundamental principles of supply and demand are the driving factors towards a prosperous economy. While it is a necessary part of modern life and the economy, transportation requires an immense amount of oil consumption which has been increasing sharply every year. According to the U.S Energy Information Administration, transportation accounts for 77% of domestic oil use with an additional 1.3 Gt of carbon dioxide each year from petroleum consumption (IEA 2009). This accounts for about 5 percent of global carbon dioxide emissions (IEA, 2009). These and other statistics raise concern towards human involvement in global climate change. There are many ways to achieve this in manufacturing, such as switching to advanced hybrid vehicles, using low-carbon fuels, or reducing the vehicle's size and weight. This statistical study analyzes the main impacts of various factors on fuel consumption. We are interested in determining if a relationship exists between fuel consumption miles per gallon and the predictors cylinders, displacement, horsepower, weight, and acceleration. In carrying out this study, we will use the Auto MPG dataset which was acquired from the StatLib library. The data set we are considering consists of 397 observations, 5 predictors and 1 response. Because of model complexity, we did not include the predictors origin and car names. We also did not include model year because it took too few values to classify as a numerical predictor and took too many to classify as a categorical predictor due to complexity. Improving upon this model and interpreting its significance will help us understand how technology can modify these factors that impact fuel consumption.

2 Methodology

The main statistical method widely used is the Least Square method which determines the best fit regression line. This method calculates the observed data by minimizing the sum of the squares of the vertical deviations from each data point to the line. Any data point represents the linear relationship between the independent variables and the dependent variable in terms of the unknown parameters of the model.

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{1}$$

In this equation, y is a vector of the dependent variable, and X is a matrix of the independent variables. The resulting estimators are called least square estimators (LSE), or ordinary

least square (OLS) estimators. The Gauss-Markov theorem states that the least square estimator $\hat{\beta}_{\mathbf{OLS}}$ is the best linear unbiased estimator (BLUE) of β if the experimental errors are uncorrelated, have a mean of zero and a constant variance. In addition, we will also discuss methods of determining a best fit model and comparing variables to see how to improve our model. These include multiple R^2 and adjusted R^2, evaluating missing data, determining multicollinearity, and using the ANOVA F test to compare bigger and smaller models of our data set.

We used various techniques of variable selection to choose the most optimal model that satisfied all of our necessary assumptions (the relationship between the response and predictors is linear; error terms are independent and identically distributed, following a normal distribution with mean 0 and a constant variance; there are no outliers or influential points). To check for a constant variance of error terms, we plotted the residuals vs the fitted values. If the plot did not depict a constant variance, we attempted to transform the response to obtain a constant variance. In doing so, we proceeded with the box-cox transformation and applied our resulting lambda to the response. Next, we experimented with the log transformation to see how it satisfied the constant variance assumption compared to the previous transformation. Additionally, we tested the normality of the errors with a qq-plot and the Shapiro-Wilk test as well tested for the presence of a linear relationship after doing variable selection. Moreover, the use of both stepwise forward and stepwise backward selection aided in finding the combination of variables that will give us the best AIC score.

The last method that was used to improve upon the model and provide a meaningful explanation of our data was principal component analysis. This analysis is widely used for finding low dimensional linear structure in higher dimensional data (Faraway 161). The purpose is to transform existing variables to orthogonality by creating new predictors that absorb the most essential information from the original predictors. In doing so, the new components are transformed to orthogonality by a linear combination of the original predictors, successfully eliminating multicollinearity by design. At first, our data set consisted of a significant amount of multicollinearity between the predictors. We discovered this using methods such as the variance inflation factor to determine the high level of correlation between the variables. As a result, it is more difficult to interpret the meaning of the original predictors and we will show how this analysis improves this interpretation by eliminating the multicollinearity. Additionally, we used principal component regression to improve the effectiveness of the model by minimizing the root mean squared error with the least amount of components. We illustrate that after partitioning the data and using cross validation, the root mean squared error is minimized with a smaller amount of components. Furthermore, the methods chosen in this statistical study provide a concrete interpretation of the different

variables on fuel consumption.

3 Results and Discussion

3.1 Missing values

```
> mpg[is.na(mpg$horsepower),]
     mpg cylinders displacement horsepower weight acceleration
32  25.0         4           98         NA   2046         19.0
126 21.0         6          200         NA   2875         17.0
330 40.9         4           85         NA   1835         17.3
336 23.6         4          140         NA   2905         14.3
354 34.5         4          100         NA   2320         15.8
374 23.0         4          151         NA   3035         20.5
```

We first observe that there are 6 missing values in the total 397 observations. Some of these NA values in the horsepower dataset were most likely missing at random. In this case, we can delete these observations without losing much information. Let's see what happens when we use the lm function which automatically deletes missing values.

```
> lmod <- lm(mpg~  cylinders + displacement + horsepower + weight
+ acceleration, data=mpg)
> sumary(lmod)
                Estimate  Std. Error t value  Pr(>|t|)
(Intercept)   4.6279e+01  2.6745e+00 17.3036 < 2.2e-16
cylinders    -3.9310e-01  4.1226e-01 -0.9535  0.340919
displacement -9.0725e-05  9.0834e-03 -0.0100  0.992036
horsepower   -4.5380e-02  1.6702e-02 -2.7170  0.006885
weight       -5.1901e-03  8.1794e-04 -6.3453 6.241e-10
acceleration -3.0127e-02  1.2610e-01 -0.2389  0.811295

n = 391, p = 6, Residual SE = 4.25244, R-Squared = 0.71
```

We observe how cylinders, displacement and acceleration are all insignificant based on their greater p-values. For the missing data, we did not input the mean as a replacement because it adds additional uncertainty. Furthermore, we have a large enough sample size and a small enough number of missing values that we can afford to remove missing observations and still fit an accurate model.

3.2 Multicollinearity

```
> round(cor(removed_NA_data[,2:6]),3)
             cylinders displacement horsepower weight acceleration
cylinders        1.000        0.951      0.843  0.898       -0.502
displacement     0.951        1.000      0.897  0.933       -0.542
horsepower       0.843        0.897      1.000  0.864       -0.689
weight           0.898        0.933      0.864  1.000       -0.416
acceleration    -0.502       -0.542     -0.689 -0.416        1.000
```

After observing our data set, notice how the correlation of cylinders and displacement is 0.951, and the correlation between weight and displacement is 0.933. The high correlation among some of the predictors suggests that multicollinearity exists. Engine displacement is the swept volume of pistons inside the cylinders of an engine (Leanse). Therefore displacement, cylinders and weight are related and can contain similar information (they are highly correlated). By adding displacement and cylinders to the simple linear regression model, not much new important information is obtained other than the model becoming more complex. Let's look at the variance inflation factors.

```
> vif(lmodel)
   cylinders displacement   horsepower       weight acceleration
   10.632621    19.477654     8.926310    10.426639     2.605708
```

It is obvious that there is serious multicollinearity because $VIF > 10$. We then observe the effect of removing the variables cylinders, displacement, and acceleration.

```
> reducemodel <- lm(mpg ~horsepower +weight, removed_NA_data)
> sumary(reducemodel)
               Estimate  Std. Error  t value  Pr(>|t|)
(Intercept) 45.63711421  0.79421367  57.4620 < 2.2e-16
horsepower  -0.04726287  0.01109915  -4.2582 2.588e-05
weight      -0.00579350  0.00050293 -11.5195 < 2.2e-16

n = 391, p = 3, Residual SE = 4.24520, R-Squared = 0.71
> vif(reducemodel)
horsepower     weight
  3.955415   3.955415
```

It is clear that the VIF decreased after removing the highly correlated variables. To conclude, removing these variables does improve our model and can be better used to explain the

relationships that the variables have on fuel consumption. However, we will later explore more methods in better reducing this multicollinearity.

3.3 Comparing the model

In order to further determine a better model to represent a relationship, we conduct hypothesis tests to check the coefficients of cylinders, displacement and acceleration based on the ANOVA F test. We will proceed by constructing a bigger and a smaller model as shown below with the removal of highly correlated variables.

```
Analysis of Variance Table

Model 1: mpg ~ cylinders + displacement + horsepower + weight + acceleration
Model 2: mpg ~ horsepower + weight
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    385 6962.0
2    388 6992.4 -3   -30.389 0.5602 0.6416
```

$H_0 : \beta_{cylinders} = \beta_{displacement} = \beta_{acceleration} = 0$
H_a :At least one of the $\beta_{cynlinders}, \beta_{displacement}, \beta_{acceleration} \neq 0$

We see that $p = 0.6416 > 0.05$ at the 95% significance level, so we fail to reject H_0. We can interpret this as the smaller model, not including those highly correlated predictors being cylinders, displacement and acceleration, being just as good as the full model. Therefore, to improve our model of the data set, we could remove these variables. In addition, we know that the least square estimates of β are obtained by minimizing the Residual Sum of Squares. Let's take a look at the R-squared formula which is a measure of goodness of fit for the linear regression model.

$$R^2 = 1 - \frac{RSS}{TSS}.$$

R-squared is the percentage of variation of the dependent variable that is explained by the predictors. An important note is that R-squared always increases by adding new predictors. In general, it does not satisfy a good model when there are excess predictors. Because of the defect of R-squared, it may not be a good measure for evaluating the goodness of fit. One of other methods for measuring the goodness of fit is the adjusted R-squared.

$$\bar{R}^2 = 1 - \frac{RSS/(n-p-1)}{TSS/(n-p-1)}.$$

These two methods are another way of determining how many predictors would better fit the model. We can observe both values of R^2 in figure 1. We notice that the maximum R^2 value is achieved with 5 predictors and the maximum adjusted R^2 value is achieved with three predictors. This is significant because it shows how the two methods contrast in comparing the model for a better fit.

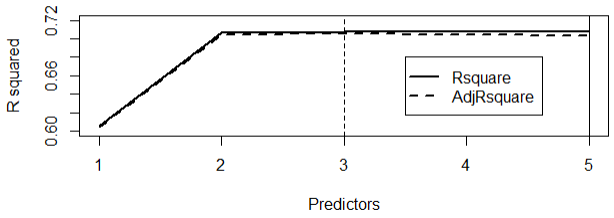

Figure 1: R^2

3.4 Checking model assumptions, transformations

3.4.1 Checking constant variance, transformations, log transformation

Using the first step of checking our model assumptions, we investigate if the errors satisfy the constant variance assumption. See (2); it looks somewhat fan-shaped. We tried the Box-cox transformation as it makes the data more normally distributed. See (3); -0.5 is a feasible value for lambda. After applying the box-cox transformation, our plot of residuals vs fitted values still looks too fan-shaped (see 4). Next, we tried the log transformation on our response. This time, the constant variance assumption appears to be satisfied because the points on our residual vs. fitted value plot look random (see5). So, from this point forward, our model will be the one with the log transformation applied to the response.

```
logmod <- lm(log(mpg) ~ cylinders + displacement + horsepower + weight +
 acceleration, data=mpgdata)
```

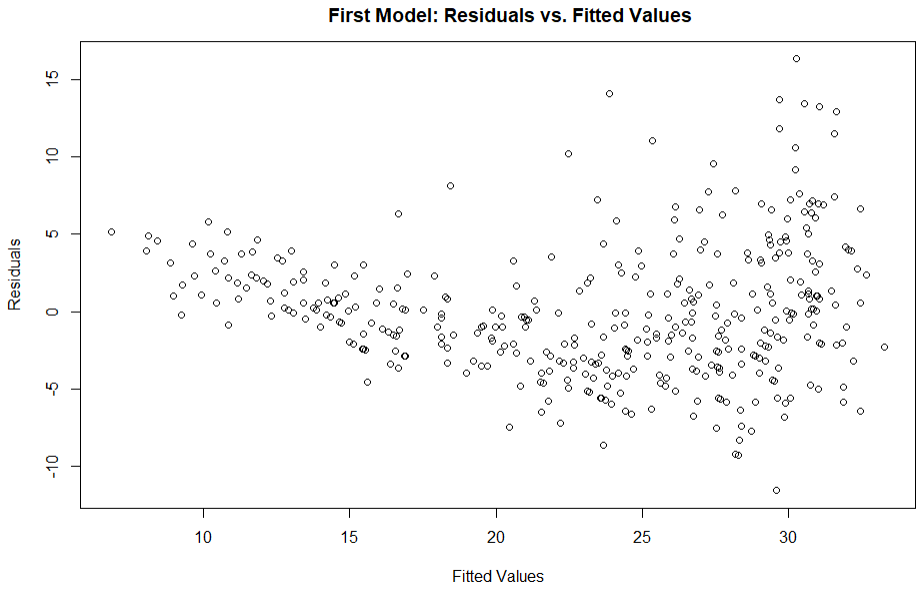

Figure 2: First Model Residuals vs Fitted Values

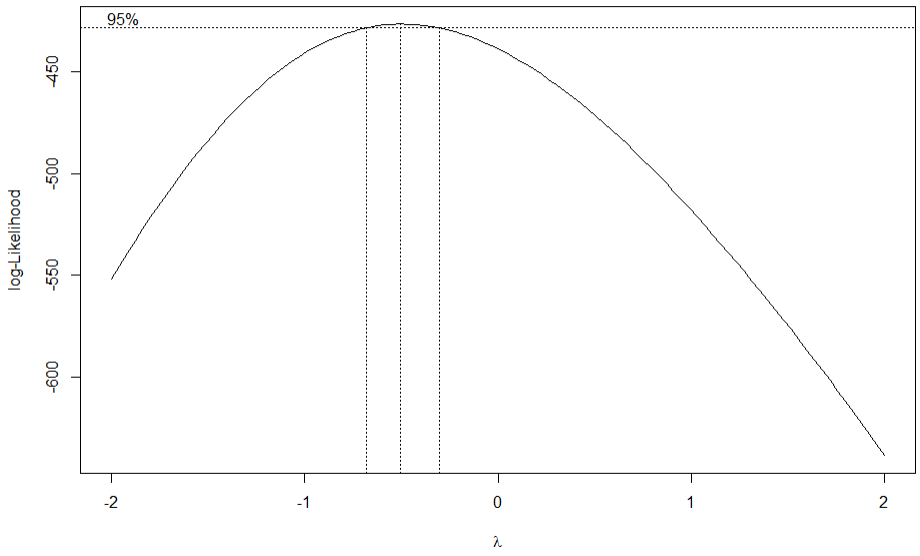

Figure 3: Choosing lambda for Box-Cox Transformation

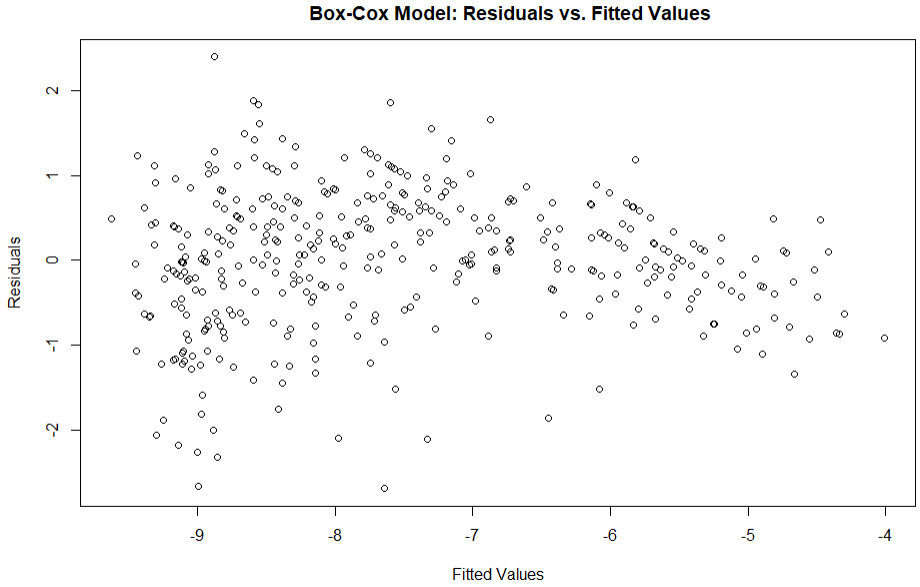

Figure 4: Box-Cox Model: Residuals vs Fitted Values

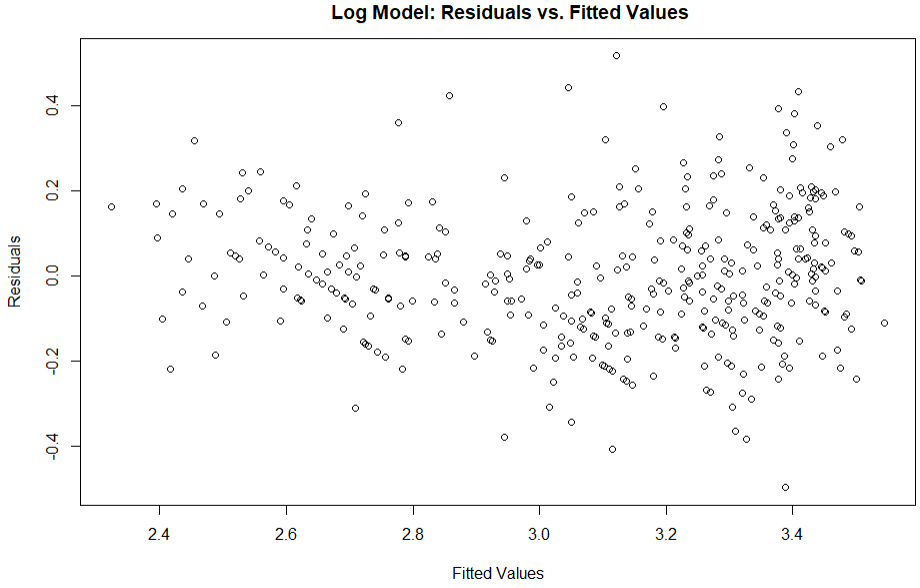

Figure 5: Log Transformation: Residuals vs Fitted Values

3.4.2 Checking for normality of errors

To check if the errors are normally distributed, we did a qq-plot of the residuals and the Shapiro-Wilk test. See (6). Our p-value for the Shapiro-Wilk test is 0.1524, so we cannot reject the null hypothesis that the data is normally distributed. The qq-plot looks like a straight line, and we did not reject the null hypothesis for the Shapiro-Wilk test (which rejects at very small departures from normality for large sample sizes). Thus, the model satisfies the normality assumption.

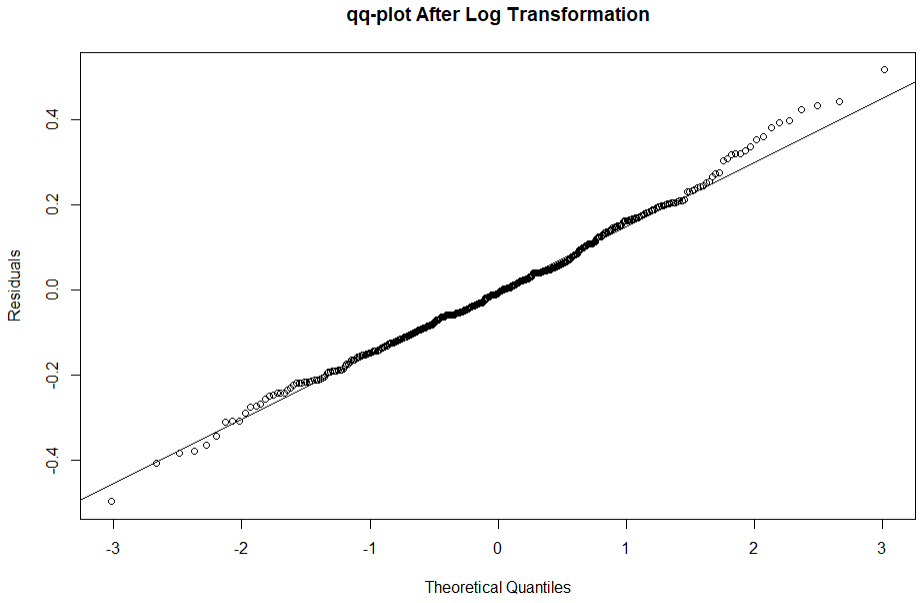

Figure 6: QQ plot after Log Transformation

3.4.3 Checking for influential points and outliers

First, we looked for potential outliers; these points have studentized residuals of absolute value greater than or equal to 3. Two observations satisfied this condition.

```
> rstudent(logmod)[which(abs(rstudent(logmod))>=3)]
      111       387
-3.248421  3.386415
```

To check for influential points, we looked at half-normal plots of the leverages and Cook's distances using the `halfnorm()` function in the `faraway` package. (see 7 and 8). While observation 13 had a leverage that required investigation, the Cook's distances were all extremely low, the highest of which was under 0.06 and none of which stood out too far from the others. Thus, we concluded that there were no influential points ("Identifying Influential Data Points").

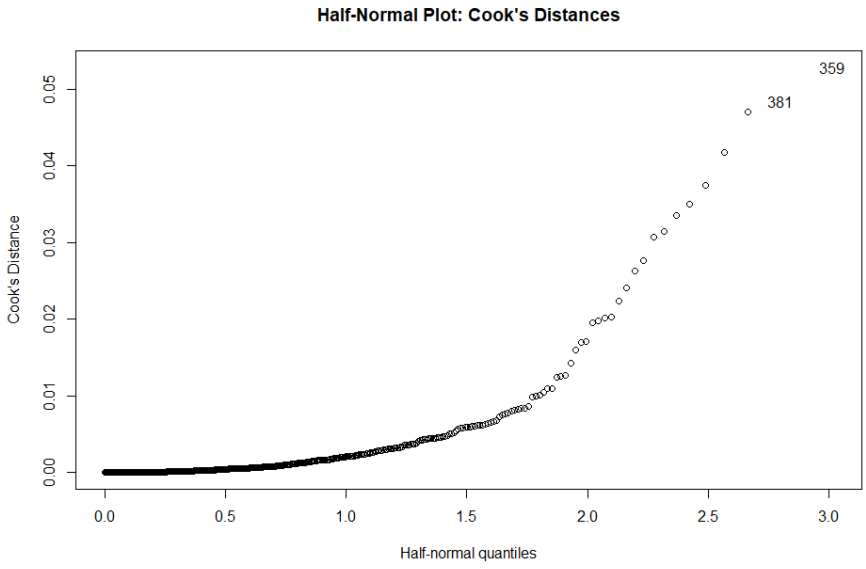

Figure 7: Half Normal Plot, Cooks distance

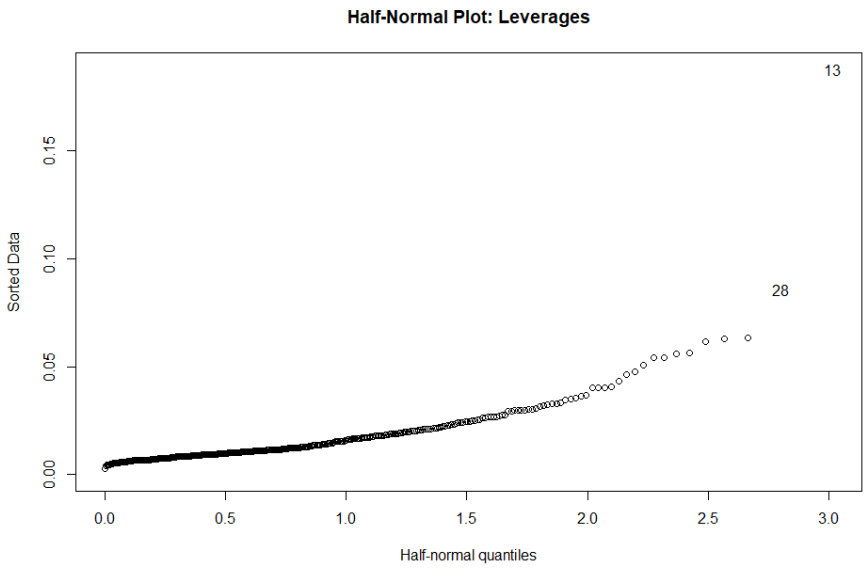

Figure 8: Half Normal Plot, Leverage

3.5 Variable selection with and without outliers

We performed forward stepwise selection and backward stepwise selection using the AIC criterion with the `stepwise()` function in the `RcmdrMisc` library. Using this, we found that with or without the outliers, the chosen model always had the predictors `weight`, `horsepower`, and `cylinders`. These were the only predictors when outliers were excluded. When including outliers, `acceleration` was also in the chosen model. Forward and backward stepwise selection yielded the same results. Below is the output for the `stepwise()` function.

With outliers:

```
Step:  AIC=-1446.82
log(mpg) ~ cylinders + horsepower + weight

               Df Sum of Sq     RSS     AIC
<none>                      9.4682 -1446.8
+ acceleration  1   0.03690  9.4313 -1446.3
+ displacement  1   0.00014  9.4681 -1444.8
```

```
- cylinders     1   0.11877  9.5870 -1444.0
- horsepower    1   0.68504 10.1532 -1421.5
- weight        1   2.05058 11.5188 -1372.2

Call:
lm(formula = log(mpg) ~ cylinders + horsepower + weight, data = mpgdata)

Coefficients:
(Intercept)    cylinders   horsepower       weight
  4.1170889   -0.0244521   -0.0022706   -0.0002176
```

Without outliers:

```
Step:  AIC=-1459.04
log(mpg) ~ weight + horsepower + cylinders + acceleration

               Df Sum of Sq     RSS     AIC
<none>                      8.9097 -1459.0
- acceleration  1   0.04713  8.9568 -1459.0
+ displacement  1   0.00817  8.9015 -1457.4
- cylinders     1   0.19294  9.1026 -1452.7
- horsepower    1   0.48441  9.3941 -1440.4
- weight        1   1.33556 10.2452 -1406.7

Call:
lm(formula = log(mpg) ~ weight + horsepower + cylinders + acceleration,
    data = NoOutlierData)

Coefficients:
 (Intercept)        weight    horsepower     cylinders  acceleration
   4.2435486    -0.0002002    -0.0026427    -0.0317231    -0.0064113
```

We also used subset selection (with `regsubsets()`, in the `leaps` package) to find the best model with one predictor, two predictors, and three predictors:

Below is the output from calling the `summary(regsubsets())` function.

With outliers:

```
Subset selection object
Call: regsubsets.formula(log(mpg) ~ cylinders + displacement + horsepower +
    weight + acceleration, data = mpgdata)
5 Variables  (and intercept)
             Forced in Forced out
cylinders        FALSE      FALSE
displacement     FALSE      FALSE
horsepower       FALSE      FALSE
weight           FALSE      FALSE
acceleration     FALSE      FALSE
1 subsets of each size up to 5
Selection Algorithm: exhaustive
         cylinders displacement horsepower weight acceleration
1  ( 1 ) " "       " "          " "        "*"    " "
2  ( 1 ) " "       " "          "*"        "*"    " "
3  ( 1 ) "*"       " "          "*"        "*"    " "
4  ( 1 ) "*"       " "          "*"        "*"    "*"
5  ( 1 ) "*"       "*"          "*"        "*"    "*"
```

Without outliers:

```
Subset selection object
Call: regsubsets.formula(log(mpg) ~ cylinders + displacement + horsepower +
    weight + acceleration, data = NoOutlierData)
5 Variables  (and intercept)
             Forced in Forced out
cylinders        FALSE      FALSE
displacement     FALSE      FALSE
horsepower       FALSE      FALSE
weight           FALSE      FALSE
acceleration     FALSE      FALSE
```

```
1 subsets of each size up to 5
Selection Algorithm: exhaustive
         cylinders displacement horsepower weight acceleration
1  ( 1 ) " "       " "          " "        "*"    " "
2  ( 1 ) " "       " "          "*"        "*"    " "
3  ( 1 ) "*"       " "          "*"        "*"    " "
4  ( 1 ) "*"       " "          "*"        "*"    "*"
5  ( 1 ) "*"       "*"          "*"        "*"    "*"
```

Note that the models chosen with `regsubsets()` were the same, with or without outliers.

3.6 Choosing a final model

Note that from section 3.2, models with more than two variables had a problem with multicollinearity, which was solved by removing multiple predictors. The predictors used in that section, `weight` and `horsepower`, had no issues with multicollinearity. Moreover, the results of subset selection indicate that using exactly these predictors resulted in the best model with two predictors.

Below, we compared the AICs and adjusted R^2 values for the best models of each size as chosen by subset selection earlier. Note that removing `displacement`, `cylinders`, and (when including outliers) `acceleration` did not cause any significant losses in AIC score or adjusted R^2, so we chose this best two-predictor model as our final model. The two-predictor model has the added bonus of being a smaller model, which is easier for interpretation and is most practical for real-world testing and data collection.

Note that the scores were slightly better for the model without the outliers, so our final model also has the outliers removed. Also note that the log transformation was used, so the scores may differ from when they were calculated with a non-transformed model in the earlier sections.

With outliers:

```
[1] "For the model with  2 predictors: Adjusted R^2 is  0.786647818233046
and AIC is  -332.336104379735"
[2] "For the model with  3 predictors: Adjusted R^2 is  0.788746613178948
and AIC is  -335.21053872808"
[3] "For the model with  4 predictors: Adjusted R^2 is  0.788478332052383
and AIC is  -332.740214783723"
```

Without outliers:

```
[1] "For the model with  2 predictors: Adjusted R^2 is  0.795418578005746
and AIC is  -347.827370797289"
[2] "For the model with  3 predictors: Adjusted R^2 is  0.798659901249664
and AIC is  -353.048988999157"
[3] "For the model with  4 predictors: Adjusted R^2 is  0.798858048393737
and AIC is  -351.458054787715"
```

So our final model is:

```
Bestmod <- lm(log(mpg)~horsepower+weight, data=NoOutlierData)
```

3.7 Checking final model assumptions

Examining the first diagnostic plot (see 9), we see that the constant variance assumption is satisfied. Examining the qq-plot of the residuals, (see 10) and since the p-value for the Shapiro-Wilk test is .4925, we see that the normality assumption is satisfied.

```
Shapiro-Wilk normality test


data:  residuals(Bestmod)
W = 0.99625, p-value = 0.4925
```

Note that there are no points that could be outliers, as the studentized residuals have absolute value less than 3.

```
> which(abs(rstudent(Bestmod))>=3)
named integer(0)
```

Examining the fourth diagnostic plot with standardized residuals vs. leverage and Cook's distance as contour lines, we see that there may not be influential points (see 11). This is because there is nothing on the top right or bottom right and the Cook's distances are small. Examining the half-normal plots of the leverages (see 12) and Cook's distances (see 13), almost everything looks good. Only two points have abnormal leverages. One of these two only has a slightly abnormal leverage and is the only point to have an abnormally high Cook's distance (around 0.07). While it is substantially larger than the second-largest Cook's distance of around .04, it is still a very small value and thus is arguably not an influential point (see: "Identifying Influential Data Points").

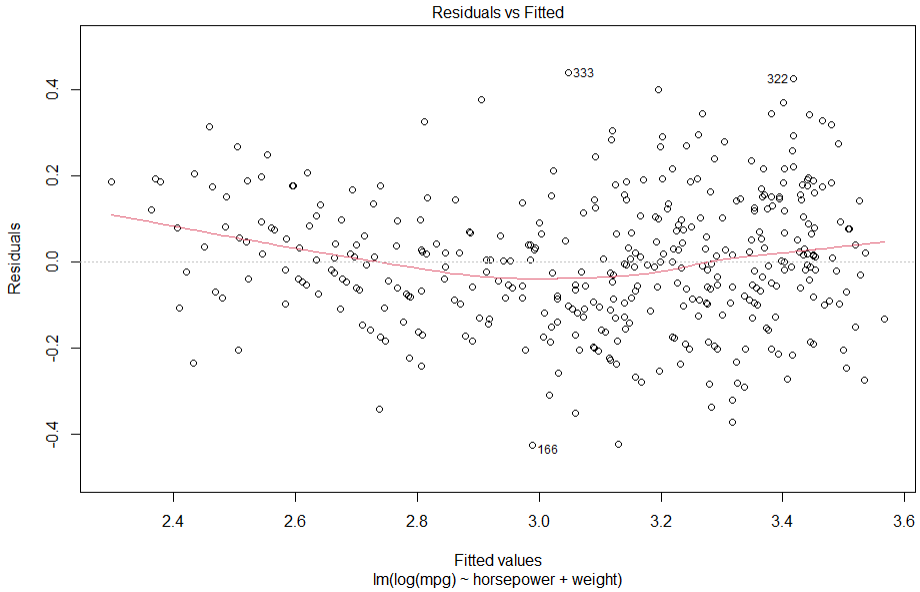

Figure 9: Final model Residuals vs Fitted

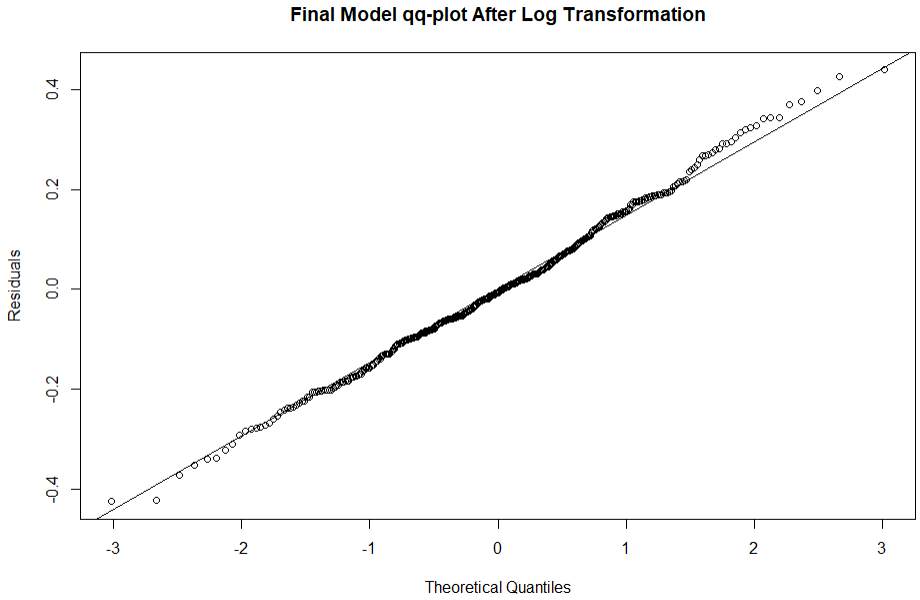

Figure 10: Final model normal QQ-plot

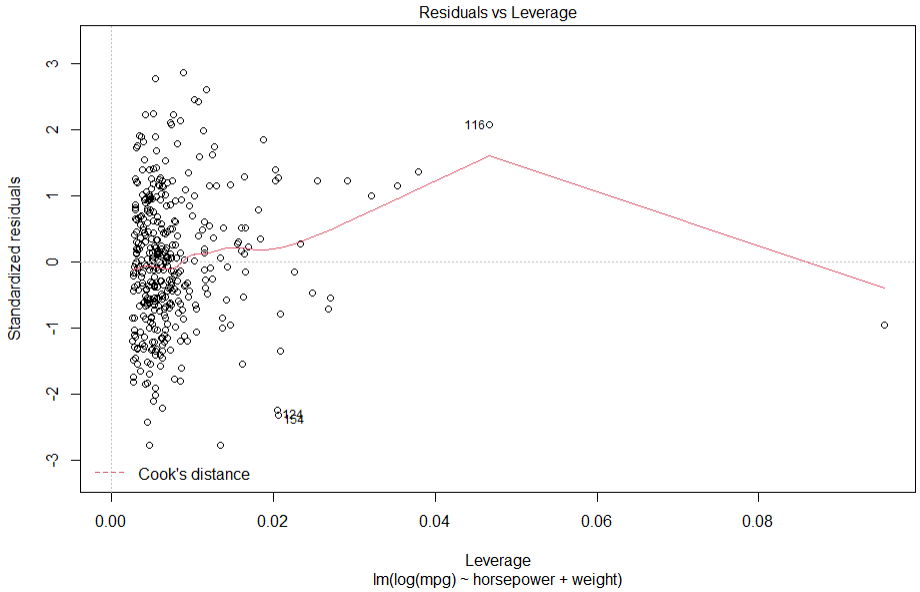

Figure 11: Final model Residuals, Leverage, and Cook's Distance

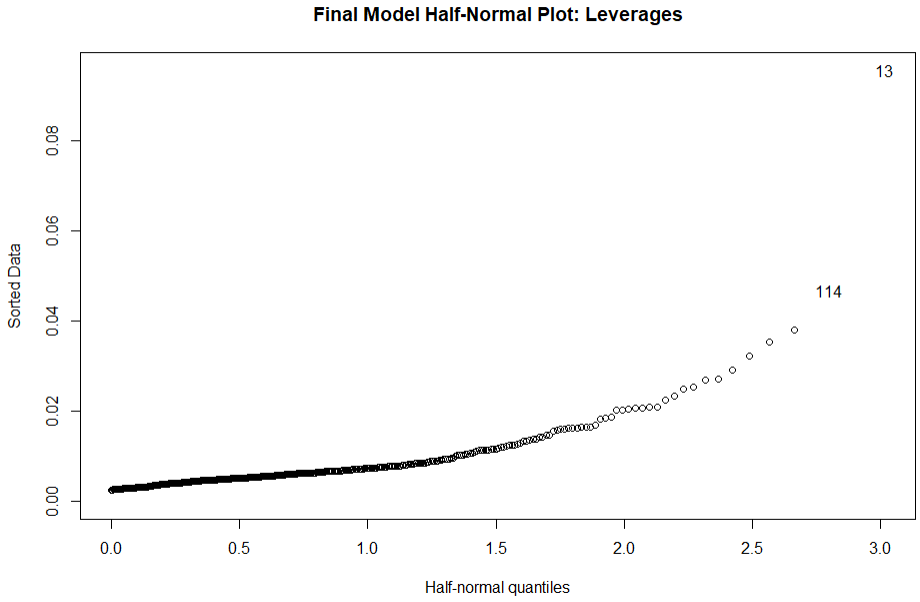

Figure 12: Final Model Half Normal Plot, Leverages

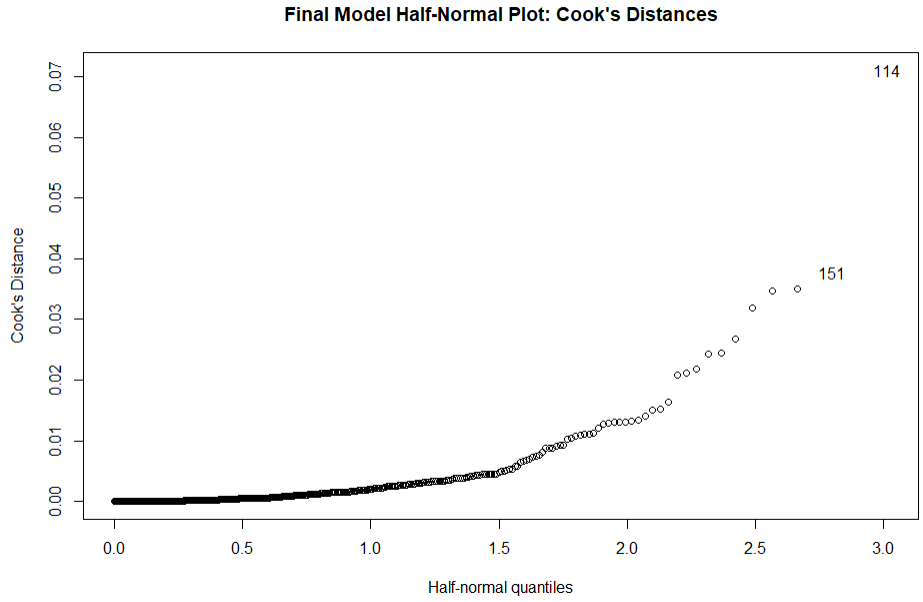

Figure 13: Final Model Half Normal Plot, Cook's Distance

Next, we show that there is an approximately linear relationship between the predictors and the transformed response. Note from the model summary below that, because the p-values associated with each predictor are well below .05 and even .01, each predictor is somehow related to the response.

```
    Call:
lm(formula = log(mpg) ~ horsepower + weight, data = NoOutlierData)

Residuals:
     Min       1Q   Median       3Q      Max
-0.42461 -0.09819 -0.00635  0.10017  0.43854

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.117e+00  2.884e-02 142.767  < 2e-16 ***
horsepower  -2.425e-03  4.031e-04  -6.017 4.14e-09 ***
weight      -2.570e-04  1.828e-05 -14.059  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 0.1537 on 386 degrees of freedom
Multiple R-squared:  0.7965,Adjusted R-squared:  0.7954
F-statistic: 755.3 on 2 and 386 DF,  p-value: < 2.2e-16
```

It remains to be shown that this relationship with the (transformed) response is indeed linear. By plotting the response against each predictor, there is in both cases a clear negative trend that looks approximately linear (see 14 and 15). Moreover, this shows that our final model approximately satisfies all of our model assumptions.

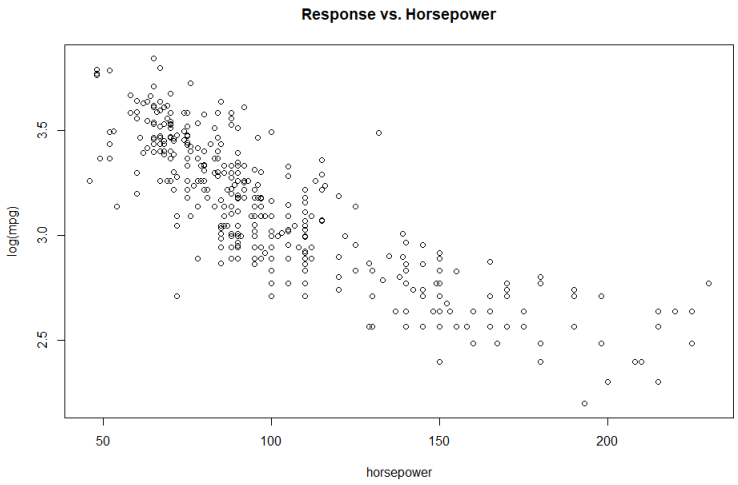

Figure 14: Response vs Horsepower

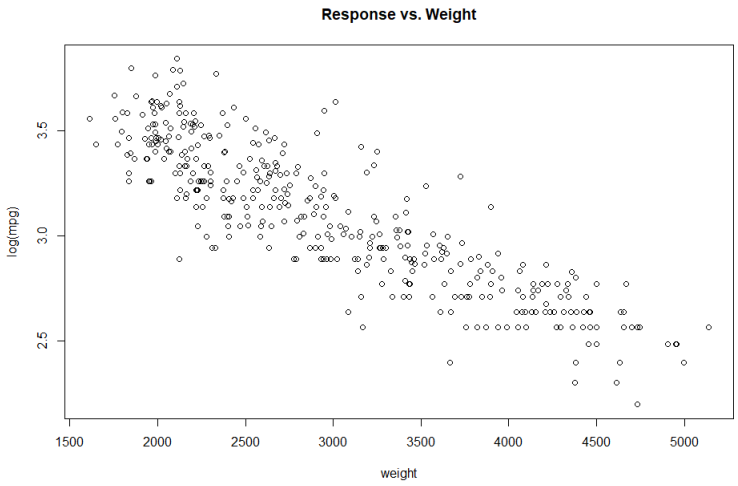

Figure 15: Response vs Weight

3.8 Principal Component Analysis

3.8.1 Variation and Relationship

As mentioned in the Methodology, principal component analysis plays a huge role in understanding large data sets. For the purposes of conducting principal component analysis, we removed the 6 missing data points. The reason for this is because since there are only 6 missing data points out of 397, we are only removing 1.5% of the data. After the removal, there will be little to no impact on the analysis because we are still left with many observations as mentioned earlier. As we construct our model, we include the variable cylinders to see the effect it has on the principal components. Before we use principal components, we define a new variable to be equal to our auto mpg data set without the 6 missing data points and then conduct the analysis.

```
> cmpg <- mpgdata[,2:6] # 6 NA points removed
> summary(cmpg)
   cylinders      displacement     horsepower        weight      acceleration
 Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00
 1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2224   1st Qu.:13.80
 Median :4.000   Median :151.0   Median : 93.0   Median :2800   Median :15.50
 Mean   :5.465   Mean   :194.1   Mean   :104.4   Mean   :2976   Mean   :15.55
 3rd Qu.:8.000   3rd Qu.:264.5   3rd Qu.:125.0   3rd Qu.:3616   3rd Qu.:17.05
 Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :24.80
> prcmpg <- prcomp(cmpg) # including cylinders
> summary(prcmpg)
```

```
Importance of components:
                            PC1      PC2      PC3   PC4    PC5
Standard deviation     856.3222 38.86338 16.15882 1.703 0.5222
Proportion of Variance   0.9976  0.00205  0.00036 0.000 0.0000
Cumulative Proportion    0.9976  0.99964  1.00000 1.000 1.0000
```

We see that the first principal component explains 99.76% of the variation in the data while the last few components account for very little of the variation. Let's observe the first principal component.

```
> round(prcmpg$rot[,1],2)
   cylinders displacement   horsepower       weight acceleration
        0.00        -0.11        -0.04        -0.99         0.00
```

We see that the data is of uneven variance and needs standardization. We can scale in R using the prcomp function.

```
> #standardized PC
> prcmpg2 <- prcomp(cmpg,scale=TRUE)
> summary(prcmpg2)
Importance of components:
                          PC1    PC2     PC3     PC4     PC5
Standard deviation     2.0176 0.8342 0.36392 0.25372 0.19139
Proportion of Variance 0.8141 0.1392 0.02649 0.01287 0.00733
Cumulative Proportion  0.8141 0.9533 0.97980 0.99267 1.00000
> round(prcmpg2$rot[,1],2)#now they are all standardized around the same value,
weights in the first pc become even now
   cylinders displacement   horsepower       weight acceleration
       -0.47        -0.48        -0.47        -0.46         0.33
```

We can see that, after scaling, the proportion of variability explained by the first component drops to 81.41%. The remaining variation is more evenly spread over the other components and now the first principal component has very similar coefficients for all the variables as opposed to before. One interpretation of this dominant first principal component is that the effects of cylinders, displacement, horsepower and weight are all inversely proportional to the response. This meaning that an increase in each of these predictors contributes towards a lower mpg which makes sense intuitively. However, we see a positive relationship with acceleration and this also makes intuitive sense. The greater amount of time a car takes to accelerate, the more miles per gallon it has and therefore will have an inverse relationship with the rest of the predictors. Now we can take a look at the second principal component.

```
> round(prcmpg2$rot[,2],2) #2nd PC
   cylinders displacement   horsepower       weight acceleration
        0.22         0.18        -0.12         0.34         0.89
```

We see a strong contrast in the relationships now between the variables and the response. Despite this contrast to the first principal component, we see that the second component only makes up for 13.92% of the variation in the data, so it is not very significant. Also, the scree plot in figure 16 reveals that about 3 principal components are sufficient in explaining the total variation as opposed to the full model.

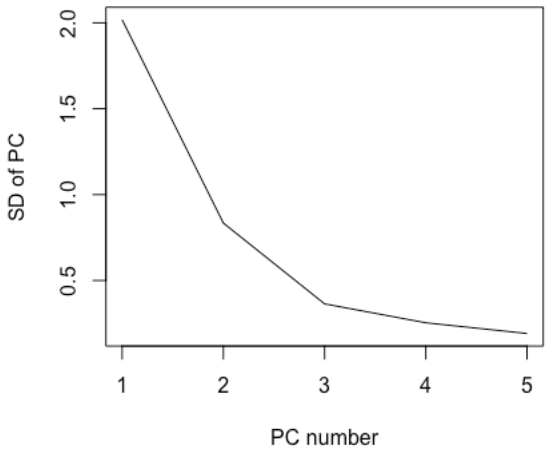

Figure 16: Scree Plot of the first five principal components

3.8.2 Principal Component Regression and Multicollinearity

It is important to show how principal components can be more reliable than full regression models due to reduced multicollinearity. As we recall from 3.2,

```
Call:
lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
    acceleration, data = mpgdata)

Residuals:
     Min       1Q   Median       3Q      Max
-11.5789  -2.8724  -0.3379   2.2690  16.3410

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   4.628e+01  2.675e+00  17.304  < 2e-16 ***
cylinders    -3.931e-01  4.123e-01  -0.954  0.34092
displacement -9.072e-05  9.083e-03  -0.010  0.99204
horsepower   -4.538e-02  1.670e-02  -2.717  0.00688 **
weight       -5.190e-03  8.179e-04  -6.345 6.24e-10 ***
acceleration -3.013e-02  1.261e-01  -0.239  0.81130
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.252 on 385 degrees of freedom
Multiple R-squared:  0.7073,Adjusted R-squared:  0.7035
F-statistic: 186.1 on 5 and 385 DF,  p-value: < 2.2e-16

> round(vif(lmod),2) ##Find VIF, serious multicollinearity
   cylinders displacement   horsepower       weight acceleration
       10.63        19.48         8.93        10.43         2.61
```

The data above shows R^2 to be 70.73% and the variance inflation factor depicts severe multicollinearity. As stated before, principal components in the model help our understanding of the relationship between the predictors and the response by reducing this severe multicollinearity. In the following output, we see how the principal components do so.

```
Call:
```

```
lm(formula = mpgdata$mpg ~ prcmpg2$x[, 1:2])

Residuals:
     Min       1Q   Median       3Q      Max
-12.9458  -2.8543  -0.3287   2.1070  16.6149

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)          23.4598     0.2197 106.772  < 2e-16 ***
prcmpg2$x[, 1:2]PC1   3.1576     0.1090  28.958  < 2e-16 ***
prcmpg2$x[, 1:2]PC2  -1.5316     0.2637  -5.808 1.32e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.345 on 388 degrees of freedom
Multiple R-squared:  0.6921,Adjusted R-squared:  0.6905
F-statistic: 436.1 on 2 and 388 DF,  p-value: < 2.2e-16

> round(vif(lmodpcr),2)
prcmpg2$x[, 1:2]PC1 prcmpg2$x[, 1:2]PC2
                  1                   1
```

Now we observe the R^2 value has decreased to 69% and the variance inflation factor decreased to 1 for both new variables, indicating no multicollinearity. This is due to the new variables being orthogonal and linear combinations of the original predictors which by design eliminate multicollinearity, thus providing us with a better interpretation of the data.

3.8.3 Cross Validation

Additionally, principal components serve an important purpose in improving our model by utilizing cross validation. Cross Validation is the method of testing the effectiveness of a model and a resampling procedure to evaluate a model (Sanjay). In doing so, we split our model into training data and testing data to conduct the analysis. If this method is successful, then it allows us to make better predictions with a smaller number of components as opposed to the full model. We proceed by partitioning our data set and observing the root mean squared error (rmse).

```
trainmpg <- mpgdata[1:300,]
> testmpg <- mpgdata[300:391,]
> #PCR
> pcrmod <- pcr(mpg~ cylinders+displacement + horsepower + weight
+ acceleration, data=trainmpg)
> pcrmse <- RMSEP(pcrmod, newdata=testmpg)
> plot(pcrmse,xlim=c(0,10),ylim=c(0,10))
> which.min(pcrmse$val)
[1] 6
> pcrmse$val[6]
[1] 7.950746
```

The output reveals that with using principal component regression, the rmse is minimized with 6 principal components. However, we see that cross validation can decrease the amount of needed principal components even further.

```
pcrmod <- pcr(mpg ~ cylinders+displacement + horsepower + weight + acceleration,
data=trainmpg, validation="CV")
> pcrCV <- RMSEP(pcrmod, estimate="CV")
> plot(pcrCV,xlim=c(0,10),ylim=c(0,10))
> which.min(pcrCV$val)
[1] 4
> ypred <- predict(pcrmod, testmpg, ncomp=4)
> rmse(ypred, testmpg$mpg)
[1] 7.964397
>
```

We note that the rmse, with cross validation, is minimized with 4 principal components as opposed to 6 previously. This demonstrates how cross validation can aid in the improvement of a regression model using the least amount of principal components. Despite the benefits utilizing this method, it is important to recognize the potential negatives of the data changing over time due to the partitions being random. However, this method of analysis still provides valuable information for enhancing a model.

4 Conclusion

We found by checking the model assumptions that after using the log transformation on the response and using only horsepower and weight as predictors, we satisfied our model

assumptions quite well and had good adjusted R^2 and AIC scores. Using this model, we see that there is a clear negative relationship between log(mpg) and each predictor, therefore there is a negative relationship between mpg and each predictor. Principal components were also used for improving the model by providing a more credible explanation of the relationship than the full model. In doing so, we noticed a negative relationship with our variables cylinders, displacement, horsepower and weight. However, we determined that acceleration had a positive relationship in the sense that a greater acceleration time is accompanied with more miles per gallon. It is important to note that satisfying the model assumptions are crucial. Without them, methods such as least square estimation and the overall meaning of the data set would be uncertain. We also had to reduce multicollinearity, remove missing data, and improve the overall model prevented this uncertainty, allowing us to determine the relationship between the predictors and mpg. With this understanding, automobile industries and manufacturers can utilize similar data sets to develop new innovations that preserve fuel consumption. Based on the study, this might include higher acceleration times and decreased amount of cylinders, displacement, weight, and horsepower for automobiles, leading towards higher miles per gallon and thus, less fuel consumption.

5 References

Dataset: UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/auto+mpg

Faraway, Julian James. Linear Models with R. Chapman & Hall/CRC, 2005.

IEA (2009). Transport, Energy and CO2: Moving Toward Sustainability. International Energy Agency, Paris, France, 418 pp

Leanse, Alex. "What Is Engine Displacement?" YourMechanic Advice, 1 Dec. 2015, www.yourmechanic.com/article/what-is-engine-displacement.

"9.5 - Identifying Influential Data Points." 9.5 - Identifying Influential Data Points — STAT 462, online.stat.psu.edu/stat462/node/173/.

Sanjay.M. "Why and How to Cross Validate a Model?" Medium, Towards Data Science, 19 Aug. 2020, towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f.

6 Appendix

6.0.1 Thi Van Nguyen's Code

```
Thi Van Nguyen
library(faraway)
linkdata= "https://raw.githubusercontent.com/vannguyen26248/
Regression-Project/main/auto-mpg.data"

mpgdata <- read.table(linkdata,header= TRUE,
                      col.names = c("mpg","cylinders", "displacement",
                                    "horsepower","weight", "acceleration",
                                    "model_year", "origin", "car_name"),na.strings="?")

mpg <- subset(mpgdata[,1:6])

###deal with missing data
##data with NA
mpg[is.na(mpg$horsepower),]
lmod <- lm(mpg~  cylinders + displacement + horsepower +
 weight + acceleration, data=mpg)
sumary(lmod)
#The standard errors are larger for the missing version because
we have less data to fit the model and so the estimates are less precise.
#deletion method
removed_NA_data <-mpg[complete.cases(mpg),]
lmodel <-lm(mpg~cylinders+displacement + horsepower + weight +
acceleration, removed_NA_data)
sumary(lmodel)
#We see that cyliders ,displacement and acceleration are not
significatn based on p_values

#Pairwise scatter plots for all predictors
pairs(~ cylinders +displacement +horsepower+weight +
acceleration,data =removed_NA_data, pch=16, cex= .5,
      main= "cylinders ,displacement, horsepower,weight,acceleration")
#We see that weight and displacement appear to be strongly related.
```

```
round(cor(removed_NA_data[,2:6]),3)
#Notify that The high correlation among some of the predictors suggests that
#multicollinearity exists.
#But displacement, cylinders and Weight contains almost the same
information (they are highly correlated).
#By adding displacement, cylinders to the simple linear regression model,
not much new information is added but the model becomes more complicated

vif(lmodel)
reducemodel <- lm(mpg ~horsepower +weight, removed_NA_data)
sumary(reducemodel)
vif(reducemodel)

#VIF <4 after removing high VIF. It doesn't work well.We try reduce
multilinearity in the regression model by using ridge regression
#and partial leaast square regresion, Principal components regression later.
##F-test check coefficent of  cylinders ,displacement ,acceleration =0
anova(lmodel, reducemodel)

#We see that p_value= 0.6416: Fail to reject H_0.


#####R^2
Rsquare <- AdjRsquare <- numeric()
########First model##########
obj1 <- lm(mpg ~horsepower, removed_NA_data)
Rsquare[1] <- summary(obj1)$r.squared
AdjRsquare[1] <- summary(obj1)$adj.r.squared
########Second model##########
obj2 <- lm(mpg ~horsepower +weight, removed_NA_data)
Rsquare[2] <- summary(obj2)$r.squared
AdjRsquare[2] <- summary(obj2)$adj.r.squared
########Third model##########
obj3 <- lm(mpg ~horsepower +weight +cylinders, removed_NA_data)
Rsquare[3] <- summary(obj3)$r.squared
AdjRsquare[3] <- summary(obj3)$adj.r.squared
```

```
########4th model##########
obj4 <- lm(mpg ~horsepower +weight + cylinders+ acceleration,
removed_NA_data)
Rsquare[4] <- summary(obj4)$r.squared
AdjRsquare[4] <- summary(obj4)$adj.r.squared
########5th model##########
obj5 <- lm(mpg ~horsepower +weight+ cylinders+acceleration+
displacement, removed_NA_data)
Rsquare[5] <- summary(obj5)$r.squared
AdjRsquare[5] <- summary(obj5)$adj.r.squared

plot(Rsquare,type="l",lwd=2, ylim = c(0.6, 0.72), xlab = "Predictors",
ylab = "R squared")
lines(AdjRsquare,lty=2,lwd=2)
abline(v=which.max(Rsquare))
abline(v=which.max(AdjRsquare),lty=2)
legend(0.65,4, c("Rsquare","AdjRsquare"), lty = c(1,2),lwd=c(2,2), merge = TRUE)
```

6.0.2 Matthew Pressimone's Code

```
rm(list=ls())
library(faraway)
library(leaps)
library(RcmdrMisc)
mpgdata <- na.omit(read.table("auto-mpg.data",header= TRUE,
                      col.names = c("mpg","cylinders", "displacement",
                                    "horsepower","weight", "acceleration",
                                    "model_year", "origin", "car_name"),
                                    na.strings="?"))
lmod <- lm(mpg~  cylinders + displacement + horsepower + weight
+ acceleration,
data=mpgdata)
```

```
summary (lmod)
plot(lmod$residuals ~lmod$fitted.values, xlab="Fitted Values",
ylab="Residuals", main="
     First Model: Residuals vs. Fitted Values")
#constant variance assumption isn't satisfied, fan shape


#chose to do a transformation to get more constant variance over
weighted least squares.


#Box-Cox transformation, full model
boxcox(lmod, plotit=T)
boxcoxmod <- lm(-2*(sqrt(mpg)-1)~ cylinders+ displacement +
horsepower + weight +
acceleration, data=mpgdata)
plot(boxcoxmod$residuals ~ boxcoxmod$fitted.values,xlab="Fitted Values",
    ylab="Residuals", main="
        Box-Cox Model: Residuals vs. Fitted Values")




#log transformation, full model
logmod <- lm(log(mpg) ~ cylinders+ displacement + horsepower +
weight + acceleration,
    data=mpgdata)
plot(logmod$residuals ~ logmod$fitted.values, xlab="Fitted Values",
ylab="Residuals",
    main="
         Log Model: Residuals vs. Fitted Values")
#constant variance is better
```

```
#check for normality

qqnorm(residuals(logmod),ylab = "Residuals",main="qq-plot After Log
Transformation")
qqline(residuals(logmod))
shapiro.test(residuals(logmod))


#check for outliers
range(rstudent(logmod))
rstudent(logmod)[which(abs(rstudent(logmod))>=3)] #see which studentized
residuals >=3,
#observations 111 and 387 are outliers


halfnorm(hatvalues(logmod), main="Half-Normal Plot: Leverages")
#some observations (13) need investigation
halfnorm(cooks.distance(logmod),
    ylab="Cook's Distance",
        main="Half-Normal Plot: Cook's Distances") #faraway
#no cook's distances need investigation... no potential influential points

#remove the outliers, and see what happens; need to see what happens
when we remove

NoOutlierData <-mpgdata[-which(abs(rstudent(logmod))>=3),]

logmodNoOutliers <- lm(log(mpg) ~ cylinders+ displacement +
horsepower + weight +
    acceleration, data=NoOutlierData)


#variable selection
stepwise(logmod,direction="backward/forward",criterion="AIC") #RcmdrMisc
stepwise(logmod,direction="forward/backward",criterion="AIC")
stepwise(logmodNoOutliers,direction="backward/forward",criterion="AIC")
```

```
stepwise(logmodNoOutliers,direction="forward/backward",criterion="AIC")

#subset selection, with outliers then without outliers
summary(regsubsets(log(mpg) ~ cylinders+ displacement +
horsepower + weight +
    acceleration, data=mpgdata))
summary(regsubsets(log(mpg) ~ cylinders+ displacement +
horsepower + weight +
    acceleration, data=NoOutlierData))
#best models of each size are the same
ModelScores <- function(Dataset,lmodel){

  TwoVarMod <- lm(log(mpg)~horsepower+weight, data=Dataset)
  ThreeVarMod <- lm(log(mpg)~cylinders+horsepower+weight,data=
  Dataset)
  FourVarMod <- lm(log(mpg)~cylinders+horsepower+acceleration+weight,
  data=Dataset)
  print("Two Variable Model VIF:")
  print(vif(TwoVarMod))
  print("Three Variable Model VIF:")
  print(vif(ThreeVarMod))
  print("Four Variable Model VIF:")
  print(vif(FourVarMod))

  paste("For the model with ",c(2,3,4),"predictors: Adjusted R^2 is ",
      c(summary(TwoVarMod)$adj.r.squared,
        summary(ThreeVarMod)$adj.r.squared,summary(lmodel)$adj.r.squared),
            "and AIC is ",
      c(AIC(TwoVarMod),
        AIC(ThreeVarMod),AIC(lmodel)))
}

#WITH outliers
ModelScores(mpgdata,logmod)
#two variable model virtually ties for best AIC and R^2, but also
#has as many predictors as possible without collinearity
```

```
#NO outliers
ModelScores(NoOutlierData,logmodNoOutliers)
#serious multicollinearity in the three/four variable model, both variables
#in the two variable model are significant. so we choose the two variable one.
#two variable has best AIC, very close to best R^2.


Bestmod <- lm(log(mpg)~horsepower+weight, data=NoOutlierData)
summary(Bestmod)
plot(Bestmod)




qqnorm(residuals(Bestmod),ylab = "Residuals",
    main="Final Model qq-plot After Log Transformation")
qqline(residuals(Bestmod))
shapiro.test(residuals(Bestmod))
which(abs(rstudent(Bestmod))>=3)
#checking for linear relationship between transformed response and predictors
plot(log(mpg)~horsepower, data=mpgdata, main="Response vs. Horsepower")
plot(log(mpg)~weight, data=mpgdata, main="Response vs. Weight")

halfnorm(hatvalues(Bestmod),
    main="Final Model Half-Normal Plot: Leverages")
halfnorm(cooks.distance(Bestmod), ylab="Cook's Distance", main="
Final Model Half-Normal Plot: Cook's Distances")
#similar story; only one point has concerning leverage. However
obs 114 has slightly
#high leverage and unusually high Cook's distance.
#approximately satisfy assumption of no influential points.
```

```
#note that removing outliers resulted in slightly improved R^2 and AIC,
#but other than that little changes.

#Conclusion: in both cases we chose model with weight and
horsepower as predictors.
#in order to best satisfy model assumptions, we used the log
 transformation on the response.
```

6.0.3 Matthew Heym's Code

```
cmpg <- mpgdata[,2:6] # 6 NA points removed
summary(cmpg)
prcmpg <- prcomp(cmpg) # including cylinders
summary(prcmpg)
#We see that the first principal component explains 86.7$\%$
of the variation in the data
#while the last few components account for very little of the variation.
 Instead, we could use just a single variable, formed by a linear
#combination described by the first principal component
round(prcmpg$rot[,1],2)
round(apply(cmpg,2,var),2) #We see the variances for displacement,
horsepower and weight dominate cylinders, could be just due
#due to the larger values resulting in higher weight, so we will standardize

#standardized PC
prcmpg2 <- prcomp(cmpg,scale=TRUE)
summary(prcmpg2)
#We can see that, after scaling, the proportion of variability explained
by the first
#component drops to 80.19$\%$. The remaining variation is more evenly
spread over
#the other components. The first principal component has very similar coefficients
#for all the variables. One interpretation of this dominant first principal component
 is that the effects of
#cylinders,displacement, weight, and horsepower are all proportional in t
he sense that an increase in each of these predictors
#contributes towards a lower mpg which makes sense intuitively.
```

```
round(prcmpg2$rot[,1],2)#now they are all standardized around the
same value, weights in the first pc become even now
#First principal component explains that cylinders, displacement,
horsepower, and weight all negative rlshtnp with mpg
round(prcmpg2$rot[,2],2) #2nd PC
round(apply(cmpg,2,var),2)

#After standardizing, we can observe how many PC's are necessary
for our model using a scree plot
plot(prcmpg2$sdev,type="l",ylab="SD of PC", xlab="PC number")
# By the looks of it, three principal components are sufficient to better the model

#The PC is better to interpret the meaning of the predictors because
the full regression model can be unclear in the sense
#that there is a lot of multicollinearity present with R^2=70$\%$. Therefore
because the PCA reduces it, it is more meaningful
#Observe below

lmod <- lm(mpg~ cylinders + displacement + horsepower + weight
+ acceleration, data=mpgdata)
summary(lmod)
round(vif(lmod),2) ##Find VIF, serious multicollinearity


##PC regression with first two PCs, with removal of 6 missing data
lmodpcr <- lm(mpgdata$mpg ~ prcmpg2$x[,1:2])
summary(lmodpcr)
round(vif(lmodpcr),2)
lmodpcr <- lm(mpgdata$mpg ~ prcmpg2$x[,1])




trainmpg <- mpgdata[1:300,]
testmpg <- mpgdata[300:391,]
```

```
newlmod <- lm(mpg~ displacement + horsepower + weight
+ acceleration, data=trainmpg)
rmse <- function(x,y) sqrt(mean((x-y)^2))
##RMSE on the training data >> 2.879217
rmse(fitted(newlmod), trainmpg$mpg)

##RMSE on the test data >> 7.192799
rmse(predict(newlmod,testmpg), testmpg$mpg)
#RMSE is much worse for the test data, indication of overfitting,
we can reduce this with PCR

##compute the PCA on the training sample predictors
pcrmpgtrain <- prcomp(trainmpg[,2:6])
##examine the standard deviations of the principal components
round(pcrmpgtrain$sdev,3) #most of the variation can be
explained using the first few pc's


require(pls)
#PCR
pcrmod <- pcr(mpg~ cylinders+displacement + horsepower +
weight + acceleration, data=trainmpg)
pcrmse <- RMSEP(pcrmod, newdata=testmpg)
plot(pcrmse,xlim=c(0,10),ylim=c(0,10))
which.min(pcrmse$val)
pcrmse$val[6]
rmse(predict(pcrmod,trainmpg), trainmpg$mpg)
rmse(predict(pcrmod,testmpg), testmpg$mpg)


pcrmod <- pcr(mpg ~ cylinders+displacement + horsepower +
weight + acceleration, data=trainmpg, validation="CV")
pcrCV <- RMSEP(pcrmod, estimate="CV")
plot(pcrCV,xlim=c(0,10),ylim=c(0,10))
which.min(pcrCV$val)
ypred <- predict(pcrmod, testmpg, ncomp=4)
```

```
rmse(ypred, testmpg$mpg)

ytpred <- predict(pcrmod, trainmpg, ncomp=4)
rmse(ytpred, trainmpg$mpg)
```