Student Name: Brandon Russell

Submission Date: 7/3/18

DBST 667 – Data Mining

Dr. Irene Tsapara

**Week 7 Individual Exercise**

**Deliverables:** Two Files: (1) Submit this lab report with answers to all questions including output screenshots into the ‘Individual Exercises Week 7’ assignment folder. (2) Submit an R script that contains all commands with comments that briefly describe each commands purpose.

**Grading: This exercise is worth 2% of the course grade.** All questions must be answered in your own words with any paraphrased references properly cited using in-text citations and a reference list as needed. In addition, grammatical and spelling errors may affect the grade.

**Part 2** – **Run an exercise on the *imports-85* dataset from imports-85.csv (note again that we are NOT using the credit approval nor the vertebral column dataset this week), completing this report and providing the commands, output screenshots, and discussion/interpretation as requested. Ensure that all commands are saved in this report AND in an R script.**

**For Reference:** [**UCI Machine Learning Repository: Imports 85**](http://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.names)

1. **Introduction:** 
   1. **Identify the dependent variable and independent variables in the imports-85 data set.**

After performing a visual analysis of the 26 variables in the imports-85 dataset, it appears our dependent variable is the price. I came to this conclusion based on the other attribute values. Most of the other variables, except for symbolling and normalized\_losses which are calculated statistics, are tangible attributes defining the characteristics of the car. It is these characteristics we can use to infer the price of the vehicle.

* 1. **Based on what you have learned this week about multiple linear regression, provide a one-paragraph masters-level response describing what you anticipate that the lm algorithm will accomplish for the imports-85 data? Be specific about the behavior and structure of multiple linear regression model.**

Simple linear regression is an algorithm we can apply to a set of data with the hope that knowing one variable, X, will allow us to define, or predict, an unknown value Y based on what has been seen in other results from a training set. For example, if cars with a similar size engine in a dataset also have similar fuel efficiency we might be able to apply a linear model to these variables, stating that fuel efficiency shows a linear growth when plotted with engine size. Multiple linear regression does the same function, but with the addition of multiple X values (James, Witten, Hastie, & Tibshirani, 2013). For the imports-85 dataset, I believe there will be some correlation between Price and the other attributes, though not necessarily all. For example, we might see that cars with 4 doors, a gas engine, and better highway miles per gallon have a certain price, while two-door diesel engine cars have another price.

1. **Data Pre-Processing: Load the imports-85 data into R Studio using the read.csv command (*do not use File > Import Dataset > From CSV in the R Studio GUI as this uses read\_csv() resulting in significant different variable types!!!*).**
   1. **Run the commands to remove the following variables: engine\_type, make, num\_of\_cylinders, fuel\_system. Include the commands and output screenshot.**

**Command(s): >**

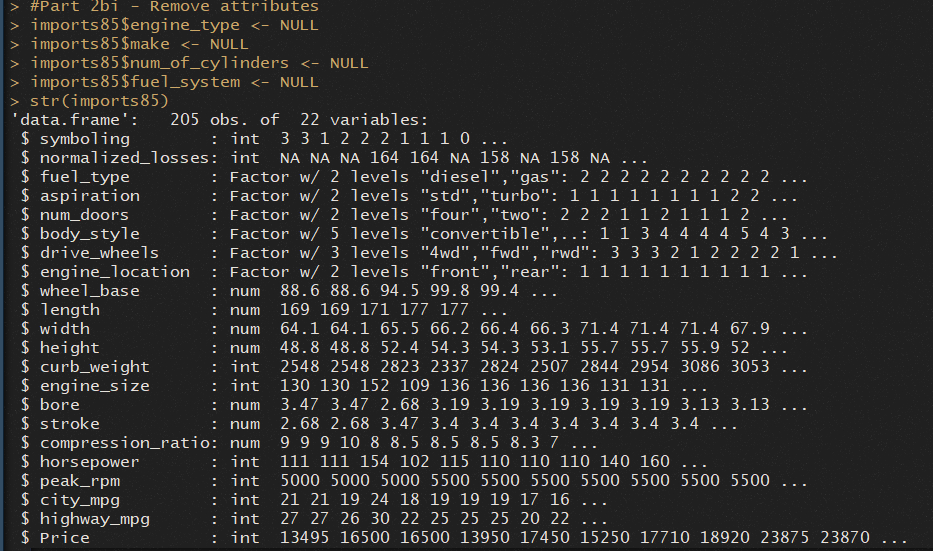
**imports85$engine\_type <- NULL**

**imports85$make <- NULL**

**imports85$num\_of\_cylinders <- NULL**

**imports85$fuel\_system <- NULL**

**Output:**



* 1. **What additional data pre-processing (if any) does the lm() method require for the imports-85 data? Include the commands you ran and the output screenshot.**

**Command(s): >**

**#Verify no empty fields**

**apply(imports85, 2, function (imports85) sum(is.na(imports85)))**

**#Part 2bii - Replace empty fields with attribute mean**

**imports85$normalized\_losses[is.na(imports85$normalized\_losses)]<-0**

**imports85$bore[is.na(imports85$bore)]<-mean(imports85$bore, na.rm=TRUE)**

**imports85$stroke[is.na(imports85$stroke)]<-mean(imports85$stroke, na.rm=TRUE)**

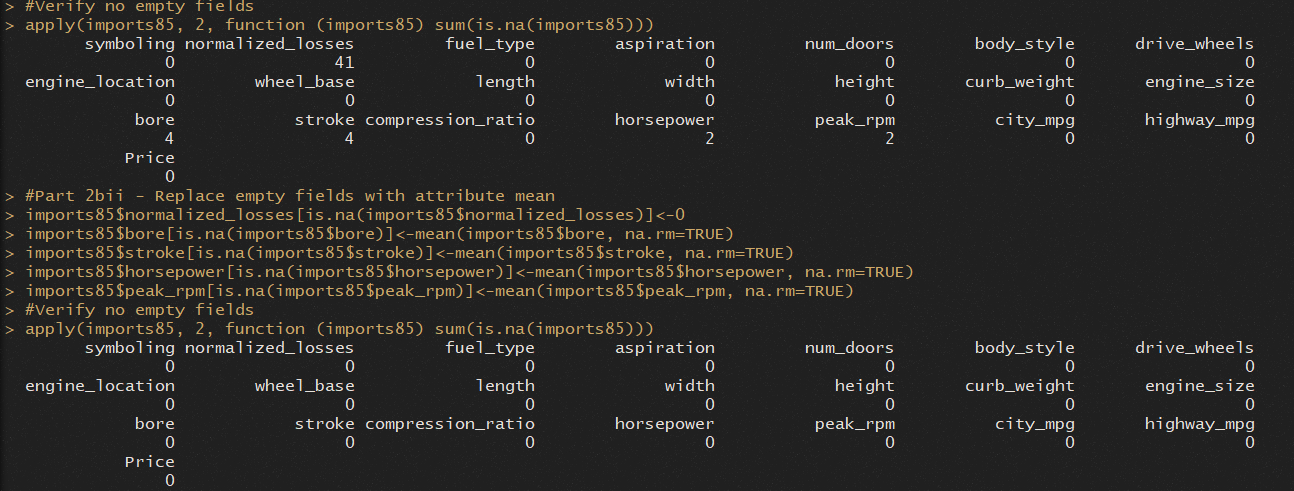
**imports85$horsepower[is.na(imports85$horsepower)]<-mean(imports85$horsepower, na.rm=TRUE)**

**imports85$peak\_rpm[is.na(imports85$peak\_rpm)]<-mean(imports85$peak\_rpm, na.rm=TRUE)**

**#Verify no empty fields**

**apply(imports85, 2, function (imports85) sum(is.na(imports85)))**

**Output:**



1. **Multiple Linear Regression – Running the Method with Training Data:**
   1. **Run ‘set.seed(12345)’ and then split the data into a training set consisting of 70% of the instances and a test set containing the remaining 30% of the instances. Includes the commands below.**

**Commands: >**

**set.seed(12345)**

**ind <- sample(2, nrow(imports85), replace = TRUE, prob = c(0.7, 0.3))**

**train.data <- imports85[ind == 1, ]**

**test.data <- imports85[ind == 2, ]**

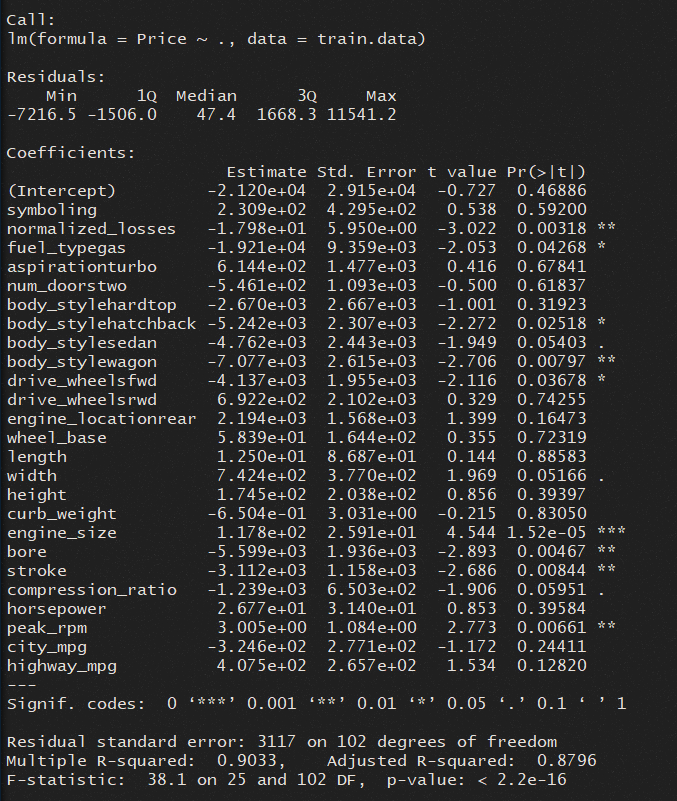
* 1. **Run the lm() function to build the multiple linear regression model storing the results in a variable called ‘mlr\_model’*.* Include the command you ran and a brief discussion about the default input parameters used.**

**Command: > mlr\_model <- lm(Price~., train.data)**

**Discussion:** In the above command we are assigning our linear model to the mlr\_model variable. Inside of the lm() command are a few components. The first part, “Price~.”, is the formula for our model itself. This is telling us Price is the dependent variable. Then on the other side of the tilde, we see a period. This is shorthand for include all other variables in the provided dataset. Since this is our first linear model it makes sense to include all variables. After the comma, we specify we want to use our training dataset.

* 1. **Run the command ‘summary(mlr\_model)’. Include the output screenshot and answer the following questions:**

**Output:**



**How does the model represent the relationship between dependent and independent variables in the import-85 dataset?**

Looking at the summary output, there is a lot of information, but it all relates to how we represent our standard linear regression model, see in formula 1 below (James, Witten, Hastie, & Tibshirani, 2013). We do not see the dependent variable Price since this is the variable we are attempting to predict based on the other X variables, i.e. symboling, length, and so on. The values we find here would be added to the linear regression model to calculate with X to determine Y.

(1)

**How does the method handle categorical variables?**

When we supply categorical data fields to a linear regression model it creates dummy variables that can be calculated as continuous variables in the linear model (James, Witten, Hastie, & Tibshirani, 2013). For example, the body\_style attribute there are 5 attributes. In the summary we see body\_stylehardtop, body\_stylehatchback, body\_stylesedan, and body\_stylewagon. These are the dummy variables created for each factor. We do not see one for the “convertible” level because it is the reference value and can be calculated with just the intercept.

**What does the residuals section of the output mean?**

At the top of the summary, we see the residuals section. Residuals are the difference between an actual versus predicted value, in other words the amount of error between the plotted regression line and plotted data points, if we visualize it (James, Witten, Hastie, & Tibshirani, 2013). The summary provides the minimum, 1st and 3rd quartiles, median and maximum values seen in the calculated residuals of our model.

**What are the coefficients and what do they mean?**

The coefficients are the numbers below the estimate column, except for the intercept value. These are the slopes for each independent attribute and are used in the linear model formula to calculate Y when multiplied by X, as seen in formula 1 above (James, Witten, Hastie, & Tibshirani, 2013).

**What is an intercept and what does it mean?**

The intercept is the first value in the Estimate column of the summary. As mentioned before, this is the value in the linear model formula 1 which is added to the slope times the X value to determine Y. This is the Y value when X is equal to 0 (James, Witten, Hastie, & Tibshirani, 2013).

**What do the p-values tell about the significance of each variable?**

The p-value tells us the probability of finding a value equal to or greater than the t value. If the p-value is small it is telling us there is a smaller chance the relationship is occurring by chance. The default threshold for determining significance is 95%, so anything below .05 will have an asterisk next to it, with other codes as noted at the bottom of the coefficient section (James, Witten, Hastie, & Tibshirani, 2013).

**What is the overall accuracy of the model?**

In our summary, there are three values which will assist with determining our model accuracy: residual standard error (RSE), R2/adjusted R2, and the F-statistic. RSE is telling us the lack of fit for our model, in other words the amount of error we can expect to see when predicting price. In this case, we have 3,117 which will be the amount in the price we might be off by. R2/adjusted R2 tells us how much variance in Y can be explained by X, in other words how well the model fits the data. The adjusted version accounts for multiple independent variables to prevent the output from growing too much. The F-statistic assesses how strong the relationship is between the dependent to independent variables. Higher values are better, though how much higher than a value of 1 to know if the model is performing well is dependent on how much data is being tested (James, Witten, Hastie, & Tibshirani, 2013). In our model, we have an RSE of 3,117, R2 of 0.90, adjusted R2 of 0.88, and an F-statistic of 38.1. These might be good indicators of an accurate model, but we would need to run additional test for comparison.

1. **Multiple Linear Regression – Evaluate the Model with Test Data:** 
   1. **Run the command to evaluate the ‘mlr\_model’ on the imports-85 *test* data Include the command below.**

**Command: > pred <- predict(mlr\_model, test.data)**

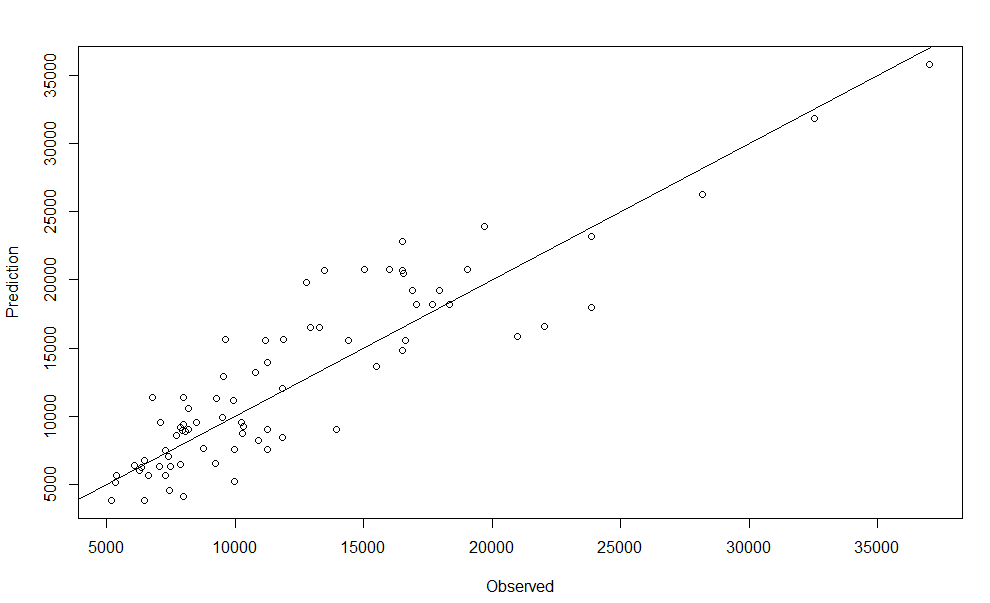
* 1. **Run the command to build the predicted vs. actual (observed) value scatter plot. Add a diagonal line to this plot. Include the commands and the final plot with the diagonal line below.**

**Commands: >**

**plot(test.data$Price, pred, xlab = "Observed", ylab = "Prediction")**

**abline(a = 0, b = 1)**

**Output:**



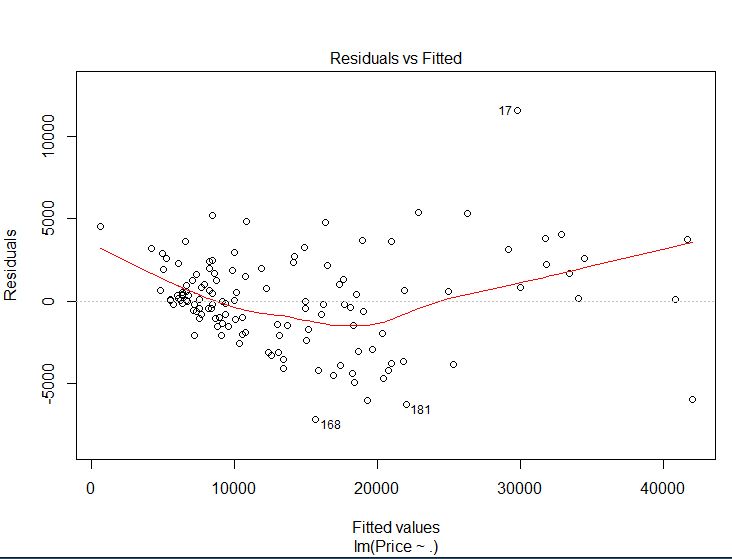
* 1. **What does the distance between points and the diagonal line tell us about the accuracy of the prediction?**

The plot above shows us our actual values on the x plane and predicted values from our model on the y plane. With our diagonal line, any points here are where x and y are equal, in other words where our actual and predicted values are equal. This means the closer our points are to the line the more accurate of a prediction. The vertical distance from the line indicates the error. Overall, the more points we have near the line the more accurate our model is.

1. **Multiple Linear Regression – Residual Plots:**
   1. **Run the ‘plot(mlr\_model)’ command to build the residuals plots. Interpret at least one of the plots. Include the command, the plot, and the interpretation of that plot below.**

**Command: > plot (mlr\_model)**

**Output:**



**Interpretation:**

In the above plot, our residuals vs fitted plot gives us an indication of whether a model has a linear relationship or not. For the most part, we want to see our data points equally spread out with no patterns and a horizontal line. If there were an indication of a distinct pattern it might indicate a non-linear relationship (James, Witten, Hastie, & Tibshirani, 2013). In our plot above we see a slight U-shape in the red line, but I believe it is not enough and our data does, in fact, exhibit a linear pattern.

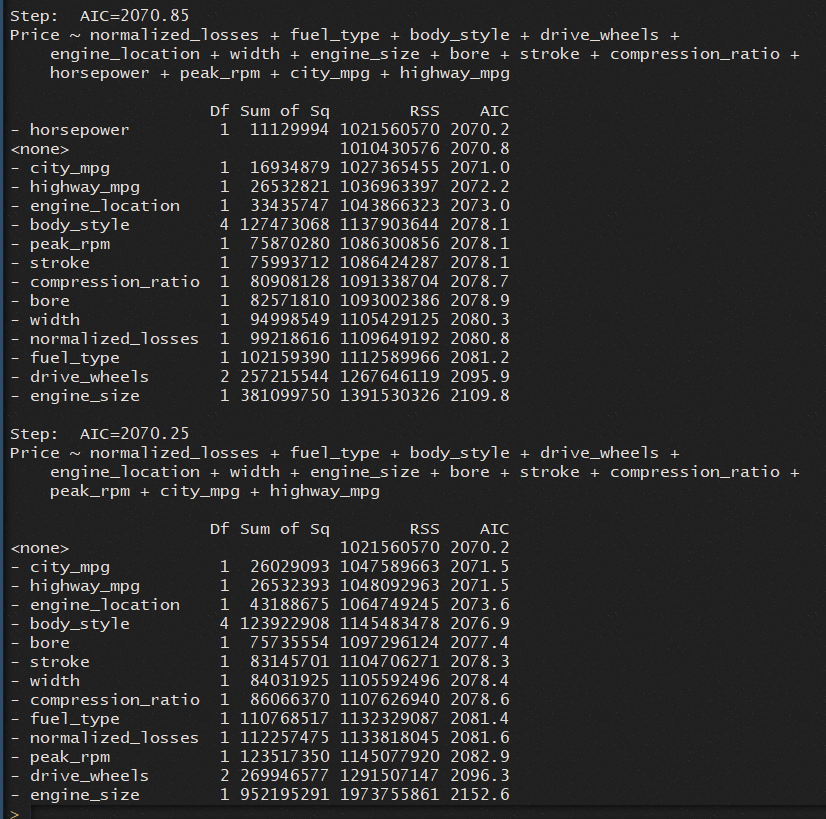
1. **Multiple Linear Regression – Minimum Adequate Model:**
   1. **What is the minimal adequate model? Why do we build it? Provide a one-paragraph, masters-level response.**

Finding our minimal adequate model is simply a process of determining which variables should be used in a model to provide the best accuracy, without using any unnecessary variables that might introduce randomness. There are different ways of performing this process and different ways to measure it. For example, we can use the R-squared value and the Akaike information criterion (AIC) to measure a model’s accuracy. To complete the process, we could apply a forward selection method which begins with the null model and then begins adding variables one at a time and assess the results. Another option is backward selection where you start with all the variables in the model and begin removing them one at a time, based on their p-value, while assessing for accuracy after each action. There is also a mixed option that starts with forward selection but also assesses the p-value (James, Witten, Hastie, & Tibshirani, 2013).

* 1. **Run the command to build the minimum adequate model and store the model in a variable named ‘mlr\_model\_min’. Include the command and output screenshot.**

**Command: > mlr\_model\_min <- step(mlr\_model, direction="backward")**

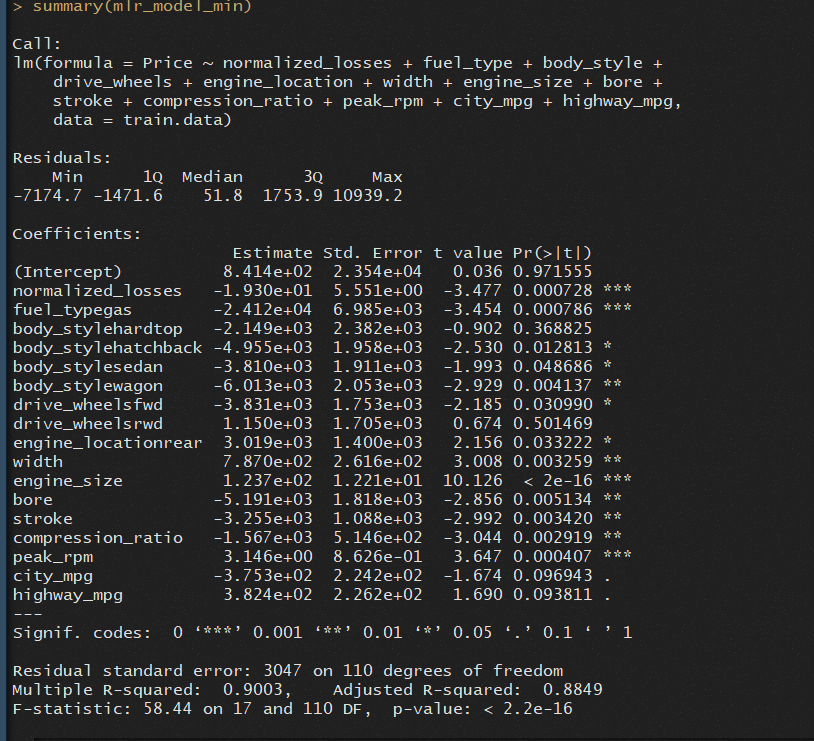
**Output:**



* 1. **Run the ‘summary(mlr\_model\_min)’ command. Include the command, output screenshot, and answers to the following questions:**

**Command: > summary(mlr\_model\_min)**

**Output:**



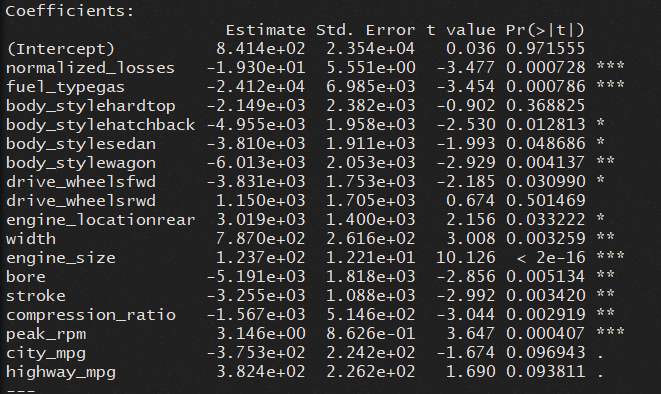
**Which variables were eliminated and which variables remain?**

The table below shows the original regression model on the left with 25 variables and on the right the minimum adequate model with 17 variables. I highlighted the ones on the left in yellow that were removed in the model on the right for readability.

|  |  |
| --- | --- |
| mlr\_model | mlr\_model\_min |
| aspirationturbo |  |
| body\_stylehardtop | body\_stylehardtop |
| body\_stylehatchback | body\_stylehatchback |
| body\_stylesedan | body\_stylesedan |
| body\_stylewagon | body\_stylewagon |
| bore | bore |
| city\_mpg | city\_mpg |
| compression\_ratio | compression\_ratio |
| curb\_weight |  |
| drive\_wheelsfwd | drive\_wheelsfwd |
| drive\_wheelsrwd | drive\_wheelsrwd |
| engine\_locationrear | engine\_locationrear |
| engine\_size | engine\_size |
| fuel\_typegas | fuel\_typegas |
| height |  |
| highway\_mpg | highway\_mpg |
| horsepower |  |
| length |  |
| normalized\_losses | normalized\_losses |
| num\_doorstwo |  |
| peak\_rpm | peak\_rpm |
| stroke | stroke |
| symboling |  |
| wheel\_base |  |
| width | width |

**What are the coefficients and the intercept? What do the coefficient and intercept mean?**

The image below shows the intercept and all the coefficients for the new model. The intercept is what Y is equal to if X is 0 and is the first part of the linear formula we discussed in formula 1 above. The slope is a multiple of X, indicated by the numbers next to each variable under the Estimate attribute. This is the second number used in formula 1 to determine the value of Y in our linear model (James, Witten, Hastie, & Tibshirani, 2013).



**Compare the prediction accuracy of the minimum adequate model with the prediction accuracy of the original model. Provide a one-paragraph, masters-level response.**

In the original model, our RSE was 3,117, the R-squared and adjusted R-squared were 0.90 and 0.88, and the F-statistic was 38.1. These three values are good indicators of the overall accuracy of our model. RSE tells us the lack of fit, R-squared the variance of Y based on X, and F-statistic the relationship between the dependent and independent variables. In our new model, the RSE was 3,047, R-squared and adjusted R-squared were 0.90 and 0.88, and the F-statistic was 58.44. Although the R-squared and adjusted R-squared values did not change, our RSE dropped by 70 and the F-statistic went up 20. This indicates our second model did improve in overall accuracy (James, Witten, Hastie, & Tibshirani, 2013).

1. **New Instance:**
   1. **Suppose that we have a new car added to the imports-85 data set. We know the values of the independent variables. How would you use the model to predict the value of the dependent variable for the new car? (Hint: Use the lessons learned and hints from the prior week to complete this exercise). Include the command you would run below:**

**Command: >**

**#Part 2gi - New Instance**

**NewCarTest <- imports85[1,] #Move row 1 into new data frame**

**str(NewCarTest) #Get some of the values**

**NewCarTest$Price <- NULL #Remove price**

**NewCarTest$symboling <- 2 #Change some of the values around**

**NewCarTest$wheel\_base <- 99.8**

**NewCarTest$width <- 70**

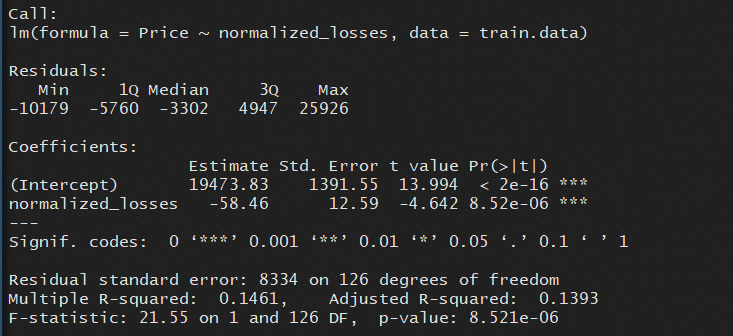
**NewCarTest$height <- 50**

**newpred <- predict(mlr\_model\_min, NewCarTest) #Run prediction**

**newpred #Get prediction - 26,368.86**

1. **Summary:**
   1. **Is the multiple linear regression method appropriate for predicting the values of dependent variables in the imports-85 dataset? Explain why or why not. Provide a one-paragraph, masters-level response.**

Based on our analysis I would say multiple linear regression is a suitable method for the imports-85 dataset. When analyzing our residuals vs fitted plot from before we saw an indication for a linear model. Additionally, if we examine the 17 variables in the minimum adequate model we will see that 13 of these variables had a p-value of less than 0.05, indicating their significance. We also saw an increase in accuracy when using backward stepwise regression to remove 8 insignificant variables. As additional proof, if we were to conduct simple linear regression of price with only the normalized losses variable we would see a significant drop in accuracy. Looking at the image below, we can see our RSE would be 8,334, R-squared and adjusted R-squared 0.15 and 0.14, and the F-statistic would be 21.55. Based on this information we can conclude multiple linear regression is the best model for this data.



* 1. **(Not graded) Which part of this exercise did you find the most challenging and what steps did you take to resolve the challenge?**

In section g it seemed like you were asking for a single command, but I did not know how to create new data and run a prediction with a single command. Provided the answer based on how I built a new test and then ran the prediction.

# References

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.