**Analysis of the Automotive Industry**

**Authors:** Nicole Assenza, Andrew Taylor and Ranjan Karki

In 2021, the worldwide automotive industry had an estimated value of almost 3 trillion dollars. More than 60 international manufacturers compete for their share of the market which sells roughly 76 million cars every year. As the amount of relevant data surrounding the industry continues to grow at an exponential rate, these companies are continually looking for reliable trends that yield even the slightest competitive advantage. This comprehensive report is intended to aid one such company in analysis and interpretation of the available data through multiple regression and nonparametric modeling.

The data set of interest includes 11,914 records of discrete automobiles in production between 1990 and 2017. Each record hosts data points for 16 variables (9 numerical and 7 categorical) such as sale price (MSRP), Engine HP, Manufacturer, and “popularity score” to name a few. The full list of variables and their descriptions can be seen in **Table T-1** below.

**Table T-1**

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| MSRP | Numeric | Sale Price (Response variable) |
| Car Make | Factor | The company that made the car. Ex: Honda, Toyota, etc. |
| Car Model | Factor | The model of the car. Ex: 4Runner, Accord, etc. |
| Year | Numeric | Year the car was produced |
| Engine Fuel Type | Factor | Type of fuel the car accepts. Ex: Regular unleaded, Diesel, etc. |
| Engine HP | Numeric | Horsepower of the car’s engine. |
| Engine Cylinders | Factor | Number of cylinders in the car’s engine. |
| Transmission Type | Factor | Type of transmission in the car. Ex: manual, automatic, etc. |
| Driven Wheels | Numeric | Wheels powered by engine. Ex: Front Wheel, Rear Wheel, etc. |
| Number of Doors | Numeric | The number of doors that the car has. Ex: 2, 4, etc. |
| Market Category | Factor | Various special factors for each car. Ex: Exotic, Luxury, etc. |
| Vehicle Size | Factor | The size of the vehicle. Ex: Midsize, Large, Compact, etc. |
| Vehicle Style | Factor | Body type of the vehicle. Ex: Coupe, Convertible, etc. |
| Highway MPG | Numeric | Fuel efficiency on the highway in MPG |
| City MPG | Numeric | Fuel efficiency in the city in MPG |
| Popularity | Numeric | A numerical popularity score for each car. |

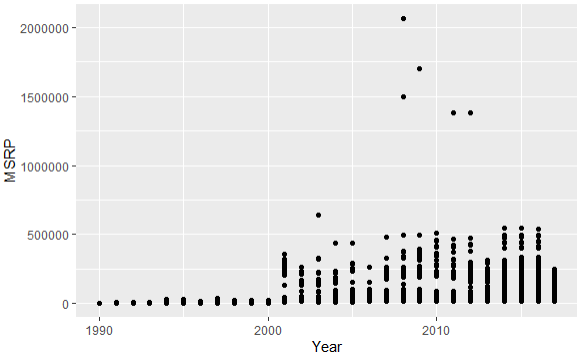
**Objective #1**

For the first objective of our analysis, we will assess the correlations of these variables to our response, the MSRP sale price. Then using various selection methods, we will build a simple regression model that use the identified variables to predict a car’s MSRP with moderate accuracy, but more importantly, in a way that is easy to interpret. To close out our first objective we will pick apart the chosen model and describe its characteristics and practical meaning.

In order to have confidence in any future analysis or model building, the data must first be cleaned of errors, blanks, N/As, etc. For this particular dataset there were few enough of these null values that the records in question could be verified online at various car manufacturer websites to obtain and insert the correct values. In total we identified N/As in 69 records for “Engine HP”, 30 records for “Engine Cylinders”, and 6 records for “Number of Doors”. Furthermore, there were 3 records with missing “Engine Fuel Type” and 1 ecord with an incorrect value for “Highway MPG”. Once the missing data was resolved, the data was further cleaned by altering the data types of certain variables to fit our analysis strategy. By converting the values of “Engine Cylinders” for all electric vehicles to “0”, this variable is able to retain its numerical quality. Other variables were converted to factors to allow for analysis of particular levels including: Make, Engine Fuel Type, Transmission Type, Driven Wheels, Vehicle Size, Vehicle Style, and Market Category.

Now that the data is wrangled and cleaned, the exploratory data analysis can begin to better understand the effects each variable has on the response. After reviewing a matrix of correlation plots, the team was able to identify a few particulars that could be altered to assist in our final models. One such example is the relationship between “Year” and “MSRP”. As seen below in **Plot P-1**, the MSRP values after the year 2000 appear drastically different than those before it. It appears there was a change after the year 2000 in how this set of data was captured or recorded which poses an issue. Therefore, the data with “Year” less than or equal to 2000 was dropped from the dataset for the following practical reason: The consumers of this analysis are most likely interested in using these models to apply and evaluate current and future cars. The results of trends seen more than 20 years ago would probably have little effect (compared to recent trends) on strategic actions and analysis in the current automotive landscape

**Plot P-1**



Correlations to MSRP from other variables are not as visually obvious in their current portrayal. By taking the log transform of MSRP, much of the relationships become clearer and easier to see. However, this will make the interpretation of our results harder to comprehend by consumers which are not familiar with various types of statistical interpretations. Therefore, in the interest of making our model as easy to understand for the masses, we will save any suggested transforms for objective #2. Taking the log of our response still maintains a level the of interpretability sought after, however, any further transforming of explanatory variables will make it more difficult to understand the models’ overall practical interpretation. So, we will avoid any further transformations (at least in Objective #1) even though they may increase the models’ fit and predictive power. Similarly, certain variables will also be too complex to interpret simply based on their data type and vast number of factor levels. For this reason, categorical variables such as “Make” “Model” and “Market Category” will also be temporarily dropped from the analysis.

The remaining data variables can now be used to build a few simple models for comparison. In order to determine which of the remaining 12 variables should be included in the regression models, step-wise, forward, and backward selection techniques were used to evaluate each variable’s statistical significance as it relates to the impact it has on MSRP. The table below (**Table T-2**) shows the suggested variables using each selection method:

**Table T-2**

|  |  |  |
| --- | --- | --- |
| **Stepwise** | **Forward** | **Backward** |
| Transmission Type | Transmission Type | Transmission Type |
| City MPG | City MPG | City MPG |
| Number of Doors | Number of Doors | Number of Doors |
| Driven Wheels | Driven Wheels | Driven Wheels |
| Popularity | Popularity | Popularity |
| Engine Fuel Type | Engine Fuel Type | Engine Fuel Type |
|  | Engine Cylinders | Engine Cylinders |
|  |  | Engine HP |
| **AIC** | | |
| 12999.9 | 8871.1 | 1999.8 |
| **MSE** | | |
| 1.83 x 10^9 | 1.09 x 10^9 | 9.27 x 10^8 |

To determine which of the models has a better fit, the AIC summary statistic was compared between the three. Aside from fit, we are also interested in how each model can perform in predicting “new” MSRP values given the chosen variables. The metric to quantify this statistic is the Average (Mean) Square Error (ASE). Prior to building these models, the data was split into an 80% training set, 10% testing set, and 10% validation set. This proved useful as each of the models were built using the training set and can now be used to predict the values in the test set and compared for accuracy. Looking back at our **Table T-2**, we can conclude that the variables selected with the Backward Selection Method provides us with a better fitting model (AIC = 1999.8) and a better prediction model (MSE = 9.27 x 108).

With our final model chosen for the first objective, we can now write out our regression equation and start to apply practical meanings and interpretations:

log(MSRP) = 9.96

+ 0.26 x Engine HP\*\*\*

- 0.0908 x Transmission Type

+ 0.014 x City MPG\*\*\*

– 0.13 x Number of Doors\*\*\*

– 0.088 x Driven Wheels

– 0.00001 x Popularity\*\*\*

– 0.026 x Engine Fuel Type

Where:

Transmission Type of “Automated Manual“ = 1

Transmission Type of “Automatic” = 2

Transmission Type of “Direct Drive” = 3

Transmission Type of “Manual” = 4

Driven Wheels of AWD = 1

Driven Wheels of 4WD = 2

Driven Wheels of FWD = 3

Driven Wheels of RWD = 4

Engine Fuel Type of “Diesel” = 1

Engine Fuel Type of “Electric” = 2

Engine Fuel Type of “Flex-Fuel (premium unleaded recommended)” = 3

Engine Fuel Type of “Flex-Fuel (premium unleaded required)” = 4

Engine Fuel Type of “Flex-Fuel (unleaded)” = 5

Engine Fuel Type of “Flex-Fuel (unleaded/natural gas)” = 6

Engine Fuel Type of “Natural Gas” = 7

Engine Fuel Type of “premium unleaded (recommended)” = 8

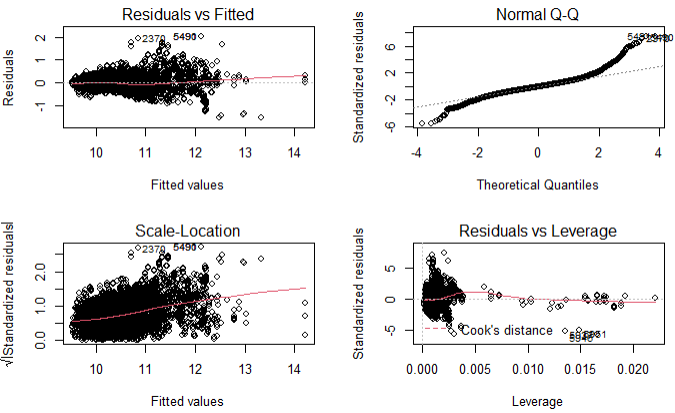
Engine Fuel Type of “premium unleaded (required)” = 9

Engine Fuel Type of “regular unleaded” = 10

\*\*\* The remaining variables are numerical and can be entered directly as seen in future records.

Before making a final interpretation of our chosen model, we’ll need to check the assumptions by looking at residual plots and determining if there are any influential outliers which are negatively skewing our model. Plots of these two checks can be seen below in **Plot P-2.**

**Plot P-2**

****

For the most part the residuals do form a random cloud but appear to spread out slightly as the fitted values increase. The Q-Q plot shows “moderate” evidence for normality, but again skews off at the tail ends. There are a few values identified as having high Cook’s D and possibly affecting the model’s power. All of these determinations might normally lead to further adjustments of the model but since the main goal for the first objective is to maintain practical interpretability of the model, we will accept these conditions with the caveat that the consumer be aware of the situation and intention of the model.

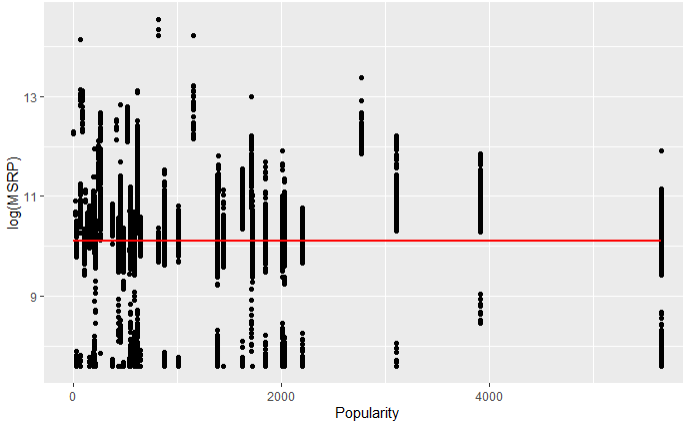
Now that we’ve “accepted” the model assumptions, we can offer an interpretation of the model which utilized the backward selection technique to choose its variables:

* “$9.96 is the unconditional mean of the log of MSRP = $21,162” – a car with no cylinders, no transmission, no fuel type, etc. would have a base price of $21k according to the current model.
* “For each 1 unit increase in engine horsepower, there is a 260% increase in the mean MSRP”
* “For each 1 unit increase in City MPG, there is a 14% increase in the mean MSRP”
* “For each additional number of doors, there is a 130% decrease in the mean MSRP”
* “For each 1 unit increase in Popularity score, there is a 0.01% increase in mean MSRP”
* “For each categorical increase in Transmission Type, there is a 91% decrease in mean MSRP”
* “For each categorical increase in Driven Wheels, there is an 88% decrease in mean MSRP”
* “For each categorical increase in Engine Fuel Type, there is a 26% decrease in mean MSRP”

On paper, this interpretation make look somewhat difficult to utilize but practically it is about as simple as it could be with the available data and requirements. If a consumer knew the numerical values for a mystery car’s engine horsepower, city mpg, number of car doors, and popularity score, as well as the lookup value associated with the transmission type, driven wheels, and engine fuel type (explained on page 4), they could simply plug in the seven numbers and receive a moderately accurate MSRP associated with that vehicle. It’s important to note that because this model is not intended for “bullseye” accurate predictions, the confidence intervals for the coefficients are quite wide. For example, the 95% confidence interval for the number of doors in our model is between (-0.071 and -0.055). This may not seem like much but when interpreted on the log(MSRP) it is a potential difference of 16%.

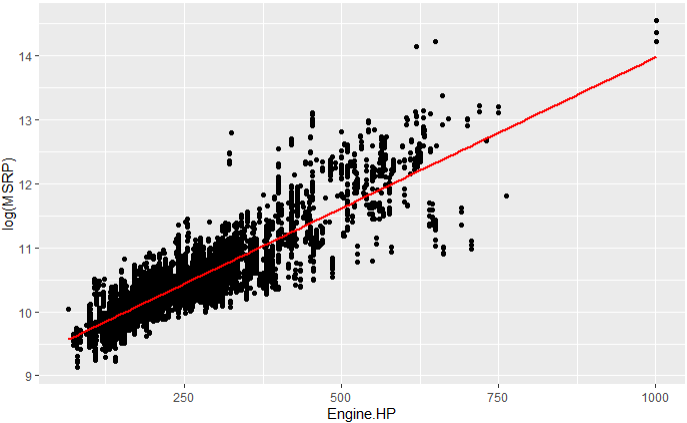
One of the interesting variables in this final model is “Popularity”. Out of the 7 variables included in the model, it was the “least significant” of the bunch as seen by its low impact coefficient (0.00001). By viewing a plot of Popularity vs MSRP (**Plot P-3**), we can see there is a slight negative correlation but with such a random cloud of points along its range, it is obviously not a good representation of MSRP and should not be used by consumers of this report to evaluate MSRPs without many of the other variables. The correlation of Popularity score and MSRP actually equates to barely 6%.

**Plot P-3**



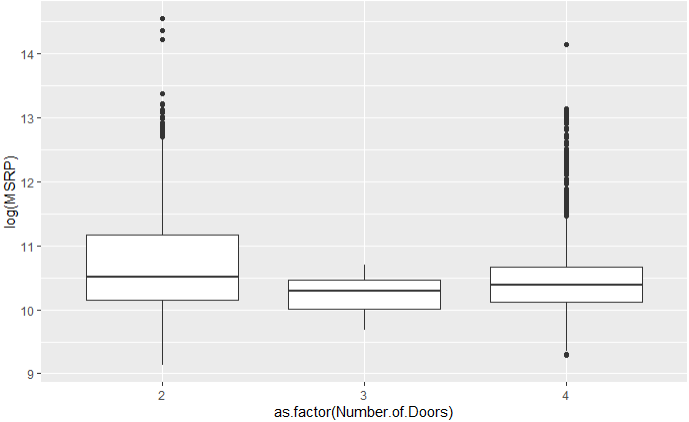
By looking at trends from the other included variables **(Plots P-3 to P-6**), we can make statements on their correlations to sale price.

**Plot P-3**



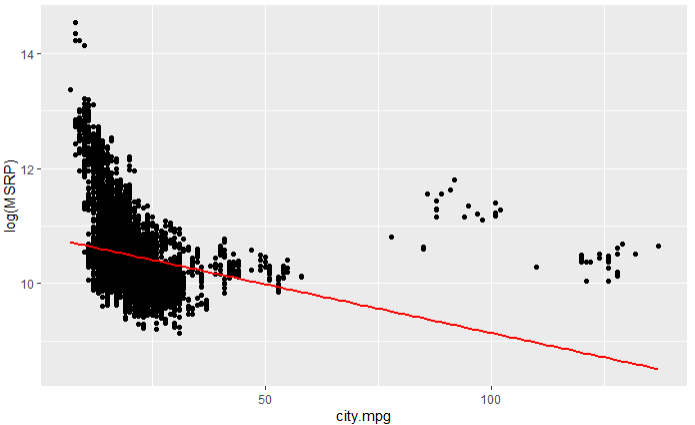
Perhaps the easiest relationship to comprehend and believe is: as the Engine Horsepower of a car increases, the resulting MSRP also increases.

**Plot P-4**



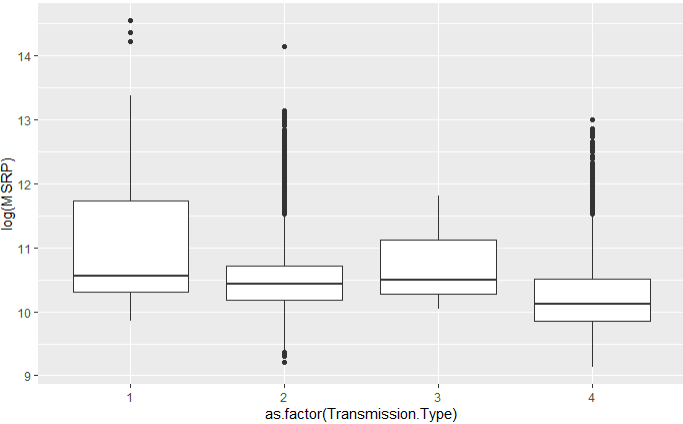
The number of doors is slightly negatively correlated with MSRP. This makes intuitive sense as high performance cars which usually come with a higher price tag tend to have less doors.

**Plot P-5**



Though it is affected by outliers (cars with city mpg > 60), there appears to be a form of quadratic relationship between city milage efficiency and MSRP. In this model we simply apply a coefficient to represent a linear relationship. In objective 2 we will attempt to capture this relationship’s added complexity.

**Plot P-6**



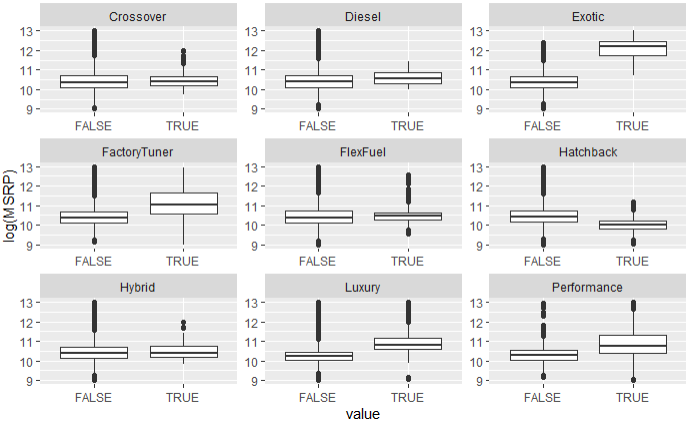
An example of one of the categorical variables: Transmission Type. Here the correlated numerical values (1-4) relate to a specific description of the type of transmission in a car. They are not ordered in any manner that relates to the domain they represent but the simply model is still able to use they’re “meaninglessness” to levels of predictive power.

**Objective #2**

In Objective #1 a simple to understand model of the car data was created. Albeit this model is easy enough to explain to the common public, it is not very good in terms of prediction power or fit of the overall data set available. In Objective #2 the goal is to forgo explainability and design two more models that can very accurately predict MSRP values of unknown cars, or in this case, our validation set mentioned above. The first attempt at a better model will be to recreate the steps taken in Objective #1 but with added complexity and variable interactions that were previously thrown out to maintain the level of practicality. The second attempt at a better model will be designed utilizing k-nearest neighbors (knn) regression.

Building upon the data wrangling and cleaning in objective #1, there are additional steps we can take to add complexity and potential predictive power to our next two models. The “Market Category” of the cars contains a combination of 11 descriptors such as “Luxury”, “Crossover”, “Hybrid”, etc. that offer good insight a particular car’s qualities and purpose as it pertains to the consumer. This information feels valuable in terms of representing certain features which are correlated to higher or lower sale prices. However, because of the way the current variable is displayed, it is not easy to comprehend or fit into regression analysis. To better utilize this portion of data, the Market Category variable was used to create new variables each representing whether or not a record contained those qualities. For example, a record with “Market Category” equal to “Luxury, Exotic, Performance” will have new variables “Luxury” “Exotic” and “Performance” equal to TRUE, while all other new variables equal to FALSE (“Hybrid” “Crossover” “Flex Fuel” “Hatchback” etc). A new list of the variables in our new data set can be seen in **Table T-3** in the Appendix and the visual representation of these variables can be seen below in **Plot P-6**.

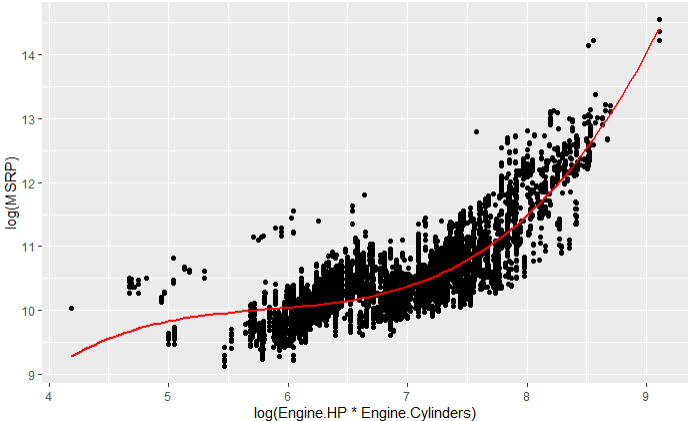
**Plot P-6**



These variables should prove useful during analysis as there are clear distinctions between TRUE and FALSE for many of them (Exotics, Factory Tuner, Hatchback, Luxury, etc.)

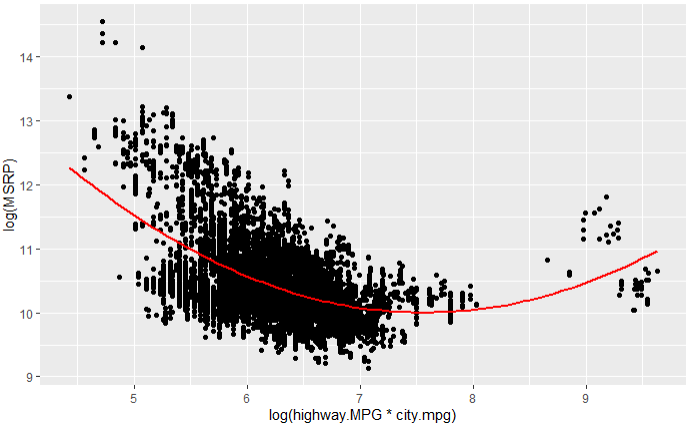
One way to add additional complexity to a model is to capture correlations between explanatory variables. Looking once again at our variable correlation matrix of plots, we identified two such relationships to include. The highway mpg and city mpg variables shared a correlation of 94% and the engine horsepower and the engine cylinders shared a correlation of 79%. Once again the relationship between log(MSRP) and each variable (and the new interaction variables) were plotted. This time we looked for potential polynomial and additional transformations which may better explain the data. A few notable interactions which resulted from this EDA are seen below in **Plots P-6 to P-8**.

**Plot P-6**



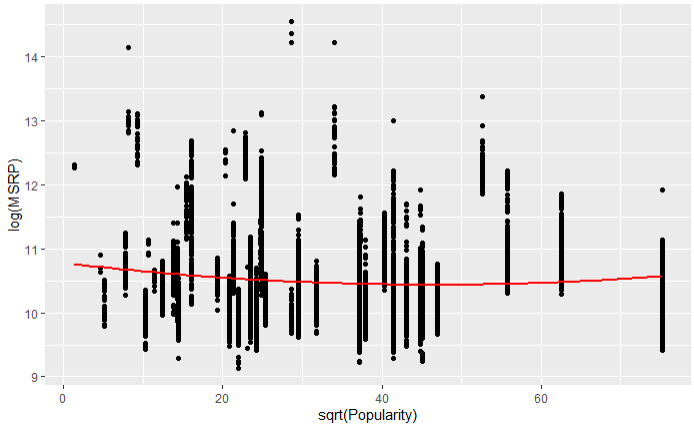
Log (MSRP) relationship to 3rd order polynomial of interaction variable log(Engine HP\*Engine Cylinders)

**Plot P-7**



Log (MSRP) relationship to 2nd order polynomial of interaction variable log(Highway.MPG\*City.MPG)

**Plot P-8**



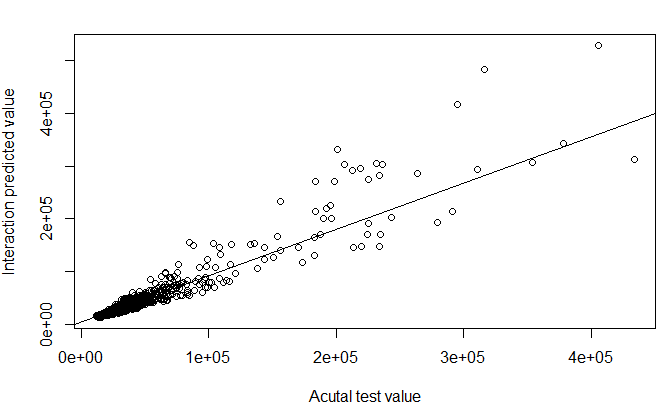
Log (MSRP) relationship to square root of Popularity score

The “Model” variable will be included in our selection techniques but the “Make” variable will be left out as it is directly correlated to the “Model”. With the interactions and transformations identified, a step-wise of selection method will again be used to narrow down the significant variables for the 2nd model. The variables chosen were all except then used to create the model equation:

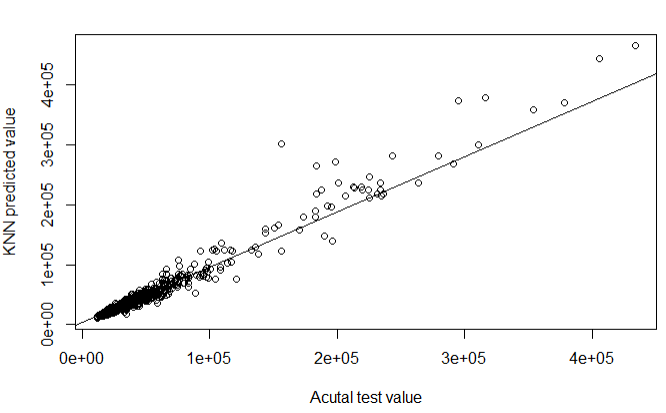
log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders + highway.MPG:city.mpg + highway.MPG + Luxury + Performance + FactoryTuner + Diesel + Crossover + Number.of.Doors + Engine.HP:Engine.Cylinders + city.mpg

The residual plots related to this model can be seen in Plot P-9 in the Appendix. Overall, the plots support our assumptions better than the model in Objective #1. The calculated AIC of the training set here came out to be -7147. A great way to see how well predictions were made is to view the predicted results vs the actual results as seen below in **Plot P-10**.

**Plot P-10**



The last model to explore is that utilizing k-nn regression. Using this method, a k value of 5 was determined to be the most effective with the data given, resulting in a well fit model with the lowest ASE statistics for both the test and validation sets of all 3 final models. Once again, we can view the actuals plotted against the predicted values to verify our success.



The final model comparisons can be seen below in **Table T-4**.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Train AIC** | **Test ASE** | **Validation ASE** |
| **Objective 1 Simple Model** | 1999.8 | 9.27 x 10^8 | N/A |
| **Objective 2 Complex Model** | -7146.755 | 2.72 x 10^8 | 1.7 x 10^8 |
| **Objective 2 KNN Model** | N/A | 9.54 x 10^7 | 5.95 x 10^7 |

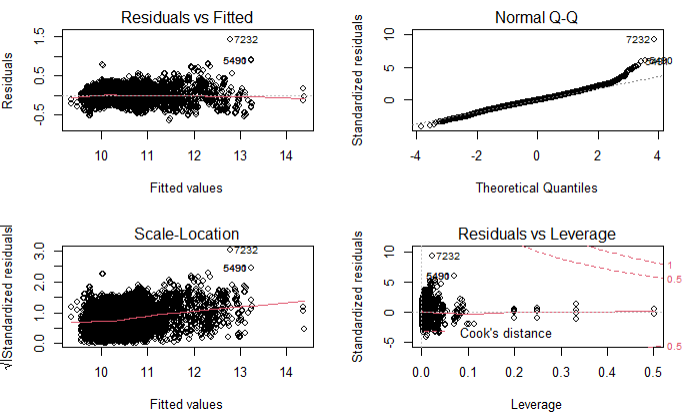
Overall, we can conclude that the knn regression method results in the best predictive model for the automotive data’s MSRP sale price but a simpler model is best used for interpretability and practical use. The Popularity score variable, while interesting, ultimately does not have much effect on the MSRP. During all 3 model representations, the data prior to year 2000 was excluded due to the inconsistency of MRSP values between those time periods. Any future assessments using this report should be aware of that as well as the fact that the “Model” and “Market Category” variables were not included although the essence of both were captured in “Make” and the newly created market category sub-variables. Though some of these models are accurate in predicting in a general sense, there are various “outliers” which seem to “break-the-mold” in terms of their pricing structure. Given more time, a more in-depth data cleaning effort could be applied to further group like cars into sub models which could more accurately predict sales prices. One such example is to create a model utilizing only electric vehicles as they can be viewed as vastly different in most of their qualities when compared to standard gas-powered cars.

**Appendix**

**Table T-3**

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Data Type** | **Description** |
| MSRP | Numeric | Sale Price (Response variable) |
| Car Make | Factor | The company that made the car. Ex: Honda, Toyota, etc. |
| Car Model | Factor | The model of the car. Ex: 4Runner, Accord, etc. |
| Year | Numeric | Year the car was produced |
| Engine Fuel Type | Factor | Type of fuel the car accepts. Ex: Regular unleaded, Diesel, etc. |
| Engine HP | Numeric | Horsepower of the car’s engine. |
| Engine Cylinders | Factor | Number of cylinders in the car’s engine. |
| Transmission Type | Factor | Type of transmission in the car. Ex: manual, automatic, etc. |
| Driven Wheels | Numeric | Wheels powered by engine. Ex: Front Wheel, Rear Wheel, etc. |
| Number of Doors | Numeric | The number of doors that the car has. Ex: 2, 4, etc. |
| Market Category | Factor | Various special factors for each car. Ex: Exotic, Luxury, etc. |
| Vehicle Size | Factor | The size of the vehicle. Ex: Midsize, Large, Compact, etc. |
| Vehicle Style | Factor | Body type of the vehicle. Ex: Coupe, Convertible, etc. |
| Highway MPG | Numeric | Fuel efficiency on the highway in MPG |
| City MPG | Numeric | Fuel efficiency in the city in MPG |
| Popularity | Numeric | A numerical popularity score for each car. |
| Crossover | Boolean | True if “Market Category” includes “Crossover”. False if not |
| Diesel | Boolean | True if “Market Category” includes “Diesel”. False if not |
| Exotic | Boolean | True if “Market Category” includes “Exotic”. False if not |
| Factory Tuner | Boolean | True if “Market Category” includes “Factory Tuner”. False if not |
| Flex Fuel | Boolean | True if “Market Category” includes “Flex Fuel”. False if not |
| Hatchback | Boolean | True if “Market Category” includes “Hatchback”. False if not |
| Hybrid | Boolean | True if “Market Category” includes “Hybrid”. False if not |
| Luxury | Boolean | True if “Market Category” includes “Luxury”. False if not |
| Performance | Boolean | True if “Market Category” includes “Performance”. False if not |

**Plot P-9**



**R-Markdown Code**

Project 1

library(naniar)  
library(caret)

## Loading required package: ggplot2

## Loading required package: lattice

library(ggplot2)  
library(mlbench)

## Warning: package 'mlbench' was built under R version 4.1.3

library(glmnet)

## Warning: package 'glmnet' was built under R version 4.1.3

## Loading required package: Matrix

## Loaded glmnet 4.1-4

library(olsrr)

## Warning: package 'olsrr' was built under R version 4.1.3

##   
## Attaching package: 'olsrr'

## The following object is masked from 'package:datasets':  
##   
## rivers

library(fmsb)

## Warning: package 'fmsb' was built under R version 4.1.3

## Registered S3 methods overwritten by 'fmsb':  
## method from  
## print.roc pROC  
## plot.roc pROC

library(corrplot)

## corrplot 0.92 loaded

library(RColorBrewer)  
library(funModeling)

## Warning: package 'funModeling' was built under R version 4.1.3

## Loading required package: Hmisc

## Loading required package: survival

##   
## Attaching package: 'survival'

## The following object is masked from 'package:caret':  
##   
## cluster

## Loading required package: Formula

##   
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':  
##   
## format.pval, units

## funModeling v.1.9.4 :)  
## Examples and tutorials at livebook.datascienceheroes.com  
## / Now in Spanish: librovivodecienciadedatos.ai

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble 3.1.6 v dplyr 1.0.7  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 2.1.1 v forcats 0.5.1  
## v purrr 0.3.4

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x tidyr::expand() masks Matrix::expand()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x purrr::lift() masks caret::lift()  
## x tidyr::pack() masks Matrix::pack()  
## x dplyr::src() masks Hmisc::src()  
## x dplyr::summarize() masks Hmisc::summarize()  
## x tidyr::unpack() masks Matrix::unpack()

library(tidyr)  
library(Hmisc)  
library(kableExtra)

## Warning: package 'kableExtra' was built under R version 4.1.3

##   
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':  
##   
## group\_rows

library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

##   
## Attaching package: 'GGally'

## The following object is masked from 'package:funModeling':  
##   
## range01

library(plyr)

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following object is masked from 'package:purrr':  
##   
## compact

## The following objects are masked from 'package:Hmisc':  
##   
## is.discrete, summarize

library(stringr)  
library(agricolae)

## Warning: package 'agricolae' was built under R version 4.1.3

library(ggplot2)  
library(GGally)  
library(lindia)

## Warning: package 'lindia' was built under R version 4.1.3

library(AICcmodavg)  
library(Metrics)

## Warning: package 'Metrics' was built under R version 4.1.3

##   
## Attaching package: 'Metrics'

## The following objects are masked from 'package:caret':  
##   
## precision, recall

library(car)

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following object is masked from 'package:purrr':  
##   
## some

library(psych)

## Warning: package 'psych' was built under R version 4.1.3

##   
## Attaching package: 'psych'

## The following object is masked from 'package:car':  
##   
## logit

## The following object is masked from 'package:Hmisc':  
##   
## describe

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

# Read in data  
cars <- read.csv("data1.csv",stringsAsFactors = T)  
  
# view data summary  
#summary(cars)  
  
# view records with N/A's for review  
cars\_NA <- cars %>% filter\_all(any\_vars(is.na(.)))  
#cars\_NA  
  
# N/A's identified in Engine HP(69), Engine Cylinders(30), & Number of Doors(6)

# Manually look up and add missing Engine HP values  
  
cars\_NA\_EngineHP <- cars[is.na(cars$Engine.HP),]  
#cars\_NA\_EngineHP  
  
# records 540 to 542 -> 2015, 2016, 2017 FIAT 500e -> 111 Hp  
cars[c(540:542),5] <- 111  
  
# records 2906 to 2909 -> 2017 Lincoln Continental (4 different trims) -> 305, 400, 335, 305 Hp  
cars[c(2906:2909),5] <- c(305,400,335,305)  
  
# records 4204 to 4207 -> 2017 Ford Escape (4 different trims) -> 168 Hp  
cars[c(4204:4207),5] <- 168  
  
# records 4706 to 4707 -> 2013, 2014 Honda Fit -> 123 Hp  
cars[c(4706:4707),5] <- 123  
  
# records 4915 to 4920 -> 2005 Ford Freestar (6 different trims) -> 193 Hp  
cars[c(4915:4920),5] <- 193  
  
# records 4786 to 4799 -> 2015, 2016, 2017 Ford Focus -> 143 Hp  
cars[c(4786:4799),5] <- 143  
  
# record 5779 -> 2014 Mitsubishi i-MiEV -> 66 Hp  
cars[c(5779),5] <- 66  
  
# records 5826 to 5841 -> 2015, 2016, 2017 Chevy Imapala (2 trims per year) -> 301 Hp  
cars[c(5826:5841),5] <- 301  
  
# records 6386 to 6395 -> 2014 to 2016 Nissan Leaf (3 trims per year) -> 107 Hp  
cars[c(6386:6395),5] <- 107  
  
# record 6579 -> 2015 Mercedes M-Class -> 200 Hp  
cars[c(6579),5] <- 200  
  
# record 6909 -> 2017 Lincoln MKZ (trim 1) -> 240 Hp  
cars[c(6909),5] <- 240  
  
# record 6911 -> 2017 Lincoln MKZ (trim 2) -> 350 Hp  
cars[c(6911),5] <- 350  
  
# record 6917 -> 2017 Lincoln MKZ (trim 3) -> 350 Hp  
cars[c(6917),5] <- 350  
  
# record 6919 -> 2017 Lincoln MKZ (trim 4) -> 400 Hp  
cars[c(6919),5] <- 400  
  
# records 6922 to 6925 -> 2014 Tesla Model S (4 trims) -> 362, 302, 691, 416 Hp  
cars[c(6922:6925),5] <- c(362,302,691,416)  
  
# records 6926 to 6930 -> 2015 Tesla Model S (5 trims) -> 323, 380, 691, 422, 380 Hp  
cars[c(6926:6930),5] <- c(323,380,691,422,380)  
  
# records 6931 to 6939 -> 2016 Tesla Model S (9 trims) -> 382, 315, 762, 422, 328, 422, 518, 691, 382 Hp  
cars[c(6931:6939),5] <- c(382,315,762,422,328,422,518,691,382)  
  
# record 8375 to 8376 -> 2013, 2014 Toyota RAV4 EV -> 154 Hp  
cars[c(8375:8376),5] <- 154  
  
# record 9851 to 9852 -> 2015 Kia Soul EV (2 trims) -> 154 Hp  
cars[c(9851:9852),5] <- 154  
  
# record 9853 to 9855 -> 2016 Kia Soul EV (3 trims) -> 109 Hp  
cars[c(9853:9855),5] <- 109  
  
# Verify no more NA's  
cars\_NA\_EngineHP <- cars[is.na(cars$Engine.HP),]  
#cars\_NA\_EngineHP  
  
# Manually look up and add missing "Engine Fuel Type" values  
cars\_blank <- cars %>% filter(Engine.Fuel.Type < 1)

## Warning in Ops.factor(Engine.Fuel.Type, 1): '<' not meaningful for factors

# records 11323:11325 -> 2004 Suzuki Verona (3 trims)  
cars[c(11322:11324),4] <- "regular unleaded"  
  
# Manually look up and add missing and "odd" Engine HP values  
cars\_NA\_EngineCylinder <- cars[is.na(cars$Engine.Cylinders),]  
#cars\_NA\_EngineCylinder  
  
# Input value of "Elec" for all electric vehicles (Engine.Fuel.Type = electric)  
cars <- within(cars, Engine.Cylinders[Engine.Fuel.Type == 'electric'] <- 0)  
  
# Change Cylinder to categorical variable  
cars$Engine.Cylinders <- as.factor(cars$Engine.Cylinders)  
  
# Change Engine Cylinders value of Mazda RX-7 cars (rotary engine) to piston equivalent -> 6  
cars[c(8696:8698),6] <- 6  
  
# Change Engine Cylinders value of Mazda RX-8 cars (rotary engine) to piston equivalent -> 8  
cars[c(8699:8715),6] <- 8  
  
# Verify no more NA's  
cars\_NA\_EngineCylinders <- cars[is.na(cars$Engine.Cylinders),]  
#cars\_NA\_EngineCylinders  
  
# Manually look up and add missing "Number of Doors" values  
cars\_NA\_Doors <- cars[is.na(cars$Number.of.Doors),]  
#cars\_NA\_Doors  
  
# record 4667 -> 2013 Ferrari FF has 2 Doors  
cars[c(4667),9] <- 2  
  
# records 6931 to 6935 -> 2016 Tesla Model S has 4 Doors  
cars[c(6931:6935),9] <- 4  
  
#audi A6 row 837 from 354mpg to 34 ##error   
cars[c(1120),13] <- 34  
  
# Verify no more NA's  
cars\_NA\_Doors <- cars[is.na(cars$Number.of.Doors),]  
#cars\_NA\_Doors

# Verify ALL NA's have been changed  
cars\_NA <- cars %>% filter\_all(any\_vars(is.na(.)))  
#cars\_NA  
#summary(cars)

# Convert applicable categorical variables to "factors"  
cars$Make <- as.factor(cars$Make)  
cars$Engine.Fuel.Type <- as.numeric(as.factor(cars$Engine.Fuel.Type))  
cars$Transmission.Type <- as.numeric(as.factor(cars$Transmission.Type))  
cars$Driven\_Wheels <- as.numeric(as.factor(cars$Driven\_Wheels))  
cars$Vehicle.Size <- as.numeric(as.factor(cars$Vehicle.Size))  
cars$Vehicle.Style <-as.numeric(as.factor(cars$Vehicle.Style))  
cars$Market.Category <-as.factor(cars$Market.Category)  
  
# Convert applicable numerical variables to "numeric"  
cars$Year <- as.numeric(cars$Year)  
cars$Engine.HP <- as.numeric(cars$Engine.HP)  
cars$highway.MPG <- as.numeric(cars$highway.MPG)  
cars$city.mpg <- as.numeric(cars$city.mpg)  
cars$Popularity <- as.numeric(cars$Popularity)  
cars$MSRP <- as.numeric(cars$MSRP)  
cars$Number.of.Doors <-as.numeric(cars$Number.of.Doors)  
cars$Engine.Cylinders <- as.numeric(cars$Engine.Cylinders)  
  
# Verify variable data types  
#str(cars)  
#summary(cars)  
  
# Too many levels to convert "Model" to Categorical/Factor. This variable will not be considered in the analysis due to it's inherent correlation to "Make".

# Review Market.Category values  
#levels(cars$Market.Category)  
plyr::count(cars,'Market.Category')

## Market.Category freq  
## 1 Crossover 1110  
## 2 Crossover,Diesel 7  
## 3 Crossover,Exotic,Luxury,High-Performance 1  
## 4 Crossover,Exotic,Luxury,Performance 1  
## 5 Crossover,Factory Tuner,Luxury,High-Performance 26  
## 6 Crossover,Factory Tuner,Luxury,Performance 5  
## 7 Crossover,Factory Tuner,Performance 4  
## 8 Crossover,Flex Fuel 64  
## 9 Crossover,Flex Fuel,Luxury 10  
## 10 Crossover,Flex Fuel,Luxury,Performance 6  
## 11 Crossover,Flex Fuel,Performance 6  
## 12 Crossover,Hatchback 72  
## 13 Crossover,Hatchback,Factory Tuner,Performance 6  
## 14 Crossover,Hatchback,Luxury 7  
## 15 Crossover,Hatchback,Performance 6  
## 16 Crossover,Hybrid 42  
## 17 Crossover,Luxury 410  
## 18 Crossover,Luxury,Diesel 34  
## 19 Crossover,Luxury,High-Performance 9  
## 20 Crossover,Luxury,Hybrid 24  
## 21 Crossover,Luxury,Performance 113  
## 22 Crossover,Luxury,Performance,Hybrid 2  
## 23 Crossover,Performance 69  
## 24 Diesel 84  
## 25 Diesel,Luxury 51  
## 26 Exotic,Factory Tuner,High-Performance 21  
## 27 Exotic,Factory Tuner,Luxury,High-Performance 52  
## 28 Exotic,Factory Tuner,Luxury,Performance 3  
## 29 Exotic,Flex Fuel,Factory Tuner,Luxury,High-Performance 13  
## 30 Exotic,Flex Fuel,Luxury,High-Performance 11  
## 31 Exotic,High-Performance 261  
## 32 Exotic,Luxury 12  
## 33 Exotic,Luxury,High-Performance 79  
## 34 Exotic,Luxury,High-Performance,Hybrid 1  
## 35 Exotic,Luxury,Performance 36  
## 36 Exotic,Performance 10  
## 37 Factory Tuner,High-Performance 106  
## 38 Factory Tuner,Luxury 2  
## 39 Factory Tuner,Luxury,High-Performance 215  
## 40 Factory Tuner,Luxury,Performance 31  
## 41 Factory Tuner,Performance 92  
## 42 Flex Fuel 872  
## 43 Flex Fuel,Diesel 16  
## 44 Flex Fuel,Factory Tuner,Luxury,High-Performance 1  
## 45 Flex Fuel,Hybrid 2  
## 46 Flex Fuel,Luxury 39  
## 47 Flex Fuel,Luxury,High-Performance 33  
## 48 Flex Fuel,Luxury,Performance 28  
## 49 Flex Fuel,Performance 87  
## 50 Flex Fuel,Performance,Hybrid 2  
## 51 Hatchback 641  
## 52 Hatchback,Diesel 14  
## 53 Hatchback,Factory Tuner,High-Performance 13  
## 54 Hatchback,Factory Tuner,Luxury,Performance 9  
## 55 Hatchback,Factory Tuner,Performance 22  
## 56 Hatchback,Flex Fuel 7  
## 57 Hatchback,Hybrid 72  
## 58 Hatchback,Luxury 46  
## 59 Hatchback,Luxury,Hybrid 3  
## 60 Hatchback,Luxury,Performance 38  
## 61 Hatchback,Performance 252  
## 62 High-Performance 199  
## 63 Hybrid 123  
## 64 Luxury 855  
## 65 Luxury,High-Performance 334  
## 66 Luxury,High-Performance,Hybrid 12  
## 67 Luxury,Hybrid 52  
## 68 Luxury,Performance 673  
## 69 Luxury,Performance,Hybrid 11  
## 70 N/A 3742  
## 71 Performance 601  
## 72 Performance,Hybrid 1

# There are too many levels (72) to convert "Market.Category" to Categorical/Factor. The values of the variable are potential correlated to MSRP and Popularity which is the main consideration of this analysis so the variable will NOT be dropped. Instead, the Market.Category variable will be read for the 11 unique string descriptors which make up all 72 combinations (levels). 11 new boolean/logical variables will be created and hold the value of whether or not the descriptor of each records shows up in its original "Market.Category" value.  
  
pattern.Crossover <- paste("Crossover", collapse = "|")  
pattern.Diesel <- paste("Diesel", collapse = "|")  
pattern.Exotic <- paste("Exotic", collapse = "|")  
pattern.Luxury <- paste("Luxury", collapse = "|")  
pattern.Performance <- paste("Performance", collapse = "|")  
pattern.FactoryTuner <- paste("Factory Tuner", collapse = "|")  
pattern.FlexFuel <- paste("Flex Fuel", collapse = "|")  
pattern.Hatchback <- paste("Hatchback", collapse = "|")  
pattern.Hybrid <- paste("Hybrid", collapse = "|")  
  
cars$Crossover <- grepl(pattern.Crossover, cars$Market.Category)  
cars$Diesel <- grepl(pattern.Diesel, cars$Market.Category)  
cars$Exotic <- grepl(pattern.Exotic, cars$Market.Category)  
cars$Luxury <- grepl(pattern.Luxury, cars$Market.Category)  
cars$Performance <- grepl(pattern.Performance, cars$Market.Category)  
cars$FactoryTuner <- grepl(pattern.FactoryTuner, cars$Market.Category)  
cars$FlexFuel <- grepl(pattern.FlexFuel, cars$Market.Category)  
cars$Hatchback <- grepl(pattern.Hatchback, cars$Market.Category)  
cars$Hybrid <- grepl(pattern.Hybrid, cars$Market.Category)  
  
cars$Crossover <- as.numeric(as.factor(cars$Crossover))  
cars$Diesel <- as.numeric(as.factor(cars$Diesel))  
cars$Exotic <- as.numeric(as.factor(cars$Exotic))  
cars$Luxury <- as.numeric(as.factor(cars$Luxury))  
cars$Performance <- as.numeric(as.factor(cars$Performance))  
cars$FactoryTuner <- as.numeric(as.factor(cars$FactoryTuner))  
cars$FlexFuel <-as.numeric(as.factor(cars$FlexFuel))  
cars$Hatchback <-as.numeric(as.factor(cars$Hatchback))  
cars$Hybrid <-as.numeric(as.factor(cars$Hybrid))  
  
#summary(cars)

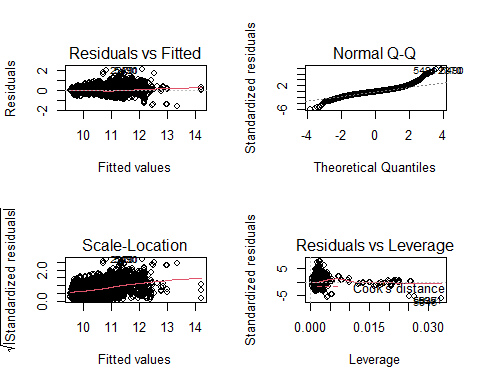
cars2000 <- cars %>% filter(cars$Year>2000)  
  
# drop variables to that allow for simpler final interpretation and increased analysis speed  
  
cars2000 = subset(cars2000, select = -c(Make, Model, Market.Category, Crossover, Diesel, Exotic, Luxury, Performance, FactoryTuner, FlexFuel, Hatchback, Hybrid))  
  
# 10,257 of the original 11,914 remain  
#summary(cars2000)

# set seed  
set.seed(123)  
  
# Split data into smaller sets: 80% training, 10% test, and 10% validate  
ss <- sample(1:3,size=nrow(cars2000),replace=TRUE,prob=c(0.8,0.1,0.1))  
train <- cars2000[ss==1,]  
test <- cars2000[ss==2,]  
valid <- cars2000[ss==3,]

# full model  
Fullmodel = lm(log(MSRP)~.,data = train)  
summary(Fullmodel)

##   
## Call:  
## lm(formula = log(MSRP) ~ ., data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.5645 -0.1543 -0.0104 0.1314 2.0086   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.017e+01 1.723e+00 5.905 3.67e-09 \*\*\*  
## Year -2.295e-04 8.580e-04 -0.267 0.78911   
## Engine.Fuel.Type -1.027e-02 1.702e-03 -6.034 1.67e-09 \*\*\*  
## Engine.HP 4.716e-03 5.185e-05 90.961 < 2e-16 \*\*\*  
## Engine.Cylinders 1.059e-02 4.939e-03 2.145 0.03202 \*   
## Transmission.Type -7.910e-02 3.888e-03 -20.341 < 2e-16 \*\*\*  
## Driven\_Wheels -3.739e-02 2.999e-03 -12.468 < 2e-16 \*\*\*  
## Number.of.Doors -5.902e-02 4.252e-03 -13.880 < 2e-16 \*\*\*  
## Vehicle.Size -1.008e-02 3.796e-03 -2.655 0.00795 \*\*   
## Vehicle.Style -1.285e-03 6.726e-04 -1.910 0.05613 .   
## highway.MPG 3.643e-03 1.319e-03 2.761 0.00577 \*\*   
## city.mpg 3.148e-03 1.064e-03 2.959 0.00310 \*\*   
## Popularity -2.176e-05 2.182e-06 -9.970 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2727 on 8253 degrees of freedom  
## Multiple R-squared: 0.7958, Adjusted R-squared: 0.7955   
## F-statistic: 2680 on 12 and 8253 DF, p-value: < 2.2e-16

par(mfrow = c(2,2))  
plot(Fullmodel)



# variable selections - step-wise  
  
linearn1select =ols\_step\_both\_p(Fullmodel, prem = 0.01, pent = 0.01, details = F, progress = T)

## Stepwise Selection Method   
## ---------------------------  
##   
## Candidate Terms:   
##   
## 1. Year   
## 2. Engine.Fuel.Type   
## 3. Engine.HP   
## 4. Engine.Cylinders   
## 5. Transmission.Type   
## 6. Driven\_Wheels   
## 7. Number.of.Doors   
## 8. Vehicle.Size   
## 9. Vehicle.Style   
## 10. highway.MPG   
## 11. city.mpg   
## 12. Popularity   
##   
## We are selecting variables based on p value...  
##   
## Variables Entered/Removed:   
##   
## - Engine.Cylinders added   
## - Transmission.Type added   
## - city.mpg added   
## - Engine.Cylinders added   
## - Number.of.Doors added   
## - Driven\_Wheels added   
## - Popularity added   
## - Engine.Fuel.Type added   
##   
## No more variables to be added/removed.  
##   
##   
## Final Model Output   
## ------------------  
##   
## Model Summary   
## -------------------------------------------------------------  
## R 0.892 RMSE 0.273   
## R-Squared 0.795 Coef. Var 2.599   
## Adj. R-Squared 0.795 MSE 0.074   
## Pred R-Squared 0.795 MAE 0.193   
## -------------------------------------------------------------  
## RMSE: Root Mean Square Error   
## MSE: Mean Square Error   
## MAE: Mean Absolute Error   
##   
## ANOVA   
## ------------------------------------------------------------------------  
## Sum of   
## Squares DF Mean Square F Sig.   
## ------------------------------------------------------------------------  
## Regression 2390.808 7 341.544 4585.41 0.0000   
## Residual 615.097 8258 0.074   
## Total 3005.905 8265   
## ------------------------------------------------------------------------  
##   
## Parameter Estimates   
## -----------------------------------------------------------------------------------------------  
## model Beta Std. Error Std. Beta t Sig lower upper   
## -----------------------------------------------------------------------------------------------  
## (Intercept) 9.772 0.032 302.714 0.000 9.708 9.835   
## Engine.HP 0.005 0.000 0.876 151.649 0.000 0.005 0.005   
## Transmission.Type -0.078 0.004 -0.109 -20.473 0.000 -0.085 -0.070   
## city.mpg 0.006 0.000 0.089 15.729 0.000 0.005 0.006   
## Number.of.Doors -0.063 0.004 -0.088 -15.996 0.000 -0.071 -0.055   
## Driven\_Wheels -0.036 0.003 -0.065 -12.521 0.000 -0.042 -0.031   
## Popularity 0.000 0.000 -0.052 -10.394 0.000 0.000 0.000   
## Engine.Fuel.Type -0.010 0.002 -0.032 -6.189 0.000 -0.014 -0.007   
## -----------------------------------------------------------------------------------------------

summary(linearn1select)

## Length Class Mode   
## orders 9 -none- character  
## method 8 -none- character  
## steps 1 -none- numeric   
## predictors 7 -none- character  
## rsquare 8 -none- numeric   
## aic 8 -none- numeric   
## sbc 8 -none- numeric   
## sbic 8 -none- numeric   
## adjr 8 -none- numeric   
## rmse 8 -none- numeric   
## mallows\_cp 8 -none- numeric   
## indvar 12 -none- character  
## betas 42 -none- numeric   
## lbetas 8 -none- numeric   
## pvalues 42 -none- numeric   
## beta\_pval 4 data.frame list   
## model 12 lm list

linearn1F = ols\_step\_forward\_p(Fullmodel, pent = 0.01, details = F, progress = T)

## Forward Selection Method   
## ---------------------------  
##   
## Candidate Terms:   
##   
## 1. Year   
## 2. Engine.Fuel.Type   
## 3. Engine.HP   
## 4. Engine.Cylinders   
## 5. Transmission.Type   
## 6. Driven\_Wheels   
## 7. Number.of.Doors   
## 8. Vehicle.Size   
## 9. Vehicle.Style   
## 10. highway.MPG   
## 11. city.mpg   
## 12. Popularity   
##   
## We are selecting variables based on p value...  
##   
## Variables Entered:   
##   
## - Engine.Cylinders   
## - Transmission.Type   
## - city.mpg   
## - Number.of.Doors   
## - Driven\_Wheels   
## - Popularity   
## - Engine.Fuel.Type   
##   
## No more variables to be added.  
##   
## Final Model Output   
## ------------------  
##   
## Model Summary   
## -------------------------------------------------------------  
## R 0.892 RMSE 0.273   
## R-Squared 0.795 Coef. Var 2.599   
## Adj. R-Squared 0.795 MSE 0.074   
## Pred R-Squared 0.795 MAE 0.193   
## -------------------------------------------------------------  
## RMSE: Root Mean Square Error   
## MSE: Mean Square Error   
## MAE: Mean Absolute Error   
##   
## ANOVA   
## ------------------------------------------------------------------------  
## Sum of   
## Squares DF Mean Square F Sig.   
## ------------------------------------------------------------------------  
## Regression 2390.892 8 298.861 4012.432 0.0000   
## Residual 615.013 8257 0.074   
## Total 3005.905 8265   
## ------------------------------------------------------------------------  
##   
## Parameter Estimates   
## -----------------------------------------------------------------------------------------------  
## model Beta Std. Error Std. Beta t Sig lower upper   
## -----------------------------------------------------------------------------------------------  
## (Intercept) 9.758 0.035 280.127 0.000 9.690 9.826   
## Engine.HP 0.005 0.000 0.869 103.468 0.000 0.005 0.005   
## Engine.Cylinders 0.004 0.004 0.010 1.058 0.290 -0.004 0.013   
## Transmission.Type -0.077 0.004 -0.108 -20.322 0.000 -0.085 -0.070   
## city.mpg 0.006 0.000 0.092 14.186 0.000 0.005 0.007   
## Number.of.Doors -0.064 0.004 -0.088 -16.025 0.000 -0.071 -0.056   
## Driven\_Wheels -0.037 0.003 -0.066 -12.514 0.000 -0.042 -0.031   
## Popularity 0.000 0.000 -0.052 -10.415 0.000 0.000 0.000   
## Engine.Fuel.Type -0.010 0.002 -0.032 -6.112 0.000 -0.014 -0.007   
## -----------------------------------------------------------------------------------------------

#summary(linearn1F)

linearnB = ols\_step\_backward\_p(Fullmodel,fit, prem = 0.01, details = F, progress = T)

## Backward Elimination Method   
## ---------------------------  
##   
## Candidate Terms:   
##   
## 1 . Year   
## 2 . Engine.Fuel.Type   
## 3 . Engine.HP   
## 4 . Engine.Cylinders   
## 5 . Transmission.Type   
## 6 . Driven\_Wheels   
## 7 . Number.of.Doors   
## 8 . Vehicle.Size   
## 9 . Vehicle.Style   
## 10 . highway.MPG   
## 11 . city.mpg   
## 12 . Popularity   
##   
## We are eliminating variables based on p value...  
##   
## Variables Removed:   
##   
## - Year   
## - Vehicle.Style   
## - Engine.Cylinders   
## - highway.MPG   
## - Vehicle.Size   
##   
## No more variables satisfy the condition of p value = 0.01  
##   
##   
## Final Model Output   
## ------------------  
##   
## Model Summary   
## -------------------------------------------------------------  
## R 0.892 RMSE 0.273   
## R-Squared 0.795 Coef. Var 2.599   
## Adj. R-Squared 0.795 MSE 0.074   
## Pred R-Squared 0.795 MAE 0.193   
## -------------------------------------------------------------  
## RMSE: Root Mean Square Error   
## MSE: Mean Square Error   
## MAE: Mean Absolute Error   
##   
## ANOVA   
## ------------------------------------------------------------------------  
## Sum of   
## Squares DF Mean Square F Sig.   
## ------------------------------------------------------------------------  
## Regression 2390.808 7 341.544 4585.41 0.0000   
## Residual 615.097 8258 0.074   
## Total 3005.905 8265   
## ------------------------------------------------------------------------  
##   
## Parameter Estimates   
## -----------------------------------------------------------------------------------------------  
## model Beta Std. Error Std. Beta t Sig lower upper   
## -----------------------------------------------------------------------------------------------  
## (Intercept) 9.772 0.032 302.714 0.000 9.708 9.835   
## Engine.Fuel.Type -0.010 0.002 -0.032 -6.189 0.000 -0.014 -0.007   
## Engine.HP 0.005 0.000 0.876 151.649 0.000 0.005 0.005   
## Transmission.Type -0.078 0.004 -0.109 -20.473 0.000 -0.085 -0.070   
## Driven\_Wheels -0.036 0.003 -0.065 -12.521 0.000 -0.042 -0.031   
## Number.of.Doors -0.063 0.004 -0.088 -15.996 0.000 -0.071 -0.055   
## city.mpg 0.006 0.000 0.089 15.729 0.000 0.005 0.006   
## Popularity 0.000 0.000 -0.052 -10.394 0.000 0.000 0.000   
## -----------------------------------------------------------------------------------------------

#summary(linearnB)

modelstep <- lm(log(MSRP) ~ Transmission.Type + city.mpg + Number.of.Doors + Driven\_Wheels + Popularity + Engine.Fuel.Type, data=train)  
   
summary(modelstep)

##   
## Call:  
## lm(formula = log(MSRP) ~ Transmission.Type + city.mpg + Number.of.Doors +   
## Driven\_Wheels + Popularity + Engine.Fuel.Type, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.4775 -0.3228 -0.0933 0.2122 3.5498   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.270e+01 5.033e-02 252.300 < 2e-16 \*\*\*  
## Transmission.Type -1.630e-01 7.295e-03 -22.342 < 2e-16 \*\*\*  
## city.mpg -1.727e-02 6.414e-04 -26.922 < 2e-16 \*\*\*  
## Number.of.Doors -1.876e-01 7.516e-03 -24.967 < 2e-16 \*\*\*  
## Driven\_Wheels -5.010e-02 5.619e-03 -8.916 < 2e-16 \*\*\*  
## Popularity -1.672e-05 4.160e-06 -4.020 5.87e-05 \*\*\*  
## Engine.Fuel.Type -6.584e-02 3.199e-03 -20.578 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5309 on 8259 degrees of freedom  
## Multiple R-squared: 0.2255, Adjusted R-squared: 0.2249   
## F-statistic: 400.8 on 6 and 8259 DF, p-value: < 2.2e-16

modelforward <-lm(log(MSRP)~ Engine.Cylinders + Transmission.Type + city.mpg + Number.of.Doors + Driven\_Wheels + Popularity + Engine.Fuel.Type ,data=train)  
  
summary(modelforward)

##   
## Call:  
## lm(formula = log(MSRP) ~ Engine.Cylinders + Transmission.Type +   
## city.mpg + Number.of.Doors + Driven\_Wheels + Popularity +   
## Engine.Fuel.Type, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.23108 -0.26563 -0.04016 0.22763 2.69984   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.015e+01 5.248e-02 193.359 < 2e-16 \*\*\*  
## Engine.Cylinders 3.234e-01 4.420e-03 73.168 < 2e-16 \*\*\*  
## Transmission.Type -9.433e-02 5.759e-03 -16.379 < 2e-16 \*\*\*  
## city.mpg 1.071e-02 6.292e-04 17.019 < 2e-16 \*\*\*  
## Number.of.Doors -1.497e-01 5.878e-03 -25.467 < 2e-16 \*\*\*  
## Driven\_Wheels -8.176e-02 4.398e-03 -18.590 < 2e-16 \*\*\*  
## Popularity -2.343e-05 3.242e-06 -7.228 5.36e-13 \*\*\*  
## Engine.Fuel.Type -2.855e-02 2.544e-03 -11.224 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4136 on 8258 degrees of freedom  
## Multiple R-squared: 0.5301, Adjusted R-squared: 0.5297   
## F-statistic: 1331 on 7 and 8258 DF, p-value: < 2.2e-16

modelbackward <- lm(log(MSRP)~ . -Year -Vehicle.Style -highway.MPG -Engine.Cylinders -Vehicle.Size ,data=train)   
  
summary(modelbackward)

##   
## Call:  
## lm(formula = log(MSRP) ~ . - Year - Vehicle.Style - highway.MPG -   
## Engine.Cylinders - Vehicle.Size, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.52643 -0.14988 -0.00996 0.13191 2.02146   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.772e+00 3.228e-02 302.714 < 2e-16 \*\*\*  
## Engine.Fuel.Type -1.043e-02 1.685e-03 -6.189 6.36e-10 \*\*\*  
## Engine.HP 4.755e-03 3.135e-05 151.649 < 2e-16 \*\*\*  
## Transmission.Type -7.763e-02 3.792e-03 -20.473 < 2e-16 \*\*\*  
## Driven\_Wheels -3.618e-02 2.890e-03 -12.521 < 2e-16 \*\*\*  
## Number.of.Doors -6.318e-02 3.950e-03 -15.996 < 2e-16 \*\*\*  
## city.mpg 5.707e-03 3.629e-04 15.729 < 2e-16 \*\*\*  
## Popularity -2.223e-05 2.139e-06 -10.394 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2729 on 8258 degrees of freedom  
## Multiple R-squared: 0.7954, Adjusted R-squared: 0.7952   
## F-statistic: 4585 on 7 and 8258 DF, p-value: < 2.2e-16

#define list of models   
models <- list(modelstep, modelforward, modelbackward)  
  
#specify model names  
mod.names <- c('stepwise\_selection', 'forward\_selection', 'backward\_selection')  
  
#calculate AIC of each model  
aictab(cand.set = models, modnames = mod.names)

##   
## Model selection based on AICc:  
##   
## K AICc Delta\_AICc AICcWt Cum.Wt LL  
## backward\_selection 9 1999.80 0.00 1 1 -990.89  
## forward\_selection 9 8871.11 6871.31 0 1 -4426.55  
## stepwise\_selection 8 12999.90 11000.10 0 1 -6491.94

Predict\_Stepwise = predict(modelstep,test)  
Predict\_Stepwise = exp(Predict\_Stepwise)  
Predict\_Stepwise = as.data.frame(Predict\_Stepwise)  
StepwisePredicted = cbind(test$MSRP,Predict\_Stepwise)  
names(StepwisePredicted)[1] = "test"  
names(StepwisePredicted)[2] = "prediction"  
  
Predict\_Forward = predict(modelforward,test)  
Predict\_Forward = exp(Predict\_Forward)  
Predict\_Forward = as.data.frame(Predict\_Forward)  
ForwardPredicted = cbind(test$MSRP,Predict\_Forward)  
names(ForwardPredicted)[1] = "test"  
names(ForwardPredicted)[2] = "prediction"  
  
Predict\_Backward = predict(modelbackward,test)  
Predict\_Backward = exp(Predict\_Backward)  
Predict\_Backward = as.data.frame(Predict\_Backward)  
BackwardPredicted = cbind(test$MSRP,Predict\_Backward)  
names(BackwardPredicted)[1] = "test"  
names(BackwardPredicted)[2] = "prediction"

MSE\_Step <- mse(StepwisePredicted$test, StepwisePredicted$prediction)  
MSE\_Fwd <- mse(ForwardPredicted$test, ForwardPredicted$prediction)  
MSE\_Back <- mse(BackwardPredicted$test, BackwardPredicted$prediction)  
  
AdjR2\_Step <- 1-(MSE\_Step/var(StepwisePredicted$test))  
AdjR2\_Fwd <- 1-(MSE\_Fwd/var(ForwardPredicted$test))  
AdjR2\_Back <- 1-(MSE\_Back/var(BackwardPredicted$test))  
  
MSE\_Step

## [1] 1831110487

MSE\_Fwd

## [1] 1090608849

MSE\_Back

## [1] 927191079

AdjR2\_Step

## [1] 0.1332462

AdjR2\_Fwd

## [1] 0.4837617

AdjR2\_Back

## [1] 0.5611153

###OBJECTIVE2##########

cars2KT <- cars %>% filter(cars$Year>2000)  
  
cars2KT = subset(cars2KT, select = -c(Model, Market.Category))  
  
#view(cars2KT)

##split data set again  
ss1 <- sample(1:3,size=nrow(cars2KT),replace=TRUE,prob=c(0.8,0.1,0.1))  
train1 <- cars2KT[ss==1,]  
test1 <- cars2KT[ss==2,]  
valid1 <- cars2KT[ss==3,]  
#view(train1)  
  
# head(train1)  
# ##complex linear model  
# model2 <- lm(log(MSRP)~.,data=train1)  
# summary(model2)  
# AIC(model2)  
cor(train1$highway.MPG,train1$city.mpg) ##0.9428157

## [1] 0.9428157

cor(train1$Engine.HP,train1$Engine.Cylinders)## 0.7859781

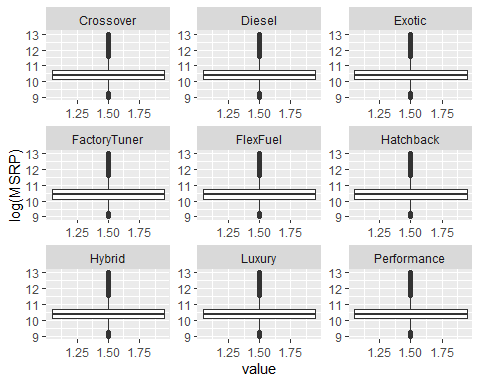
## [1] 0.7859781

# mpgc <- train$city.mpg-mean(train$city.mpg)  
# mpgh <- train$highway.MPG-mean(train$highway.MPG)  
  
##creating interaction variable.names  
#totalmpg <- mpgc \*mpgh   
  
# ols\_step\_both\_aic(model2)  
### ols\_step\_best\_subset(model2) takes time to run

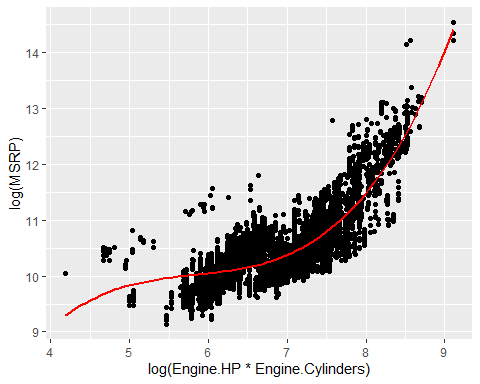
carsTypes = subset(cars, select = c(MSRP, Crossover, Diesel, Exotic, Luxury, Performance, FactoryTuner, FlexFuel, Hatchback, Hybrid))  
  
carsTypes %>%  
 gather(-MSRP, key = "var", value = "value") %>%  
 ggplot(aes(x = value, y = log(MSRP))) +  
 facet\_wrap(~var, scales = "free") +  
 geom\_boxplot()+  
 ylim(9, 13)

## Warning: Continuous x aesthetic -- did you forget aes(group=...)?

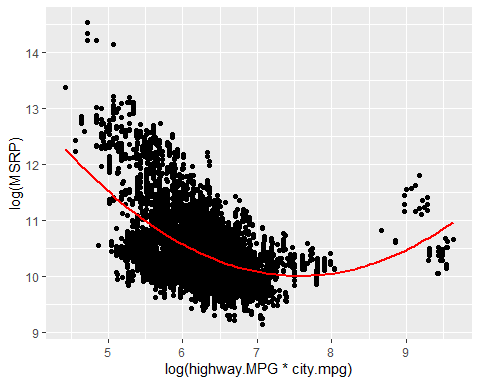
## Warning: Removed 15021 rows containing non-finite values (stat\_boxplot).



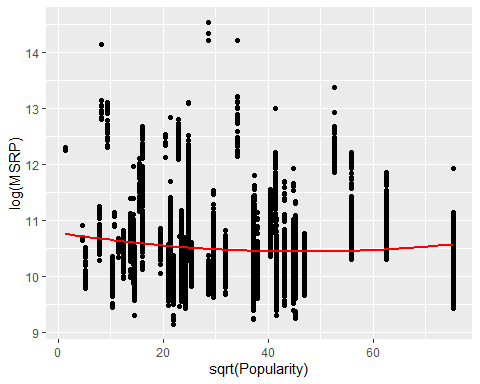
# log of highway mpg \* city mpg to 3nd order  
cars2KT %>%  
 ggplot(aes(x = log(Engine.HP\*Engine.Cylinders), y = log(MSRP))) +  
 geom\_point() +   
 stat\_smooth(method="lm", se=TRUE, fill=NA, formula=y ~ poly(x, 3, raw=TRUE),colour="red")



# log of highway mpg \* city mpg to 2nd order  
cars2KT %>%  
 ggplot(aes(x = log(highway.MPG\*city.mpg), y = log(MSRP))) +  
 geom\_point() +   
 stat\_smooth(method="lm", se=TRUE, fill=NA, formula=y ~ poly(x, 2, raw=TRUE),colour="red")



# Popularity  
cars2KT %>%  
 ggplot(aes(x = sqrt(Popularity), y = log(MSRP))) +  
 geom\_point() +   
 stat\_smooth(method="lm", se=TRUE, fill=NA, formula=y ~ poly(x, 2, raw=TRUE),colour="red")



######################################### model with interaction##############################

modelI<-lm(log(MSRP)~Make+Year+Engine.Fuel.Type+Engine.HP+Engine.Cylinders+Transmission.Type+Driven\_Wheels+Number.of.Doors+highway.MPG+city.mpg+ Popularity+Crossover+Diesel+Exotic+Luxury+Performance+FactoryTuner+FlexFuel+Hatchback+Hybrid+Engine.HP\*Engine.Cylinders+highway.MPG\*city.mpg,data=train1)  
  
summary(modelI)

##   
## Call:  
## lm(formula = log(MSRP) ~ Make + Year + Engine.Fuel.Type + Engine.HP +   
## Engine.Cylinders + Transmission.Type + Driven\_Wheels + Number.of.Doors +   
## highway.MPG + city.mpg + Popularity + Crossover + Diesel +   
## Exotic + Luxury + Performance + FactoryTuner + FlexFuel +   
## Hatchback + Hybrid + Engine.HP \* Engine.Cylinders + highway.MPG \*   
## city.mpg, data = train1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.65044 -0.09734 -0.00466 0.09393 1.42607   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.412e+01 1.227e+00 -11.510 < 2e-16 \*\*\*  
## MakeAlfa Romeo 5.233e-01 7.943e-02 6.587 4.75e-11 \*\*\*  
## MakeAston Martin 5.181e-01 4.211e-02 12.304 < 2e-16 \*\*\*  
## MakeAudi 1.905e-01 1.587e-02 12.005 < 2e-16 \*\*\*  
## MakeBentley 4.252e-01 3.462e-02 12.283 < 2e-16 \*\*\*  
## MakeBMW 1.780e-01 1.547e-02 11.502 < 2e-16 \*\*\*  
## MakeBugatti 1.128e+00 1.069e-01 10.551 < 2e-16 \*\*\*  
## MakeBuick 1.450e-01 3.089e-02 4.693 2.74e-06 \*\*\*  
## MakeCadillac 1.100e-01 1.501e-02 7.332 2.48e-13 \*\*\*  
## MakeChevrolet -6.335e-03 2.822e-02 -0.224 0.822385   
## MakeChrysler 6.619e-02 3.102e-02 2.134 0.032867 \*   
## MakeDodge -1.218e-01 2.919e-02 -4.172 3.05e-05 \*\*\*  
## MakeFerrari 6.555e-01 4.343e-02 15.096 < 2e-16 \*\*\*  
## MakeFIAT 6.064e-02 3.535e-02 1.716 0.086287 .   
## MakeFord 1.832e-02 2.851e-02 0.642 0.520607   
## MakeGenesis -1.802e-01 1.113e-01 -1.619 0.105533   
## MakeGMC 3.248e-02 2.854e-02 1.138 0.255166   
## MakeHonda 8.057e-02 2.883e-02 2.795 0.005202 \*\*   
## MakeHUMMER -2.135e-02 4.553e-02 -0.469 0.639149   
## MakeHyundai -3.479e-02 2.809e-02 -1.238 0.215569   
## MakeInfiniti -7.485e-02 1.540e-02 -4.859 1.20e-06 \*\*\*  
## MakeKia -7.678e-02 2.926e-02 -2.624 0.008698 \*\*   
## MakeLamborghini 5.744e-01 4.543e-02 12.645 < 2e-16 \*\*\*  
## MakeLand Rover 1.970e-01 1.946e-02 10.125 < 2e-16 \*\*\*  
## MakeLexus 8.777e-02 1.763e-02 4.977 6.59e-07 \*\*\*  
## MakeLincoln 3.237e-02 1.873e-02 1.728 0.084087 .   
## MakeLotus 2.762e-01 5.034e-02 5.487 4.21e-08 \*\*\*  
## MakeMaserati -3.658e-02 3.602e-02 -1.016 0.309867   
## MakeMaybach 9.636e-01 5.142e-02 18.739 < 2e-16 \*\*\*  
## MakeMazda 3.403e-02 2.911e-02 1.169 0.242487   
## MakeMcLaren 4.021e-01 7.962e-02 5.050 4.50e-07 \*\*\*  
## MakeMercedes-Benz 1.980e-01 1.611e-02 12.288 < 2e-16 \*\*\*  
## MakeMitsubishi 1.325e-02 3.053e-02 0.434 0.664304   
## MakeNissan -4.957e-03 2.876e-02 -0.172 0.863177   
## MakeOldsmobile 1.372e-01 3.667e-02 3.742 0.000184 \*\*\*  
## MakePontiac -1.578e-02 3.167e-02 -0.498 0.618359   
## MakePorsche 4.911e-01 2.027e-02 24.232 < 2e-16 \*\*\*  
## MakeRolls-Royce 8.143e-01 4.148e-02 19.628 < 2e-16 \*\*\*  
## MakeSaab 9.874e-02 2.406e-02 4.104 4.11e-05 \*\*\*  
## MakeScion -2.707e-02 3.624e-02 -0.747 0.455132   
## MakeSpyker 1.002e+00 9.786e-02 10.238 < 2e-16 \*\*\*  
## MakeSubaru 4.778e-02 2.998e-02 1.593 0.111088   
## MakeSuzuki -1.064e-01 2.945e-02 -3.614 0.000303 \*\*\*  
## MakeTesla -1.211e-01 6.907e-02 -1.753 0.079658 .   
## MakeToyota 1.505e-03 2.859e-02 0.053 0.958034   
## MakeVolkswagen 1.230e-01 2.787e-02 4.413 1.03e-05 \*\*\*  
## MakeVolvo 3.890e-02 1.773e-02 2.194 0.028253 \*   
## Year 1.162e-02 6.135e-04 18.940 < 2e-16 \*\*\*  
## Engine.Fuel.Type -1.405e-02 1.812e-03 -7.755 9.91e-15 \*\*\*  
## Engine.HP 2.315e-03 1.025e-04 22.573 < 2e-16 \*\*\*  
## Engine.Cylinders 8.467e-03 5.205e-03 1.627 0.103834   
## Transmission.Type -4.475e-02 2.347e-03 -19.068 < 2e-16 \*\*\*  
## Driven\_Wheels -1.577e-02 1.950e-03 -8.086 7.03e-16 \*\*\*  
## Number.of.Doors -8.828e-03 2.537e-03 -3.480 0.000505 \*\*\*  
## highway.MPG -1.066e-02 9.270e-04 -11.498 < 2e-16 \*\*\*  
## city.mpg -3.422e-03 1.666e-03 -2.054 0.040049 \*   
## Popularity NA NA NA NA   
## Crossover 2.616e-02 5.159e-03 5.071 4.05e-07 \*\*\*  
## Diesel 1.091e-01 1.729e-02 6.308 2.98e-10 \*\*\*  
## Exotic 5.469e-01 2.708e-02 20.197 < 2e-16 \*\*\*  
## Luxury 2.263e-01 2.495e-02 9.073 < 2e-16 \*\*\*  
## Performance 5.556e-02 5.521e-03 10.063 < 2e-16 \*\*\*  
## FactoryTuner -5.585e-02 9.166e-03 -6.093 1.15e-09 \*\*\*  
## FlexFuel -8.660e-02 9.450e-03 -9.164 < 2e-16 \*\*\*  
## Hatchback -7.981e-02 6.806e-03 -11.727 < 2e-16 \*\*\*  
## Hybrid 2.122e-01 1.507e-02 14.078 < 2e-16 \*\*\*  
## Engine.HP:Engine.Cylinders 5.131e-05 1.575e-05 3.259 0.001125 \*\*   
## highway.MPG:city.mpg 1.255e-04 1.172e-05 10.704 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1564 on 8199 degrees of freedom  
## Multiple R-squared: 0.9333, Adjusted R-squared: 0.9327   
## F-statistic: 1738 on 66 and 8199 DF, p-value: < 2.2e-16

AIC(modelI)

## [1] -7146.755

ols\_step\_both\_aic(modelI, details = T, progress = F)

## Stepwise Selection Method   
## -------------------------  
##   
## Candidate Terms:   
##   
## 1 . Make   
## 2 . Year   
## 3 . Engine.Fuel.Type   
## 4 . Engine.HP   
## 5 . Engine.Cylinders   
## 6 . Transmission.Type   
## 7 . Driven\_Wheels   
## 8 . Number.of.Doors   
## 9 . highway.MPG   
## 10 . city.mpg   
## 11 . Popularity   
## 12 . Crossover   
## 13 . Diesel   
## 14 . Exotic   
## 15 . Luxury   
## 16 . Performance   
## 17 . FactoryTuner   
## 18 . FlexFuel   
## 19 . Hatchback   
## 20 . Hybrid   
## 21 . Engine.HP:Engine.Cylinders   
## 22 . highway.MPG:city.mpg   
##   
## Step 0: AIC = 15100.24   
## log(MSRP) ~ 1   
##   
##   
## Variables Entered/Removed:   
##   
## Enter New Variables   
## -------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## -------------------------------------------------------------------------------------------  
## Engine.HP 1 3041.774 2307.158 698.747 0.768 0.768   
## Engine.HP:Engine.Cylinders 1 4409.173 2181.457 824.448 0.726 0.726   
## Make 1 4981.967 2131.870 874.036 0.709 0.708   
## Engine.Cylinders 1 10306.076 1323.287 1682.618 0.440 0.440   
## Exotic 1 10667.191 1248.149 1757.756 0.415 0.415   
## Performance 1 12463.419 821.505 2184.400 0.273 0.273   
## Luxury 1 12737.265 747.926 2257.979 0.249 0.249   
## highway.MPG 1 14161.800 323.250 2682.655 0.108 0.107   
## city.mpg 1 14407.736 242.235 2763.670 0.081 0.080   
## FactoryTuner 1 14470.275 221.246 2784.659 0.074 0.073   
## Hatchback 1 14511.555 207.305 2798.600 0.069 0.069   
## Number.of.Doors 1 14702.797 141.802 2864.104 0.047 0.047   
## Transmission.Type 1 14725.304 133.992 2871.913 0.045 0.044   
## Engine.Fuel.Type 1 14811.599 103.853 2902.052 0.035 0.034   
## Year 1 14824.857 99.195 2906.710 0.033 0.033   
## highway.MPG:city.mpg 1 14949.136 55.163 2950.743 0.018 0.018   
## Driven\_Wheels 1 15064.822 13.575 2992.330 0.005 0.004   
## Crossover 1 15087.846 5.229 3000.676 0.002 0.002   
## Popularity 1 15098.233 1.456 3004.449 0.000 0.000   
## Diesel 1 15099.890 0.854 3005.051 0.000 0.000   
## Hybrid 1 15101.001 0.450 3005.455 0.000 0.000   
## FlexFuel 1 15101.926 0.113 3005.792 0.000 0.000   
## -------------------------------------------------------------------------------------------  
##   
## - Engine.HP added   
##   
##   
## Step 1 : AIC = 3041.774   
## log(MSRP) ~ Engine.HP   
##   
## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Make 1 -4669.584 2734.053 271.852 0.910 0.909   
## Exotic 1 727.979 2477.887 528.018 0.824 0.824   
## Luxury 1 1651.998 2415.437 590.468 0.804 0.804   
## Transmission.Type 1 2727.958 2333.351 672.554 0.776 0.776   
## FlexFuel 1 2783.479 2328.818 677.087 0.775 0.775   
## city.mpg 1 2803.251 2327.197 678.708 0.774 0.774   
## highway.MPG 1 2821.177 2325.723 680.182 0.774 0.774   
## Diesel 1 2833.662 2324.695 681.210 0.773 0.773   
## highway.MPG:city.mpg 1 2865.516 2322.065 683.840 0.773 0.772   
## Driven\_Wheels 1 2905.046 2318.787 687.118 0.771 0.771   
## Hybrid 1 2915.992 2317.876 688.029 0.771 0.771   
## Performance 1 2930.790 2316.644 689.262 0.771 0.771   
## Engine.Fuel.Type 1 2939.158 2315.945 689.960 0.770 0.770   
## FactoryTuner 1 2943.247 2315.604 690.301 0.770 0.770   
## Popularity 1 2962.584 2313.987 691.918 0.770 0.770   
## Engine.HP:Engine.Cylinders 1 2977.512 2312.737 693.168 0.769 0.769   
## Engine.Cylinders 1 2984.987 2312.109 693.796 0.769 0.769   
## Crossover 1 2987.582 2311.892 694.013 0.769 0.769   
## Number.of.Doors 1 3001.798 2310.697 695.208 0.769 0.769   
## Year 1 3010.201 2309.990 695.915 0.768 0.768   
## Hatchback 1 3041.992 2307.308 698.597 0.768 0.768   
## ------------------------------------------------------------------------------------------  
##   
## - Make added   
##   
##   
## Step 2 : AIC = -4669.584   
## log(MSRP) ~ Engine.HP + Make   
##   
## Remove Existing Variables   
## ------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------  
## Make 1 3041.774 2307.158 698.747 0.768 0.768   
## Engine.HP 1 4981.967 2131.870 874.036 0.709 0.708   
## ------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Transmission.Type 1 -5175.141 2750.243 255.662 0.915 0.914   
## Hybrid 1 -4941.474 2742.913 262.992 0.913 0.912   
## Exotic 1 -4880.795 2740.975 264.930 0.912 0.911   
## city.mpg 1 -4871.014 2740.662 265.243 0.912 0.911   
## highway.MPG:city.mpg 1 -4867.126 2740.537 265.368 0.912 0.911   
## Year 1 -4848.529 2739.939 265.966 0.912 0.911   
## Crossover 1 -4848.023 2739.923 265.982 0.912 0.911   
## Diesel 1 -4830.794 2739.368 266.537 0.911 0.911   
## FactoryTuner 1 -4828.422 2739.291 266.614 0.911 0.911   
## Driven\_Wheels 1 -4818.652 2738.976 266.929 0.911 0.911   
## Engine.Fuel.Type 1 -4818.134 2738.959 266.946 0.911 0.911   
## Luxury 1 -4777.869 2737.656 268.249 0.911 0.910   
## Hatchback 1 -4770.427 2737.414 268.491 0.911 0.910   
## Number.of.Doors 1 -4740.739 2736.448 269.457 0.910 0.910   
## highway.MPG 1 -4728.716 2736.056 269.849 0.910 0.910   
## Performance 1 -4680.085 2734.464 271.441 0.910 0.909   
## Engine.HP:Engine.Cylinders 1 -4673.894 2734.260 271.645 0.910 0.909   
## Engine.Cylinders 1 -4673.074 2734.233 271.672 0.910 0.909   
## FlexFuel 1 -4670.111 2734.136 271.769 0.910 0.909   
## Popularity 1 -4667.584 2734.053 271.852 0.910 0.909   
## ------------------------------------------------------------------------------------------  
##   
## - Transmission.Type added   
##   
##   
## Step 3 : AIC = -5175.141   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type   
##   
## Remove Existing Variables   
## ---------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ---------------------------------------------------------------------------------  
## Transmission.Type 1 -4669.584 2734.053 271.852 0.910 0.909   
## Make 1 2727.958 2333.351 672.554 0.776 0.776   
## Engine.HP 1 4682.104 2163.212 842.693 0.720 0.718   
## ---------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Exotic 1 -5447.487 2758.589 247.316 0.918 0.917   
## Hybrid 1 -5418.662 2757.725 248.180 0.917 0.917   
## highway.MPG:city.mpg 1 -5413.027 2757.556 248.349 0.917 0.917   
## city.mpg 1 -5407.996 2757.405 248.500 0.917 0.917   
## Diesel 1 -5330.325 2755.059 250.846 0.917 0.916   
## Engine.Fuel.Type 1 -5314.898 2754.590 251.315 0.916 0.916   
## Year 1 -5297.542 2754.062 251.843 0.916 0.916   
## Luxury 1 -5282.278 2753.597 252.308 0.916 0.916   
## FactoryTuner 1 -5271.553 2753.269 252.636 0.916 0.915   
## Crossover 1 -5270.884 2753.249 252.656 0.916 0.915   
## highway.MPG 1 -5262.226 2752.984 252.921 0.916 0.915   
## Driven\_Wheels 1 -5252.933 2752.699 253.206 0.916 0.915   
## Hatchback 1 -5237.172 2752.216 253.689 0.916 0.915   
## FlexFuel 1 -5188.364 2750.714 255.191 0.915 0.915   
## Number.of.Doors 1 -5178.687 2750.415 255.490 0.915 0.914   
## Engine.HP:Engine.Cylinders 1 -5178.328 2750.404 255.501 0.915 0.914   
## Performance 1 -5174.089 2750.273 255.632 0.915 0.914   
## Engine.Cylinders 1 -5173.153 2750.244 255.661 0.915 0.914   
## Popularity 1 -5173.141 2750.243 255.662 0.915 0.914   
## ------------------------------------------------------------------------------------------  
##   
## - Exotic added   
##   
##   
## Step 4 : AIC = -5447.487   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic   
##   
## Remove Existing Variables   
## ---------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ---------------------------------------------------------------------------------  
## Exotic 1 -5175.141 2750.243 255.662 0.915 0.914   
## Transmission.Type 1 -4880.795 2740.975 264.930 0.912 0.911   
## Make 1 116.300 2515.668 490.237 0.837 0.837   
## Engine.HP 1 4070.114 2223.539 782.367 0.740 0.738   
## ---------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Hybrid 1 -5690.623 2765.816 240.089 0.920 0.920   
## highway.MPG:city.mpg 1 -5678.817 2765.473 240.432 0.920 0.920   
## city.mpg 1 -5670.069 2765.218 240.687 0.920 0.919   
## Diesel 1 -5607.152 2763.379 242.526 0.919 0.919   
## Engine.Fuel.Type 1 -5605.700 2763.337 242.568 0.919 0.919   
## Year 1 -5602.699 2763.249 242.656 0.919 0.919   
## Luxury 1 -5567.808 2762.222 243.683 0.919 0.918   
## Driven\_Wheels 1 -5557.142 2761.908 243.998 0.919 0.918   
## Crossover 1 -5554.570 2761.832 244.073 0.919 0.918   
## FactoryTuner 1 -5541.881 2761.457 244.448 0.919 0.918   
## highway.MPG 1 -5527.345 2761.026 244.879 0.919 0.918   
## Hatchback 1 -5520.247 2760.816 245.089 0.918 0.918   
## Number.of.Doors 1 -5461.200 2759.059 246.846 0.918 0.917   
## FlexFuel 1 -5455.648 2758.893 247.012 0.918 0.917   
## Engine.HP:Engine.Cylinders 1 -5447.182 2758.640 247.265 0.918 0.917   
## Engine.Cylinders 1 -5446.988 2758.634 247.271 0.918 0.917   
## Performance 1 -5446.916 2758.632 247.273 0.918 0.917   
## Popularity 1 -5445.487 2758.589 247.316 0.918 0.917   
## ------------------------------------------------------------------------------------------  
##   
## - Hybrid added   
##   
##   
## Step 5 : AIC = -5690.623   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid   
##   
## Remove Existing Variables   
## ---------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ---------------------------------------------------------------------------------  
## Hybrid 1 -5447.487 2758.589 247.316 0.918 0.917   
## Exotic 1 -5418.662 2757.725 248.180 0.917 0.917   
## Transmission.Type 1 -5153.712 2749.642 256.263 0.915 0.914   
## Make 1 -20.581 2523.836 482.069 0.840 0.840   
## Engine.HP 1 4068.284 2223.901 782.004 0.740 0.738   
## ---------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Engine.Fuel.Type 1 -5885.546 2771.468 234.437 0.922 0.922   
## highway.MPG:city.mpg 1 -5866.306 2770.922 234.983 0.922 0.921   
## Diesel 1 -5866.147 2770.917 234.988 0.922 0.921   
## Year 1 -5836.933 2770.085 235.820 0.922 0.921   
## Crossover 1 -5809.140 2769.291 236.614 0.921 0.921   
## Driven\_Wheels 1 -5807.434 2769.242 236.663 0.921 0.921   
## city.mpg 1 -5802.950 2769.114 236.791 0.921 0.921   
## FactoryTuner 1 -5791.674 2768.791 237.114 0.921 0.921   
## Hatchback 1 -5789.592 2768.731 237.174 0.921 0.921   
## Luxury 1 -5776.978 2768.369 237.536 0.921 0.920   
## highway.MPG 1 -5725.373 2766.881 239.024 0.920 0.920   
## Number.of.Doors 1 -5699.752 2766.139 239.766 0.920 0.920   
## Engine.Cylinders 1 -5694.773 2765.995 239.910 0.920 0.920   
## FlexFuel 1 -5693.502 2765.958 239.947 0.920 0.920   
## Performance 1 -5691.504 2765.900 240.005 0.920 0.920   
## Engine.HP:Engine.Cylinders 1 -5691.039 2765.886 240.019 0.920 0.920   
## Popularity 1 -5688.623 2765.816 240.089 0.920 0.920   
## ------------------------------------------------------------------------------------------  
##   
## - Engine.Fuel.Type added   
##   
##   
## Step 6 : AIC = -5885.546   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type   
##   
## Remove Existing Variables   
## ---------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ---------------------------------------------------------------------------------  
## Engine.Fuel.Type 1 -5690.623 2765.816 240.089 0.920 0.920   
## Hybrid 1 -5605.700 2763.337 242.568 0.919 0.919   
## Exotic 1 -5592.896 2762.961 242.944 0.919 0.919   
## Transmission.Type 1 -5358.598 2755.976 249.929 0.917 0.916   
## Make 1 -245.705 2536.901 469.004 0.844 0.844   
## Engine.HP 1 3886.023 2241.140 764.765 0.746 0.744   
## ---------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## FlexFuel 1 -6070.279 2776.705 229.200 0.924 0.923   
## Crossover 1 -6024.811 2775.441 230.465 0.923 0.923   
## Driven\_Wheels 1 -6019.420 2775.290 230.615 0.923 0.923   
## Hatchback 1 -5992.420 2774.536 231.369 0.923 0.923   
## Year 1 -5990.132 2774.472 231.433 0.923 0.923   
## highway.MPG:city.mpg 1 -5978.090 2774.134 231.771 0.923 0.922   
## Luxury 1 -5976.277 2774.083 231.822 0.923 0.922   
## FactoryTuner 1 -5972.107 2773.966 231.939 0.923 0.922   
## Diesel 1 -5944.298 2773.185 232.720 0.923 0.922   
## city.mpg 1 -5933.442 2772.879 233.026 0.922 0.922   
## Number.of.Doors 1 -5896.831 2771.845 234.060 0.922 0.922   
## Engine.Cylinders 1 -5891.599 2771.696 234.209 0.922 0.922   
## highway.MPG 1 -5890.276 2771.659 234.246 0.922 0.922   
## Performance 1 -5887.405 2771.578 234.328 0.922 0.922   
## Engine.HP:Engine.Cylinders 1 -5885.211 2771.515 234.390 0.922 0.922   
## Popularity 1 -5883.546 2771.468 234.437 0.922 0.922   
## ------------------------------------------------------------------------------------------  
##   
## - FlexFuel added   
##   
##   
## Step 7 : AIC = -6070.279   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel   
##   
## Remove Existing Variables   
## ---------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ---------------------------------------------------------------------------------  
## FlexFuel 1 -5885.546 2771.468 234.437 0.922 0.922   
## Hybrid 1 -5796.682 2768.934 236.971 0.921 0.921   
## Exotic 1 -5786.947 2768.655 237.250 0.921 0.921   
## Engine.Fuel.Type 1 -5693.502 2765.958 239.947 0.920 0.920   
## Transmission.Type 1 -5475.829 2759.555 246.350 0.918 0.918   
## Make 1 -1051.530 2580.568 425.337 0.858 0.858   
## Engine.HP 1 3887.518 2241.187 764.718 0.746 0.744   
## ---------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Driven\_Wheels 1 -6196.708 2780.238 225.667 0.925 0.924   
## Hatchback 1 -6188.275 2780.008 225.897 0.925 0.924   
## Crossover 1 -6184.369 2779.901 226.004 0.925 0.924   
## FactoryTuner 1 -6175.604 2779.662 226.244 0.925 0.924   
## Year 1 -6165.529 2779.386 226.520 0.925 0.924   
## Luxury 1 -6150.752 2778.980 226.925 0.925 0.924   
## highway.MPG:city.mpg 1 -6108.071 2777.806 228.100 0.924 0.924   
## Engine.Cylinders 1 -6101.858 2777.634 228.271 0.924 0.924   
## Engine.HP:Engine.Cylinders 1 -6079.525 2777.016 228.889 0.924 0.923   
## city.mpg 1 -6078.738 2776.995 228.910 0.924 0.923   
## Number.of.Doors 1 -6078.396 2776.985 228.920 0.924 0.923   
## Diesel 1 -6075.428 2776.903 229.002 0.924 0.923   
## highway.MPG 1 -6071.356 2776.790 229.115 0.924 0.923   
## Performance 1 -6068.405 2776.708 229.197 0.924 0.923   
## Popularity 1 -6068.279 2776.705 229.200 0.924 0.923   
## ------------------------------------------------------------------------------------------  
##   
## - Driven\_Wheels added   
##   
##   
## Step 8 : AIC = -6196.708   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels   
##   
## Remove Existing Variables   
## ---------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ---------------------------------------------------------------------------------  
## Driven\_Wheels 1 -6070.279 2776.705 229.200 0.924 0.923   
## FlexFuel 1 -6019.420 2775.290 230.615 0.923 0.923   
## Hybrid 1 -5913.382 2772.313 233.592 0.922 0.922   
## Exotic 1 -5877.201 2771.288 234.617 0.922 0.921   
## Engine.Fuel.Type 1 -5808.162 2769.320 236.585 0.921 0.921   
## Transmission.Type 1 -5693.967 2766.029 239.876 0.920 0.920   
## Make 1 -1331.344 2594.825 411.080 0.863 0.863   
## Engine.HP 1 3833.892 2246.316 759.589 0.747 0.746   
## ---------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Hatchback 1 -6316.457 2783.538 222.367 0.926 0.926   
## FactoryTuner 1 -6295.752 2782.980 222.925 0.926 0.925   
## Luxury 1 -6279.863 2782.551 223.354 0.926 0.925   
## Year 1 -6276.939 2782.472 223.433 0.926 0.925   
## Crossover 1 -6251.544 2781.785 224.120 0.925 0.925   
## highway.MPG:city.mpg 1 -6239.943 2781.470 224.435 0.925 0.925   
## Engine.Cylinders 1 -6234.062 2781.310 224.595 0.925 0.925   
## Engine.HP:Engine.Cylinders 1 -6210.852 2780.679 225.226 0.925 0.925   
## city.mpg 1 -6208.197 2780.606 225.299 0.925 0.925   
## Diesel 1 -6200.897 2780.407 225.498 0.925 0.924   
## Performance 1 -6198.067 2780.330 225.575 0.925 0.924   
## highway.MPG 1 -6195.361 2780.256 225.649 0.925 0.924   
## Number.of.Doors 1 -6195.224 2780.252 225.653 0.925 0.924   
## Popularity 1 -6194.708 2780.238 225.667 0.925 0.924   
## ------------------------------------------------------------------------------------------  
##   
## - Hatchback added   
##   
##   
## Step 9 : AIC = -6316.457   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback   
##   
## Remove Existing Variables   
## ---------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ---------------------------------------------------------------------------------  
## Hatchback 1 -6196.708 2780.238 225.667 0.925 0.924   
## Driven\_Wheels 1 -6188.275 2780.008 225.897 0.925 0.924   
## FlexFuel 1 -6128.209 2778.361 227.545 0.924 0.924   
## Hybrid 1 -6001.345 2774.841 231.064 0.923 0.923   
## Exotic 1 -5981.440 2774.284 231.621 0.923 0.922   
## Engine.Fuel.Type 1 -5908.949 2772.244 233.661 0.922 0.922   
## Transmission.Type 1 -5862.059 2770.915 234.990 0.922 0.921   
## Make 1 -1393.333 2597.995 407.910 0.864 0.864   
## Engine.HP 1 3399.005 2285.420 720.485 0.760 0.759   
## ---------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Year 1 -6410.753 2786.113 219.792 0.927 0.926   
## highway.MPG:city.mpg 1 -6403.174 2785.912 219.993 0.927 0.926   
## FactoryTuner 1 -6403.141 2785.911 219.994 0.927 0.926   
## Luxury 1 -6393.948 2785.666 220.239 0.927 0.926   
## city.mpg 1 -6358.785 2784.727 221.178 0.926 0.926   
## Crossover 1 -6357.455 2784.692 221.214 0.926 0.926   
## Engine.Cylinders 1 -6344.490 2784.344 221.561 0.926 0.926   
## Engine.HP:Engine.Cylinders 1 -6335.541 2784.104 221.801 0.926 0.926   
## Performance 1 -6323.736 2783.787 222.118 0.926 0.926   
## highway.MPG 1 -6318.090 2783.636 222.270 0.926 0.926   
## Diesel 1 -6316.530 2783.594 222.312 0.926 0.926   
## Number.of.Doors 1 -6314.503 2783.539 222.366 0.926 0.926   
## Popularity 1 -6314.457 2783.538 222.367 0.926 0.926   
## ------------------------------------------------------------------------------------------  
##   
## - Year added   
##   
##   
## Step 10 : AIC = -6410.753   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year   
##   
## Remove Existing Variables   
## ---------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ---------------------------------------------------------------------------------  
## Year 1 -6316.457 2783.538 222.367 0.926 0.926   
## Driven\_Wheels 1 -6298.768 2783.061 222.844 0.926 0.925   
## Hatchback 1 -6276.939 2782.472 223.433 0.926 0.925   
## FlexFuel 1 -6230.736 2781.220 224.685 0.925 0.925   
## Hybrid 1 -6106.506 2777.818 228.088 0.924 0.924   
## Engine.Fuel.Type 1 -6059.936 2776.529 229.376 0.924 0.923   
## Exotic 1 -6051.029 2776.282 229.624 0.924 0.923   
## Transmission.Type 1 -5997.968 2774.803 231.102 0.923 0.923   
## Make 1 -1477.894 2602.244 403.661 0.866 0.866   
## Engine.HP 1 2911.722 2326.830 679.076 0.774 0.773   
## ---------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Engine.Cylinders 1 -6548.594 2789.800 216.105 0.928 0.928   
## Luxury 1 -6500.699 2788.545 217.361 0.928 0.927   
## Engine.HP:Engine.Cylinders 1 -6494.259 2788.375 217.530 0.928 0.927   
## FactoryTuner 1 -6486.518 2788.171 217.734 0.928 0.927   
## highway.MPG:city.mpg 1 -6477.346 2787.930 217.975 0.927 0.927   
## Crossover 1 -6437.930 2786.888 219.017 0.927 0.927   
## city.mpg 1 -6427.020 2786.598 219.307 0.927 0.927   
## Performance 1 -6417.296 2786.340 219.565 0.927 0.926   
## highway.MPG 1 -6414.207 2786.258 219.647 0.927 0.926   
## Diesel 1 -6411.418 2786.184 219.721 0.927 0.926   
## Number.of.Doors 1 -6411.284 2786.181 219.724 0.927 0.926   
## Popularity 1 -6408.753 2786.113 219.792 0.927 0.926   
## ------------------------------------------------------------------------------------------  
##   
## - Engine.Cylinders added   
##   
##   
## Step 11 : AIC = -6548.594   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders   
##   
## Remove Existing Variables   
## ---------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ---------------------------------------------------------------------------------  
## Driven\_Wheels 1 -6434.836 2786.753 219.152 0.927 0.927   
## Hatchback 1 -6427.702 2786.564 219.342 0.927 0.927   
## Engine.Cylinders 1 -6410.753 2786.113 219.792 0.927 0.926   
## Year 1 -6344.490 2784.344 221.561 0.926 0.926   
## FlexFuel 1 -6302.915 2783.227 222.678 0.926 0.925   
## Hybrid 1 -6216.579 2780.889 225.016 0.925 0.925   
## Transmission.Type 1 -6204.594 2780.563 225.342 0.925 0.925   
## Engine.Fuel.Type 1 -6157.613 2779.278 226.627 0.925 0.924   
## Exotic 1 -6139.764 2778.788 227.117 0.924 0.924   
## Engine.HP 1 -2446.762 2650.864 355.041 0.882 0.881   
## Make 1 -1522.266 2604.502 401.403 0.866 0.866   
## ---------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## highway.MPG:city.mpg 1 -6676.734 2793.176 212.729 0.929 0.929   
## Luxury 1 -6626.305 2791.874 214.031 0.929 0.928   
## city.mpg 1 -6621.099 2791.740 214.166 0.929 0.928   
## FactoryTuner 1 -6580.842 2790.694 215.211 0.928 0.928   
## Crossover 1 -6574.130 2790.519 215.386 0.928 0.928   
## Performance 1 -6573.717 2790.508 215.397 0.928 0.928   
## Number.of.Doors 1 -6554.065 2789.996 215.909 0.928 0.928   
## highway.MPG 1 -6551.536 2789.930 215.976 0.928 0.928   
## Engine.HP:Engine.Cylinders 1 -6546.975 2789.810 216.095 0.928 0.928   
## Diesel 1 -6546.962 2789.810 216.095 0.928 0.928   
## Popularity 1 -6546.594 2789.800 216.105 0.928 0.928   
## ------------------------------------------------------------------------------------------  
##   
## - highway.MPG:city.mpg added   
##   
##   
## Step 12 : AIC = -6676.734   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders + highway.MPG:city.mpg   
##   
## Remove Existing Variables   
## ------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------  
## Driven\_Wheels 1 -6550.375 2789.847 216.058 0.928 0.928   
## highway.MPG:city.mpg 1 -6548.594 2789.800 216.105 0.928 0.928   
## FlexFuel 1 -6512.683 2788.859 217.046 0.928 0.927   
## Hatchback 1 -6503.224 2788.611 217.294 0.928 0.927   
## Engine.Cylinders 1 -6477.346 2787.930 217.975 0.927 0.927   
## Engine.Fuel.Type 1 -6470.388 2787.746 218.159 0.927 0.927   
## Year 1 -6469.548 2787.724 218.181 0.927 0.927   
## Hybrid 1 -6402.256 2785.940 219.965 0.927 0.926   
## Transmission.Type 1 -6334.458 2784.129 221.776 0.926 0.926   
## Exotic 1 -6263.464 2782.216 223.689 0.926 0.925   
## Engine.HP 1 -2549.495 2655.334 350.571 0.883 0.883   
## Make 1 -1520.350 2604.506 401.399 0.866 0.866   
## ------------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## highway.MPG 1 -6893.784 2798.739 207.166 0.931 0.931   
## city.mpg 1 -6791.720 2796.166 209.740 0.930 0.930   
## Luxury 1 -6753.628 2795.197 210.708 0.930 0.929   
## Performance 1 -6712.026 2794.134 211.771 0.930 0.929   
## Crossover 1 -6711.569 2794.122 211.783 0.930 0.929   
## FactoryTuner 1 -6705.724 2793.972 211.933 0.929 0.929   
## Number.of.Doors 1 -6688.704 2793.535 212.370 0.929 0.929   
## Diesel 1 -6679.537 2793.300 212.605 0.929 0.929   
## Engine.HP:Engine.Cylinders 1 -6677.841 2793.256 212.649 0.929 0.929   
## Popularity 1 -6674.734 2793.176 212.729 0.929 0.929   
## ------------------------------------------------------------------------------------------  
##   
## - highway.MPG added   
##   
##   
## Step 13 : AIC = -6893.784   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders + highway.MPG:city.mpg + highway.MPG   
##   
## Remove Existing Variables   
## ------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------  
## Engine.Cylinders 1 -6827.036 2797.009 208.896 0.931 0.930   
## Driven\_Wheels 1 -6825.886 2796.980 208.925 0.930 0.930   
## Hatchback 1 -6761.034 2795.335 210.571 0.930 0.929   
## highway.MPG 1 -6676.734 2793.176 212.729 0.929 0.929   
## FlexFuel 1 -6664.487 2792.861 213.044 0.929 0.929   
## Engine.Fuel.Type 1 -6642.355 2792.290 213.616 0.929 0.928   
## Transmission.Type 1 -6565.916 2790.305 215.600 0.928 0.928   
## highway.MPG:city.mpg 1 -6551.536 2789.930 215.976 0.928 0.928   
## Hybrid 1 -6525.956 2789.260 216.645 0.928 0.927   
## Year 1 -6522.766 2789.177 216.729 0.928 0.927   
## Exotic 1 -6465.075 2787.659 218.246 0.927 0.927   
## Engine.HP 1 -3142.337 2679.675 326.230 0.891 0.891   
## Make 1 -1521.417 2604.655 401.250 0.867 0.866   
## ------------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Luxury 1 -6977.122 2800.867 205.038 0.932 0.931   
## Performance 1 -6959.652 2800.433 205.472 0.932 0.931   
## FactoryTuner 1 -6921.966 2799.494 206.411 0.931 0.931   
## Diesel 1 -6906.452 2799.107 206.798 0.931 0.931   
## Number.of.Doors 1 -6905.019 2799.071 206.834 0.931 0.931   
## Crossover 1 -6900.683 2798.962 206.943 0.931 0.931   
## city.mpg 1 -6896.268 2798.852 207.053 0.931 0.931   
## Engine.HP:Engine.Cylinders 1 -6894.203 2798.800 207.105 0.931 0.931   
## Popularity 1 -6891.784 2798.739 207.166 0.931 0.931   
## ------------------------------------------------------------------------------------------  
##   
## - Luxury added   
##   
##   
## Step 14 : AIC = -6977.122   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders + highway.MPG:city.mpg + highway.MPG + Luxury   
##   
## Remove Existing Variables   
## ------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------  
## Engine.Cylinders 1 -6919.644 2799.387 206.519 0.931 0.931   
## Driven\_Wheels 1 -6908.770 2799.115 206.790 0.931 0.931   
## Luxury 1 -6893.784 2798.739 207.166 0.931 0.931   
## Hatchback 1 -6849.574 2797.628 208.277 0.931 0.930   
## FlexFuel 1 -6761.681 2795.402 210.503 0.930 0.929   
## highway.MPG 1 -6753.628 2795.197 210.708 0.930 0.929   
## Engine.Fuel.Type 1 -6731.801 2794.640 211.265 0.930 0.929   
## Hybrid 1 -6650.029 2792.539 213.366 0.929 0.929   
## Transmission.Type 1 -6648.234 2792.493 213.412 0.929 0.929   
## highway.MPG:city.mpg 1 -6628.646 2791.987 213.918 0.929 0.928   
## Year 1 -6593.219 2791.068 214.837 0.929 0.928   
## Exotic 1 -6534.501 2789.536 216.369 0.928 0.928   
## Make 1 -3820.431 2702.153 303.753 0.899 0.899   
## Engine.HP 1 -3236.213 2683.437 322.468 0.893 0.892   
## ------------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Performance 1 -7040.463 2802.482 203.424 0.932 0.932   
## FactoryTuner 1 -7002.151 2801.537 204.369 0.932 0.932   
## Diesel 1 -6992.127 2801.289 204.617 0.932 0.931   
## Number.of.Doors 1 -6991.700 2801.278 204.627 0.932 0.931   
## Crossover 1 -6985.755 2801.131 204.774 0.932 0.931   
## city.mpg 1 -6977.477 2800.926 204.980 0.932 0.931   
## Engine.HP:Engine.Cylinders 1 -6976.704 2800.906 204.999 0.932 0.931   
## Popularity 1 -6975.122 2800.867 205.038 0.932 0.931   
## ------------------------------------------------------------------------------------------  
##   
## - Performance added   
##   
##   
## Step 15 : AIC = -7040.463   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders + highway.MPG:city.mpg + highway.MPG + Luxury + Performance   
##   
## Remove Existing Variables   
## ------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------  
## Performance 1 -6977.122 2800.867 205.038 0.932 0.931   
## Engine.Cylinders 1 -6965.127 2800.569 205.336 0.932 0.931   
## Luxury 1 -6959.652 2800.433 205.472 0.932 0.931   
## Driven\_Wheels 1 -6957.054 2800.369 205.536 0.932 0.931   
## Hatchback 1 -6895.089 2798.822 207.083 0.931 0.931   
## FlexFuel 1 -6841.625 2797.478 208.427 0.931 0.930   
## Engine.Fuel.Type 1 -6805.856 2796.575 209.330 0.930 0.930   
## highway.MPG 1 -6786.787 2796.091 209.814 0.930 0.930   
## Hybrid 1 -6691.393 2793.656 212.249 0.929 0.929   
## Transmission.Type 1 -6677.664 2793.303 212.602 0.929 0.929   
## highway.MPG:city.mpg 1 -6651.888 2792.639 213.266 0.929 0.929   
## Year 1 -6620.512 2791.828 214.077 0.929 0.928   
## Exotic 1 -6578.765 2790.744 215.161 0.928 0.928   
## Engine.HP 1 -4208.423 2719.289 286.617 0.905 0.904   
## Make 1 -3918.038 2705.791 300.114 0.900 0.900   
## ------------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## FactoryTuner 1 -7074.691 2803.371 202.534 0.933 0.932   
## Diesel 1 -7071.874 2803.302 202.603 0.933 0.932   
## Crossover 1 -7058.081 2802.964 202.941 0.932 0.932   
## Number.of.Doors 1 -7046.828 2802.687 203.218 0.932 0.932   
## Engine.HP:Engine.Cylinders 1 -7044.657 2802.634 203.271 0.932 0.932   
## city.mpg 1 -7041.208 2802.549 203.356 0.932 0.932   
## Popularity 1 -7038.463 2802.482 203.424 0.932 0.932   
## ------------------------------------------------------------------------------------------  
##   
## - FactoryTuner added   
##   
##   
## Step 16 : AIC = -7074.691   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders + highway.MPG:city.mpg + highway.MPG + Luxury + Performance + FactoryTuner   
##   
## Remove Existing Variables   
## ------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------  
## FactoryTuner 1 -7040.463 2802.482 203.424 0.932 0.932   
## Engine.Cylinders 1 -7025.739 2802.119 203.786 0.932 0.932   
## Performance 1 -7002.151 2801.537 204.369 0.932 0.932   
## Luxury 1 -6997.658 2801.425 204.480 0.932 0.931   
## Driven\_Wheels 1 -6992.957 2801.309 204.596 0.932 0.931   
## Hatchback 1 -6936.087 2799.897 206.008 0.931 0.931   
## FlexFuel 1 -6874.470 2798.355 207.550 0.931 0.930   
## Engine.Fuel.Type 1 -6843.023 2797.564 208.341 0.931 0.930   
## highway.MPG 1 -6819.499 2796.970 208.935 0.930 0.930   
## Transmission.Type 1 -6724.839 2794.564 211.341 0.930 0.929   
## Hybrid 1 -6724.024 2794.543 211.362 0.930 0.929   
## Year 1 -6705.069 2794.058 211.847 0.930 0.929   
## highway.MPG:city.mpg 1 -6686.597 2793.584 212.321 0.929 0.929   
## Exotic 1 -6626.326 2792.030 213.875 0.929 0.928   
## Engine.HP 1 -4376.477 2725.125 280.780 0.907 0.906   
## Make 1 -3977.533 2708.015 297.890 0.901 0.901   
## ------------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Diesel 1 -7108.704 2804.252 201.653 0.933 0.932   
## Crossover 1 -7091.521 2803.832 202.073 0.933 0.932   
## Number.of.Doors 1 -7080.961 2803.574 202.331 0.933 0.932   
## Engine.HP:Engine.Cylinders 1 -7080.400 2803.560 202.345 0.933 0.932   
## city.mpg 1 -7074.026 2803.404 202.501 0.933 0.932   
## Popularity 1 -7072.691 2803.371 202.534 0.933 0.932   
## ------------------------------------------------------------------------------------------  
##   
## - Diesel added   
##   
##   
## Step 17 : AIC = -7108.704   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders + highway.MPG:city.mpg + highway.MPG + Luxury + Performance + FactoryTuner + Diesel   
##   
## Remove Existing Variables   
## ------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------  
## Diesel 1 -7074.691 2803.371 202.534 0.933 0.932   
## Engine.Cylinders 1 -7072.081 2803.307 202.598 0.933 0.932   
## FactoryTuner 1 -7071.874 2803.302 202.603 0.933 0.932   
## Engine.Fuel.Type 1 -7043.957 2802.617 203.288 0.932 0.932   
## Luxury 1 -7028.752 2802.242 203.663 0.932 0.932   
## Driven\_Wheels 1 -7027.790 2802.219 203.686 0.932 0.932   
## Performance 1 -7017.872 2801.974 203.931 0.932 0.932   
## FlexFuel 1 -7014.370 2801.888 204.017 0.932 0.932   
## Hatchback 1 -6974.511 2800.902 205.003 0.932 0.931   
## highway.MPG 1 -6831.456 2797.323 208.582 0.931 0.930   
## Hybrid 1 -6761.504 2795.550 210.355 0.930 0.929   
## Transmission.Type 1 -6758.466 2795.473 210.432 0.930 0.929   
## Year 1 -6739.911 2795.000 210.905 0.930 0.929   
## highway.MPG:city.mpg 1 -6685.491 2793.607 212.298 0.929 0.929   
## Exotic 1 -6664.129 2793.058 212.848 0.929 0.929   
## Engine.HP 1 -4374.510 2725.126 280.779 0.907 0.906   
## Make 1 -4018.442 2709.558 296.347 0.901 0.901   
## ------------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Crossover 1 -7127.338 2804.754 201.151 0.933 0.933   
## Number.of.Doors 1 -7115.791 2804.473 201.432 0.933 0.932   
## Engine.HP:Engine.Cylinders 1 -7114.997 2804.454 201.451 0.933 0.932   
## city.mpg 1 -7108.194 2804.288 201.617 0.933 0.932   
## Popularity 1 -7106.704 2804.252 201.653 0.933 0.932   
## ------------------------------------------------------------------------------------------  
##   
## - Crossover added   
##   
##   
## Step 18 : AIC = -7127.338   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders + highway.MPG:city.mpg + highway.MPG + Luxury + Performance + FactoryTuner + Diesel + Crossover   
##   
## Remove Existing Variables   
## ------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------  
## Crossover 1 -7108.704 2804.252 201.653 0.933 0.932   
## Diesel 1 -7091.521 2803.832 202.073 0.933 0.932   
## FactoryTuner 1 -7091.294 2803.826 202.079 0.933 0.932   
## Engine.Cylinders 1 -7085.726 2803.690 202.215 0.933 0.932   
## Driven\_Wheels 1 -7072.098 2803.357 202.548 0.933 0.932   
## Engine.Fuel.Type 1 -7067.518 2803.244 202.661 0.933 0.932   
## FlexFuel 1 -7047.176 2802.745 203.160 0.932 0.932   
## Luxury 1 -7045.034 2802.692 203.213 0.932 0.932   
## Performance 1 -7025.724 2802.217 203.688 0.932 0.932   
## Hatchback 1 -6998.492 2801.545 204.360 0.932 0.931   
## highway.MPG 1 -6887.344 2798.779 207.127 0.931 0.931   
## Year 1 -6797.735 2796.521 209.384 0.930 0.930   
## Transmission.Type 1 -6794.019 2796.427 209.478 0.930 0.930   
## Hybrid 1 -6782.946 2796.146 209.759 0.930 0.930   
## highway.MPG:city.mpg 1 -6730.652 2794.815 211.090 0.930 0.929   
## Exotic 1 -6683.188 2793.599 212.306 0.929 0.929   
## Engine.HP 1 -4376.436 2725.259 280.646 0.907 0.906   
## Make 1 -4019.904 2709.682 296.223 0.901 0.901   
## ------------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Number.of.Doors 1 -7137.867 2805.059 200.846 0.933 0.933   
## Engine.HP:Engine.Cylinders 1 -7134.310 2804.973 200.932 0.933 0.933   
## city.mpg 1 -7127.495 2804.807 201.098 0.933 0.933   
## Popularity 1 -7125.338 2804.754 201.151 0.933 0.933   
## ------------------------------------------------------------------------------------------  
##   
## - Number.of.Doors added   
##   
##   
## Step 19 : AIC = -7137.867   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders + highway.MPG:city.mpg + highway.MPG + Luxury + Performance + FactoryTuner + Diesel + Crossover + Number.of.Doors   
##   
## Remove Existing Variables   
## ------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------  
## Number.of.Doors 1 -7127.338 2804.754 201.151 0.933 0.933   
## Crossover 1 -7115.791 2804.473 201.432 0.933 0.932   
## FactoryTuner 1 -7101.975 2804.136 201.769 0.933 0.932   
## Diesel 1 -7100.899 2804.110 201.795 0.933 0.932   
## Engine.Cylinders 1 -7092.344 2803.901 202.004 0.933 0.932   
## Engine.Fuel.Type 1 -7080.300 2803.606 202.299 0.933 0.932   
## Driven\_Wheels 1 -7075.043 2803.478 202.427 0.933 0.932   
## FlexFuel 1 -7057.375 2803.045 202.860 0.933 0.932   
## Luxury 1 -7052.280 2802.920 202.986 0.932 0.932   
## Performance 1 -7046.761 2802.784 203.121 0.932 0.932   
## Hatchback 1 -7002.970 2801.705 204.200 0.932 0.932   
## highway.MPG 1 -6903.139 2799.224 206.681 0.931 0.931   
## Year 1 -6797.873 2796.575 209.330 0.930 0.930   
## Transmission.Type 1 -6792.019 2796.427 209.478 0.930 0.930   
## Hybrid 1 -6790.717 2796.394 209.511 0.930 0.930   
## highway.MPG:city.mpg 1 -6740.047 2795.106 210.800 0.930 0.929   
## Exotic 1 -6704.596 2794.200 211.706 0.930 0.929   
## Engine.HP 1 -4407.800 2726.390 279.515 0.907 0.906   
## Make 1 -4121.725 2713.379 292.526 0.903 0.902   
## ------------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Engine.HP:Engine.Cylinders 1 -7144.505 2805.269 200.636 0.933 0.933   
## city.mpg 1 -7138.058 2805.112 200.793 0.933 0.933   
## Popularity 1 -7135.867 2805.059 200.846 0.933 0.933   
## ------------------------------------------------------------------------------------------  
##   
## - Engine.HP:Engine.Cylinders added   
##   
##   
## Step 20 : AIC = -7144.505   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders + highway.MPG:city.mpg + highway.MPG + Luxury + Performance + FactoryTuner + Diesel + Crossover + Number.of.Doors + Engine.HP:Engine.Cylinders   
##   
## Remove Existing Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Engine.Cylinders 1 -7141.014 2805.136 200.770 0.933 0.933   
## Engine.HP:Engine.Cylinders 1 -7137.867 2805.059 200.846 0.933 0.933   
## Number.of.Doors 1 -7134.310 2804.973 200.932 0.933 0.933   
## Crossover 1 -7121.766 2804.667 201.238 0.933 0.933   
## FactoryTuner 1 -7106.958 2804.307 201.598 0.933 0.932   
## Diesel 1 -7106.902 2804.305 201.600 0.933 0.932   
## Engine.Fuel.Type 1 -7086.215 2803.800 202.105 0.933 0.932   
## Driven\_Wheels 1 -7081.896 2803.694 202.211 0.933 0.932   
## FlexFuel 1 -7062.165 2803.211 202.694 0.933 0.932   
## Luxury 1 -7061.024 2803.183 202.722 0.933 0.932   
## Performance 1 -7046.716 2802.832 203.073 0.932 0.932   
## Hatchback 1 -7003.129 2801.758 204.147 0.932 0.932   
## highway.MPG 1 -6901.671 2799.237 206.668 0.931 0.931   
## Year 1 -6796.754 2796.597 209.308 0.930 0.930   
## Hybrid 1 -6795.331 2796.561 209.344 0.930 0.930   
## Transmission.Type 1 -6792.032 2796.478 209.427 0.930 0.930   
## Exotic 1 -6744.471 2795.269 210.636 0.930 0.929   
## highway.MPG:city.mpg 1 -6738.729 2795.123 210.782 0.930 0.929   
## Engine.HP 1 -6593.828 2791.395 214.510 0.929 0.928   
## Make 1 -4379.759 2722.438 283.467 0.906 0.905   
## ------------------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

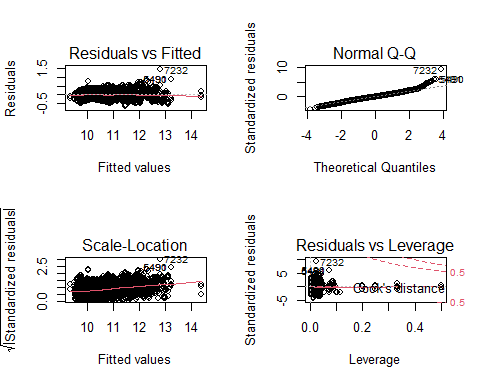
## Enter New Variables   
## --------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## --------------------------------------------------------------------------  
## city.mpg 1 -7146.755 2805.372 200.533 0.933 0.933   
## Popularity 1 -7142.505 2805.269 200.636 0.933 0.933   
## --------------------------------------------------------------------------  
##   
## - city.mpg added   
##   
##   
## Step 21 : AIC = -7146.755   
## log(MSRP) ~ Engine.HP + Make + Transmission.Type + Exotic + Hybrid + Engine.Fuel.Type + FlexFuel + Driven\_Wheels + Hatchback + Year + Engine.Cylinders + highway.MPG:city.mpg + highway.MPG + Luxury + Performance + FactoryTuner + Diesel + Crossover + Number.of.Doors + Engine.HP:Engine.Cylinders + city.mpg   
##   
## Remove Existing Variables   
## ------------------------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## ------------------------------------------------------------------------------------------  
## Engine.Cylinders 1 -7146.088 2805.307 200.598 0.933 0.933   
## city.mpg 1 -7144.505 2805.269 200.636 0.933 0.933   
## Engine.HP:Engine.Cylinders 1 -7138.058 2805.112 200.793 0.933 0.933   
## Number.of.Doors 1 -7136.558 2805.076 200.829 0.933 0.933   
## Crossover 1 -7122.872 2804.743 201.162 0.933 0.933   
## FactoryTuner 1 -7111.406 2804.464 201.441 0.933 0.932   
## Diesel 1 -7108.738 2804.399 201.506 0.933 0.932   
## Engine.Fuel.Type 1 -7088.349 2803.901 202.004 0.933 0.932   
## Driven\_Wheels 1 -7083.096 2803.773 202.132 0.933 0.932   
## Luxury 1 -7066.177 2803.359 202.546 0.933 0.932   
## FlexFuel 1 -7064.516 2803.318 202.587 0.933 0.932   
## Performance 1 -7047.296 2802.895 203.010 0.932 0.932   
## highway.MPG:city.mpg 1 -7034.047 2802.570 203.335 0.932 0.932   
## highway.MPG 1 -7016.539 2802.139 203.766 0.932 0.932   
## Hatchback 1 -7011.268 2802.009 203.896 0.932 0.932   
## Hybrid 1 -6951.315 2800.524 205.381 0.932 0.931   
## Year 1 -6794.798 2796.598 209.307 0.930 0.930   
## Transmission.Type 1 -6790.101 2796.479 209.426 0.930 0.930   
## Exotic 1 -6747.407 2795.395 210.510 0.930 0.929   
## Engine.HP 1 -6650.378 2792.909 212.996 0.929 0.929   
## Make 1 -4385.961 2722.719 283.186 0.906 0.906   
## ------------------------------------------------------------------------------------------

## Warning in b \* sx: longer object length is not a multiple of shorter object  
## length

## Enter New Variables   
## --------------------------------------------------------------------------  
## Variable DF AIC Sum Sq RSS R-Sq Adj. R-Sq   
## --------------------------------------------------------------------------  
## Popularity 1 -7144.755 2805.372 200.533 0.933 0.933   
## --------------------------------------------------------------------------  
##   
##   
## No more variables to be added or removed.  
##   
## Final Model Output   
## ------------------  
##   
## Model Summary   
## -------------------------------------------------------------  
## R 0.966 RMSE 0.156   
## R-Squared 0.933 Coef. Var 1.489   
## Adj. R-Squared 0.933 MSE 0.024   
## Pred R-Squared 0.932 MAE 0.119   
## -------------------------------------------------------------  
## RMSE: Root Mean Square Error   
## MSE: Mean Square Error   
## MAE: Mean Absolute Error   
##   
## ANOVA   
## ------------------------------------------------------------------------  
## Sum of   
## Squares DF Mean Square F Sig.   
## ------------------------------------------------------------------------  
## Regression 2805.372 66 42.506 1737.886 0.0000   
## Residual 200.533 8199 0.024   
## Total 3005.905 8265   
## ------------------------------------------------------------------------  
##   
## Parameter Estimates   
## -----------------------------------------------------------------------------------------------------------  
## model Beta Std. Error Std. Beta t Sig lower upper   
## -----------------------------------------------------------------------------------------------------------  
## (Intercept) -14.124 1.227 -11.510 0.000 -16.529 -11.719   
## Engine.HP 0.002 0.000 0.426 22.573 0.000 0.002 0.003   
## MakeAlfa Romeo 0.523 0.079 0.019 6.587 0.000 0.368 0.679   
## MakeAston Martin 0.518 0.042 0.082 12.304 0.000 0.436 0.601   
## MakeAudi 0.191 0.016 0.053 12.005 0.000 0.159 0.222   
## MakeBentley 0.425 0.035 0.061 12.283 0.000 0.357 0.493   
## MakeBMW 0.178 0.015 0.051 11.502 0.000 0.148 0.208   
## MakeBugatti 1.128 0.107 0.036 10.551 0.000 0.918 1.338   
## MakeBuick 0.145 0.031 0.030 4.693 0.000 0.084 0.206   
## MakeCadillac 0.110 0.015 0.035 7.332 0.000 0.081 0.139   
## MakeChevrolet -0.006 0.028 -0.003 -0.224 0.822 -0.062 0.049   
## MakeChrysler 0.066 0.031 0.014 2.134 0.033 0.005 0.127   
## MakeDodge -0.122 0.029 -0.040 -4.172 0.000 -0.179 -0.065   
## MakeFerrari 0.656 0.043 0.088 15.096 0.000 0.570 0.741   
## MakeFIAT 0.061 0.035 0.008 1.716 0.086 -0.009 0.130   
## MakeFord 0.018 0.029 0.008 0.642 0.521 -0.038 0.074   
## MakeGenesis -0.180 0.111 -0.005 -1.619 0.106 -0.398 0.038   
## MakeGMC 0.032 0.029 0.011 1.138 0.255 -0.023 0.088   
## MakeHonda 0.081 0.029 0.026 2.795 0.005 0.024 0.137   
## MakeHUMMER -0.021 0.046 -0.001 -0.469 0.639 -0.111 0.068   
## MakeHyundai -0.035 0.028 -0.010 -1.238 0.216 -0.090 0.020   
## MakeInfiniti -0.075 0.015 -0.022 -4.859 0.000 -0.105 -0.045   
## MakeKia -0.077 0.029 -0.019 -2.624 0.009 -0.134 -0.019   
## MakeLamborghini 0.574 0.045 0.070 12.645 0.000 0.485 0.663   
## MakeLand Rover 0.197 0.019 0.037 10.125 0.000 0.159 0.235   
## MakeLexus 0.088 0.018 0.019 4.977 0.000 0.053 0.122   
## MakeLincoln 0.032 0.019 0.006 1.728 0.084 -0.004 0.069   
## MakeLotus 0.276 0.050 0.024 5.487 0.000 0.178 0.375   
## MakeMaserati -0.037 0.036 -0.005 -1.016 0.310 -0.107 0.034   
## MakeMaybach 0.964 0.051 0.066 18.739 0.000 0.863 1.064   
## MakeMazda 0.034 0.029 0.010 1.169 0.242 -0.023 0.091   
## MakeMcLaren 0.402 0.080 0.016 5.050 0.000 0.246 0.558   
## MakeMercedes-Benz 0.198 0.016 0.054 12.288 0.000 0.166 0.230   
## MakeMitsubishi 0.013 0.031 0.003 0.434 0.664 -0.047 0.073   
## MakeNissan -0.005 0.029 -0.002 -0.172 0.863 -0.061 0.051   
## MakeOldsmobile 0.137 0.037 0.017 3.742 0.000 0.065 0.209   
## MakePontiac -0.016 0.032 -0.003 -0.498 0.618 -0.078 0.046   
## MakePorsche 0.491 0.020 0.089 24.232 0.000 0.451 0.531   
## MakeRolls-Royce 0.814 0.041 0.078 19.628 0.000 0.733 0.896   
## MakeSaab 0.099 0.024 0.014 4.104 0.000 0.052 0.146   
## MakeScion -0.027 0.036 -0.003 -0.747 0.455 -0.098 0.044   
## MakeSpyker 1.002 0.098 0.032 10.238 0.000 0.810 1.194   
## MakeSubaru 0.048 0.030 0.011 1.593 0.111 -0.011 0.107   
## MakeSuzuki -0.106 0.029 -0.030 -3.614 0.000 -0.164 -0.049   
## MakeTesla -0.121 0.069 -0.008 -1.753 0.080 -0.256 0.014   
## MakeToyota 0.002 0.029 0.001 0.053 0.958 -0.055 0.058   
## MakeVolkswagen 0.123 0.028 0.054 4.413 0.000 0.068 0.178   
## MakeVolvo 0.039 0.018 0.009 2.194 0.028 0.004 0.074   
## Transmission.Type -0.045 0.002 -0.063 -19.068 0.000 -0.049 -0.040   
## Exotic 0.547 0.027 0.198 20.197 0.000 0.494 0.600   
## Hybrid 0.212 0.015 0.065 14.078 0.000 0.183 0.242   
## Engine.Fuel.Type -0.014 0.002 -0.043 -7.755 0.000 -0.018 -0.010   
## FlexFuel -0.087 0.009 -0.046 -9.164 0.000 -0.105 -0.068   
## Driven\_Wheels -0.016 0.002 -0.028 -8.086 0.000 -0.020 -0.012   
## Hatchback -0.080 0.007 -0.040 -11.727 0.000 -0.093 -0.066   
## Year 0.012 0.001 0.085 18.940 0.000 0.010 0.013   
## Engine.Cylinders 0.008 0.005 0.019 1.627 0.104 -0.002 0.019   
## highway.MPG -0.011 0.001 -0.150 -11.498 0.000 -0.012 -0.009   
## Luxury 0.226 0.025 0.170 9.073 0.000 0.177 0.275   
## Performance 0.056 0.006 0.043 10.063 0.000 0.045 0.066   
## FactoryTuner -0.056 0.009 -0.021 -6.093 0.000 -0.074 -0.038   
## Diesel 0.109 0.017 0.025 6.308 0.000 0.075 0.143   
## Crossover 0.026 0.005 0.017 5.071 0.000 0.016 0.036   
## Number.of.Doors -0.009 0.003 -0.012 -3.480 0.001 -0.014 -0.004   
## city.mpg -0.003 0.002 -0.053 -2.054 0.040 -0.007 0.000   
## highway.MPG:city.mpg 0.000 0.000 0.190 10.704 0.000 0.000 0.000   
## Engine.HP:Engine.Cylinders 0.000 0.000 0.077 3.259 0.001 0.000 0.000   
## -----------------------------------------------------------------------------------------------------------

##   
##   
## Stepwise Summary   
## --------------------------------------------------------------------------------------------------  
## Variable Method AIC RSS Sum Sq R-Sq Adj. R-Sq   
## --------------------------------------------------------------------------------------------------  
## Engine.HP addition 3041.774 698.747 2307.158 0.76754 0.76751   
## Make addition -4669.584 271.852 2734.053 0.90956 0.90904   
## Transmission.Type addition -5175.141 255.662 2750.243 0.91495 0.91445   
## Exotic addition -5447.487 247.316 2758.589 0.91772 0.91723   
## Hybrid addition -5690.623 240.089 2765.816 0.92013 0.91964   
## Engine.Fuel.Type addition -5885.546 234.437 2771.468 0.92201 0.92152   
## FlexFuel addition -6070.279 229.200 2776.705 0.92375 0.92327   
## Driven\_Wheels addition -6196.708 225.667 2780.238 0.92493 0.92444   
## Hatchback addition -6316.457 222.367 2783.538 0.92602 0.92554   
## Year addition -6410.753 219.792 2786.113 0.92688 0.92639   
## Engine.Cylinders addition -6548.594 216.105 2789.800 0.92811 0.92762   
## highway.MPG:city.mpg addition -6676.734 212.729 2793.176 0.92923 0.92874   
## highway.MPG addition -6893.784 207.166 2798.739 0.93108 0.93059   
## Luxury addition -6977.122 205.038 2800.867 0.93179 0.93130   
## Performance addition -7040.463 203.424 2802.482 0.93233 0.93183   
## FactoryTuner addition -7074.691 202.534 2803.371 0.93262 0.93212   
## Diesel addition -7108.704 201.653 2804.252 0.93291 0.93241   
## Crossover addition -7127.338 201.151 2804.754 0.93308 0.93257   
## Number.of.Doors addition -7137.867 200.846 2805.059 0.93318 0.93266   
## Engine.HP:Engine.Cylinders addition -7144.505 200.636 2805.269 0.93325 0.93272   
## city.mpg addition -7146.755 200.533 2805.372 0.93329 0.93275   
## --------------------------------------------------------------------------------------------------

par(mfrow = c(2,2))  
plot(modelI) ## residual plot is somewhat normal distribution . There is no points that is low residual and high leverage. The model looks good



### making prediction with interaction model

Predict\_I = predict(modelI,test1)

## Warning in predict.lm(modelI, test1): prediction from a rank-deficient fit may  
## be misleading

Predict\_I = exp(Predict\_I)  
Predict\_I = as.data.frame(Predict\_I)  
Predicted\_I\_N = cbind(test1$MSRP,Predict\_I)  
names(Predicted\_I\_N)[1] = "test"  
names(Predicted\_I\_N)[2] = "prediction"

## metrics of interaction model

MSE\_I <- mse(Predicted\_I\_N$test, Predicted\_I\_N$prediction)  
MSE\_I

## [1] 271835863

AdjR2\_I <- 1-(MSE\_I/var(Predicted\_I\_N$test))  
AdjR2\_I

## [1] 0.8713268

## validation set prediction by inteaction model

valid1<- valid1[!(valid1$Make %in% c("Plymouth")),] ### Need to drop Plymouth make because it was only on validation set  
  
Predict\_IV = predict(modelI,valid1)

## Warning in predict.lm(modelI, valid1): prediction from a rank-deficient fit may  
## be misleading

Predict\_IV = exp(Predict\_IV)  
Predict\_IV= as.data.frame(Predict\_IV)  
Predicted\_I\_NV = cbind(valid1$MSRP,Predict\_IV)  
names(Predicted\_I\_NV)[1] = "valid"  
names(Predicted\_I\_NV)[2] = "prediction"

##metrics of interaction model FOR Validation set

MSE\_I <- mse(Predicted\_I\_NV$valid, Predicted\_I\_NV$prediction)  
MSE\_I

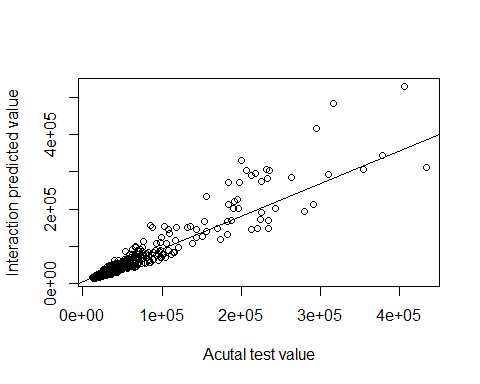
## [1] 169955911

AdjR2\_I <- 1-(MSE\_I/var(Predicted\_I\_NV$valid))  
AdjR2\_I

## [1] 0.9017187

##plot actual vs predicted value Interaction

plot(Predicted\_I\_N$test, Predicted\_I\_N$prediction, xlab = "Acutal test value",ylab="Interaction predicted value")  
abline(lm(Predicted\_I\_N$test~Predicted\_I\_N$prediction))



################################LASSO###############################################

#x=model.matrix(MSRP~.,train1)[,-1]  
#y=log(train1$MSRP)  
  
#xtest<-model.matrix(MSRP~.,test1)[,-1]  
#ytest<-log(test1$MSRP)  
  
#xvalid<-model.matrix(MSRP~.,valid1)[,-1] ##for validation set  
#yvalid<-log(valid1$MSRP)  
  
#grid=10^seq(10,-2, length =100)  
#lasso.mod=glmnet(x,y,alpha=1, lambda =grid)  
  
#cv.out=cv.glmnet(x,y,alpha=1) #alpha=1 performs LASSO  
#plot(cv.out)

#bestlambda<-cv.out$lambda.min   
#bestlambda

#lasso.pred=predict (lasso.mod ,s=bestlambda ,newx=xtest)  
#lasso.pred

#testMSE\_LASSO<-mean((ytest-lasso.pred)^2)  
#testMSE\_LASSO  
  
##0.03036748 ASE(MSE)

###MSE

#lasso.pred =exp(lasso.pred)  
#lasso.pred= as.data.frame(lasso.pred)  
#LPredicted = cbind(test1$MSRP,lasso.pred)  
#names(LPredicted)[1] = "test"  
#names(LPredicted)[2] = "prediction"  
# head(LPredicted)  
# view(LPredicted)  
  
#mean((LPredicted$test-LPredicted$prediction)^2)  
 ##312204479

##metrics of lassomodel

#SSE\_L <- sse(LPredicted$test,LPredicted$prediction)  
#SSE\_L  
  
#MSE\_L <- mse(LPredicted$test, LPredicted$prediction)  
#MSE\_L   
  
#AdjR2\_L <- 1-(MSE\_Step/var(LPredicted$test))  
#AdjR2\_L

#coef(lasso.mod,s=bestlambda)

## validation set prediction by lasso model

#lasso.predV=predict (lasso.mod ,s=bestlambda ,newx=xvalid)  
#lasso.predV =exp(lasso.predV)  
#lasso.predV= as.data.frame(lasso.predV)  
#LPredictedV = cbind(valid1$MSRP,lasso.predV)  
#names(LPredictedV)[1] = "valid"  
#names(LPredictedV)[2] = "prediction"

##metrics of Lasso model FOR Validation set

#SSE\_I <- sse(LPredictedV$valid, LPredictedV$prediction)  
#SSE\_I  
  
#MSE\_I <- mse(LPredictedV$valid, LPredictedV$prediction)  
#MSE\_I   
  
#AdjR2\_I <- 1-(MSE\_Step/var(LPredictedV$valid1))  
#AdjR2\_I

##plot actual vs predicted value laSSO

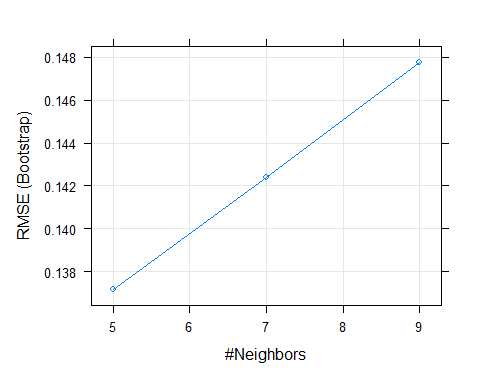
#plot(test1$MSRP,LPredicted$prediction, xlab = "Acutal test value",ylab="Lasso predicted value")  
#abline(lm(test1$MSRP~LPredicted$prediction))

#########obj 2 KNN###################

m1 <-train(log(MSRP)~.,data=train1,method='knn')  
m1

## k-Nearest Neighbors   
##   
## 8266 samples  
## 22 predictor  
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 8266, 8266, 8266, 8266, 8266, 8266, ...   
## Resampling results across tuning parameters:  
##   
## k RMSE Rsquared MAE   
## 5 0.1371763 0.9480110 0.0974647  
## 7 0.1423991 0.9439232 0.1009818  
## 9 0.1477516 0.9396078 0.1044672  
##   
## RMSE was used to select the optimal model using the smallest value.  
## The final value used for the model was k = 5.

plot(m1)



test\_pred <- predict(m1, newdata = test1)  
test\_pred= exp(test\_pred)  
test\_pred= as.data.frame(test\_pred)  
KNNPredicted = cbind(test1$MSRP,test\_pred)  
names(KNNPredicted)[1] = "test"  
names(KNNPredicted)[2] = "prediction"  
  
  
  
#head(KNNPredicted)  
## calcaulte MSE  
#mean((KNNPredicted$test-KNNPredicted$prediction)^2)  
# 95373406

##metrics of knn

MSE\_K <- mse(KNNPredicted$test, KNNPredicted$prediction)  
MSE\_K

## [1] 95373406

AdjR2\_K <- 1-(MSE\_K/var(KNNPredicted$test))  
AdjR2\_K

## [1] 0.9548551

AIC(m1)

## Error in UseMethod("logLik"): no applicable method for 'logLik' applied to an object of class "c('train', 'train.formula')"

## validation set prediction by KNN model

##valid1<- valid1[!(valid1$Make %in% c("Plymouth")),] ### Need to drop Plymouth make because it was only on validation set   
  
test\_predV <- predict(m1, newdata = valid1)  
test\_predV= exp(test\_predV)  
test\_predV= as.data.frame(test\_predV)  
KNNPredictedV= cbind(valid1$MSRP,test\_predV)  
names(KNNPredictedV)[1] = "valid"  
names(KNNPredictedV)[2] = "prediction"

##metrics of knn for validation set

MSE\_K <- mse(KNNPredictedV$valid, KNNPredictedV$prediction)  
MSE\_K

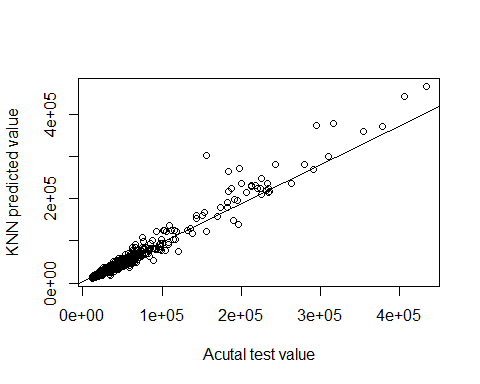
## [1] 59460543

AdjR2\_K <- 1-(MSE\_K/var(KNNPredictedV$valid))  
AdjR2\_K

## [1] 0.9656155

##plot

plot(test1$MSRP,KNNPredicted$prediction, xlab = "Acutal test value",ylab="KNN predicted value")  
  
abline(lm(test1$MSRP~KNNPredicted$prediction))



PRESS <- function(model) {  
 i <- residuals(model)/(1 - lm.influence(model)$hat)  
 sum(i^2)  
}  
  
PRESS(modelI)

## [1] 204.9277

PRESS(lasso.mod)

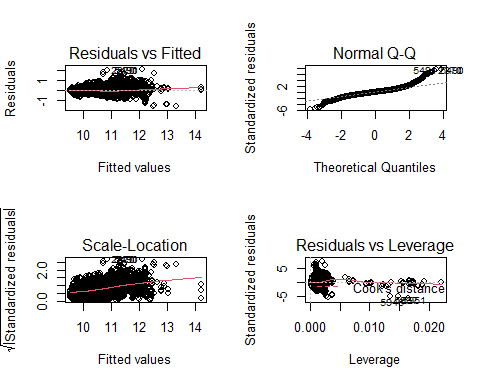
## Error in residuals(model): object 'lasso.mod' not found

PLOTS

# assumption plots of final model (objective 1)  
  
summary(modelbackward)

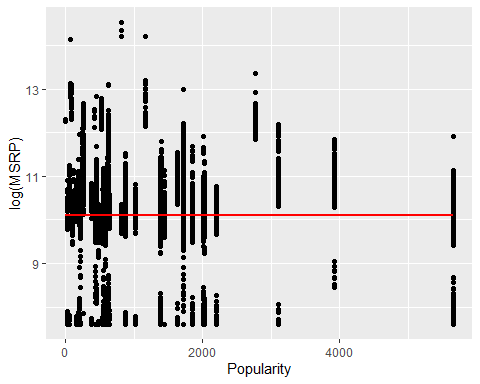
##   
## Call:  
## lm(formula = log(MSRP) ~ . - Year - Vehicle.Style - highway.MPG -   
## Engine.Cylinders - Vehicle.Size, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.52643 -0.14988 -0.00996 0.13191 2.02146   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 9.772e+00 3.228e-02 302.714 < 2e-16 \*\*\*  
## Engine.Fuel.Type -1.043e-02 1.685e-03 -6.189 6.36e-10 \*\*\*  
## Engine.HP 4.755e-03 3.135e-05 151.649 < 2e-16 \*\*\*  
## Transmission.Type -7.763e-02 3.792e-03 -20.473 < 2e-16 \*\*\*  
## Driven\_Wheels -3.618e-02 2.890e-03 -12.521 < 2e-16 \*\*\*  
## Number.of.Doors -6.318e-02 3.950e-03 -15.996 < 2e-16 \*\*\*  
## city.mpg 5.707e-03 3.629e-04 15.729 < 2e-16 \*\*\*  
## Popularity -2.223e-05 2.139e-06 -10.394 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2729 on 8258 degrees of freedom  
## Multiple R-squared: 0.7954, Adjusted R-squared: 0.7952   
## F-statistic: 4585 on 7 and 8258 DF, p-value: < 2.2e-16

par(mfrow = c(2,2))  
plot(modelbackward)



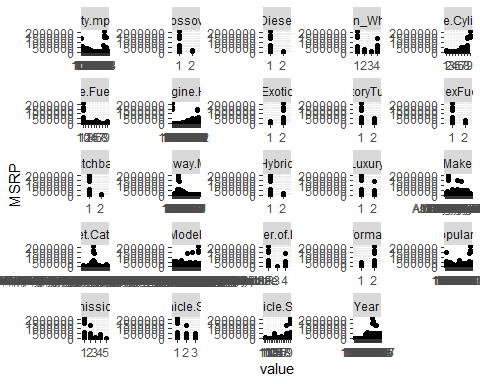
# correlation of MSRP and Popularity  
cars %>%  
 ggplot(aes(x = Popularity, y = log(MSRP))) +  
 geom\_point() +  
 geom\_smooth(method = "lm", colour = "red", se = FALSE)

## `geom\_smooth()` using formula 'y ~ x'

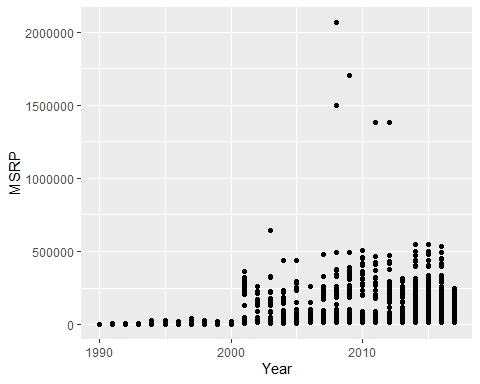


# visualize each variable against response variable (MSRP)  
cars %>%  
 gather(-MSRP, key = "var", value = "value") %>%  
 ggplot(aes(x = value, y = MSRP)) +  
 facet\_wrap(~var, scales = "free") +  
 geom\_point()

## Warning: attributes are not identical across measure variables;  
## they will be dropped

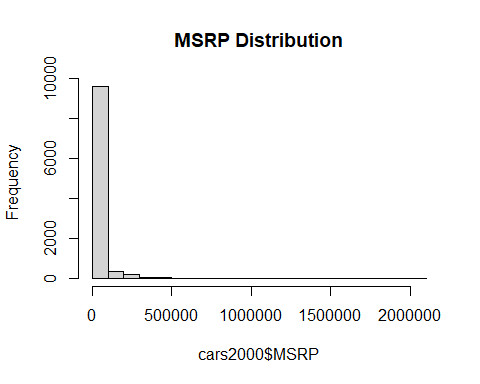


# visualize change in car prices (MSRP) over time  
cars %>%  
 ggplot(aes(x = Year, y = MSRP)) +  
 geom\_point()

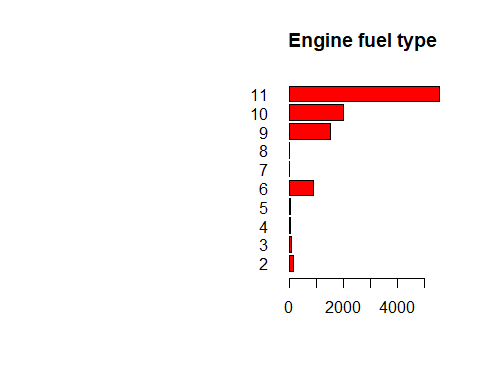


## There is a clear change (possibly in how the data is collected or calculated) in MSRP from years <= 2000 and >2000. There are plenty of records on each side to complete our analysis so we will exclude all cars with "Year"<2000 as the "customer" is undoubtedly more interested in current/recent trends and correlations over the past 20 years rather than those older than 20 years.

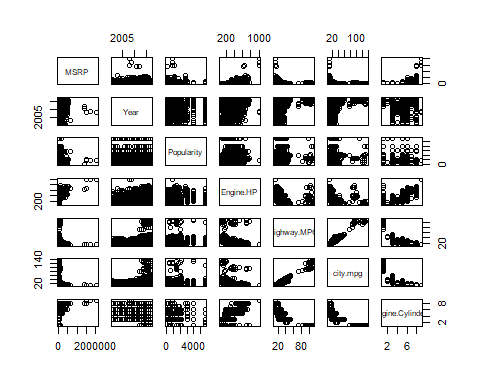
#visualizing MSRP distribution  
hist(cars2000$MSRP,main="MSRP Distribution") ### since this is heavily right skewed



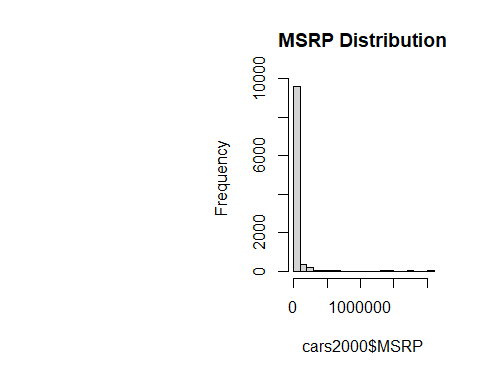
# visualizing categorical variable   
par(mar = c(5.5,15,4.1,2.1))  
barplot(plyr::count(cars2000,'Engine.Fuel.Type')[,2], names.arg=plyr::count(cars2000,'Engine.Fuel.Type')[,1], horiz=TRUE, col='red',las=1, main = "Engine fuel type")



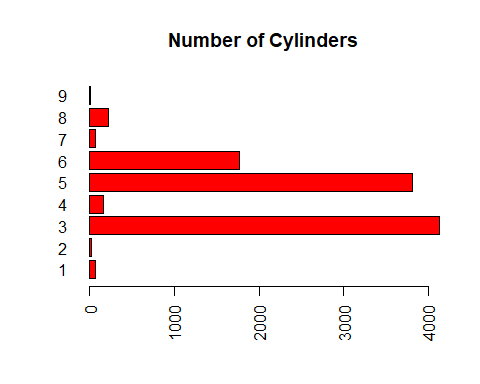
## more number of cars with regular unleaded gas  
  
# scatter plot for numeric variable  
pairs(~MSRP+Year+Popularity+Engine.HP+highway.MPG+city.mpg+Engine.Cylinders, data=cars2000)



## Engine HP and MSRP is correlated   
## city and highway mpg is highly correlated.  
  
hist(cars2000$MSRP,main="MSRP Distribution") ## this is better distribution of MSRP

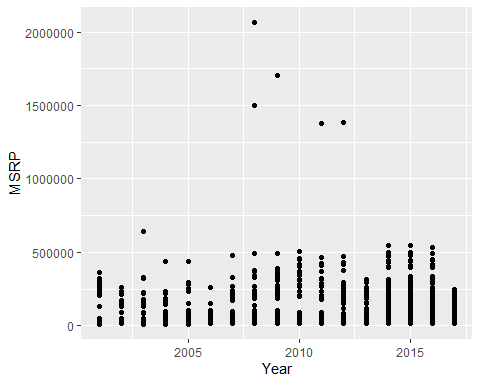


par(mar = c(5.1,4.5,4.1,2.1))  
barplot(plyr::count(cars2000,'Engine.Cylinders')[,2], names.arg=plyr::count(cars2000,'Engine.Cylinders')[,1], horiz=TRUE, col='red',las=2, main = 'Number of Cylinders')

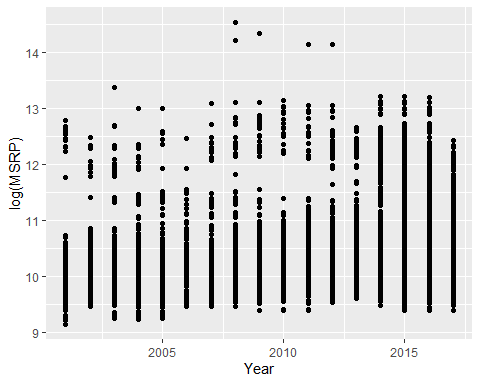


### 4,6,8 are highest cyclinders  
  
#par(mfrow = c(2,2))  
#par(mar = c(5.1,15,4.1,2.1))  
#barplot(plyr::count(cars2000,'Transmission.Type')[,2], names.arg=plyr::count(cars2000,'Transmission.Type')[,1], horiz=TRUE, col='red',las=2, main = 'Transmission Type')  
#barplot(plyr::count(cars2000,'Number.of.Doors')[,2], names.arg=plyr::count(cars2000,'Number.of.Doors')[,1], horiz=TRUE, col='red',las=2, main = 'Number of Doors')  
#barplot(plyr::count(cars2000,'Driven\_Wheels')[,2], names.arg=plyr::count(cars2000,'Driven\_Wheels')[,1], horiz=TRUE, col='red',las=2, main = 'Driven Wheels')  
#barplot(plyr::count(cars2000,'Vehicle.Size')[,2], names.arg=plyr::count(cars2000,'Vehicle.Size')[,1], horiz=TRUE, col='red',las=2, main = 'Vehicle Size')  
#par(mar = c(5.1,10,4.1,2.1))  
#barplot(plyr::count(cars2000,'Vehicle.Style ')[,2], names.arg=plyr::count(cars2000,'Vehicle.Style ')[,1], horiz=TRUE, col='red',las=2, main = 'Vehicle Style ')

# visualize change in car prices (MSRP) since 2000  
  
cars2000 %>%  
 ggplot(aes(x = Year, y = MSRP)) +  
 geom\_point()

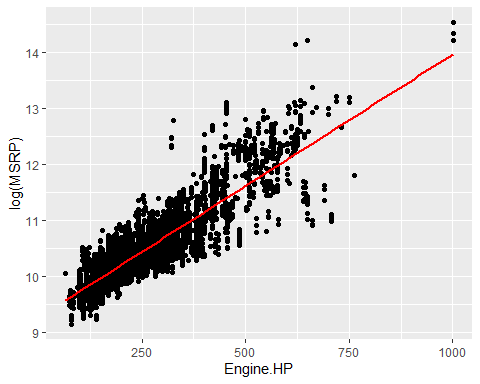


# take log(MRSP)  
cars2000 %>%  
 ggplot(aes(x = Year, y = log(MSRP))) +  
 geom\_point()



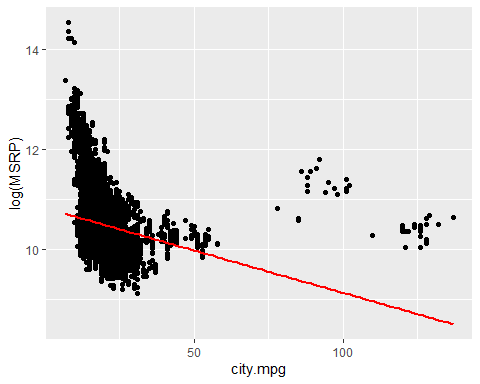
# visualize all individual variable relationships to MSRP  
  
# engine.hp  
cars2000 %>%  
 ggplot(aes(x = Engine.HP, y = log(MSRP))) +  
 geom\_point() +  
 geom\_smooth(method = "lm", colour = "red", se = FALSE)

## `geom\_smooth()` using formula 'y ~ x'

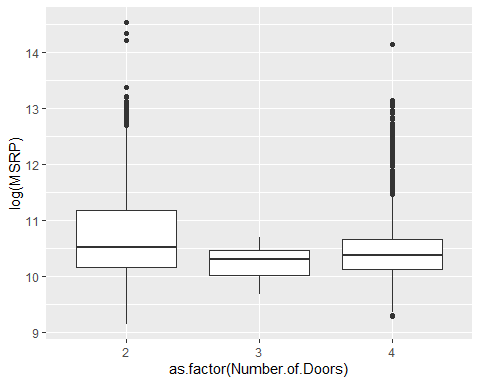


# citympg  
  
cars2000 %>%  
 ggplot(aes(x = city.mpg, y = log(MSRP))) +  
 geom\_point() +  
 geom\_smooth(method = "lm", colour = "red", se = FALSE)

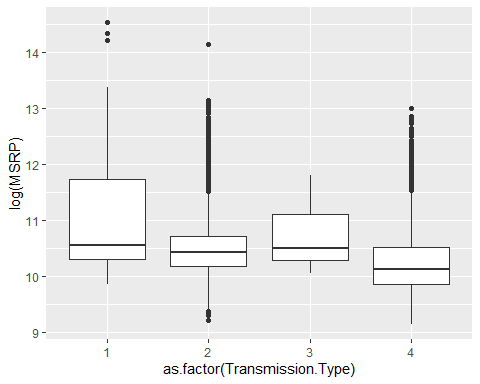
## `geom\_smooth()` using formula 'y ~ x'



# Number of Doors  
  
cars2000 %>%  
ggplot(aes(x=as.factor(Number.of.Doors), y=log(MSRP))) +   
 geom\_boxplot()

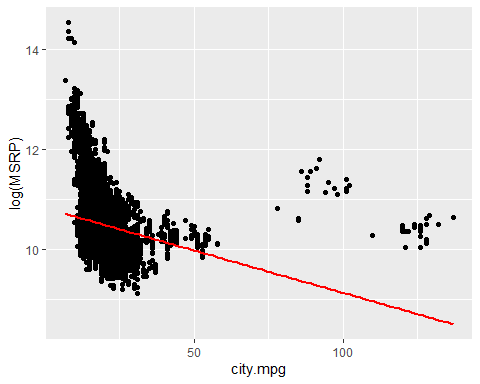


# Transmission type  
  
cars2000 %>%  
ggplot(aes(x=as.factor(Transmission.Type), y=log(MSRP))) +   
 geom\_boxplot()

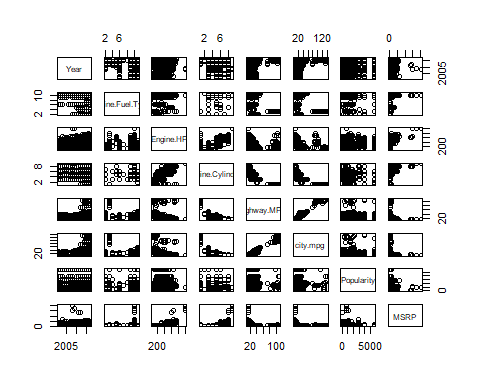


# citympg  
  
cars2000 %>%  
 ggplot(aes(x = city.mpg, y = log(MSRP))) +  
 geom\_point() +  
 geom\_smooth(method = "lm", colour = "red", se = FALSE)

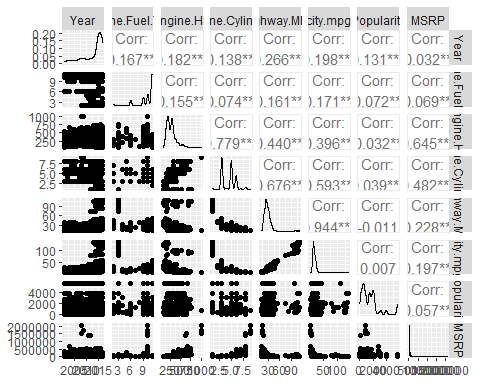
## `geom\_smooth()` using formula 'y ~ x'



pairs((cars2000[,c(1,2,3,4,10,11,12,13)]))



ggpairs((cars2000[,c(1,2,3,4,10,11,12,13)]))



plot\_num(cars2000)

## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =  
## "none")` instead.

