

FinalPaper

2022-12-07

Description/Introduction

Running Project

We have two major files to run. One is the DataCleaning.Rmd and the other is ShinyR.R. The former shows the different visualizations and models while the latter displays our Shiny application. The Presentation.Rmd also outputs our final presentation.

Purpose

Our team has a significant interest in cars: the faster and more luxurious the better. Therefore, it would make sense that our keen interest was in German cars which are known to be the best-engineered cars in the world. Obviously, though, we are all students and on a tight budget that makes it difficult to afford such luxurious vehicles. Still, there are likely some German cars that may be just as fast and nice but cheaper in price. Therefore, our team aimed at finding what effects the price of different German cars. Specifically, we were looking for specific factors that effect the price in hopes of finding some specific model or set of models that are available and fit our budget and driving needs. Below, we perform such analysis in an effort to find our dream cars while still being able to afford staying in school. In addition to that, we noticed that such a problem really applies to the general American public as a whole. Specifically, Covid-19 has significantly disrupted supply chains and the availability of different car parts, ad, thus, different cars. Not only has the price of new cars from different dealerships increased significantly over recent years, but also the price of rentals through services like AVIS, Hertz, and Budget have seen significant increases at multiples of 5 times as high as they were just a year or two before Covid-19 started. Therefore, we found our data analysis crucial to those Americans that still want to travel in luxury at an affordable price.

Data Retrieval

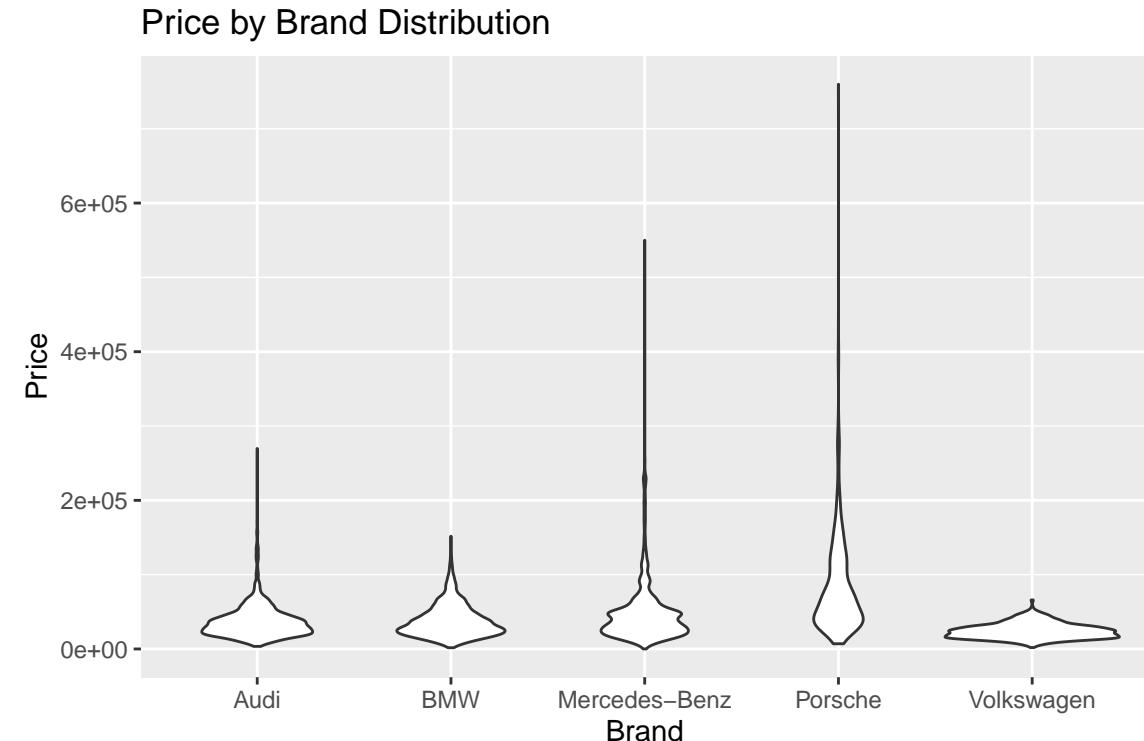
We decided to retrieve used car data from www.cars.com and focus on 5 German brands including Audi, BMW, Mercedes, Porsche and Volkswagen. In addition, we filter only cars listed within 200 miles around Champaign (ZIP: 61820) to make it relevant to us who study at UIUC. Thus, we come up with API [https://www.cars.com/shopping/results/list_price_max=&makes\[\]=%20volkswagen&maximum_distance=200](https://www.cars.com/shopping/results/list_price_max=&makes[]=%20volkswagen&maximum_distance=200) which we will use for data scraping. In order to retrieve all the data, we use **repeat** loop and increase **page** until no data is returned. Then, we change makes and rerun the code. Technically, we use library **rvest** to get HTML data and point to specific HTML elements which contain required information. The data from this process has 3 fields including Detail, Price, and Mileage. While Price and Mileage can be used easily, Detail contains year, model and sub-model which is very specific to brands so we will need data manipulation technique in the next step.

R Shiny

To make users better interacting with the data and customize the data visualization based on their needs, we created the ShinyR. In the ShinyR, we mainly have three sections: overall plot, statistics summary, and depreciation trend graph. For the overall plot section, users can choose the specific car brand, car make, and year they want look into, and then the plot for price versus mileage will come up. Users can either choose to click on the data points or brush through an area to get more specific information behind the data. For the statistics summary section, users will get an overview of the minimum, average, mean, and max price and mileage for different brand and make to have a general idea of where the price falls into. As for the depreciation trend graph, it clearly shows how the price of a car will decrease as the age of the car increase. The smaller the slope of the graph indicates a better value-maintain and a better investment choice. Lastly, we also created a download button, enable users to download the data and have a further analysis based on their needs.

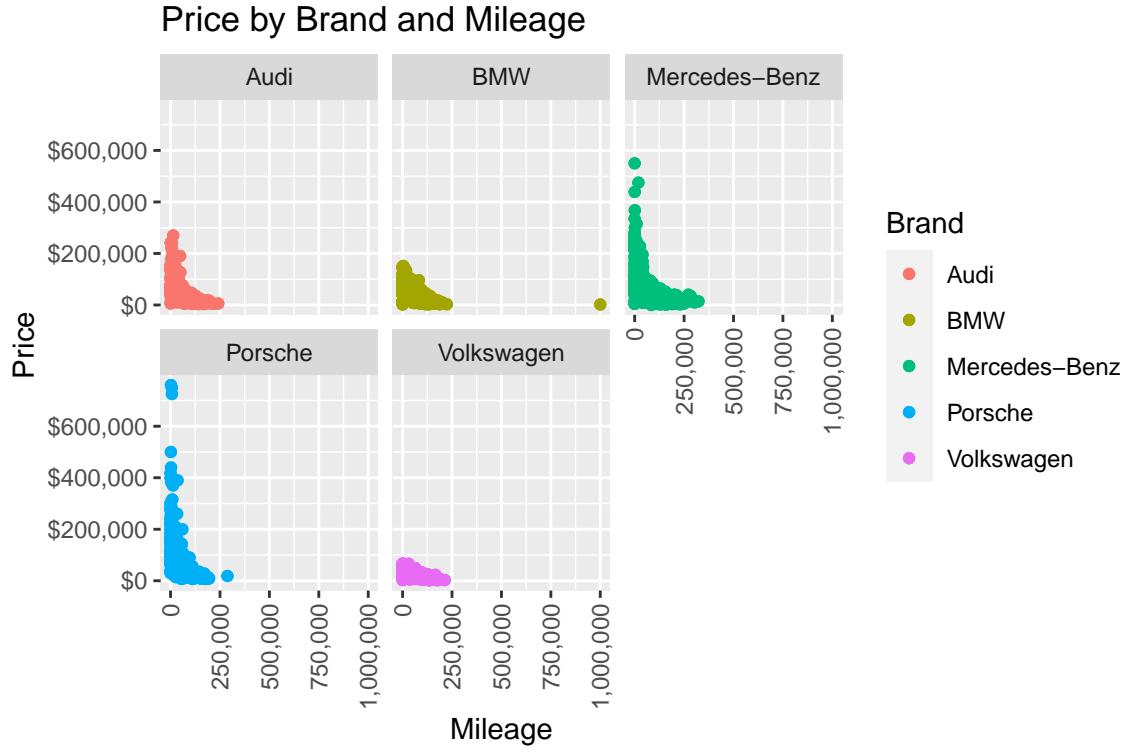
Exploratory Data Analysis

We began our work by analyzing the differences between brands. Below, we have a violin plot which gives a general idea about the distribution of the price based on the car brand. From that plot, we learn a lot about how the brand effects the price. Specifically, we get a more clear idea about how the price data is distributed across our different car brands. We see that Porsche and Mercedes have some specific models that are likely very high-end and much more expensive. Also, Audi likely has some models that are not quite as luxurious but still far from average. Meanwhile, BMW and Volkswagen have almost no such cars. We see that there is a point where all brands have a very highly distributed amount of variables that is at nearly the same spot for all three brands except Mercedes and Volkswagen. For Mercedes we see both a lower and higher peak instead, and for Volkswagen we see that peak in the distribution at a slightly lower price point.



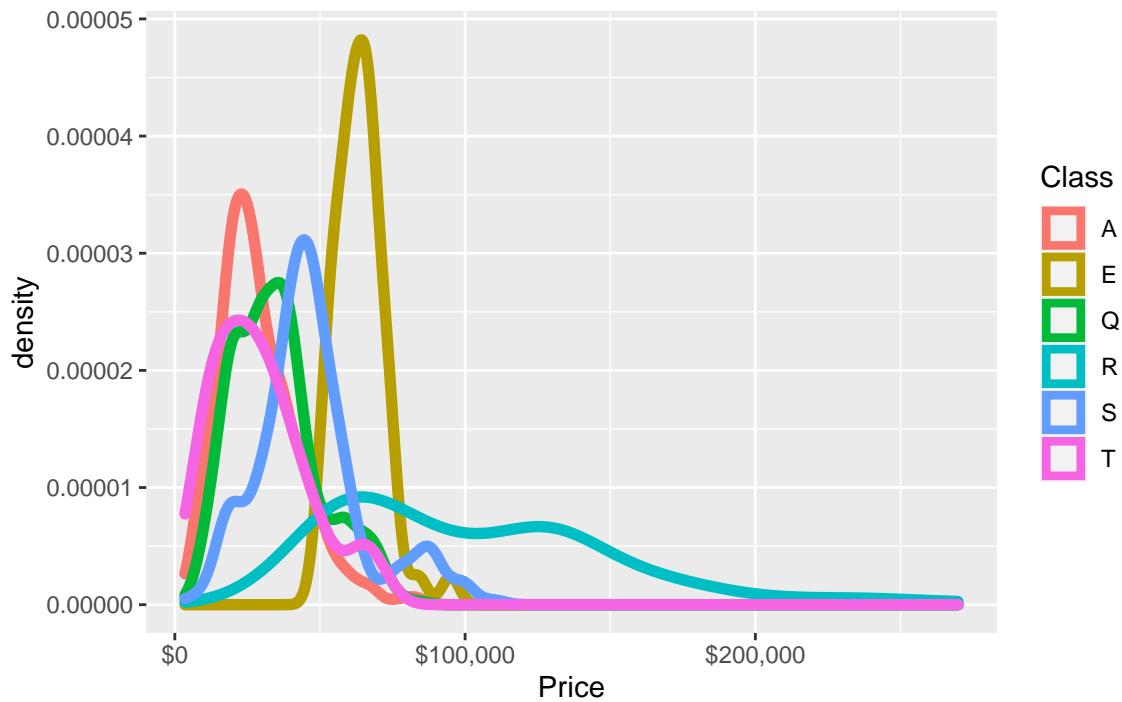
The following graph lays out data points by the mileage that cars have driven and the associated cost for that car, which is color-coded by its brand, we have a few initial takeaways about how our different variables might be effecting the price. Firstly, we notice an exponential decay in cost as mileage increases. Additionally, we

see that Volkswagen is by far the cheapest and has a relatively small range in comparison to its counterparts. This makes sense because Volkswagen is typically not thought of as a luxury brand so we would not expect many super high-priced vehicles. Above that, we see a sea of blue followed by a sea of green. This means that Porsche tends to be a little cheaper than Mercedes. However, we see that some new Porsches can be quite expensive while those with more miles on them tends to be a little cheaper to a comparable Mercedes.



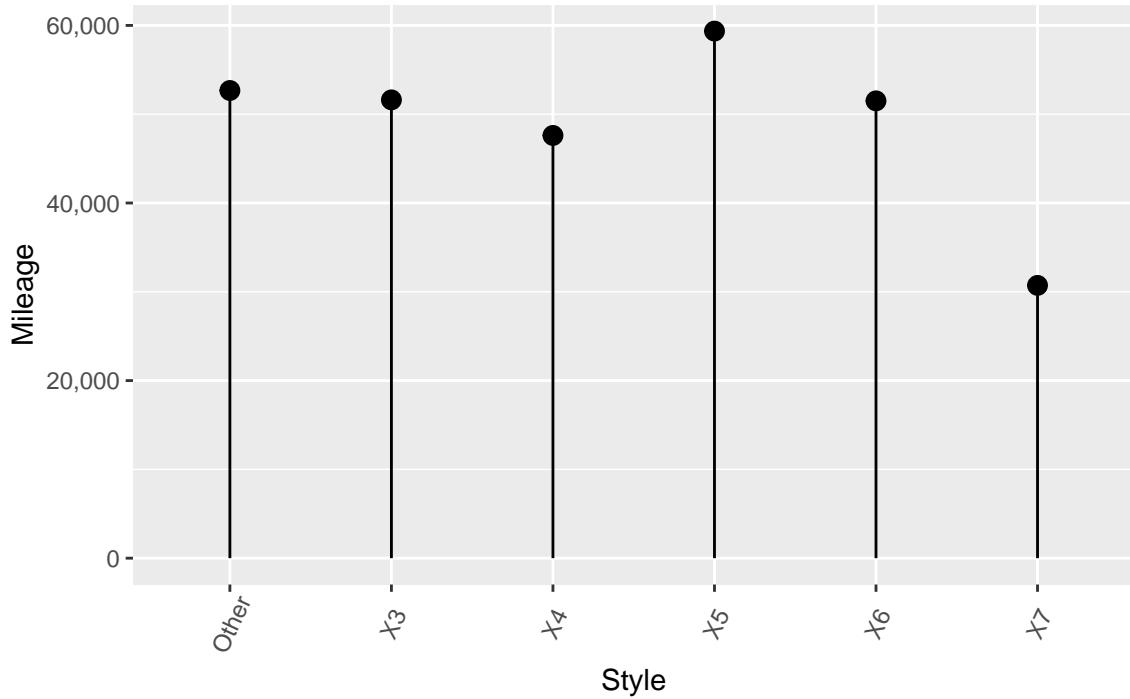
Audi This plot gives us a good idea of how the price differs across the class of the car for different Audi's. We can see pretty strong right-skews for A, R, and T class cars, meaning that they likely have some very high-end cars. On the other hand we can see for e-class cars (representing etrons) that there is a left skew, meaning that these cars tend to be priced on the specific model of car. Moreover, it means that electronic cars being made by Audi tend to be priced in a small range. While, this range is higher than the middle for other car classes, we see that there are also a lot of cars on the lower end of the scale of the distribution. For A, T, Q, and S class cars, we see that their mean tends to be close to the same but we have a much larger variability for the T-class cars. We find that R-class cars have the most variability as the data is almost bimodal with two different peaks. This suggests that there are two sets of R-class cars, one whose mean is near (but still higher) than the other classes and another that is much higher end and expensive than the other classes.

Audi: Density by Price and Class



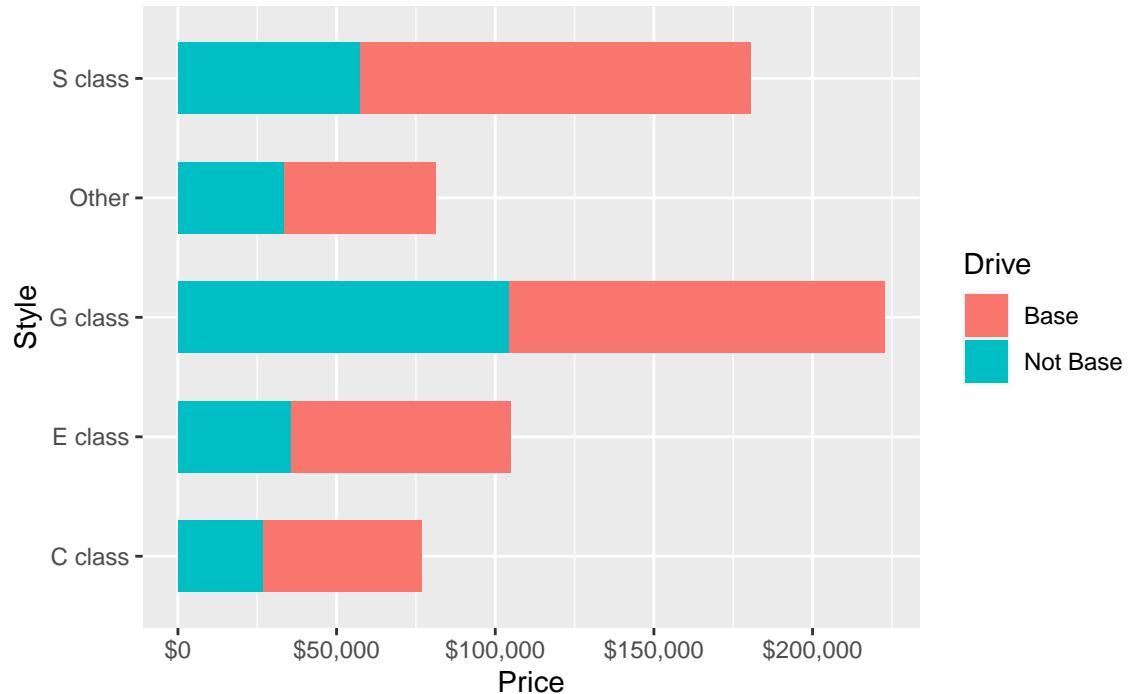
BMW From this graphic, we see that x7's have the least mileage on them by far and also are representative of much of the newer cars as they have the lowest mean average age. Therefore, from the graphs we can not deem that x7's are higher priced just due to being x7's because other factors like mileage and age might also be playing into their higher price.

BMW: Average Mileage by Style



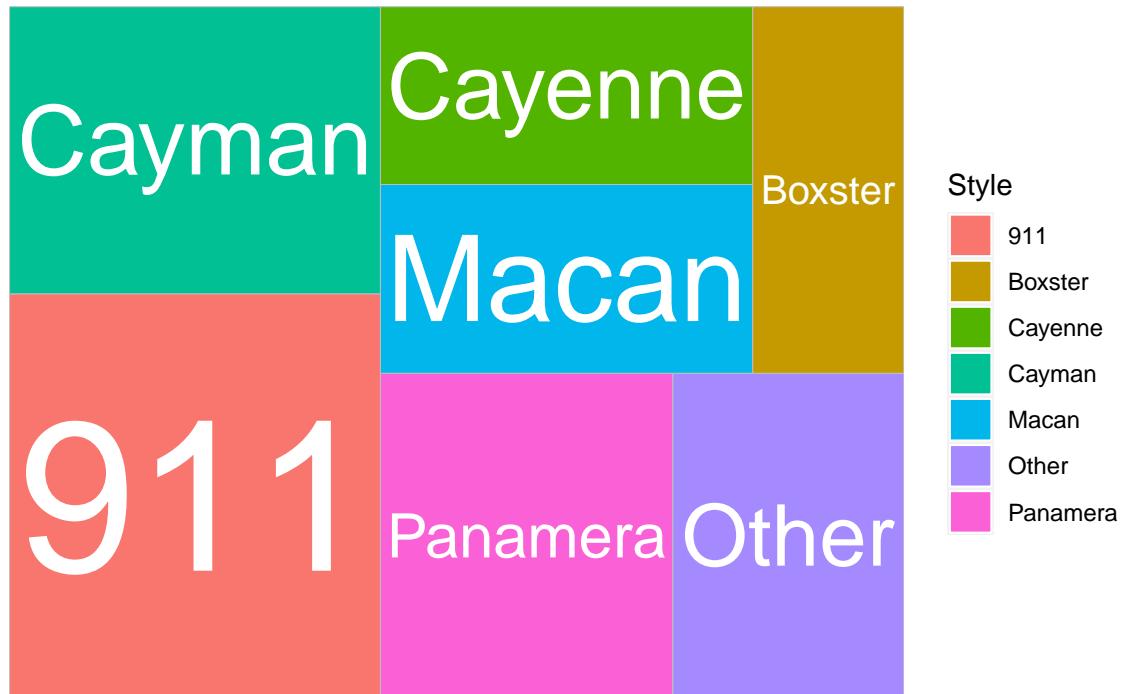
Mercedes Benz From the horizontal bar chart above, we get an idea of how the mean price differs by style of car in general, as well as how the impact of the car being a base model versus not a base model also impacts the price. We can see that in general the other category is by far the most expensive, followed by the g class and s class. Also, we see for S, E, and C classes, the average price of the base models is significantly lower than that of other models but for the G class and other vehicles, the Base class actually ends up being more expensive on average. This is definitely something we will want to take into account for our Mercedes-specific pricing model. We want to ensure that this is something that we want to capture in our model, so we will make sure that other variables are not impacting and skewing this data.

Mercedes-Benz: Average Price by Drive Type and Style Class



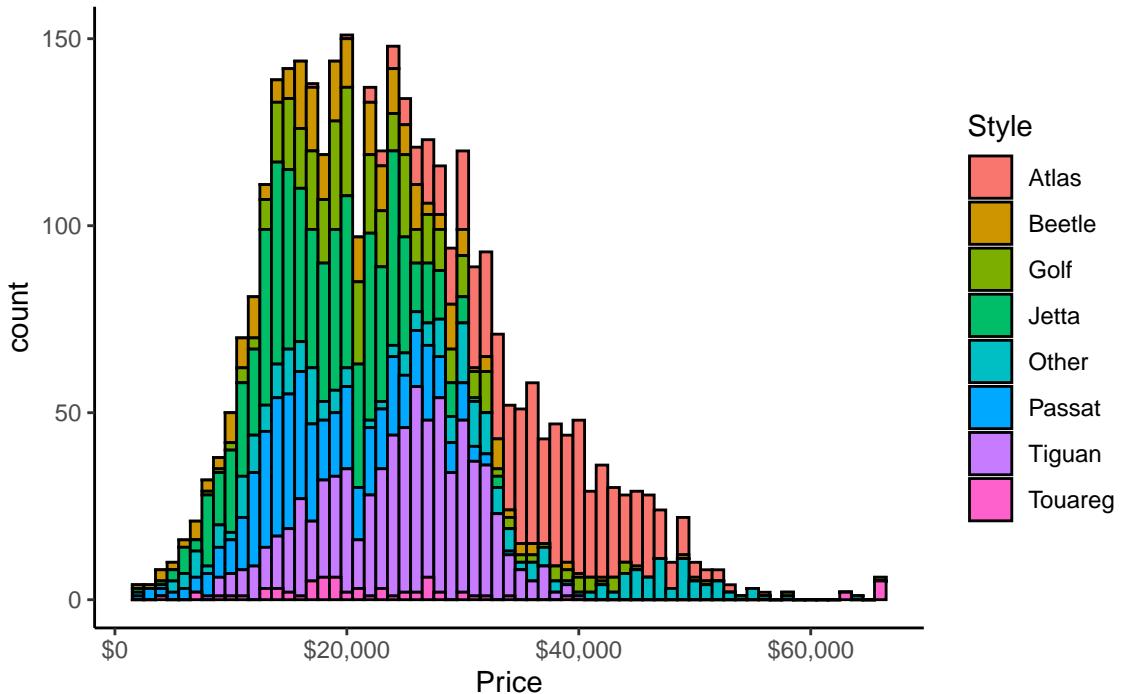
Porsche This treemap displays the average price of each Porsche model in relation to all other models and in general. Specifically, the map is configured such that we find that average price of each model and list each model in the map, and the larger the box is for the model, the more expensive it is on average. Therefore, we find that the 911 tends to be significantly more expensive than all other models, while the Cayenne and Macan are relatively comparable and the Cayman and Panamera are also relatively with one another, respectively. Lastly, we find that the Boxster tends to be significantly cheaper than all other models.

Porsche: Treemap of Price by Model Style



Volkswagen We can clearly see that there are large differences in price across the different German brands. In order to learn more about each brand, we decided to perform price analysis within every brand. The graph below details how price differs across style of Volkswagen. Specifically, for different prices on the x-axis, we see the count of how many of each model lies within that price range. Therefore, we can see right away that Atlas' tend to lie more on the pricier side as they have greater area when the price extends past \$40,000. We can also see a large area for both Jetta and Tiguan in the middle price range of about \$20,000-30,000. We also see the Golf and Touareg lie in this range. Meanwhile, we see both the Golf and Beetle fall in bottom third of the price range at less than \$20,000 which is where a significant amount of their area lies. Thus, there seems to be a strong association with price by style (model) of car for Volkswagens. It is also worth noting that some Touarges seem to be on the very price side around \$60,000 which might skew its specific data a bit.

Volkswagen: Count of Price by Model Style



Modeling Techniques and Results

To begin our modeling process, the first thing we did was created two models with the full dataset: one model including all of the predictors, and one model including all of the predictors as well as interaction terms. We see that the model with interactions tends to perform better than the one with all variables fitted. This is seen in multiple ways: 1. The anova test rejects the non-interaction model in favor of the interaction model 2. The p-values of all predictors in the interaction model show a statistical significance for all variables 3. The R² and Adjusted R² increase for the interaction term model over the model without interaction terms.

```
## [1] "Simple Model:"
```

```
##
## Call:
## lm(formula = Price ~ Years_Old + Brand + Mileage, data = all.clean)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -57171 -12883 -4275   5818 666006
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.104e+04  6.442e+02  94.749 <2e-16 ***
## Years_Old              -8.364e+02  4.693e+01 -17.823 <2e-16 ***
## BrandBMW                6.027e+02  7.153e+02   0.843   0.399
## BrandMercedes-Benz    9.168e+03  7.267e+02  12.616 <2e-16 ***
## BrandPorsche            4.284e+04  9.946e+02  43.075 <2e-16 ***
## BrandVolkswagen         -1.366e+04  7.336e+02 -18.617 <2e-16 ***
```

```

## Mileage           -3.570e-01  7.028e-03 -50.793   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28390 on 15079 degrees of freedom
## Multiple R-squared:  0.3883, Adjusted R-squared:  0.3881
## F-statistic:  1596 on 6 and 15079 DF,  p-value: < 2.2e-16

## [1] "Interaction Model:"

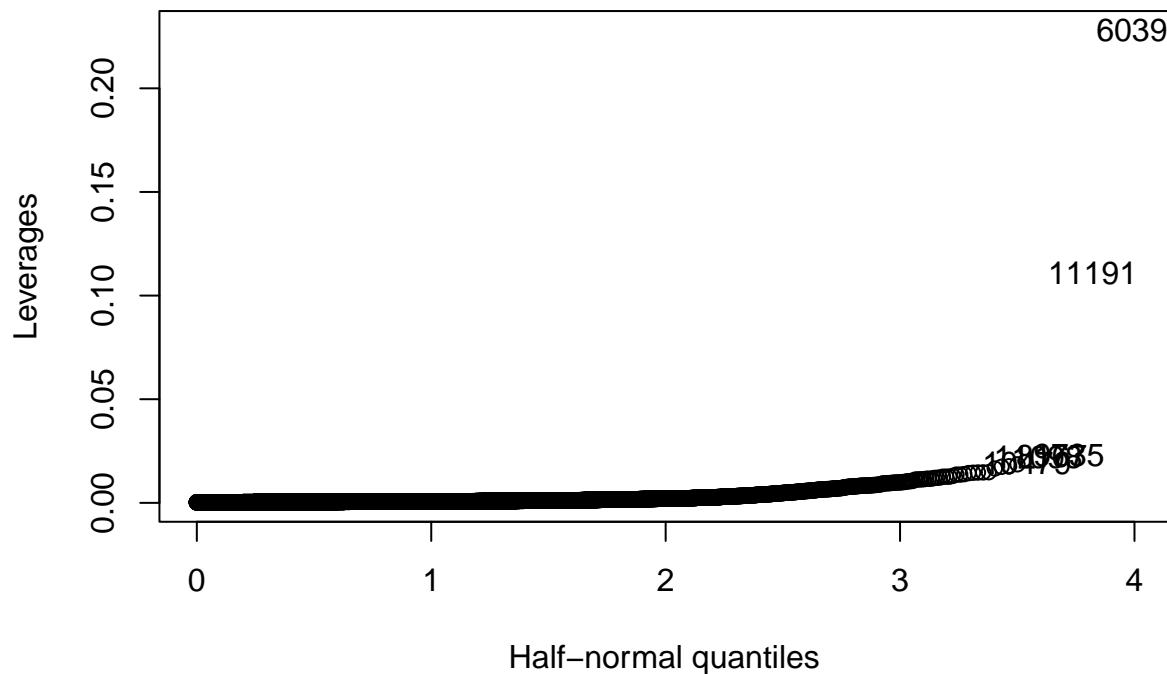

## 
## Call:
## lm(formula = Price ~ Years_Old + Brand + Mileage + Mileage *
##     Years_Old + Brand * Mileage, data = all.clean)
## 
## Residuals:
##    Min      1Q Median      3Q      Max 
## -77551 -10635  -3041   4599  642671 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            6.518e+04  9.478e+02  68.777   < 2e-16 ***
## Years_Old             -1.797e+03  6.494e+01 -27.669   < 2e-16 ***
## BrandBMW              1.038e+03  1.167e+03   0.889    0.374  
## BrandMercedes-Benz    1.274e+04  1.137e+03  11.206   < 2e-16 ***
## BrandPorsche           7.415e+04  1.498e+03  49.494   < 2e-16 ***
## BrandVolkswagen        -2.363e+04  1.206e+03 -19.585   < 2e-16 ***
## Mileage                -4.766e-01  1.533e-02 -31.093   < 2e-16 ***
## Years_Old:Mileage     1.967e-02  8.652e-04  22.731   < 2e-16 ***
## BrandBMW:Mileage      -1.875e-02  1.792e-02  -1.046    0.296  
## BrandMercedes-Benz:Mileage -1.016e-01  1.749e-02  -5.809  6.42e-09 ***
## BrandPorsche:Mileage   -8.001e-01  2.683e-02 -29.825   < 2e-16 ***
## BrandVolkswagen:Mileage  1.919e-01  1.856e-02  10.340   < 2e-16 ***
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 26780 on 15074 degrees of freedom
## Multiple R-squared:  0.4558, Adjusted R-squared:  0.4554
## F-statistic:  1148 on 11 and 15074 DF,  p-value: < 2.2e-16

## [1] "Anova Test:"


## Analysis of Variance Table
## 
## Response: Price
##                               Df   Sum Sq  Mean Sq F value Pr(>F)    
## Years_Old                  1 1.7309e+12 1.7309e+12 2413.46   < 2.2e-16 ***
## Brand                      4 3.9045e+12 9.7612e+11 1361.05   < 2.2e-16 ***
## Mileage                     1 2.0788e+12 2.0788e+12 2898.56   < 2.2e-16 ***
## Years_Old:Mileage          1 2.6601e+11 2.6601e+11 370.91   < 2.2e-16 ***
## Brand:Mileage               4 1.0733e+12 2.6834e+11 374.16   < 2.2e-16 ***
## Residuals                  15074 1.0811e+13 7.1718e+08 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

The model with interaction terms had a better R-squared value (0.4554 v. 0.3881), so we began model diagnostics on this model. The first step here was to check high leverage points. We see that we have about 1,102 out of our 15,000 observations marked as high leverage points. That means that nearly 6% of our data is considered to have some predictor have an extreme value which can have large effects on our modeling efforts being very accurate. Out of which only about 189 are considered bad leverage point which is only about 1% of the data, making it seem that our model will be more accurate than we may have initially thought.



```
## [1] "High Leverage Points:"
## [1] 1102
## [1] "Bad Leverage Points:"
## [1] 189
```

Additionally, we see that our data only outputs about 10 outliers which is an extremely small percentage of the data and is to be expected. While we would like to remove those points from our model and it would make sense (since they might be the multi-hundred thousand dollar Porsche's that our predictors can't account for), we unfortunately can not without consulting with an industry professional. Still, with only such a small percentage of observations being outliers we do not expect them to have too big of an effect on the accuracy of our model.

```
## [1] "Number of Outliers:"
```

```
## [1] 10
```

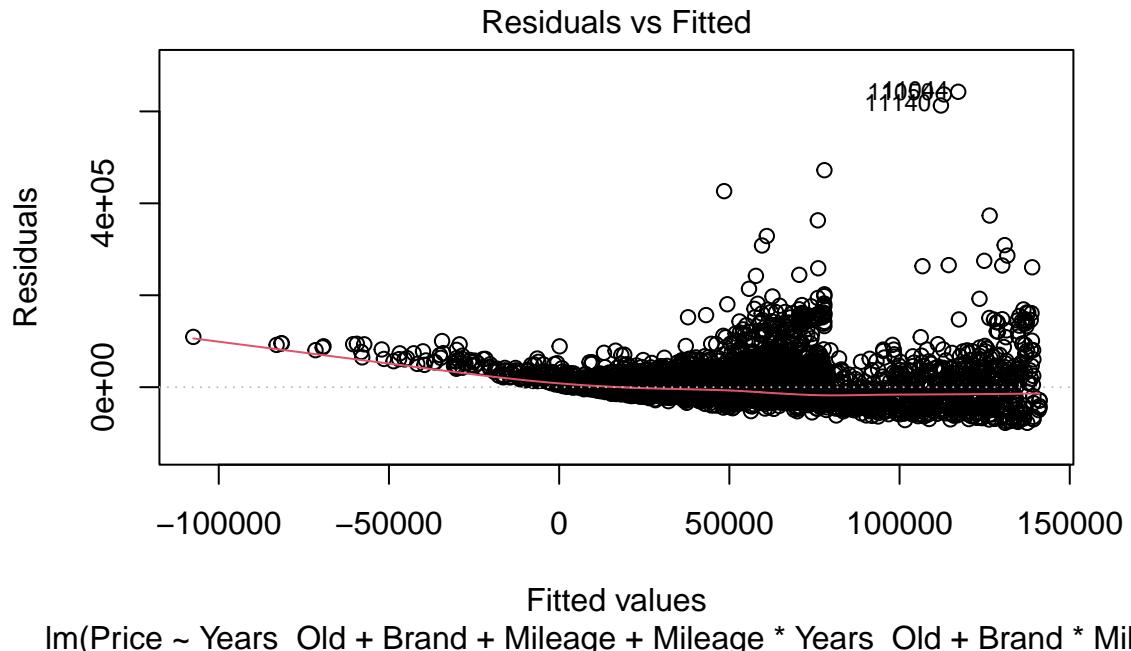
We also see that we have no influential points which is another good sign that none of the outliers appear to have a significant impact on the regression model that we have fitted, and we feel confident moving towards additional steps of the model selection process.

```
## [1] "Influential Points:"
```

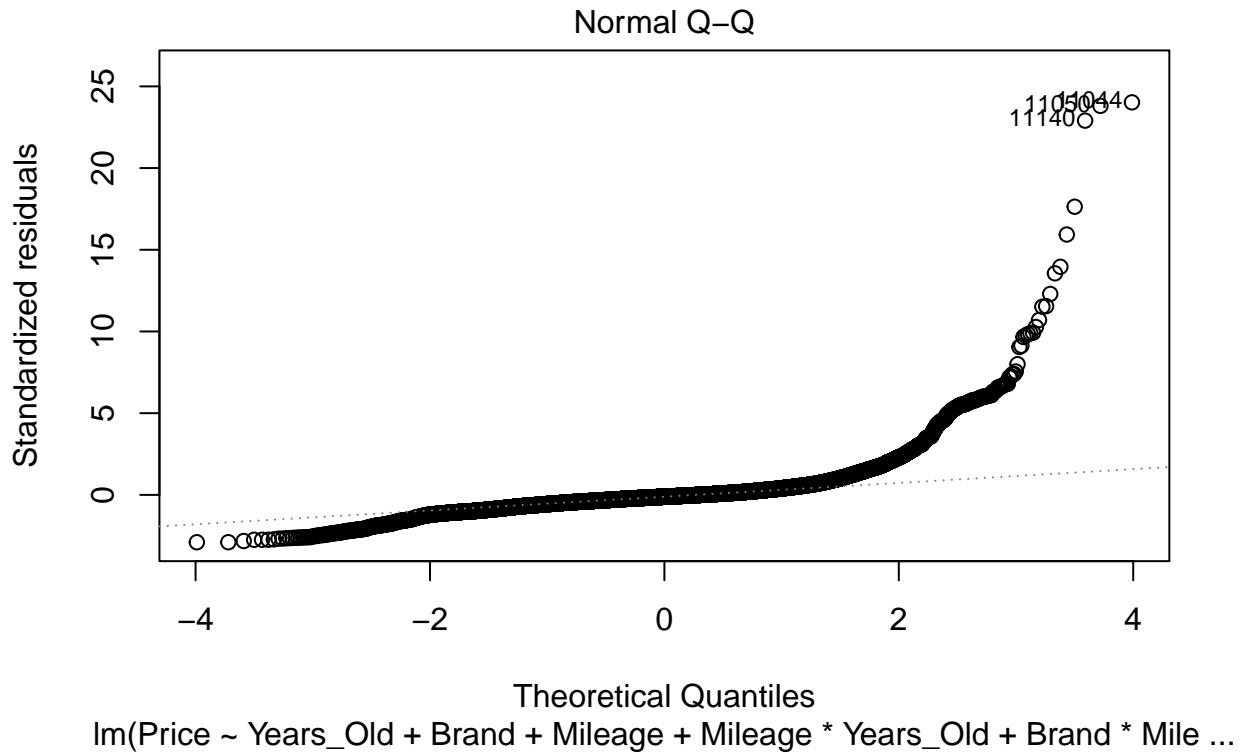
```
## [1] 0
```

We see that the constant variance assumption fails as the variance in error stands to increase and has a fan shape, additionally we fail the BP test. This heteroskedacity must be fixed because otherwise we will obtain p-values smaller than what they are in reality, resulting in us having a model with predictors we do not need, and, thus, losing its simplicity.

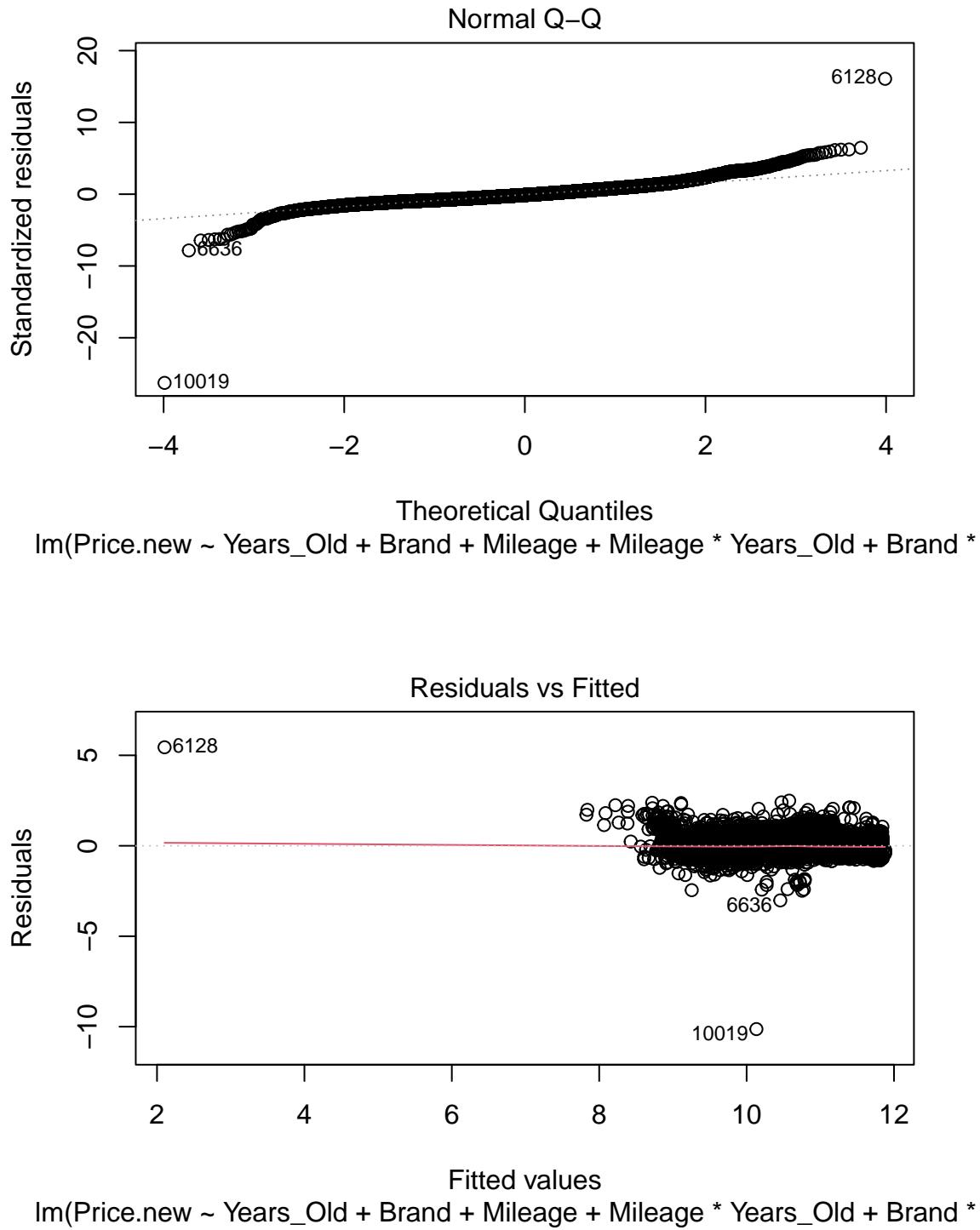
```
##  
## studentized Breusch-Pagan test  
##  
## data: all.model.interactions  
## BP = 480.98, df = 11, p-value < 2.2e-16
```



We also see that our model currently fails the normality assumption. This will again effect the values of our p-values across our variable predictors when we test them and may deviate our results from their true meaning, unless fixed. Therefore, we will try to turn to transformations to fix these problems.



Since the optimal lambda from the box-cox transformation was very close to zero, we decided to do a log transformation on our y-variable (Price). While the new model still fails the tests, we still see great improvement in the graphs as the variance in error appears to be relatively constant and many points on the QQ-plot fit along the linear line. Additionally, we see that our Adjusted R² also improves a significant amount. Therefore, we will continue to use this model with the transformed y through the rest of this study.



The final model that we are choosing is the one outputted after running the BIC criterion. Even though Volkswagen is not significant on its own, its interaction terms are, so we will keep it in our model as well. Therefore, our final model is the log price which is equal to the coefficients listed in the table above. We find that this model has an only slightly worse error for our training and testing dataset. Additionally, this model removed ones that were found insignificant in our original model, so we feel confident with this model.

describing the log price of all of our data across the different brands, ages of cars, and mileage of cars. We will perform similar analysis below on each specific brand dataset individually as well.

```

## [1] "Final Model:"


##
## Call:
## lm(formula = Price.new ~ Years_Old + Brand + Years_Old:Mileage +
##      Brand:Mileage, data = train)
##
## Residuals:
##    Min      1Q   Median      3Q     Max
## -10.1738 -0.2492 -0.0486  0.1973  4.1541
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                1.054e+01  1.259e-02 836.992 < 2e-16 ***
## Years_Old                  -2.424e-02  1.514e-03 -16.012 < 2e-16 ***
## BrandBMW                   5.227e-01  2.190e-02  23.868 < 2e-16 ***
## BrandMercedes-Benz         5.852e-01  2.029e-02  28.841 < 2e-16 ***
## BrandPorsche                1.254e+00  2.966e-02  42.289 < 2e-16 ***
## BrandVolkswagen             5.132e-03  2.188e-02   0.235   0.815
## Years_Old:Mileage          -1.301e-07 1.858e-08  -7.006 2.66e-12 ***
## BrandBMW:Mileage           -9.160e-06 3.120e-07 -29.361 < 2e-16 ***
## BrandMercedes-Benz:Mileage -8.373e-06 2.869e-07 -29.181 < 2e-16 ***
## BrandPorsche:Mileage       -1.184e-05 5.311e-07 -22.290 < 2e-16 ***
## BrandVolkswagen:Mileage    -6.956e-06 2.982e-07 -23.326 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4213 on 7532 degrees of freedom
## Multiple R-squared:  0.6106, Adjusted R-squared:  0.6101
## F-statistic:  1181 on 10 and 7532 DF,  p-value: < 2.2e-16

## [1] "Training Error:"

## [1] 0.4210168

## [1] "Testing Error:"

## [1] 0.4133444

```

Audi Model We actually see that the model fit improves only slightly with the addition of all predictors and interaction terms (Adjusted R² increases from .74 to .78). Again, we significantly fail both the normality and constant variance tests and move to a linear transformation. Again, we see that a box-cox transformation with a log transformation of the y improves these two issues. Additionally, we see that the BIC model has almost the exact same training and testing errors and has a slightly less adjusted R² at .89 versus .9. Therefore, our final model will be the one produced by our BIC criterion above. We will follow a similar process for the rest of our models and shortcut some steps and explanation unless we see some different statistical results that deviate from the pattern we have seen thus far.

```
## [1] "Simple Model:"
```

```

## 
## Call:
## lm(formula = Price ~ Years_Old + Mileage + Class + Type, data = audi.clean)
## 
## Residuals:
##      Min     1Q Median     3Q    Max 
## -49925 -5422 -1436  3584 158583 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 6.695e+04 1.106e+03 60.563 < 2e-16 ***
## Years_Old   -2.652e+03 1.061e+02 -24.993 < 2e-16 *** 
## Mileage     -1.580e-01 9.510e-03 -16.618 < 2e-16 *** 
## ClassE       1.767e+04 1.733e+03 10.194 < 2e-16 *** 
## ClassQ       2.709e+03 5.531e+02  4.898 1.03e-06 *** 
## ClassR       5.167e+04 1.402e+03 36.854 < 2e-16 *** 
## ClassS       1.015e+04 8.423e+02 12.056 < 2e-16 *** 
## ClassT      -7.021e+02 1.915e+03 -0.367  0.714  
## TypePremium  -1.602e+04 1.033e+03 -15.511 < 2e-16 *** 
## TypePremium_Plus -1.487e+04 9.915e+02 -14.999 < 2e-16 *** 
## TypePrestige -1.052e+04 1.110e+03 -9.484 < 2e-16 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 12180 on 2603 degrees of freedom
## Multiple R-squared:  0.7454, Adjusted R-squared:  0.7444 
## F-statistic: 762 on 10 and 2603 DF, p-value: < 2.2e-16

## [1] "Interaction Model:" 

## 
## Call:
## lm(formula = Price ~ Years_Old + Mileage + Class + Type + Years_Old * 
##      Class + Years_Old * Class + Years_Old * Type + Mileage * 
##      Class + Mileage * Type + Class * Type, data = audi.clean)
## 
## Residuals:
##      Min     1Q Median     3Q    Max 
## -59183 -5032 -1255  3026 148510 
## 
## Coefficients: (4 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 6.565e+04 1.876e+03 34.987 < 2e-16 *** 
## Years_Old   -2.516e+03 2.414e+02 -10.420 < 2e-16 *** 
## Mileage     -1.764e-01 2.889e-02 -6.104 1.19e-09 *** 
## ClassE       2.203e+04 4.905e+03  4.492 7.37e-06 *** 
## ClassQ      -6.789e+02 4.592e+03 -0.148  0.88246  
## ClassR       7.430e+04 2.504e+03 29.674 < 2e-16 *** 
## ClassS       2.503e+04 2.753e+03  9.095 < 2e-16 *** 
## ClassT      -9.335e+03 3.963e+03 -2.355  0.01858 *  
## TypePremium -2.063e+04 2.082e+03 -9.912 < 2e-16 *** 
## TypePremium_Plus -1.701e+04 2.080e+03 -8.181 4.38e-16 *** 
## TypePrestige -6.924e+03 2.458e+03 -2.817  0.00489 ** 
## Years_Old:ClassE -3.285e+03 1.780e+03 -1.846  0.06506 .

```

```

## Years_Old:ClassQ      -4.812e+02  2.564e+02  -1.876  0.06071 .
## Years_Old:ClassR      1.084e+03  3.993e+02   2.715  0.00667 **
## Years_Old:ClassS     -7.271e+02  3.535e+02  -2.057  0.03983 *
## Years_Old:ClassT      6.035e+02  4.389e+02   1.375  0.16926
## Years_Old>TypePremium 5.013e+02  3.202e+02   1.566  0.11756
## Years_Old>TypePremium_Plus 1.207e+02  3.180e+02   0.380  0.70431
## Years_Old>TypePrestige -3.138e+02  4.073e+02  -0.770  0.44111
## Mileage:ClassE        -3.958e-02  1.433e-01  -0.276  0.78248
## Mileage:ClassQ        -9.148e-03  2.052e-02  -0.446  0.65583
## Mileage:ClassR       -1.010e+00  6.177e-02 -16.348 < 2e-16 ***
## Mileage:ClassS       -9.323e-02  3.244e-02  -2.874  0.00409 **
## Mileage:ClassT        4.489e-02  7.419e-02   0.605  0.54517
## Mileage>TypePremium   4.807e-02  3.334e-02   1.442  0.14951
## Mileage>TypePremium_Plus 6.277e-02  3.248e-02   1.932  0.05342 .
## Mileage>TypePrestige  4.183e-02  3.832e-02   1.092  0.27510
## ClassE>TypePremium   2.967e+03  5.589e+03   0.531  0.59550
## ClassQ>TypePremium   9.170e+03  4.495e+03   2.040  0.04145 *
## ClassR>TypePremium    NA        NA        NA        NA
## ClassS>TypePremium   -9.052e+03  2.858e+03  -3.167  0.00156 **
## ClassT>TypePremium    NA        NA        NA        NA
## ClassE>TypePremium_Plus 7.652e+03  3.918e+03   1.953  0.05090 .
## ClassQ>TypePremium_Plus 4.639e+03  4.454e+03   1.042  0.29774
## ClassR>TypePremium_Plus  NA        NA        NA        NA
## ClassS>TypePremium_Plus -5.842e+03  2.487e+03  -2.349  0.01888 *
## ClassT>TypePremium_Plus 4.388e+03  4.134e+03   1.062  0.28855
## ClassE>TypePrestige   NA        NA        NA        NA
## ClassQ>TypePrestige   3.568e+03  4.600e+03   0.776  0.43810
## ClassR>TypePrestige   -3.077e+03  4.150e+03  -0.741  0.45857
## ClassS>TypePrestige   -1.230e+04  2.893e+03  -4.252  2.20e-05 ***
## ClassT>TypePrestige   -1.356e+03  1.142e+04  -0.119  0.90550
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11080 on 2576 degrees of freedom
## Multiple R-squared:  0.7914, Adjusted R-squared:  0.7884
## F-statistic: 264.2 on 37 and 2576 DF,  p-value: < 2.2e-16

## [1] "Anova Test:"
```

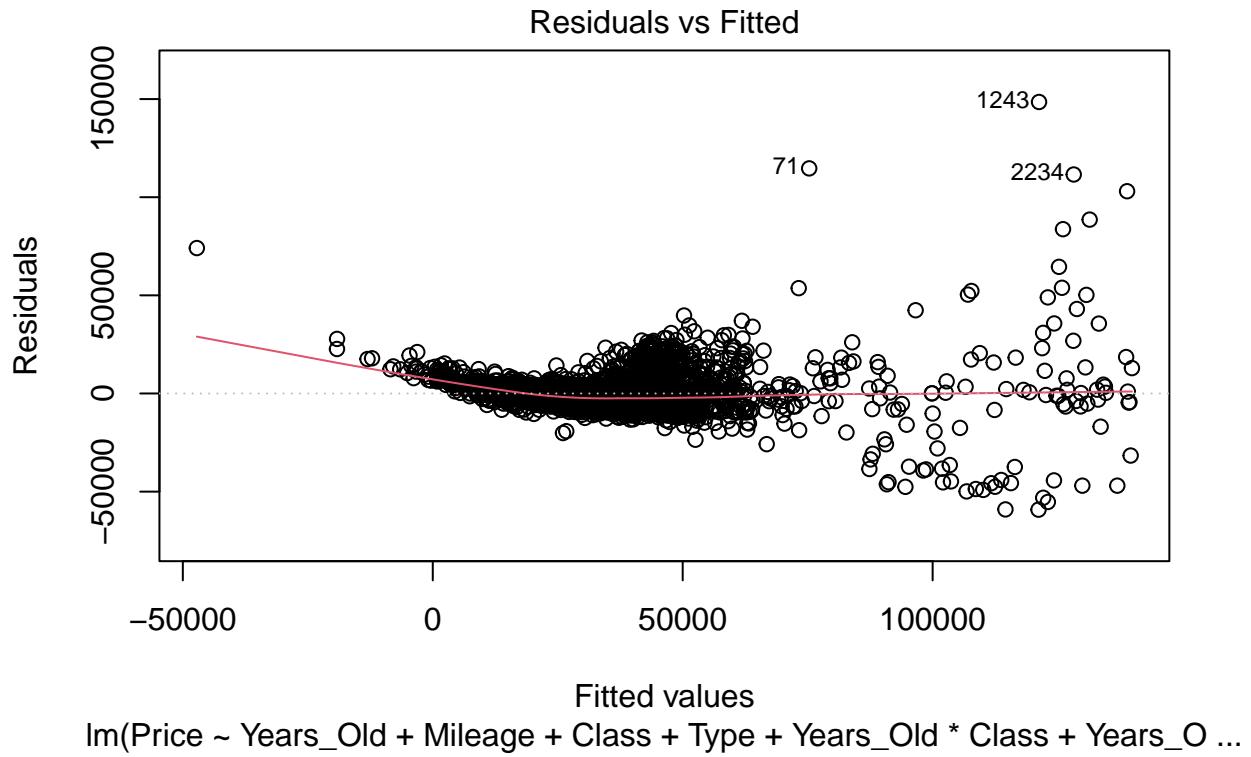
Analysis of Variance Table

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) |
|--------------------|------|------------|------------|-----------|---------------|
| ## Years_Old | 1 | 4.6150e+11 | 4.6150e+11 | 3757.8131 | < 2.2e-16 *** |
| ## Mileage | 1 | 1.5406e+11 | 1.5406e+11 | 1254.4160 | < 2.2e-16 *** |
| ## Class | 5 | 4.7431e+11 | 9.4862e+10 | 772.4246 | < 2.2e-16 *** |
| ## Type | 3 | 4.0762e+10 | 1.3587e+10 | 110.6356 | < 2.2e-16 *** |
| ## Years_Old:Class | 5 | 1.2825e+10 | 2.5650e+09 | 20.8854 | < 2.2e-16 *** |
| ## Years_Old>Type | 3 | 4.0326e+09 | 1.3442e+09 | 10.9453 | 3.846e-07 *** |
| ## Mileage:Class | 5 | 4.6149e+10 | 9.2297e+09 | 75.1538 | < 2.2e-16 *** |
| ## Mileage>Type | 3 | 5.4410e+08 | 1.8137e+08 | 1.4768 | 0.2189 |
| ## Class>Type | 11 | 6.3259e+09 | 5.7509e+08 | 4.6827 | 3.982e-07 *** |
| ## Residuals | 2576 | 3.1636e+11 | 1.2281e+08 | | |
| ## --- | | | | | |

```

## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## [1] "Variance Assumption:"

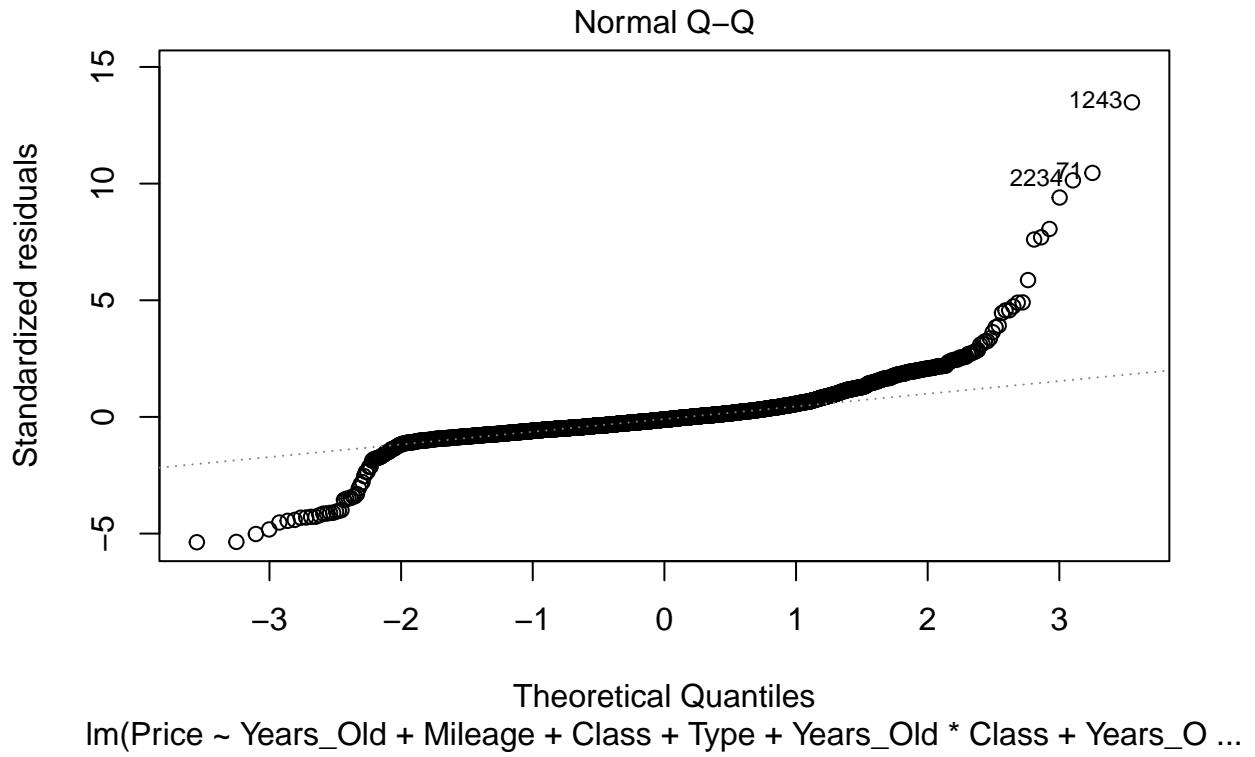
```



```

##
## studentized Breusch-Pagan test
##
## data: audi.model.interactions
## BP = 510.16, df = 37, p-value < 2.2e-16
## [1] "Normality Assumption:"

```



```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data: audi.model.interactions$residuals
## D = 0.58148, p-value < 2.2e-16
## alternative hypothesis: two-sided

## [1] "Transformed Model:"
```

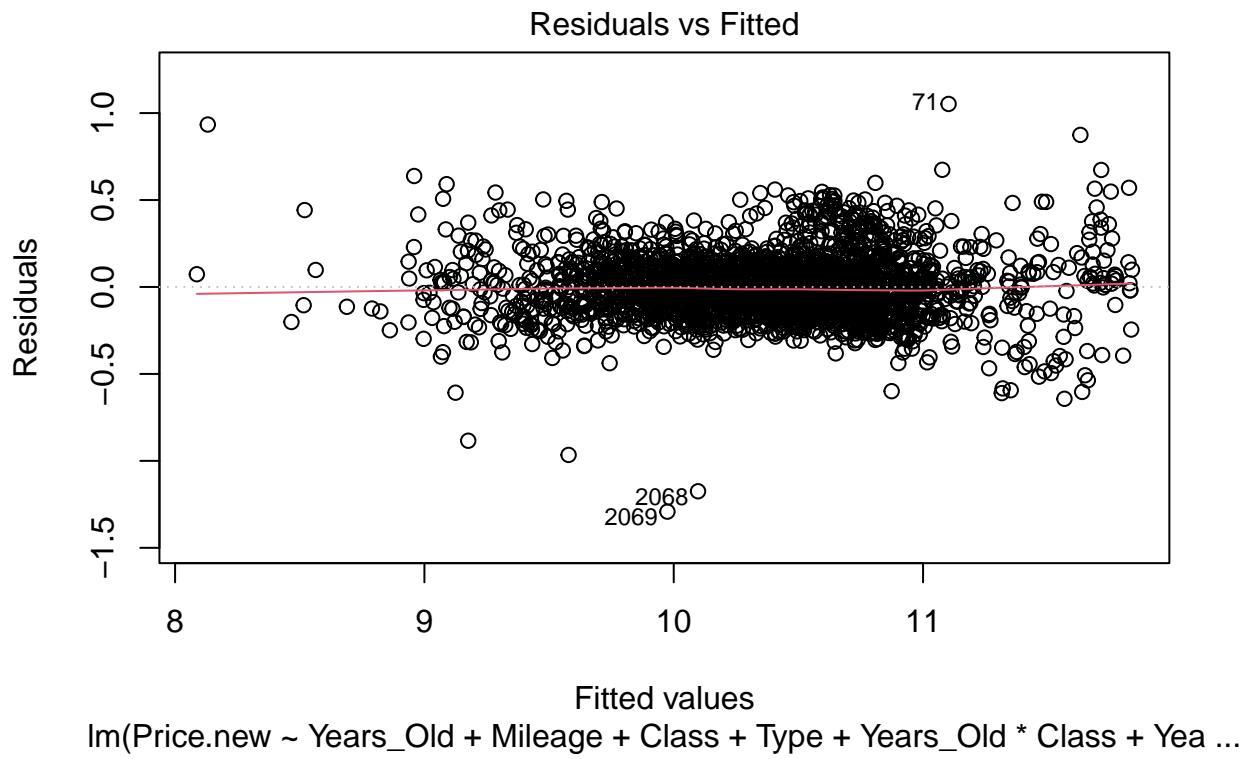
```
##
## Call:
## lm(formula = Price.new ~ Years_Old + Mileage + Class + Type +
##     Years_Old * Class + Years_Old * Class + Years_Old * Type +
##     Mileage * Class + Mileage * Type + Class * Type, data = audi.clean2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.29185 -0.11810 -0.01578  0.09818  1.05242
##
## Coefficients: (4 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.136e+01 3.127e-02 363.413 < 2e-16 ***
## Years_Old   -1.123e-01 4.023e-03 -27.914 < 2e-16 ***
## Mileage    -4.817e-06 4.814e-07 -10.005 < 2e-16 ***
## ClassE      1.457e-01 8.174e-02   1.782 0.074844 .
```

```

## ClassQ          -4.660e-02  7.651e-02  -0.609  0.542576
## ClassR          4.762e-01  4.173e-02  11.414  < 2e-16 ***
## ClassS          2.724e-01  4.587e-02   5.938  3.27e-09 ***
## ClassT          -2.247e-01  6.604e-02  -3.402  0.000679 ***
## TypePremium     -5.571e-01  3.469e-02 -16.060  < 2e-16 ***
## TypePremium_Plus -4.036e-01  3.465e-02 -11.647  < 2e-16 ***
## TypePrestige    -1.974e-01  4.096e-02  -4.818  1.53e-06 ***
## Years_Old:ClassE -1.401e-03  2.966e-02  -0.047  0.962326
## Years_Old:ClassQ -2.043e-02  4.273e-03  -4.780  1.85e-06 ***
## Years_Old:ClassR  1.037e-01  6.654e-03  15.578  < 2e-16 ***
## Years_Old:ClassS  1.634e-02  5.891e-03   2.774  0.005572 **
## Years_Old:ClassT  3.286e-02  7.314e-03   4.492  7.35e-06 ***
## Years_Old:TypePremium 4.249e-02  5.336e-03   7.964  2.47e-15 ***
## Years_Old:TypePremium_Plus 2.894e-02  5.298e-03   5.463  5.14e-08 ***
## Years_Old:TypePrestige  2.154e-02  6.787e-03   3.175  0.001518 **
## Mileage:ClassE      2.624e-06  2.388e-06   1.099  0.272042
## Mileage:ClassQ      9.949e-07  3.420e-07   2.909  0.003656 **
## Mileage:ClassR      -9.487e-06  1.029e-06  -9.217  < 2e-16 ***
## Mileage:ClassS      -8.513e-07  5.406e-07  -1.575  0.115428
## Mileage:ClassT      4.401e-07  1.236e-06   0.356  0.721900
## Mileage:TypePremium -7.097e-07  5.556e-07  -1.277  0.201582
## Mileage:TypePremium_Plus -4.421e-07  5.413e-07  -0.817  0.414207
## Mileage:TypePrestige -2.042e-07  6.386e-07  -0.320  0.749200
## ClassE:TypePremium  2.040e-01  9.313e-02   2.191  0.028545 *
## ClassQ:TypePremium  2.084e-01  7.490e-02   2.782  0.005448 **
## ClassR:TypePremium NA          NA          NA          NA
## ClassS:TypePremium -2.684e-03  4.763e-02  -0.056  0.955062
## ClassT:TypePremium NA          NA          NA          NA
## ClassE:TypePremium_Plus 1.681e-01  6.528e-02   2.576  0.010062 *
## ClassQ:TypePremium_Plus 1.192e-01  7.421e-02   1.606  0.108429
## ClassR:TypePremium_Plus NA          NA          NA          NA
## ClassS:TypePremium_Plus -2.986e-02  4.144e-02  -0.721  0.471206
## ClassT:TypePremium_Plus 4.257e-02  6.889e-02   0.618  0.536613
## ClassE:TypePrestige NA          NA          NA          NA
## ClassQ:TypePrestige  8.987e-02  7.666e-02   1.172  0.241124
## ClassR:TypePrestige  3.199e-02  6.916e-02   0.463  0.643669
## ClassS:TypePrestige -1.743e-01  4.821e-02  -3.616  0.000305 ***
## ClassT:TypePrestige -1.203e-01  1.903e-01  -0.632  0.527426
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1847 on 2576 degrees of freedom
## Multiple R-squared:  0.8936, Adjusted R-squared:  0.8921
## F-statistic:  585 on 37 and 2576 DF,  p-value: < 2.2e-16

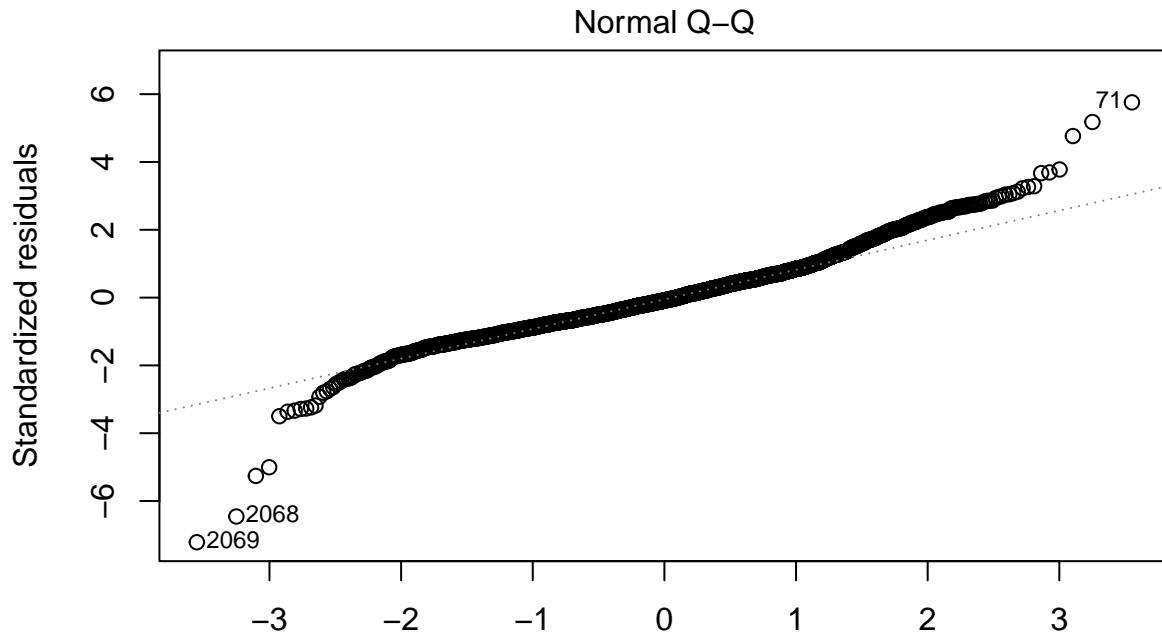
## [1] "Transformed Model: Variance Assumption:"

```



```
##
## studentized Breusch-Pagan test
##
## data: audi.model.transformed
## BP = 322.9, df = 37, p-value < 2.2e-16

## [1] "Transformed Model: Normality Assumption:"
```



```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
##  data: audi.model.transformed$residuals
## D = 0.35849, p-value < 2.2e-16
## alternative hypothesis: two-sided

## Reordering variables and trying again:

## [1] "Final Model:"
```

```
##
## Call:
## lm(formula = Price.new ~ Years_Old + Mileage + Class + Type +
##     Years_Old:Class + Years_Old>Type + Mileage:Class + Mileage>Type +
##     Class>Type, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.94971 -0.10936 -0.01867  0.10098  0.59783
##
## Coefficients: (5 not defined because of singularities)
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.142e+01  4.372e-02 261.165  < 2e-16 ***
## Years_Old          -1.132e-01  6.455e-03 -17.543  < 2e-16 ***
```

```

## Mileage          -5.393e-06 7.562e-07 -7.132 1.65e-12 ***
## ClassE           2.592e-01 1.070e-01  2.421 0.015596 *
## ClassQ          -5.625e-02 1.100e-01 -0.511 0.609208
## ClassR           4.252e-01 6.208e-02  6.849 1.16e-11 ***
## ClassS           2.399e-01 6.198e-02  3.870 0.000114 ***
## ClassT          -2.580e-01 1.078e-01 -2.393 0.016854 *
## TypePremium      -6.199e-01 4.850e-02 -12.781 < 2e-16 ***
## TypePremium_Plus -4.804e-01 4.874e-02 -9.856 < 2e-16 ***
## TypePrestige     -3.223e-01 5.779e-02 -5.576 3.00e-08 ***
## Years_Old:ClassE -2.380e-02 3.825e-02 -0.622 0.533944
## Years_Old:ClassQ -1.526e-02 5.670e-03 -2.692 0.007205 **
## Years_Old:ClassR  1.106e-01 9.183e-03 12.044 < 2e-16 ***
## Years_Old:ClassS  5.487e-03 9.477e-03  0.579 0.562745
## Years_Old:ClassT  3.807e-02 1.091e-02  3.489 0.000500 ***
## Years_Old>TypePremium 4.721e-02 8.024e-03  5.884 5.11e-09 ***
## Years_Old>TypePremium_Plus 3.063e-02 8.116e-03  3.774 0.000168 ***
## Years_Old>TypePrestige  2.555e-02 9.871e-03  2.589 0.009749 **
## Mileage:ClassE     1.522e-06 3.384e-06  0.450 0.653036
## Mileage:ClassQ     3.819e-07 4.674e-07  0.817 0.414039
## Mileage:ClassR    -1.172e-05 1.583e-06 -7.402 2.44e-13 ***
## Mileage:ClassS    -9.358e-07 8.343e-07 -1.122 0.262248
## Mileage:ClassT     1.985e-07 2.031e-06  0.098 0.922191
## Mileage>TypePremium -7.258e-07 8.453e-07 -0.859 0.390739
## Mileage>TypePremium_Plus 1.631e-07 8.488e-07  0.192 0.847682
## Mileage>TypePrestige  8.692e-07 9.827e-07  0.885 0.376592
## ClassE>TypePremium  1.120e-01 1.188e-01  0.943 0.345863
## ClassQ>TypePremium  2.494e-01 1.083e-01  2.303 0.021454 *
## ClassR>TypePremium   NA        NA        NA        NA
## ClassS>TypePremium  6.333e-02 6.467e-02  0.979 0.327668
## ClassT>TypePremium   NA        NA        NA        NA
## ClassE>TypePremium_Plus 1.828e-01 9.477e-02  1.929 0.053908 .
## ClassQ>TypePremium_Plus 1.647e-01 1.076e-01  1.530 0.126182
## ClassR>TypePremium_Plus   NA        NA        NA        NA
## ClassS>TypePremium_Plus  7.609e-02 5.659e-02  1.344 0.179037
## ClassT>TypePremium_Plus  1.532e-01 1.031e-01  1.486 0.137593
## ClassE>TypePrestige    NA        NA        NA        NA
## ClassQ>TypePrestige   1.310e-01 1.110e-01  1.180 0.238202
## ClassR>TypePrestige   3.420e-01 1.350e-01  2.534 0.011402 *
## ClassS>TypePrestige  -5.193e-02 6.487e-02 -0.801 0.423557
## ClassT>TypePrestige   NA        NA        NA        NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1756 on 1270 degrees of freedom
## Multiple R-squared:  0.9013, Adjusted R-squared:  0.8985
## F-statistic: 322.1 on 36 and 1270 DF,  p-value: < 2.2e-16

## [1] "Training Error:"
## [1] 0.1731243
## [1] "Testing Error:"
## [1] 0.1986552

```

BMW Model We see similar model selection trends for BMW, Mercedes, Porsche, and Volkswagen, and, thus, choose the same BIC model with the y transformed by log for all of these brand-specific models as well. Continually, we see this BIC model perform quite well with a high adjusted R², fewer used variables, and a low Root Mean Squared Error. The rest of the models (and their diagnostics) continue and are labeled as such. Overall, we feel confident with using these models in our data analysis.

```

## [1] "Simple Model:"


##      Year          Brand          Make          Price
##  Length:3994    Length:3994    Length:3994    Min.   : 1700
##  Class :character  Class :character  Class :character  1st Qu.: 21996
##  Mode  :character  Mode  :character  Mode  :character  Median  : 32998
##                                         Mean   : 38226
##                                         3rd Qu.: 49256
##                                         Max.   :151674
##      Mileage        Type          Style        Years_Old
##  Min.   :     0  Length:3994    Length:3994    Min.   :-1.000
##  1st Qu.: 25385  Class :character  Class :character  1st Qu.: 2.000
##  Median  : 45367  Mode  :character  Mode  :character  Median  : 4.000
##  Mean   : 52347
##  3rd Qu.: 72871
##  Max.   :9999999  Years_Old:50.000

## [1] "Interaction Model:"

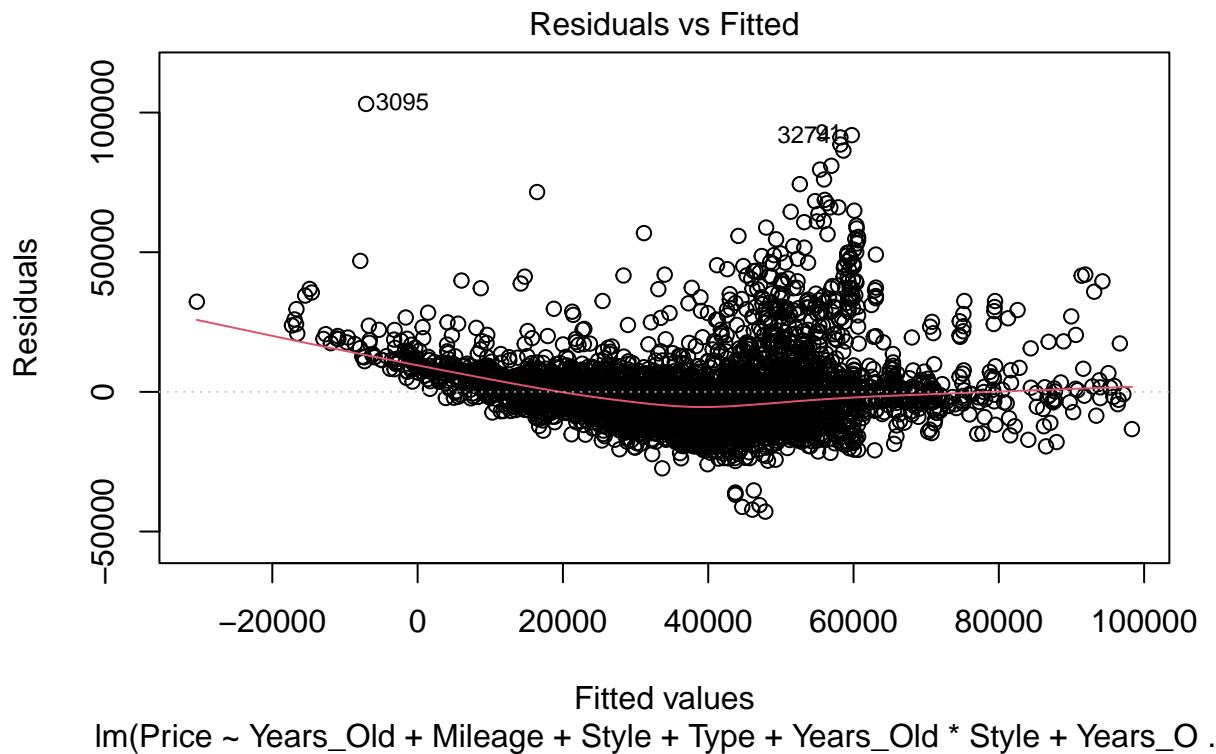

## 
## Call:
## lm(formula = Price ~ Years_Old + Mileage + Style + Type + Years_Old *
##      Style + Years_Old * Style + Years_Old * Type + Mileage *
##      Style + Mileage * Type + Style * Type, data = bmw.clean)
## 
## Residuals:
##      Min      1Q Median      3Q      Max
## -42873 -7549 -1794  3988 103081
## 
## Coefficients: (2 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 5.740e+04 7.871e+02  72.925 < 2e-16 ***
## Years_Old   -7.948e+02 7.038e+01 -11.292 < 2e-16 ***
## Mileage     -3.005e-01 1.184e-02 -25.372 < 2e-16 ***
## StyleX3     1.660e+03 1.614e+03   1.028 0.303781  
## StyleX4     7.500e+03 3.305e+03   2.269 0.023318 *  
## StyleX5     2.220e+04 1.889e+03   11.752 < 2e-16 ***
## StyleX6     3.484e+04 2.972e+03   11.723 < 2e-16 ***
## StyleX7     4.196e+04 2.698e+03   15.551 < 2e-16 ***
## TypesDrive -1.050e+04 3.069e+03  -3.422 0.000629 *** 
## TypexDrive  3.231e+03 9.121e+02   3.543 0.000400 *** 
## Years_Old:StyleX3 -5.046e+02 3.205e+02  -1.574 0.115527  
## Years_Old:StyleX4 -9.377e+02 1.083e+03  -0.866 0.386629  
## Years_Old:StyleX5 -3.190e+03 2.140e+02 -14.907 < 2e-16 ***
## Years_Old:StyleX6 -2.066e+03 6.362e+02  -3.248 0.001174 ** 
## Years_Old:StyleX7 -1.063e+03 1.168e+03  -0.910 0.362860  
## Years_Old:TypesDrive 1.914e+02 4.221e+02   0.453 0.650289

```

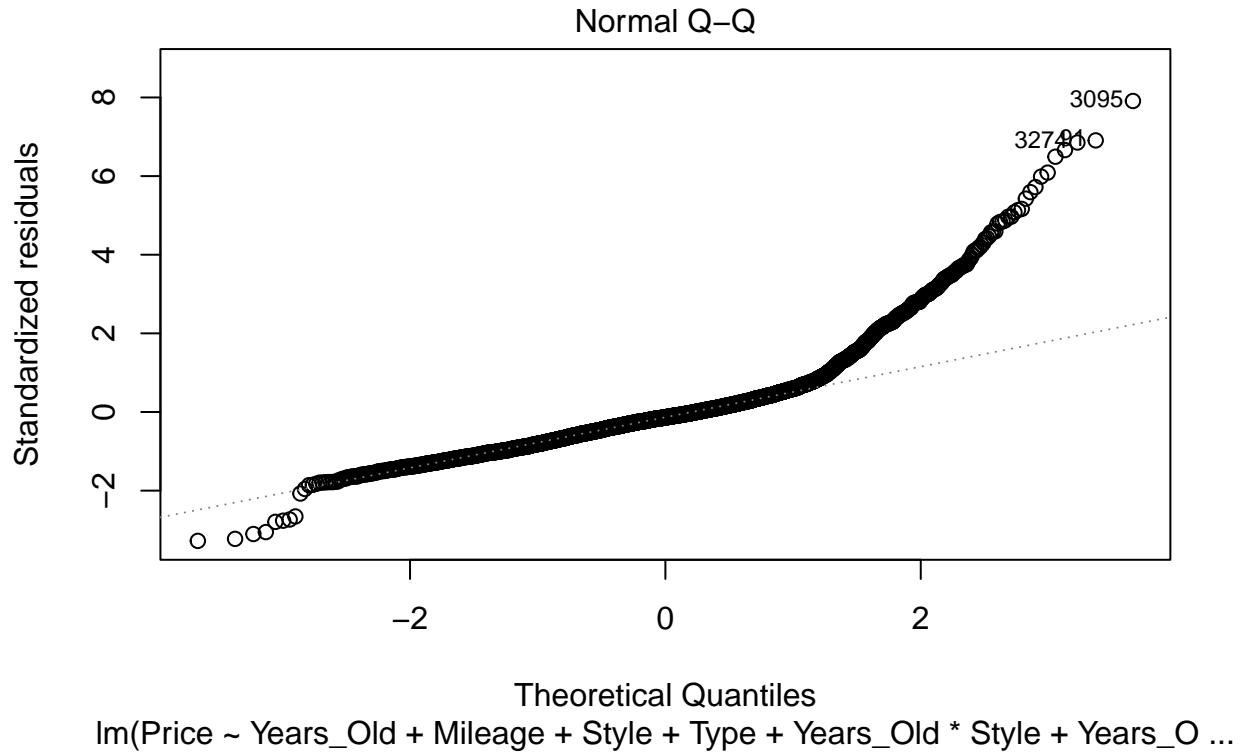
```

## Years_Old:TypexDrive -1.656e+03 1.732e+02 -9.558 < 2e-16 ***
## Mileage:StyleX3      1.202e-01 3.250e-02  3.699 0.000219 ***
## Mileage:StyleX4      1.414e-01 8.287e-02  1.706 0.088061 .
## Mileage:StyleX5      2.623e-01 1.679e-02 15.617 < 2e-16 ***
## Mileage:StyleX6      2.574e-02 6.794e-02  0.379 0.704856
## Mileage:StyleX7     -5.026e-02 7.037e-02 -0.714 0.475177
## Mileage:TypesDrive   3.248e-03 5.453e-02  0.060 0.952500
## Mileage:TypexDrive   4.883e-02 1.716e-02  2.846 0.004449 **
## StyleX3:TypesDrive -4.722e+03 3.772e+03 -1.252 0.210781
## StyleX4:TypesDrive    NA        NA        NA        NA
## StyleX5:TypesDrive  -1.357e+04 3.888e+03 -3.490 0.000488 ***
## StyleX6:TypesDrive  -1.161e+04 7.450e+03 -1.558 0.119281
## StyleX7:TypesDrive    NA        NA        NA        NA
## StyleX3:TypexDrive  -1.125e+04 1.609e+03 -6.991 3.18e-12 ***
## StyleX4:TypexDrive  -1.215e+04 3.578e+03 -3.394 0.000695 ***
## StyleX5:TypexDrive  -1.741e+04 1.715e+03 -10.152 < 2e-16 ***
## StyleX6:TypexDrive  -1.637e+04 3.079e+03 -5.316 1.12e-07 ***
## StyleX7:TypexDrive  -1.864e+04 2.535e+03 -7.351 2.38e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13320 on 3962 degrees of freedom
## Multiple R-squared:  0.6421, Adjusted R-squared:  0.6393
## F-statistic: 229.3 on 31 and 3962 DF,  p-value: < 2.2e-16
##
## [1] "Variance Assumption:"

```



```
## [1] "Normality Assumption:"
```



```
## [1] "Transformed Model:"
```

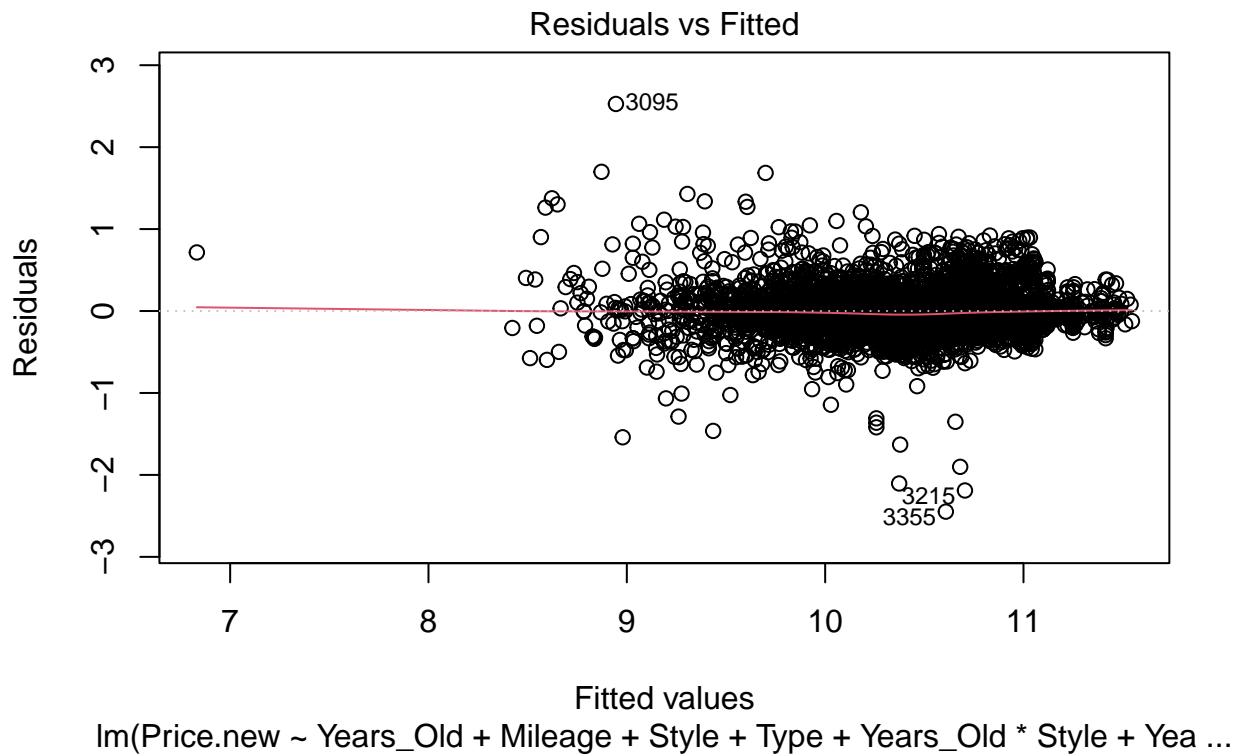
```
## 
## Call:
## lm(formula = Price.new ~ Years_Old + Mileage + Style + Type +
##     Years_Old * Style + Years_Old * Style + Years_Old * Type +
##     Miles + Miles * Style + Miles * Type + Style * Type, data = bmw.clean2)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.44792 -0.16409 -0.02069  0.13173  2.52694 
## 
## Coefficients: (2 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.099e+01 1.759e-02 625.144 < 2e-16 ***
## Years_Old   -2.410e-02 1.573e-03 -15.324 < 2e-16 ***
## Mileage    -1.025e-05 2.647e-07 -38.728 < 2e-16 ***
```

```

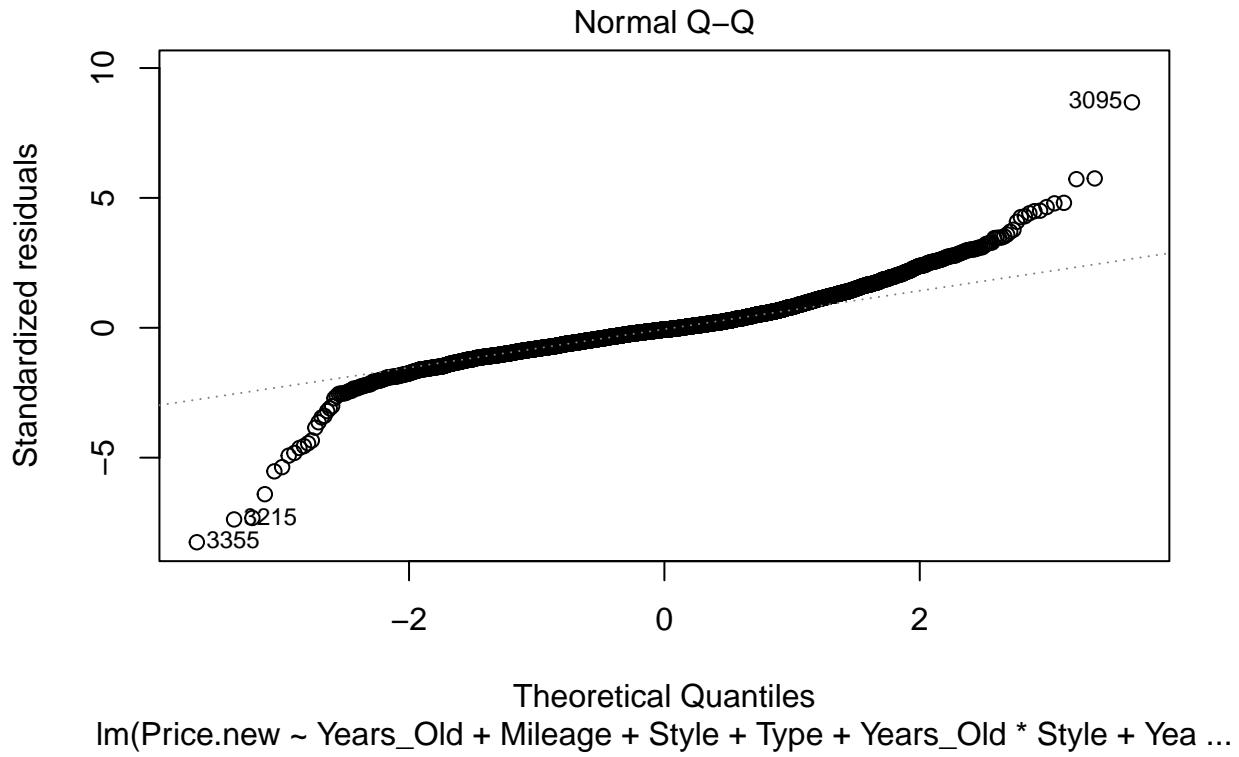
## TypexDrive      6.568e-02 2.038e-02  3.223 0.001280 **
## Years_Old:StyleX3 -5.087e-02 7.162e-03 -7.103 1.44e-12 ***
## Years_Old:StyleX4 -2.120e-02 2.420e-02 -0.876 0.381041
## Years_Old:StyleX5 -1.038e-01 4.781e-03 -21.711 < 2e-16 ***
## Years_Old:StyleX6 -3.504e-02 1.421e-02 -2.465 0.013737 *
## Years_Old:StyleX7  1.821e-02 2.609e-02  0.698 0.485204
## Years_Old:TypesDrive 8.183e-03 9.431e-03  0.868 0.385623
## Years_Old:TypexDrive -3.787e-02 3.870e-03 -9.783 < 2e-16 ***
## Mileage:StyleX3   3.555e-06 7.261e-07  4.895 1.02e-06 ***
## Mileage:StyleX4   4.128e-06 1.851e-06  2.229 0.025851 *
## Mileage:StyleX5   7.975e-06 3.752e-07 21.253 < 2e-16 ***
## Mileage:StyleX6   2.677e-06 1.518e-06  1.763 0.077920 .
## Mileage:StyleX7   3.852e-06 1.572e-06  2.450 0.014331 *
## Mileage:TypesDrive 7.763e-07 1.218e-06  0.637 0.524024
## Mileage:TypexDrive 2.372e-06 3.833e-07  6.188 6.72e-10 ***
## StyleX3:TypesDrive -9.381e-02 8.429e-02 -1.113 0.265784
## StyleX4:TypesDrive      NA       NA       NA       NA
## StyleX5:TypesDrive -1.223e-01 8.686e-02 -1.408 0.159249
## StyleX6:TypesDrive -9.706e-02 1.665e-01 -0.583 0.559837
## StyleX7:TypesDrive      NA       NA       NA       NA
## StyleX3:TypexDrive -2.018e-01 3.594e-02 -5.613 2.12e-08 ***
## StyleX4:TypexDrive -2.700e-01 7.995e-02 -3.377 0.000740 ***
## StyleX5:TypexDrive -2.265e-01 3.832e-02 -5.911 3.69e-09 ***
## StyleX6:TypexDrive -2.645e-01 6.879e-02 -3.845 0.000123 ***
## StyleX7:TypexDrive -2.708e-01 5.665e-02 -4.780 1.81e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2976 on 3962 degrees of freedom
## Multiple R-squared:  0.7568, Adjusted R-squared:  0.7549
## F-statistic: 397.7 on 31 and 3962 DF,  p-value: < 2.2e-16

## [1] "Transformed Model: Variance Assumption:"

```



```
## [1] "Transformed Model: Normality Assumption:"
```



```
## Reordering variables and trying again:
```

```
## [1] "Final Model:"
```

```
## 
## Call:
## lm(formula = Price.new ~ Years_Old + Mileage + Style + Type +
##     Years_Old:Style + Years_Old>Type + Mileage:Style + Mileage>Type +
##     Style>Type, data = train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.52105 -0.16308 -0.02114  0.13921  2.42875 
## 
## Coefficients: (2 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.103e+01 2.576e-02 428.108 < 2e-16 ***
## Years_Old   -2.167e-02 2.239e-03 -9.681 < 2e-16 ***
## Mileage    -1.095e-05 3.992e-07 -27.423 < 2e-16 ***
## StyleX3    4.620e-02 5.595e-02  0.826 0.409077  
## StyleX4    1.874e-01 1.241e-01  1.510 0.131275  
## StyleX5    4.286e-01 5.888e-02  7.278 4.86e-13 ***
## StyleX6    5.759e-01 9.043e-02  6.368 2.38e-10 ***
## StyleX7    4.851e-01 8.863e-02  5.474 4.97e-08 ***
## TypesDrive -3.029e-01 1.010e-01 -2.998 0.002747 **
```

```

## TypexDrive      5.348e-02 3.030e-02  1.765 0.077684 .
## Years_Old:StyleX3 -6.447e-02 9.960e-03 -6.472 1.21e-10 ***
## Years_Old:StyleX4 -2.909e-02 3.762e-02 -0.773 0.439521
## Years_Old:StyleX5 -1.029e-01 8.174e-03 -12.590 < 2e-16 ***
## Years_Old:StyleX6 -3.111e-02 1.891e-02 -1.645 0.100159
## Years_Old:StyleX7  2.951e-02 4.056e-02  0.728 0.466947
## Years_Old:TypesDrive 8.985e-03 1.461e-02  0.615 0.538767
## Years_Old:TypexDrive -3.890e-02 6.064e-03 -6.415 1.76e-10 ***
## Mileage:StyleX3   5.264e-06 1.024e-06  5.141 3.01e-07 ***
## Mileage:StyleX4   4.578e-06 2.920e-06  1.568 0.117028
## Mileage:StyleX5   7.660e-06 8.562e-07  8.946 < 2e-16 ***
## Mileage:StyleX6   2.125e-06 2.103e-06  1.010 0.312382
## Mileage:StyleX7   4.311e-06 2.293e-06  1.880 0.060277 .
## Mileage:TypesDrive 1.693e-06 1.897e-06  0.892 0.372374
## Mileage:TypexDrive 2.854e-06 6.511e-07  4.384 1.23e-05 ***
## StyleX3:TypesDrive -2.004e-02 1.211e-01 -0.165 0.868590
## StyleX4:TypesDrive       NA        NA        NA        NA
## StyleX5:TypesDrive -4.735e-02 1.235e-01 -0.383 0.701398
## StyleX6:TypesDrive  2.461e-02 3.345e-01  0.074 0.941357
## StyleX7:TypesDrive       NA        NA        NA        NA
## StyleX3:TypexDrive -1.737e-01 5.485e-02 -3.166 0.001567 **
## StyleX4:TypexDrive -2.528e-01 1.179e-01 -2.144 0.032136 *
## StyleX5:TypexDrive -2.457e-01 5.543e-02 -4.432 9.86e-06 ***
## StyleX6:TypexDrive -2.547e-01 9.915e-02 -2.569 0.010271 *
## StyleX7:TypexDrive -2.687e-01 7.977e-02 -3.368 0.000771 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3083 on 1965 degrees of freedom
## Multiple R-squared:  0.7515, Adjusted R-squared:  0.7476
## F-statistic: 191.7 on 31 and 1965 DF,  p-value: < 2.2e-16

## [1] "Training Error:"
## [1] 0.3058221

## [1] "Testing Error:"
## [1] 0.2894066

```

Mercedes Benz Model

```

## [1] "Simple Model:"
```

| | Year | Brand | Make | Price |
|----|------------------|------------------|------------------|----------------|
| ## | Length:3733 | Length:3733 | Length:3733 | Min. : 1 |
| ## | Class :character | Class :character | Class :character | 1st Qu.: 23990 |
| ## | Mode :character | Mode :character | Mode :character | Median : 37991 |
| ## | | | | Mean : 48070 |
| ## | | | | 3rd Qu.: 54545 |
| ## | | | | Max. :550000 |
| ## | Mileage | Type | Drive | Style |

```

## Min. : 0 Length:3733      Length:3733      Length:3733
## 1st Qu.: 13649 Class :character  Class :character  Class :character
## Median : 40632 Mode  :character  Mode  :character  Mode  :character
## Mean   : 48675
## 3rd Qu.: 74664
## Max.   :322971
## Years_Old
## Min. :-1.000
## 1st Qu.: 1.000
## Median : 4.000
## Mean   : 5.692
## 3rd Qu.: 8.000
## Max.   :63.000

## [1] "Interaction Model:"


##
## Call:
## lm(formula = Price ~ Years_Old + Mileage + Style + Type + Drive +
##     Years_Old * Style + Years_Old * Style + Years_Old * Type +
##     Mileage * Style + Mileage * Type + Style * Type + Drive *
##     Years_Old + Drive * Mileage + Drive * Style + Drive * Type,
##     data = mercedes.clean)
##
## Residuals:
##    Min      1Q  Median      3Q      Max
## -109269 -9739 -3831  5529 429990
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)               5.636e+04  4.213e+03 13.379 < 2e-16 ***
## Years_Old                  1.120e+02  6.599e+02  0.170 0.865205
## Mileage                   -2.666e-01  5.591e-02 -4.769 1.92e-06 ***
## StyleE class                2.408e+04  7.693e+03  3.130 0.001760 **
## StyleG class                5.287e+04  4.941e+03 10.701 < 2e-16 ***
## StyleOther                  -1.191e+03  4.142e+03 -0.288 0.773730
## StyleS class                 7.816e+04  6.666e+03 11.725 < 2e-16 ***
## TypeOther                   1.234e+04  3.521e+03  3.504 0.000463 ***
## DriveNot Base              -4.681e+03  4.472e+03 -1.047 0.295204
## Years_Old:StyleE class     -3.890e+02  5.758e+02 -0.676 0.499349
## Years_Old:StyleG class     -6.245e+03  8.904e+02 -7.013 2.76e-12 ***
## Years_Old:StyleOther       -4.414e+02  2.254e+02 -1.958 0.050268 .
## Years_Old:StyleS class     -1.500e+03  3.694e+02 -4.060 5.01e-05 ***
## Years_Old>TypeOther        3.574e+03  3.267e+02 10.941 < 2e-16 ***
## Mileage:StyleE class      -1.251e-01  6.134e-02 -2.040 0.041424 *
## Mileage:StyleG class      -3.706e-01  8.548e-02 -4.336 1.49e-05 ***
## Mileage:StyleOther         -7.220e-02  3.701e-02 -1.951 0.051131 .
## Mileage:StyleS class      -2.903e-01  5.330e-02 -5.447 5.45e-08 ***
## Mileage>TypeOther          -1.368e-01  3.362e-02 -4.071 4.78e-05 ***
## StyleE class>TypeOther    -7.994e+03  3.648e+03 -2.192 0.028465 *
## StyleG class>TypeOther    4.470e+04  4.038e+03 11.070 < 2e-16 ***
## StyleOther>TypeOther       3.565e+02  2.955e+03  0.121 0.903972
## StyleS class>TypeOther    -5.314e+02  3.977e+03 -0.134 0.893732
## Years_Old:DriveNot Base  -3.823e+03  6.333e+02 -6.037 1.72e-09 ***

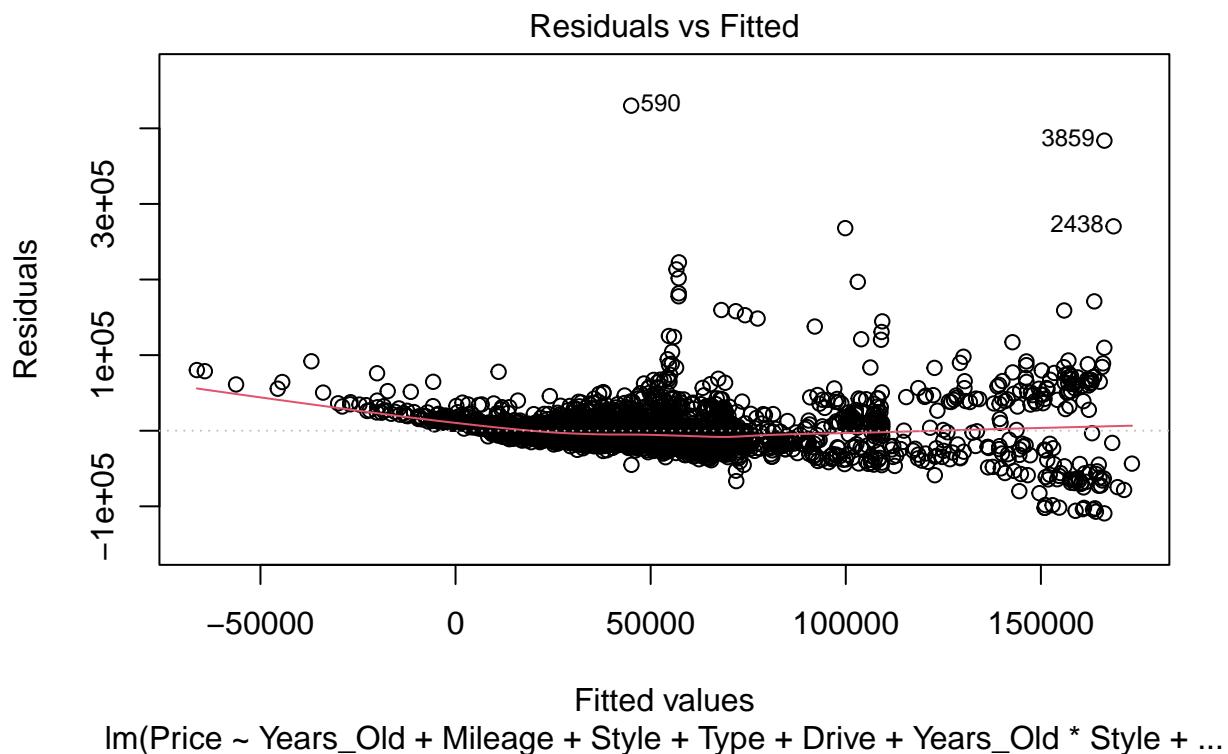
```

```

## Mileage:DriveNot Base      2.265e-01  4.673e-02   4.846 1.31e-06 ***
## StyleE class:DriveNot Base -3.337e+03  8.019e+03  -0.416 0.677351
## StyleG class:DriveNot Base  3.822e+04  5.353e+03   7.140 1.12e-12 ***
## StyleOther:DriveNot Base   1.874e+04  4.457e+03   4.203 2.69e-05 ***
## StyleS class:DriveNot Base -2.048e+04  6.940e+03  -2.951 0.003182 **
## TypeOther:DriveNot Base   -2.468e+04  2.748e+03  -8.982 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26040 on 3703 degrees of freedom
## Multiple R-squared:  0.6094, Adjusted R-squared:  0.6064
## F-statistic: 199.2 on 29 and 3703 DF,  p-value: < 2.2e-16

## [1] "Variance Assumption:"

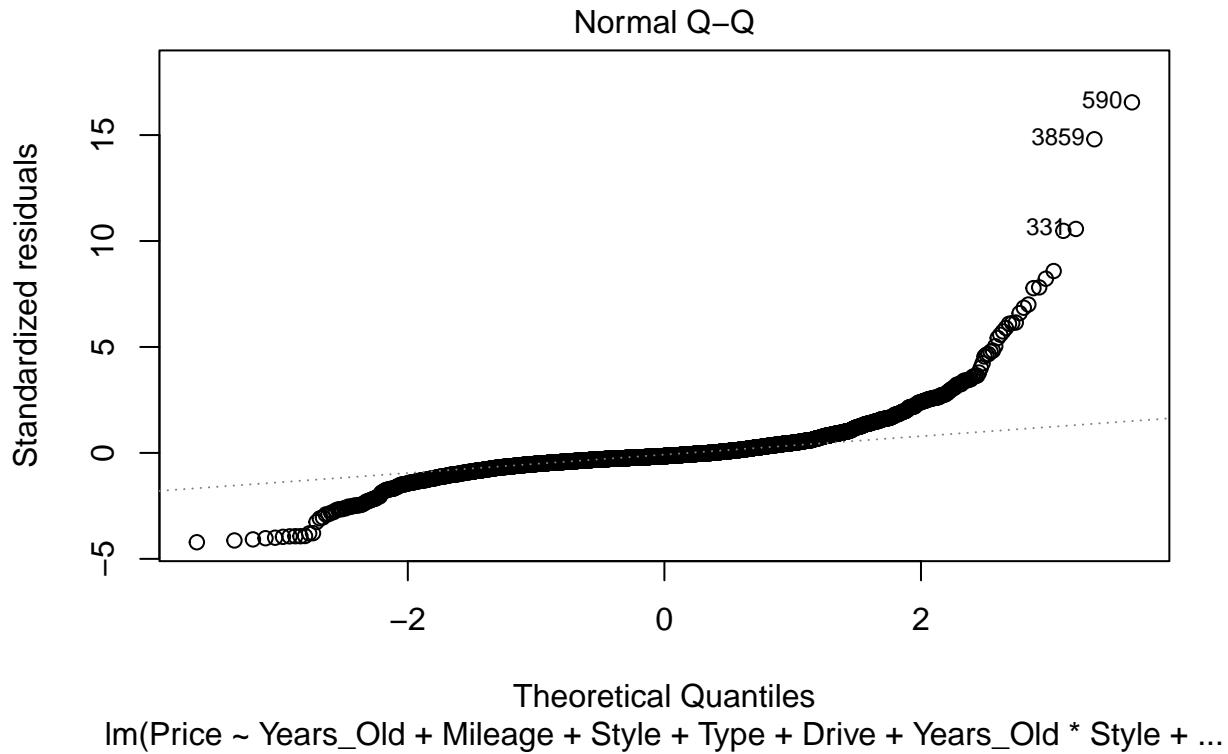
```



```

## [1] "Normality Assumption:"

```



```

## [1] "Transformed Model:"
```

```

##
## Call:
## lm(formula = Price.new ~ Years_Old + Mileage + Style + Type +
##     Drive + Years_Old * Style + Years_Old * Style + Years_Old *
##     Type + Mileage * Style + Mileage * Type + Style * Type +
##     Drive * Years_Old + Drive * Mileage + Drive * Style + Drive *
##     Type, data = mercedes.clean2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4377  -0.1736  -0.0533   0.1561   2.5621
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.117e+01  5.932e-02 188.336 < 2e-16 ***
## Years_Old                -5.103e-02  9.292e-03  -5.492 4.25e-08 ***
## Mileage                  -6.631e-06  7.872e-07  -8.423 < 2e-16 ***
## StyleE class              3.041e-01  1.083e-01   2.807 0.005027 **
## StyleG class              2.585e-01  6.957e-02   3.715 0.000206 ***
## StyleOther                -2.583e-01  5.833e-02  -4.429 9.76e-06 ***
## StyleS class              8.524e-01  9.387e-02   9.081 < 2e-16 ***
## TypeOther                 -1.947e-02  4.958e-02  -0.393 0.694596
## DriveNot Base             -2.000e-01  6.297e-02  -3.177 0.001502 **

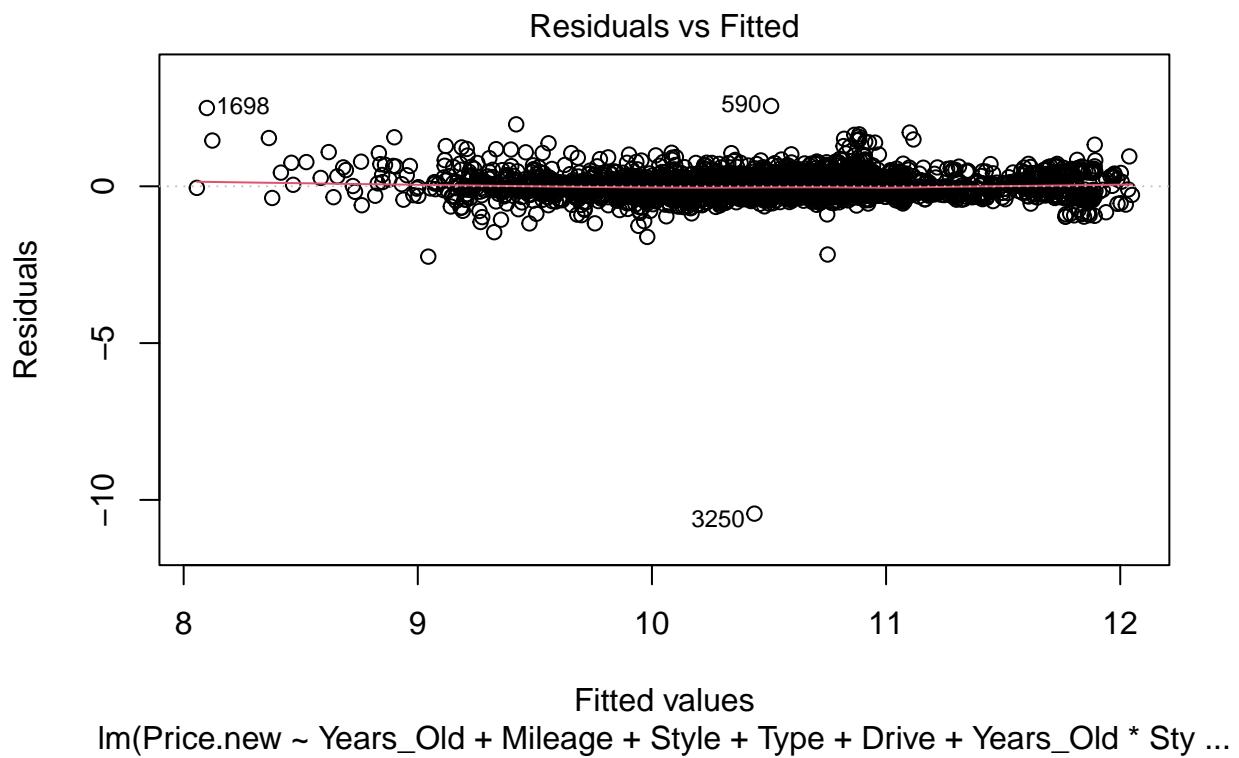
```

```

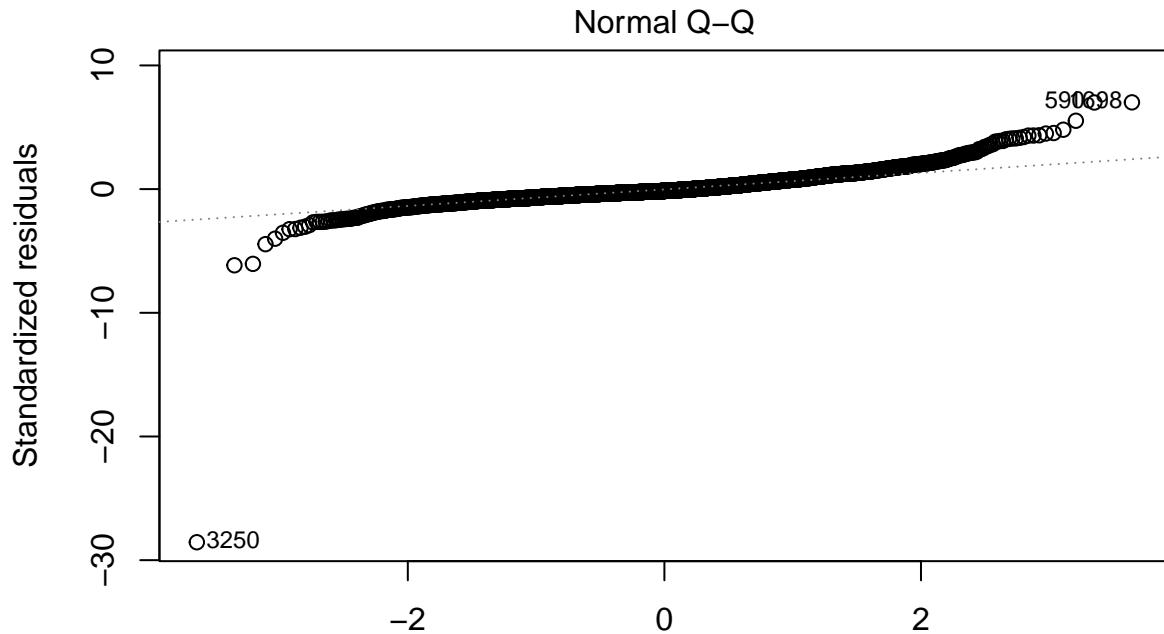
## Years_Old:StyleE class      -3.542e-02 8.108e-03 -4.368 1.29e-05 ***
## Years_Old:StyleG class     -1.859e-02 1.254e-02 -1.483 0.138292
## Years_Old:StyleOther       -4.770e-03 3.174e-03 -1.503 0.132948
## Years_Old:StyleS class    -3.244e-02 5.202e-03 -6.237 4.98e-10 ***
## Years_Old>TypeOther        7.208e-02 4.600e-03 15.669 < 2e-16 ***
## Mileage:StyleE class      1.118e-06 8.637e-07 1.294 0.195704
## Mileage:StyleG class      1.404e-06 1.204e-06 1.167 0.243460
## Mileage:StyleOther         2.069e-06 5.211e-07 3.969 7.34e-05 ***
## Mileage:StyleS class      6.386e-07 7.505e-07 0.851 0.394881
## Mileage>TypeOther          -2.513e-06 4.734e-07 -5.307 1.18e-07 ***
## StyleE class>TypeOther     -1.352e-01 5.136e-02 -2.633 0.008500 **
## StyleG class>TypeOther     4.802e-01 5.686e-02 8.445 < 2e-16 ***
## StyleOther>TypeOther        5.079e-02 4.161e-02 1.221 0.222329
## StyleS class>TypeOther     -2.936e-02 5.601e-02 -0.524 0.600178
## Years_Old:DriveNot Base   -3.575e-02 8.917e-03 -4.009 6.23e-05 ***
## Mileage:DriveNot Base      5.445e-07 6.580e-07 0.827 0.408052
## StyleE class:DriveNot Base 9.542e-02 1.129e-01 0.845 0.398163
## StyleG class:DriveNot Base 6.067e-01 7.538e-02 8.049 1.11e-15 ***
## StyleOther:DriveNot Base   3.648e-01 6.277e-02 5.813 6.67e-09 ***
## StyleS class:DriveNot Base -9.375e-02 9.773e-02 -0.959 0.337473
## TypeOther:DriveNot Base    -2.242e-01 3.869e-02 -5.795 7.40e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3667 on 3703 degrees of freedom
## Multiple R-squared:  0.7364, Adjusted R-squared:  0.7344
## F-statistic: 356.8 on 29 and 3703 DF,  p-value: < 2.2e-16

## [1] "Transformed Model: Variance Assumption:"

```



```
## [1] "Transformed Model: Normality Assumption:"
```



Theoretical Quantiles
lm(Price.new ~ Years_Old + Mileage + Style + Type + Drive + Years_Old * Sty ...

```

## [1] "Final Model:"


##
## Call:
## lm(formula = Price.new ~ Years_Old + Mileage + Style + Type +
##     Drive + Years_Old:Style + Years_Old:Type + Mileage:Style +
##     Mileage:Type + Style:Type + Years_Old:Drive + Mileage:Drive +
##     Style:Drive + Type:Drive, data = train)
##
## Residuals:
##      Min        1Q    Median        3Q       Max
## -10.3516  -0.1728  -0.0504   0.1602   2.4851
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)               1.120e+01  8.148e-02 137.447 < 2e-16 ***
## Years_Old                -4.765e-02  1.432e-02  -3.327 0.000895 ***
## Mileage                  -7.748e-06  1.201e-06  -6.451 1.42e-10 ***
## StyleE class              2.795e-01  1.850e-01   1.511 0.131011  
## StyleG class              1.932e-01  9.969e-02   1.938 0.052735 .  
## StyleOther                -3.053e-01  8.005e-02  -3.814 0.000141 ***
## StyleS class              8.775e-01  1.464e-01   5.995 2.44e-09 ***
## TypeOther                 -3.503e-02  7.639e-02  -0.459 0.646597  
## DriveNot Base             -1.853e-01  8.920e-02  -2.078 0.037886 *  
## Years Old:StyleE class   -4.184e-02  1.285e-02  -3.256 0.001149 ** 
## 
```

```

## Years_Old:StyleG class      -1.159e-03  2.082e-02 -0.056  0.955587
## Years_Old:StyleOther       3.739e-03  5.385e-03  0.694  0.487514
## Years_Old:StyleS class     -2.902e-02  8.690e-03 -3.340  0.000854 ***
## Years_Old>TypeOther        6.630e-02  7.023e-03  9.441  < 2e-16 ***
## Mileage:StyleE class       2.580e-06  1.327e-06  1.944  0.052076 .
## Mileage:StyleG class       2.317e-06  1.897e-06  1.221  0.222254
## Mileage:StyleOther         2.865e-06  8.220e-07  3.485  0.000503 ***
## Mileage:StyleS class       -2.638e-07  1.301e-06 -0.203  0.839421
## Mileage>TypeOther          -2.815e-06  7.235e-07 -3.891  0.000103 ***
## StyleE class>TypeOther     -1.298e-01  7.771e-02 -1.670  0.095019 .
## StyleG class>TypeOther     4.343e-01  8.740e-02  4.969  7.35e-07 ***
## StyleOther>TypeOther        1.223e-01  6.437e-02  1.901  0.057511 .
## StyleS class>TypeOther     -5.085e-02  9.363e-02 -0.543  0.587096
## Years_Old:DriveNot Base   -3.879e-02  1.337e-02 -2.902  0.003757 **
## Mileage:DriveNot Base     8.383e-07  9.525e-07  0.880  0.378894
## StyleE class:DriveNot Base 6.347e-02  1.930e-01  0.329  0.742295
## StyleG class:DriveNot Base 5.504e-01  1.124e-01  4.896  1.07e-06 ***
## StyleOther:DriveNot Base   2.989e-01  8.982e-02  3.328  0.000892 ***
## StyleS class:DriveNot Base -1.283e-01  1.550e-01 -0.828  0.407764
## TypeOther:DriveNot Base   -1.663e-01  6.067e-02 -2.740  0.006198 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4095 on 1836 degrees of freedom
## Multiple R-squared:  0.6989, Adjusted R-squared:  0.6941
## F-statistic: 146.9 on 29 and 1836 DF,  p-value: < 2.2e-16

## [1] "Training Error:"
## [1] 0.4062225

## [1] "Testing Error:"
## [1] 0.3292002

```

Porsche Model

```

## [1] "Simple Model:"


##      Year           Brand          Make          Price
## Length:1234    Length:1234    Length:1234    Min.   : 6991
## Class :character Class :character Class :character 1st Qu.: 38996
## Mode  :character Mode  :character Mode  :character Median  : 63731
##                                         Mean   : 83985
##                                         3rd Qu.:112885
##                                         Max.   :759900
##      Mileage          Style          Years_Old
## Min.   :    0  Length:1234    Min.   :-1.00
## 1st Qu.: 12850  Class :character 1st Qu.: 2.00
## Median  : 31309  Mode  :character Median  : 5.00
## Mean   : 39618                           Mean   : 6.88
## 3rd Qu.: 60088                           3rd Qu.: 8.00
## Max.   :287822                          Max.   :54.00

```

```

## [1] "Interaction Model:"
```

```

##
```

```

## Call:
## lm(formula = Price ~ Years_Old + Mileage + Style + Years_Old *
##      Style + Years_Old * Style + Mileage * Style, data = porsche.clean)
##
```

```

## Residuals:
##    Min     1Q Median     3Q    Max
## -156448 -15963 -4545    5885 570502
##
```

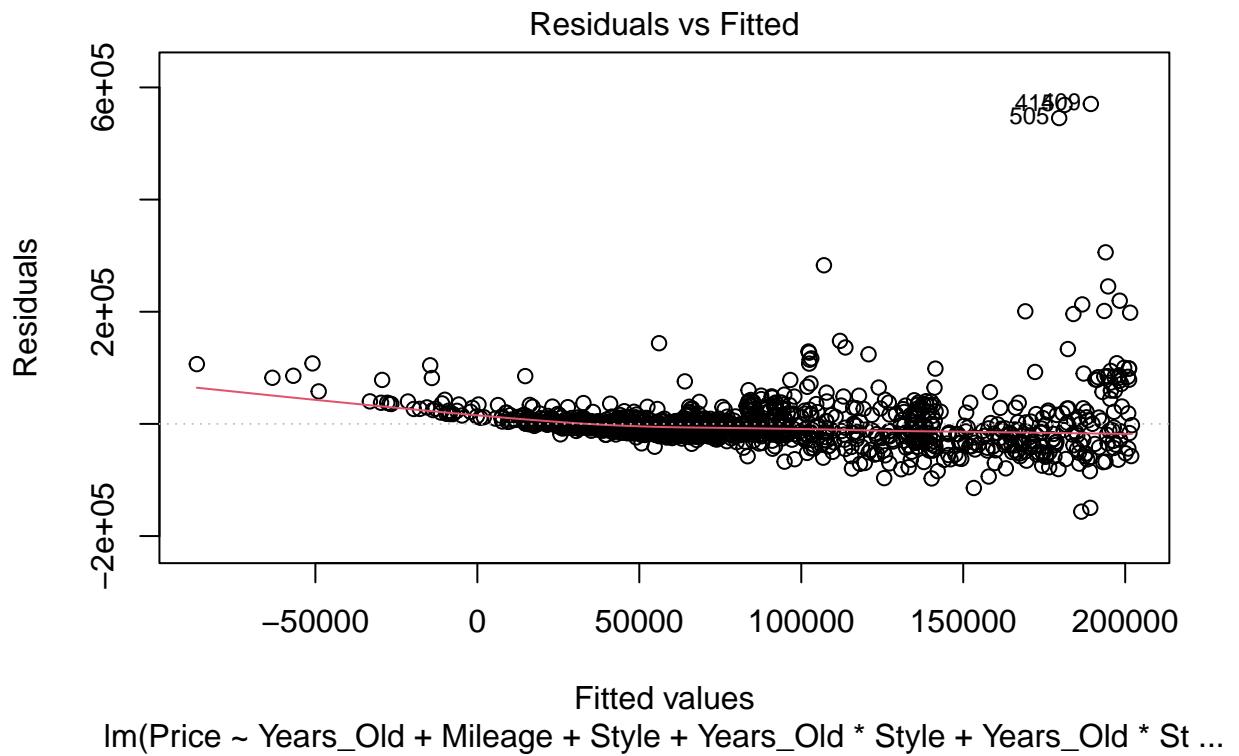
```

## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
```

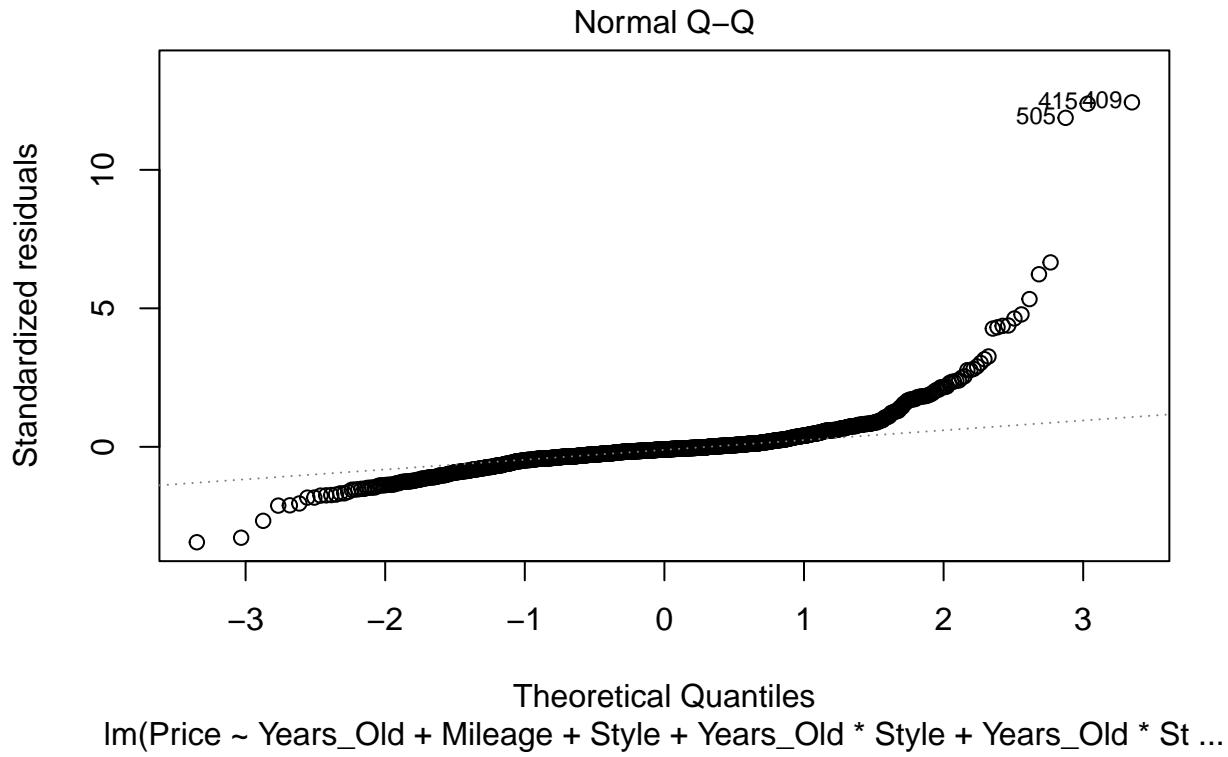
| | Estimate | Std. Error | t value | Pr(> t) |
|-----------------------------|------------|----------------------------|----------|--------------|
| ## (Intercept) | 2.021e+05 | 3.896e+03 | 51.877 | < 2e-16 *** |
| ## Years_Old | -7.464e+02 | 3.869e+02 | -1.929 | 0.05393 . |
| ## Mileage | -2.044e+00 | 1.289e-01 | -15.858 | < 2e-16 *** |
| ## StyleBoxster | -1.153e+05 | 9.733e+03 | -11.846 | < 2e-16 *** |
| ## StyleCayenne | -9.899e+04 | 6.001e+03 | -16.496 | < 2e-16 *** |
| ## StyleCayman | -7.203e+04 | 1.277e+04 | -5.639 | 2.12e-08 *** |
| ## StyleMacan | -1.315e+05 | 7.310e+03 | -17.984 | < 2e-16 *** |
| ## StyleOther | -6.063e+04 | 7.361e+03 | -8.237 | 4.53e-16 *** |
| ## StylePanamera | -7.544e+04 | 8.464e+03 | -8.913 | < 2e-16 *** |
| ## Years_Old:StyleBoxster | -1.933e+03 | 9.280e+02 | -2.083 | 0.03748 * |
| ## Years_Old:StyleCayenne | -3.831e+03 | 1.270e+03 | -3.016 | 0.00261 ** |
| ## Years_Old:StyleCayman | -3.654e+03 | 1.683e+03 | -2.171 | 0.03016 * |
| ## Years_Old:StyleMacan | -2.854e+03 | 2.490e+03 | -1.146 | 0.25191 |
| ## Years_Old:StyleOther | -1.567e+03 | 5.641e+02 | -2.778 | 0.00555 ** |
| ## Years_Old:StylePanamera | -5.463e+03 | 2.028e+03 | -2.694 | 0.00716 ** |
| ## Mileage:StyleBoxster | 1.745e+00 | 2.408e-01 | 7.245 | 7.68e-13 *** |
| ## Mileage:StyleCayenne | 1.684e+00 | 1.783e-01 | 9.447 | < 2e-16 *** |
| ## Mileage:StyleCayman | 1.256e+00 | 3.807e-01 | 3.298 | 0.00100 ** |
| ## Mileage:StyleMacan | 1.848e+00 | 2.239e-01 | 8.252 | 4.01e-16 *** |
| ## Mileage:StyleOther | 1.832e+00 | 2.010e-01 | 9.114 | < 2e-16 *** |
| ## Mileage:StylePanamera | 1.601e+00 | 2.354e-01 | 6.800 | 1.64e-11 *** |
| ## --- | | | | |
| ## Signif. codes: | 0 '***' | 0.001 '**' | 0.01 '*' | 0.05 '.' |
| ## | 0.1 , , 1 | | | |
| ## | | | | |
| ## Residual standard error: | 46100 | on 1213 degrees of freedom | | |
| ## Multiple R-squared: | 0.577 | Adjusted R-squared: | 0.57 | |
| ## F-statistic: | 82.73 | on 20 and 1213 DF, | p-value: | < 2.2e-16 |

```

## [1] "Variance Assumption:"
```



```
## [1] "Normality Assumption:"
```



```

## [1] "Transformed Model:"
```

```

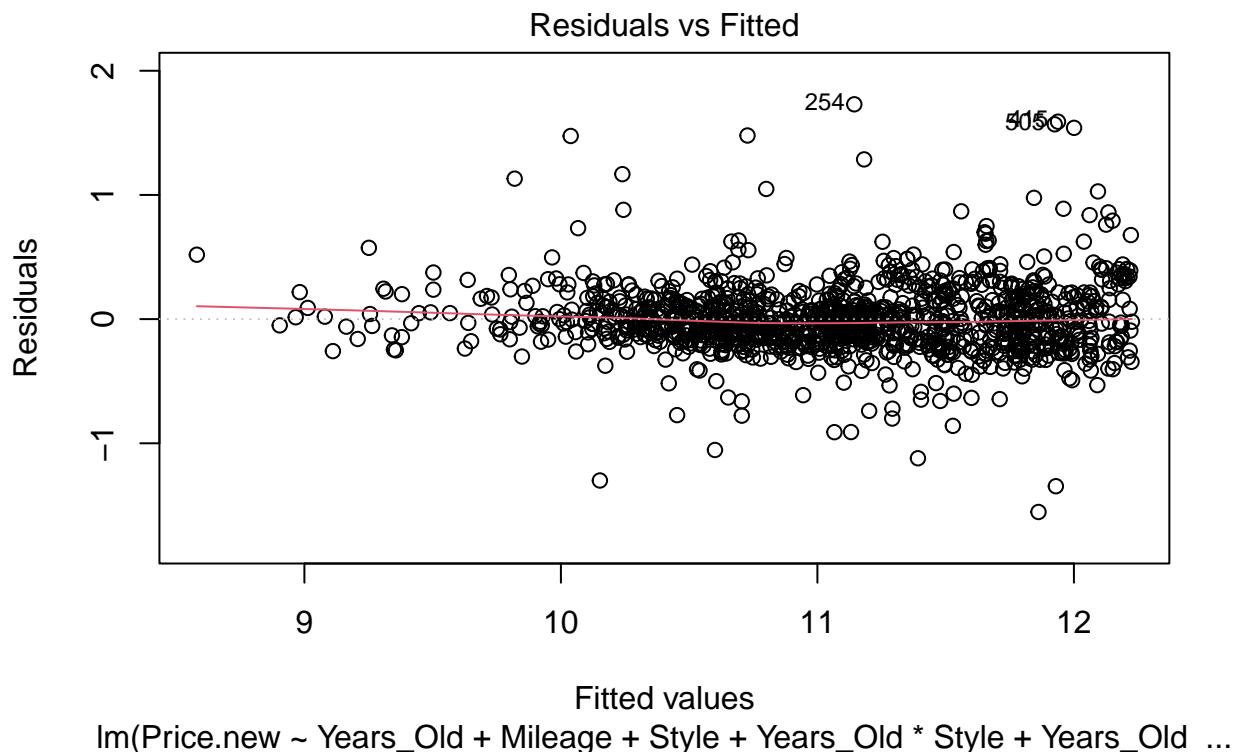
##
## Call:
## lm(formula = Price.new ~ Years_Old + Mileage + Style + Years_Old *
##      Style + Years_Old * Style + Mileage * Style, data = porsche.clean2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55311 -0.14825 -0.03068  0.12244  1.72977
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            1.223e+01  2.379e-02 514.024 < 2e-16 ***
## Years_Old             -1.736e-02  2.362e-03 -7.351 3.60e-13 ***
## Mileage                -1.579e-05  7.870e-07 -20.065 < 2e-16 ***
## StyleBoxster          -6.616e-01  5.942e-02 -11.135 < 2e-16 ***
## StyleCayenne           -5.584e-01  3.664e-02 -15.242 < 2e-16 ***
## StyleCayman            -4.324e-01  7.798e-02 -5.545 3.60e-08 ***
## StyleMacan              -1.027e+00  4.463e-02 -23.019 < 2e-16 ***
## StyleOther              -3.412e-01  4.494e-02 -7.593 6.22e-14 ***
## StylePanamera          -3.363e-01  5.167e-02 -6.509 1.10e-10 ***
## Years_Old:StyleBoxster -5.256e-02  5.666e-03 -9.277 < 2e-16 ***
## Years_Old:StyleCayenne  -8.420e-02  7.755e-03 -10.858 < 2e-16 ***
## Years_Old:StyleCayman  -4.689e-02  1.028e-02 -4.563 5.56e-06 ***

```

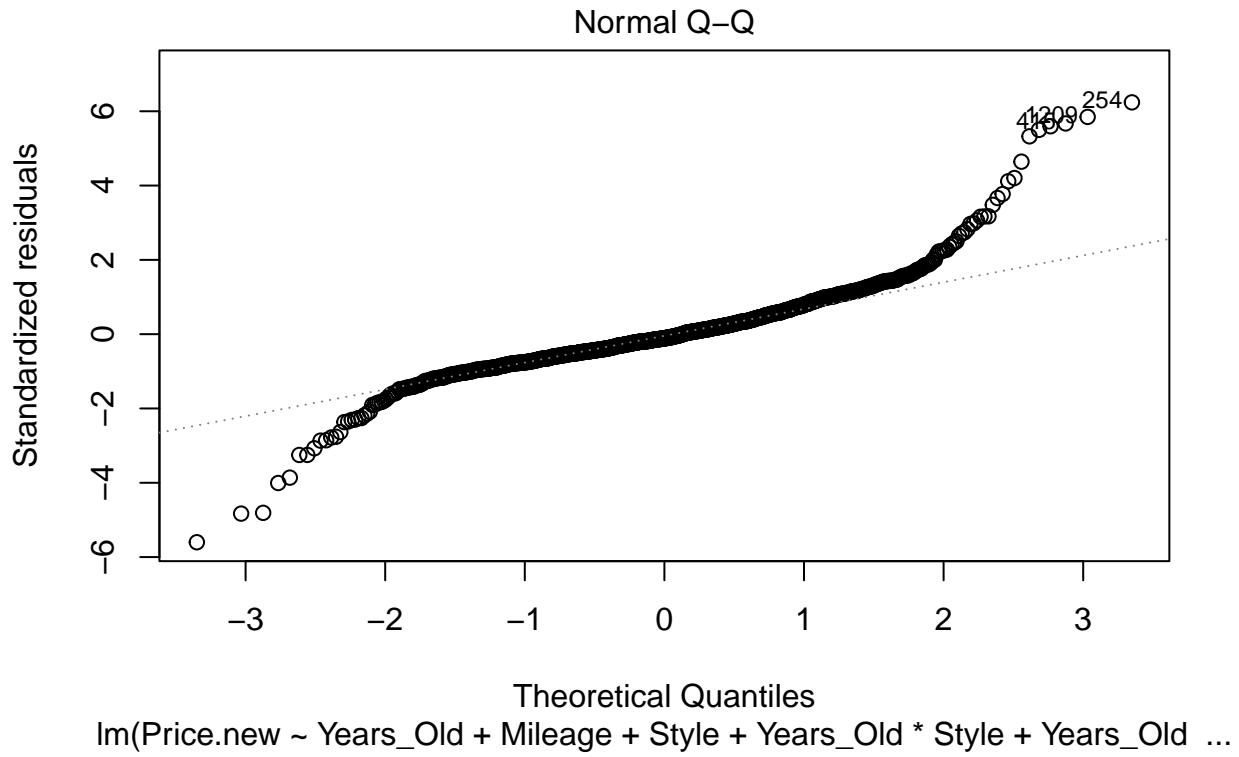
```

## Years_Old:StyleMacan    -4.511e-02  1.520e-02  -2.967  0.00306 ** 
## Years_Old:StyleOther   -1.622e-02  3.444e-03  -4.711  2.75e-06 *** 
## Years_Old:StylePanamera -7.583e-02  1.238e-02  -6.124  1.23e-09 *** 
## Mileage:StyleBoxster    9.448e-06  1.470e-06   6.426  1.88e-10 *** 
## Mileage:StyleCayenne    9.284e-06  1.088e-06   8.531 < 2e-16 *** 
## Mileage:StyleCayman    5.947e-06  2.324e-06   2.559  0.01062 *  
## Mileage:StyleMacan     1.075e-05  1.367e-06   7.865  8.14e-15 *** 
## Mileage:StyleOther     1.178e-05  1.227e-06   9.595 < 2e-16 *** 
## Mileage:StylePanamera  8.938e-06  1.437e-06   6.219  6.89e-10 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 0.2815 on 1213 degrees of freedom 
## Multiple R-squared:  0.8483, Adjusted R-squared:  0.8458 
## F-statistic: 339.1 on 20 and 1213 DF,  p-value: < 2.2e-16 

## [1] "Transformed Model: Variance Assumption:"
```



```
## [1] "Transformed Model: Normality Assumption:"
```



```

## [1] "Final Model:"
```

```

##
## Call:
## lm(formula = Price.new ~ Years_Old + Mileage + Style + Years_Old:Style +
##     Mileage:Style, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.45395 -0.14780 -0.01976  0.12066  1.58733
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            1.223e+01  3.163e-02 386.602 < 2e-16 ***
## Years_Old             -2.214e-02  2.995e-03  -7.392 4.92e-13 ***
## Mileage                -1.382e-05 1.073e-06 -12.881 < 2e-16 ***
## StyleBoxster          -6.530e-01  8.336e-02  -7.834 2.18e-14 ***
## StyleCayenne           -5.647e-01  4.817e-02 -11.721 < 2e-16 ***
## StyleCayman            -4.107e-01  1.029e-01  -3.991 7.41e-05 ***
## StyleMacan              -1.072e+00  6.081e-02 -17.623 < 2e-16 ***
## StyleOther              -3.450e-01  5.999e-02  -5.751 1.42e-08 ***
## StylePanamera          -3.531e-01  6.922e-02  -5.101 4.55e-07 ***
## Years_Old:StyleBoxster -4.732e-02  7.598e-03  -6.228 8.93e-10 ***
## Years_Old:StyleCayenne  -8.102e-02  1.047e-02  -7.739 4.30e-14 ***
## Years_Old:StyleCayman  -4.464e-02  1.343e-02  -3.324 0.000940 ***

```

```

## Years_Old:StyleMacan    -2.466e-02  2.219e-02  -1.112 0.266722
## Years_Old:StyleOther   -1.209e-02  4.609e-03  -2.624 0.008910 ***
## Years_Old:StylePanamera -8.683e-02  1.529e-02  -5.678 2.13e-08 ***
## Mileage:StyleBoxster    7.016e-06  2.096e-06   3.347 0.000868 ***
## Mileage:StyleCayenne    7.538e-06  1.438e-06   5.242 2.21e-07 ***
## Mileage:StyleCayman     4.385e-06  3.060e-06   1.433 0.152411
## Mileage:StyleMacan      8.181e-06  1.944e-06   4.208 2.97e-05 ***
## Mileage:StyleOther       7.646e-06  2.437e-06   3.137 0.001792 **
## Mileage:StylePanamera   9.072e-06  1.892e-06   4.794 2.06e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2655 on 596 degrees of freedom
## Multiple R-squared:  0.8656, Adjusted R-squared:  0.8611
## F-statistic: 191.9 on 20 and 596 DF,  p-value: < 2.2e-16

## [1] "Training Error:"

## [1] 0.2609113

## [1] "Testing Error:"

## [1] 0.3020001

```

Volkswagen Model

```

## [1] "Simple Model:"

##      Year          Brand          Make          Price
## Length:3511      Length:3511      Length:3511      Min.    : 1900
## Class :character Class :character Class :character 1st Qu.:16590
## Mode  :character Mode  :character Mode  :character Median  :23250
##                                         Mean   :24442
##                                         3rd Qu.:30504
##                                         Max.   :66000
##      Mileage         Style         Years_Old
## Min.    : 0  Length:3511      Min.    :-1.000
## 1st Qu.:21151  Class :character 1st Qu.: 2.000
## Median :44123  Mode  :character Median  : 4.000
## Mean   :52198                           Mean   : 5.148
## 3rd Qu.:78176                           3rd Qu.: 7.000
## Max.   :213255                          Max.   :68.000

## [1] "Interaction Model:"

##
## Call:
## lm(formula = Price ~ Years_Old + Mileage + Style + Years_Old *
##      Style + Years_Old * Style + Mileage * Style, data = volkswagen.clean)
##
## Residuals:

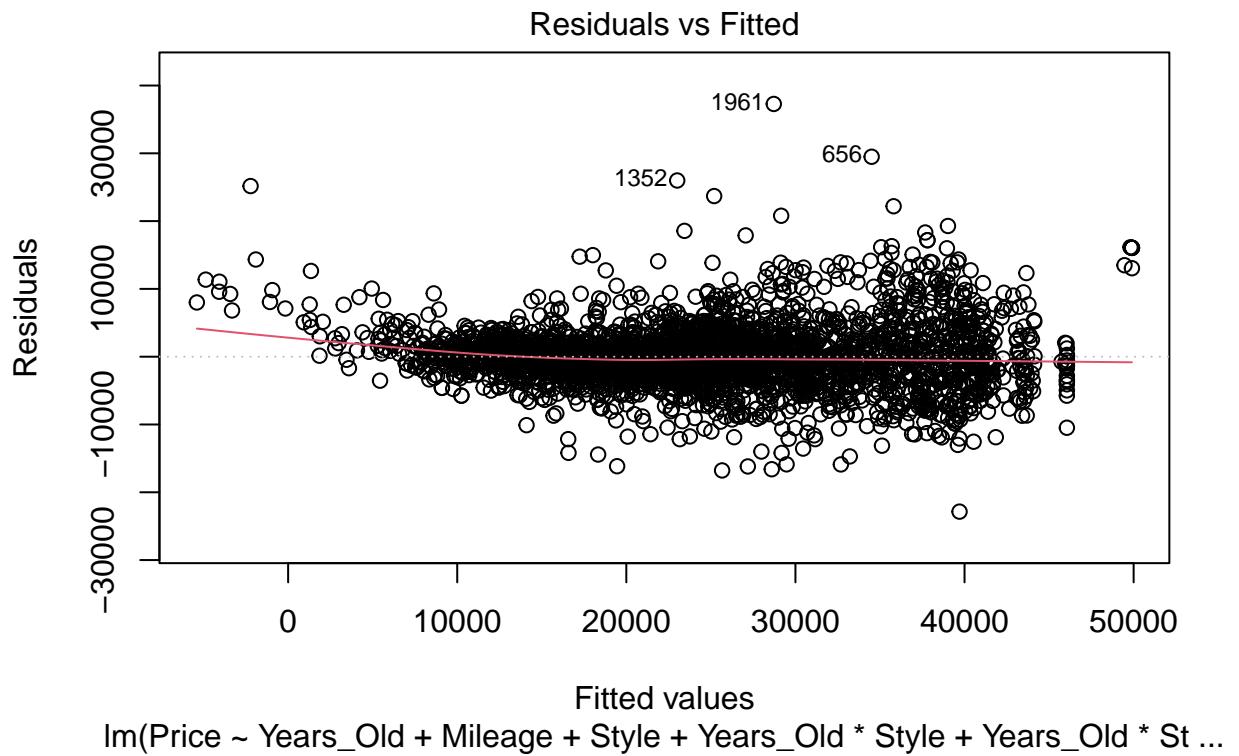
```

```

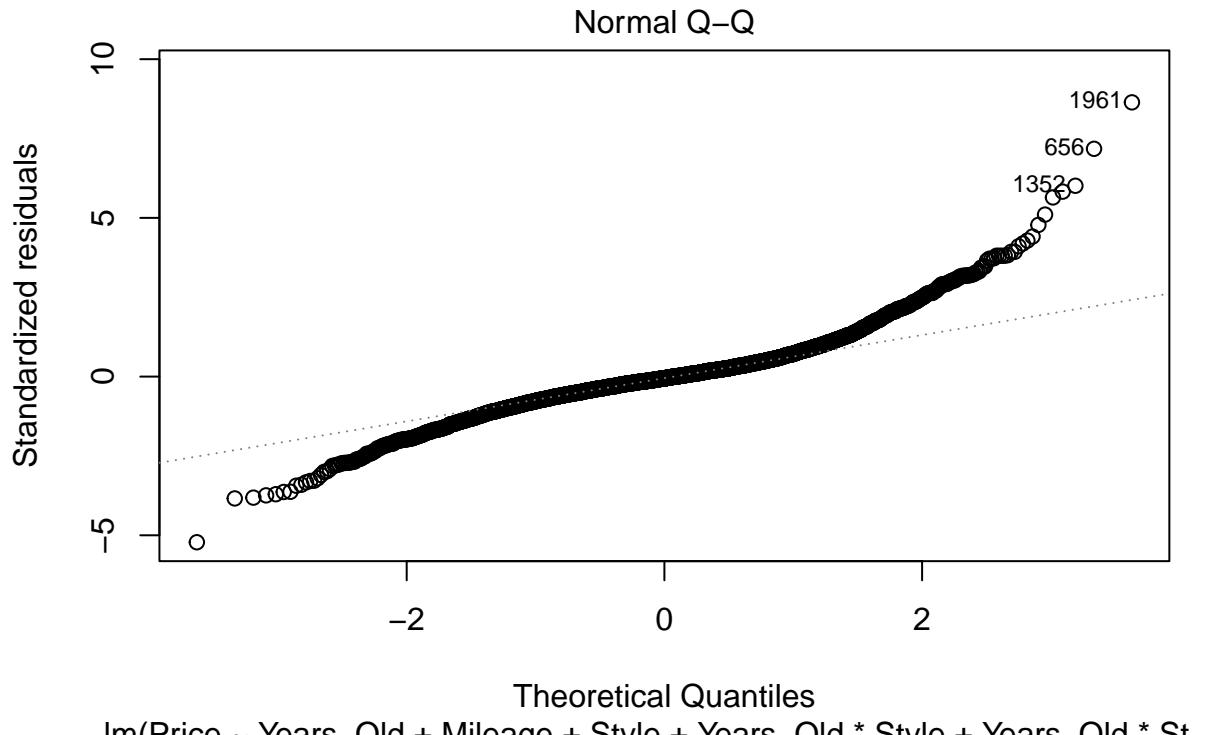
##      Min     1Q Median     3Q    Max
## -22855 -2225   -247   1785  37280
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                4.414e+04  3.112e+02 141.853 < 2e-16 ***
## Years_Old                  -1.913e+03  1.838e+02 -10.410 < 2e-16 ***
## Mileage                     -1.201e-01  1.079e-02 -11.129 < 2e-16 ***
## StyleBeetle                 -1.429e+04  6.830e+02 -20.923 < 2e-16 ***
## StyleGolf                   -7.981e+03  6.444e+02 -12.386 < 2e-16 ***
## StyleJetta                  -1.705e+04  4.529e+02 -37.639 < 2e-16 ***
## StyleOther                  -4.396e+03  4.757e+02 -9.241 < 2e-16 ***
## StylePassat                 -1.538e+04  5.279e+02 -29.129 < 2e-16 ***
## StyleTiguan                 -1.079e+04  4.341e+02 -24.844 < 2e-16 ***
## StyleTouareg                2.245e+04  1.985e+03 11.312 < 2e-16 ***
## Years_Old:StyleBeetle       1.984e+03  1.846e+02 10.745 < 2e-16 ***
## Years_Old:StyleGolf          5.621e+02  2.151e+02  2.614 0.009000 **
## Years_Old:StyleJetta         1.037e+03  1.966e+02  5.277 1.40e-07 ***
## Years_Old:StyleOther          1.958e+03  1.866e+02 10.494 < 2e-16 ***
## Years_Old:StylePassat        9.828e+02  1.993e+02  4.931 8.57e-07 ***
## Years_Old:StyleTiguan         5.663e+02  2.024e+02  2.797 0.005180 **
## Years_Old:StyleTouareg      -8.660e+02  2.836e+02 -3.054 0.002274 **
## Mileage:StyleBeetle          -4.729e-02  1.323e-02 -3.574 0.000357 ***
## Mileage:StyleGolf             1.223e-02  1.368e-02  0.894 0.371597
## Mileage:StyleJetta            4.731e-02  1.238e-02  3.820 0.000136 ***
## Mileage:StyleOther             -1.390e-01  1.256e-02 -11.066 < 2e-16 ***
## Mileage:StylePassat            4.285e-02  1.280e-02  3.347 0.000824 ***
## Mileage:StyleTiguan            3.352e-02  1.317e-02  2.545 0.010984 *
## Mileage:StyleTouareg          -1.138e-01  1.703e-02 -6.684 2.70e-11 ***
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4384 on 3487 degrees of freedom
## Multiple R-squared:  0.8181, Adjusted R-squared:  0.8169
## F-statistic: 681.9 on 23 and 3487 DF,  p-value: < 2.2e-16

## [1] "Variance Assumption:"

```



```
## [1] "Normality Assumption:"
```



```

## [1] "Transformed Model:"
```

```

##
## Call:
## lm(formula = Price.new ~ Years_Old + Mileage + Style + Years_Old *
##      Style + Years_Old * Style + Mileage * Style, data = volkswagen.clean2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.83212 -0.08782  0.00500  0.09420  1.39478
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.071e+01  1.288e-02 831.074 < 2e-16 ***
## Years_Old            -5.050e-02  7.610e-03 -6.636 3.72e-11 ***
## Mileage              -3.689e-06  4.469e-07 -8.254 < 2e-16 ***
## StyleBeetle          -2.609e-01  2.828e-02 -9.225 < 2e-16 ***
## StyleGolf             -1.426e-01  2.668e-02 -5.345 9.62e-08 ***
## StyleJetta            -4.054e-01  1.875e-02 -21.624 < 2e-16 ***
## StyleOther            -7.595e-02  1.969e-02 -3.857 0.000117 ***
## StylePassat           -3.158e-01  2.186e-02 -14.449 < 2e-16 ***
## StyleTiguan           -2.241e-01  1.797e-02 -12.467 < 2e-16 ***
## StyleTouareg          9.700e-01  8.217e-02  11.805 < 2e-16 ***
## Years_Old:StyleBeetle  5.193e-02  7.643e-03  6.795 1.27e-11 ***
## Years_Old:StyleGolf    -6.884e-03  8.904e-03 -0.773 0.439460

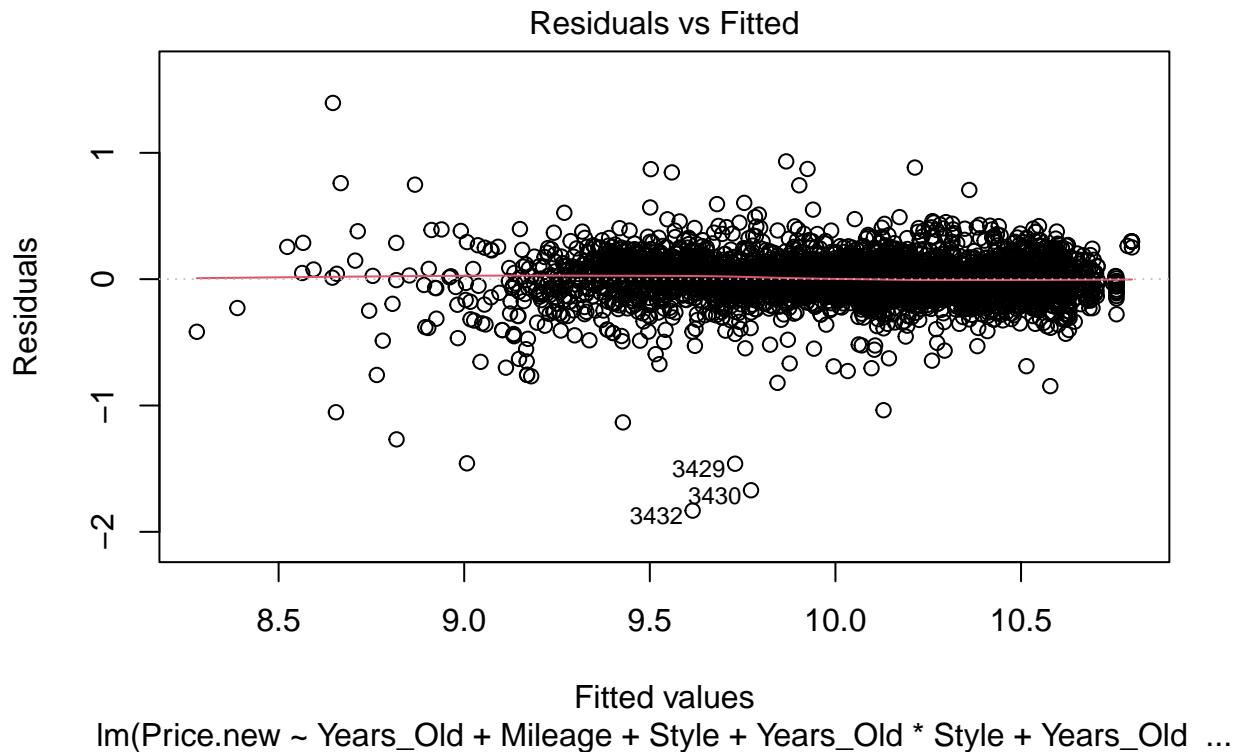
```

```

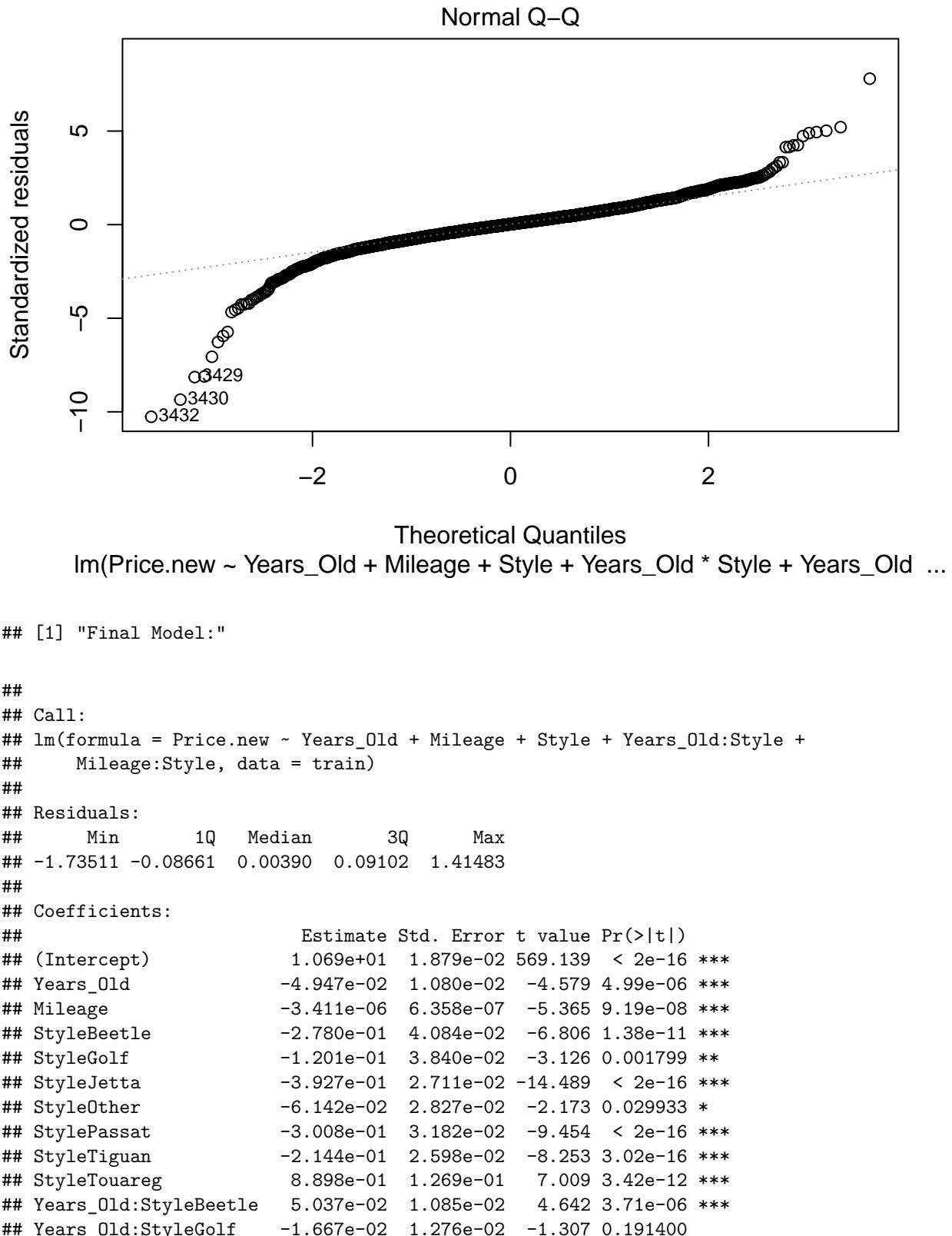
## Years_Old:StyleJetta -6.667e-03 8.138e-03 -0.819 0.412727
## Years_Old:StyleOther 5.213e-02 7.726e-03 6.748 1.75e-11 ***
## Years_Old:StylePassat -1.134e-02 8.252e-03 -1.374 0.169382
## Years_Old:StyleTiguan -1.563e-02 8.381e-03 -1.865 0.062235 .
## Years_Old:StyleTouareg -9.577e-02 1.174e-02 -8.158 4.71e-16 ***
## Mileage:StyleBeetle -6.644e-06 5.478e-07 -12.128 < 2e-16 ***
## Mileage:StyleGolf -1.149e-06 5.665e-07 -2.027 0.042705 *
## Mileage:StyleJetta -5.307e-07 5.128e-07 -1.035 0.300736
## Mileage:StyleOther -8.501e-06 5.202e-07 -16.342 < 2e-16 ***
## Mileage:StylePassat -7.016e-07 5.300e-07 -1.324 0.185690
## Mileage:StyleTiguan -1.778e-07 5.453e-07 -0.326 0.744460
## Mileage:StyleTouareg -2.505e-06 7.052e-07 -3.552 0.000388 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1815 on 3487 degrees of freedom
## Multiple R-squared: 0.8416, Adjusted R-squared: 0.8406
## F-statistic: 805.8 on 23 and 3487 DF, p-value: < 2.2e-16

```

```
## [1] "Transformed Model: Variance Assumption:"
```



```
## [1] "Transformed Model: Normality Assumption:"
```



```

## Years_Old:StyleJetta -1.562e-02 1.162e-02 -1.344 0.179019
## Years_Old:StyleOther 4.906e-02 1.097e-02 4.474 8.19e-06 ***
## Years_Old:StylePassat -1.106e-02 1.170e-02 -0.946 0.344524
## Years_Old:StyleTiguan -1.996e-02 1.197e-02 -1.668 0.095504 .
## Years_Old:StyleTouareg -7.137e-02 1.867e-02 -3.822 0.000137 ***
## Mileage:StyleBeetle -6.123e-06 7.833e-07 -7.817 9.35e-15 ***
## Mileage:StyleGolf -9.058e-07 8.073e-07 -1.122 0.262012
## Mileage:StyleJetta -9.318e-08 7.405e-07 -0.126 0.899877
## Mileage:StyleOther -8.616e-06 7.317e-07 -11.776 < 2e-16 ***
## Mileage:StylePassat -1.051e-06 7.489e-07 -1.403 0.160837
## Mileage:StyleTiguan -1.062e-07 7.860e-07 -0.135 0.892539
## Mileage:StyleTouareg -3.474e-06 1.034e-06 -3.360 0.000798 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1856 on 1731 degrees of freedom
## Multiple R-squared: 0.8347, Adjusted R-squared: 0.8325
## F-statistic: 379.9 on 23 and 1731 DF, p-value: < 2.2e-16

## [1] "Training Error:"

## [1] 0.1843383

## [1] "Testing Error:"

## [1] 0.1796327

```

Conclusion

Overall, we were able to learn a lot through our dataset about the different factors that effect the price of German cars and how we can apply that to our own personal needs and life. Specifically, one of the most important things that we saw was the agreement between our models, visualizations, and data analysis. By seeing interactions that did not look significant in a graph also have unsignificance in our final models made us quite confident about ou models ability to help us solve our problem. The creation of the different model for each brand and model with all brands combined made for great comparisons as well. We were able to see that the model selection process was almost the same, in terms of the use of a log transformation on the y and other changes to make the model stronger and more fortified. We were able to learn about the correlation between age and mileage on cars across different brands and how for specific brands price is effected by brand-specific variables. For example, with Porsche's the model typically effects the cars age and its pirce while for Volkswagen's we have two segments of data based on the age of the Volkswagen which shows that there might exist more vintage VW's adn newer ones, which are priced quite similarly.

Overall, we were able to learn about how different variables such as class, model, age, drive type, and mileage and their interactions effect the price of cars in general and across our different brands. We look forward to applying this model to our own personal needs as we shop around for some of these fast, luxorius German cars.

Statistical Conclusions

AJ: I personally am very interested in high-end German cars (specifically from Porsche or BMW, but obviously want the car to be cheap (less than \$45,000). Furthermore, I want the car to be a sedan. I do

not really care about the age or mileage on the car however. By using our model, I was able to find a set of cars that fit this mold and were likely significantly cheaper than an other model I would get. This has significant impacts for me as I am able to narrow down a potential car search and still afford a luxurious car that I can enjoy driving. Andrew: I felt like there was a lot to take away from this project, and the most interesting thing that I will take away is that data science programming methods can save someone a lot of money. Developing a model to identify the prices of luxury German cars that have the specific mileage, age, and brand of your choosing can help someone identify what cars could be over or under-priced and where one can save money. Personally, I have a new respect for data science after this project, and I will try to embrace this analytical-process in future queries. Mai: I found it fascinating that some luxury brands actually had cheaper cars that seemed inherently better based off of our model and data. For example, I found some newer Porsche's with little mileage on them that were actually significantly cheaper than older Volkswagen's with some decent mileage on them. It really makes me reevaluate how car's can be priced and gives more insight into how I would choose a potential German car. Jamie: I was amazed by the outputs of our modeling. I did not expect all of the interaction terms to be significant. I was also super surprised that a lot of the variables that we manual created from the initial dataset ended up having great significance. It goes to show the importance of evaluating your data and observing it properly prior to hopping into statistical analysis right away.

Personal Takeaways

AJ: Realizing the power of combining topics across different STAT courses. Specifically, I was able to take a lot from STAT 425 (Statistical Modeling) and use it in conjunction with topics from this course. For example, prior to creating our statistical models, which largely used topics I learned from STAT 425, I had to clean and evaluate the data, as well as create visualizations to better help me understand what the data looked like and where relationships may exist. This allowed me to create better models with the creation of new variables and removal of unnecessary ones that would not have been caught by ordinary model testing but was found through the initial analysis and visualizations of our data. Andrew: Through this project and this class, I believe that I was able to grow as a data engineer, data scientist, and data analyst. We started to start the class with the basics of shell programming in order to understand the basics of data systems and structures. We used the knowledge and experience from this unit to retrieve and store the data correctly so that everyone in our group could work with it. In addition to this, I felt like the data manipulation skills that we learned in this class were crucial for our success in this project. From this project, I feel like I have been able to implement the skills that we learned in class and practiced in assignments and quizzes to ultimately create a complete analysis of something that I am passionate about. Mai: I loved seeing how the capabilities of R compared to those of Python. I had always used Python for statistical analysis and building things like web scrapers, so I was amazed by how easy it was to implement similar functionalities in R. I am starting to see the power of R as a statistical language more and more. Jamie: I could not believe how much I was able to grow and learn from this project. I had no prior experience in Shiny applications or anything related to it, but I was able to take just a small lecture about it and create some pretty cool application that has some actual use applications. It goes to show how you can just take some small pieces of a course like this and really build on it at deeper levels.