MA575 Linear Models C1 Group1
Lab Report #4

Xudong Han, Jialin (Ella) Hu, Youngju (Jenna) Lee, Kejing (Skylar) Yan, Samantha DiMarco

Modeling Price Determinant Features of Used BMW Cars Sold at Auction in 2008

Abstract

This article aims to investigate the factors that affect the price of used BMW cars sold at auction in 2008 using multiple linear regression principles. In this study, 14 selected predictor variables, *Mileage*, *Engine Power*, *Age*, *Car Type*, *Paint Color*, *Features 1-8*, and *Model Key*, are used to predict the auction price based on detailed reasons and background interests. Diagnostic methods such as scatter plots, standardized residual plots, and partial F-tests are deployed to test the appropriateness of the constructed model. The results of this study may provide valuable insights into the factors that influence the selling price of used BMW cars.

1. Introduction

1.1. Background Information

We aim to investigate the factors that affect the prices of used BMW cars sold at an auction in 2008 using multiple linear regression principles. The dataset includes several explanatory variables, such as mileage, engine power, date sold, register date, model key, and others. To construct a model with multiple covariates, we will first select p predictors, where $p \leq n$. We will not randomly select variables to fit the model but rather provide detailed reasons and background interest of why we chose them as variables. We will deploy various steps of multiple linear regression diagnostics to test the appropriateness of the model constructed. These diagnostics will include scatter plots between variables, standardized residual plots, and partial F tests. The results of this study may provide valuable insights into the factors that influence the price of used BMW cars and may help individuals and companies make informed decisions about buying and selling these cars.

1.2. Initial Variable Selection

There are a total of 18 variables in the given BMW dataset. As our goal in discovering this dataset is to find out what factors will affect the price change of BMW cars, we decided to set the 2008 auction price as our response variable, y. For the remaining 17 covariates (aside from the indexing variable) , according to their respective features, we selected 6 of them as our predictor variables to construct our initial multiple linear regression model. The reasons for selecting each variable are given below:

1. Mileage: Some of our group members had experience buying or selling second-handed vehicles, and according to their experience, the price of a vehicle will normally drop when the mileage is high. So, we include mileage as a predictor variable here to explore its relationship with price.

2. Engine power: The engine is an important component of a car that people willing to buy a car will constantly assess. As a car's engine power improves, the car might be more valuable. For this reason, we also include engine power as an explanatory variable in our model.
3. Registration date & Date sold: Instead of directly using the two string-type variables, we transformed the date into a numeric time variable and calculated the age of the car. The new variable, “age,” will be a more reasonable covariate since it keeps track of the number of days the car is being used. We believe that if a car’s age is older, the cheaper it will be.
4. Car type: Specific car types include convertible, coupe, estate, hatchback, sedan, subcompact, SUV, and van. When considering the price difference between a coupe and a van, we always think that a coupe will be more expensive. So we add car type as another covariate to explain and estimate the price.
5. Paint color: We can see vehicles with different colors on the road daily. But some colors might appear more often than others. For example, black cars are certainly appearing more than green ones. So we want to know whether the consumer demand difference will also affect the price.

Based on the above thoughts and reasons, we decided on these covariates as explanatory variables and price as the response variable in our initial model. In the following part, we will construct scatter plots or box plots according to the data type to briefly see whether they seem to be a useful predictor for price. After that, we will construct multiple linear regression and do various diagnostic tests to check the appropriateness of our model.

1.3. Data Cleaning

The original BMW dataset contains 4,841 observations of 22 variables, including the 2008 auction price of each used BMW car and various factors that could influence the sold price. Before any statistical analysis, we cleaned the dataset. We looked for unrealistic observations within our data, such as cars with mileage and engine power less than zero, and removed them from our dataset.

Using the original *registration date* and *sold at* variables, we created a new numerical variable, *age*, representing the number of days since the car was first registered and when it was sold at auction.

Moreover, *model keys* such as ‘125’, ‘330 Gran Turismo’, ‘650’, and other subcategories have comparatively small observation numbers. So we decided to categorize the *model keys* with observations smaller than 100 into a sub-category called *other key*. For covariate *car type*, we found that ‘convertible,’ ‘coupe,’ ‘subcompact,’ ‘hatchback,’ and ‘van’ have a small number of observations compared with the other three types. So, we combined these five sub-categories into one, called *others*.

Lastly, we split our dataset into two; a training dataset to curate our model and a validation dataset to test our final model. We split based on the given assigned index.

2. Modeling and Analysis

2.1. Initial Full Model

As mentioned above, we formed scatter plots and box plots based on our training dataset to subjectively check whether the covariates we suggested are useful and meaningful to our response variable.

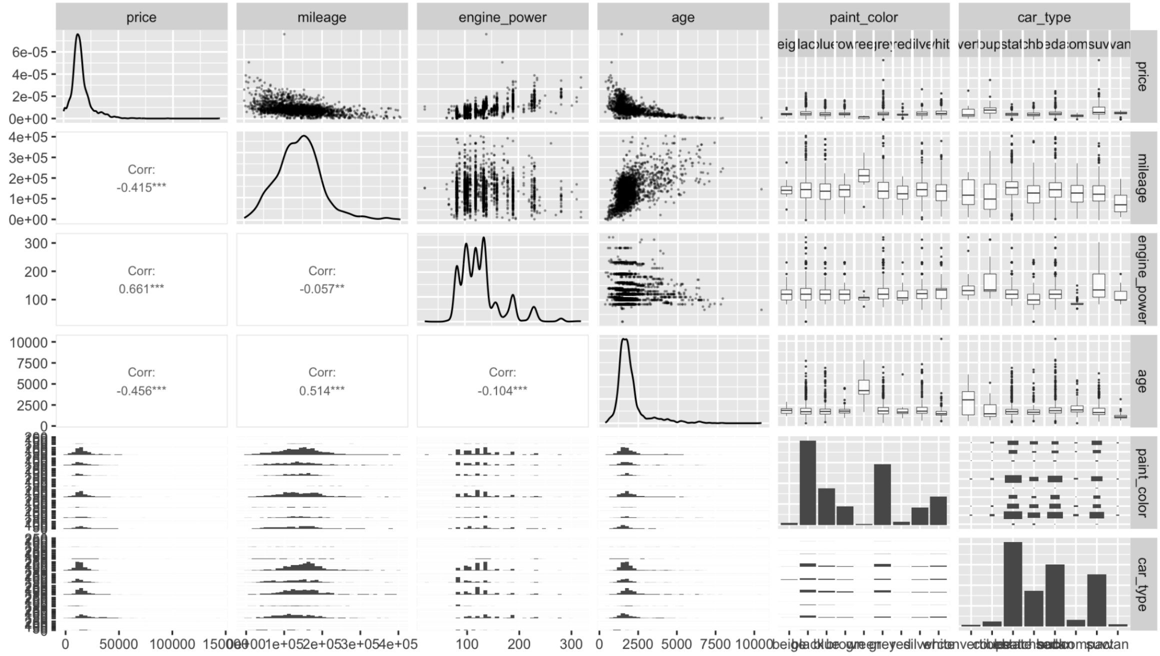

Figure 1: Scatterplot matrix for all five covariates and price

It is obvious that *mileage* and *engine power* have a linear relationship with *price*. But *age* has a non-linear relationship with price. Thus, we decided to perform a transformation on *age*, which we will use *log(age)* in our later modeling and analysis. The relationship between *paint color* and *car type* with the price is hard to check from the box plots of many sub-categories. So we decided to combine some sub-categories of *car type* with a small number of observations into a reference group and conduct partial F tests to check the significance.

2.2. Reduced Model

The p-value of the partial F test with the null hypothesis: $\beta_{paint\ color} = 0$ is 0.06016. As we select $\alpha = 0.05$ before the test, we cannot reject the null hypothesis, and we thus do not include paint color as an explanatory variable in our model (*Table a1*).

The partial F test relative to *car type* rejects the null hypothesis. The p-value produced is 2.2e-16, which is nearly 0. So under any reasonable alpha value, we reject the null hypothesis. Thus, we continued to include the covariate *car type* to estimate *price* in our later models (*Table a1*).

As we removed paint color, we still want to know if other useful covariates can help to estimate and predict the *price*. So we perform another partial F test regarding *model keys* and create box plots for *features 1-8* separately to check if they are significant (*Table 3*). Results showed that *model keys* are significant, and *features 1-6* and *8* are also helpful in estimating the *price*.

This is our model so far (*Table 4*):

$$price \sim mileage + engine\_power + \log(age) + car\_type + model\_key + feature1 + feature2 + feature3 + feature4 + feature5 + feature6 + feature8$$

And below is the according to the standardized residual plot:

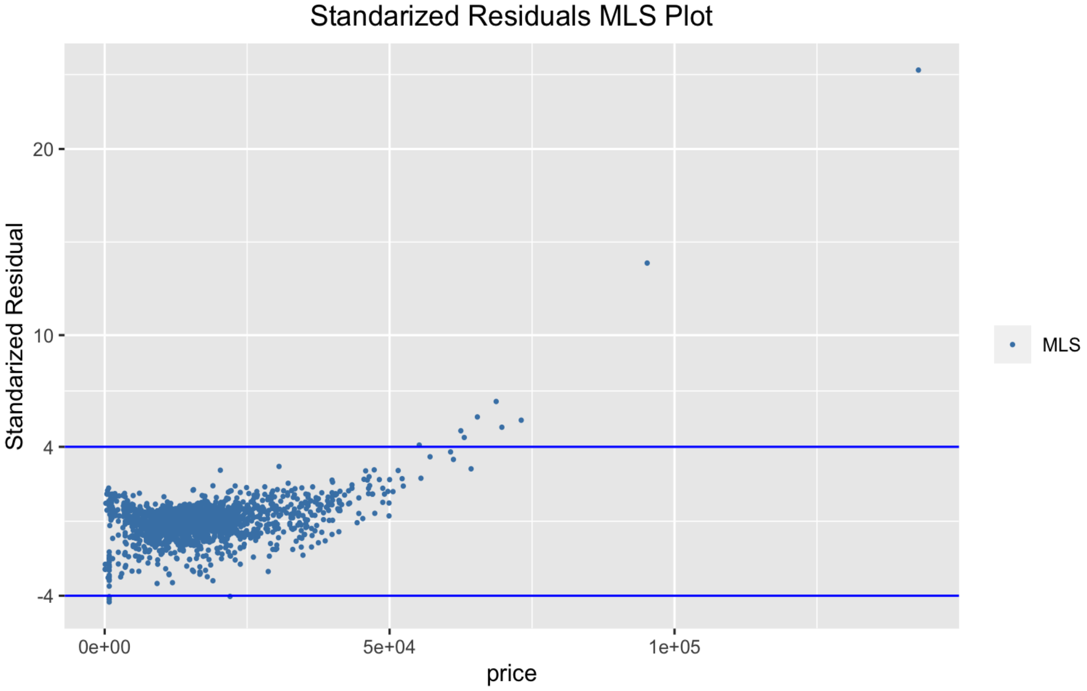

Figure 2: Standardized Residual vs. Price for full model

Based on Figure 2, we further analyzed the potential outlier seen in the upper righthand corner of the plot. We identified this potential outlier as observation #2338, an approximately four year old gray model X4 SUV with more than 103,00 miles sold for $142,800. This is a very high auction price compared to the other cars in our training dataset. To confirm that this car is an outlier, we calculated the Cook's Distance, 0.419205, and removed the observation from our dataset (See Figure a13).

2.3. Transformed Model

From the standardized residual plot, we can see a non-linear trend, and the assumption of homoscedasticity seems not to hold. In this case, we plan to conduct strategies such as weighted least squares and transformation of the response variable to improve the model.

BoxCox is a strong transformation when non-linearity happens. The λ produced from the BoxCox function in the MASS package is around 0.6, so in the following models, we will use $price^{0.6}$ as the response variable (*Figure a4*). The violation of the assumption of linearity has been resolved mainly by transforming the response variable. We still intend to use weighted least to refine the problem of non-constant variance.

We tried various weights to apply to the WLS. As mileage and engine power have a linear relationship with the response variable, we tried these two as weights.

We are using engine power as weight makes the distribution of standardized residuals more random. There is no noticeable trend, so we will set the following form as our transformed model:

$$\begin{aligned} price^{0.6} \sim\ & mileage + engine\_power\ + log(age)\ + car\_type\ + model\_key + feature1 \\ & + feature2 + feature3 + feature4 + feature5 + feature6 + feature8, \\ & weights\ =\ engine\_power \end{aligned}$$

The diagnostic plots showed that the assumptions of multiple linear regression are achieved, despite the lower and upper quantiles in the Normal Q-Q plot deviating from the line a bit (*Figure a6*). But since the sample size is large, we think the normality assumption is still achieved.

2.4. Final Model

To avoid overfitting, we applied the LASSO variable selection strategy. LASSO (Least Absolute Shrinkage and Selection Operation) is a regularization method used for variable selection in regression analysis. In this case, we applied LASSO to a model fitted with our predictor variable, *price^0.5*, and the covariates *mileage*, *engine power*, *log(age)*, *car type*, *features 1-8*, *other key*, and *binary variables* for the BMW models not grouped as *others* (*Table a5*).

LASSO variable selection tells us that while most of our covariates have regression coefficients different from zero, some of the covariates have coefficients that shrink away to zero (See Table 5). Based on our application of LASSO, we fitted a new model to be referred to as the “LASSO model.”

After fitting the new model based on LASSO variable selection, we constructed various diagnostic plots to evaluate our model. The Standardized Residuals vs. Fitted Values plot (*Figure a9*) for the LASSO model shows a random pattern, indicating that our model's constant error variance assumption holds. Constructing a QQ-plot for the LASSO model (*Figure a11*) shows deviation from the linear pattern in the theoretical quantiles outside of (-1, 2). However, since we are dealing with a huge dataset, some deviation from the ideal linear pattern is to be expected. The histogram of the standardized residuals for the LASSO model (*Figure a8*) appears to be unskewed and normal, providing evidence that the normality assumption for our model holds. The diagnostic plots in (*Figure a11*) show random patterns in the Residuals vs. Fitted and $\left|\sqrt{Standardized\ Residuals}\right|$ vs. Fitted Values plots, providing further evidence that our LASSO model is a good fit.

Based on these results, we've selected the LASSO model as our final model.

3. Prediction and Validation

We did some basic data cleaning processes for the validation data set, such as removing data entries with *mileage*, *engine power*, and *price* lower than 0. We also move the sub-categories for *car type* into the reference group. We removed data entries with *car type*s that did not appear in the training data set.

We use the explanatory variables in the validation data set and the final model to predict the estimated price based on 50 data points. *Figure a12* shows the complete comparison between actual and predicted price, and *Figure 3* shows a close view of comparing the predicted and actual prices of the validation data set.

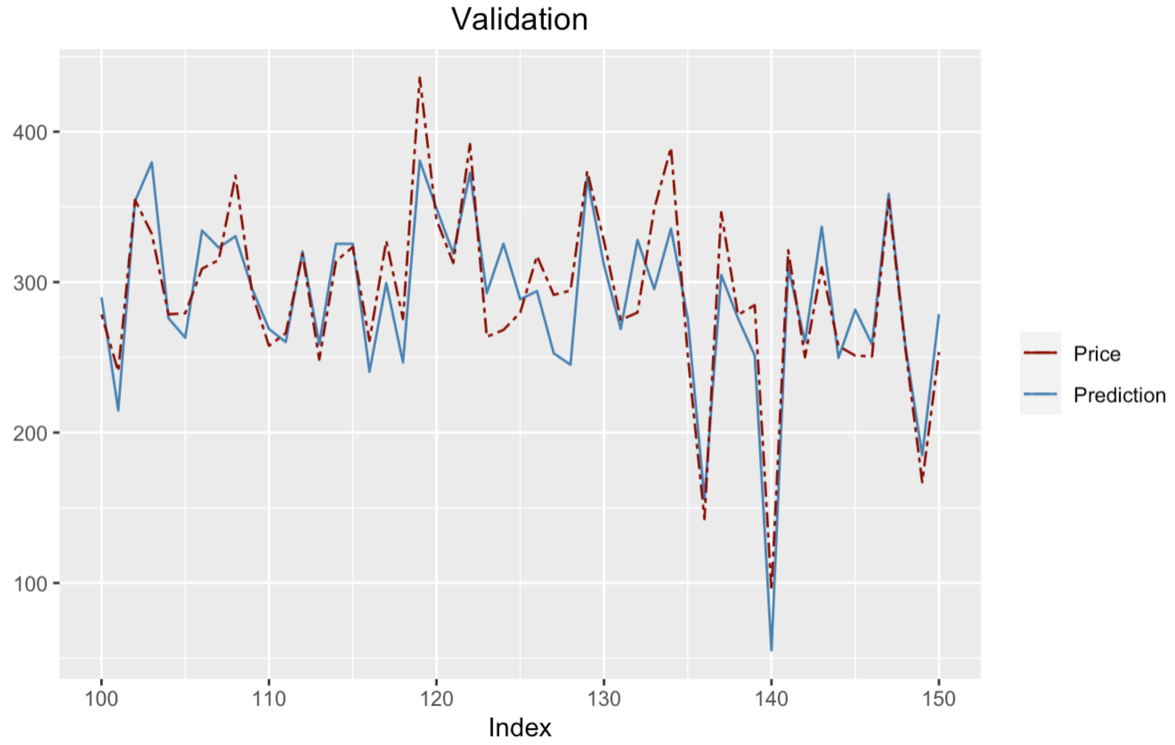

Figure 3: Validation Plot for the final model (based on 50 data points)

From the graph, we can observe that our model generally captures the basic fluctuation trend of the actual price, indicating that our model has some predictive power.

However, it is also important to note that our model tends to perform more extreme predictions when the actual price drops. This suggests that our model may not accurately capture the nuances of the relationship between the different features and car prices under certain conditions.

Despite these limitations, the overall performance of our model is good, as evidenced by the comparison between actual and predicted prices.

4. Discussion

Our model has some limitations. Since we did data cleaning, we removed all entries with price, mileage, and engine power less than 0. So that our model provides appropriate prediction toward data with corresponding properties.

Moreover, we categorized some car types with fewer observations into a reference group. This may lead to inaccuracies in prediction. Also, if the car types of a new observation are not included in our current categories, our model could not provide an appropriate fit for the model.

In addition, we included features 1, 2, 3, 4, 5, 6, and 8 into our model, but we don't know what these features specifically are. These features might be whether there is Apple car play, Bluetooth, and other functions. In our case, we can only introduce them as feature 1 but not specific functions to clients or people who are interested in them.

We removed model keys to avoid overfitting, i.e., to avoid tracking data points but not the trend. However, this may have resulted in the omission of significant variables, leading to an incomplete understanding of the relationship between different features and car prices.

Also, we can see in the QQ-plot for our final (LASSO) model (*Figure a7*) that the data points begin to deviate from the linear pattern in the outer quantiles. While this is expected when analyzing a large dataset, it is important to acknowledge that this deviation may impact the accuracy of our model and lead to some degree of uncertainty in our predictions.

5. Conclusion

The standardized residual plot showed a relatively random pattern. Still, some potential outliers are out of the cut-off boundaries ± 4 (large sample size). However, none of the observations have Cook's distance larger than 0.5. So we conclude there are no influence points.

Our study suggests that covariates such as *mileage*, *engine power*, *age*, *car type*, *model keys*, and different *features* all have significant contributions to the price of used BMW cars. We recognize that other variables not included in our final model, such as fuel type, might also affect the sale price. However, we believe that the relative significance of these variables might be smaller than the ones we included in our model.

Furthermore, other combinations of the covariates might perform better than our final model. As such, further research could explore the inclusion of other variables or the use of more advanced modeling techniques to enhance the predictive power of our model.

6. Contribution

Xudong Han: Modeling & Figures & Coding
Jialin Hu: Modeling & analysis

Youngju Lee: Analysis & editing
Kejing Yan: Abstract, Introduction; Modeling & Analysis; Discussion; Conclusion
Samantha DiMarco: LASSO Variable Selection, Final model analysis

7. Appendix

Table a1: partial F test regarding *paint color*

<table>
<tr><td colspan="5">Reduced Model: lm(price ~ mileage + engine_power + age + car_type)</td><td colspan="5">Reduced Model: lm(price ~ mileage + engine_power + age)</td></tr>
<tr><td colspan="5">Full Model: lm(price ~ mileage + engine_power + age + car_type + paint_color)</td><td colspan="5">Full Model: lm(price ~ mileage + engine_power + age + car_type)</td></tr>
<tr><td>Res. Df</td><td>RSS</td><td>Df</td><td>Sum of Sq</td><td>P-value</td><td>Res. Df</td><td>RSS</td><td>Df</td><td>Sum of Sq</td><td>P-value</td></tr>
<tr><td>2412</td><td>7.2344e+10</td><td></td><td></td><td></td><td>2415</td><td>7.9113e+10</td><td></td><td></td><td></td></tr>
<tr><td>2404</td><td>7.1896e+10</td><td>8</td><td>447891975</td><td>0.06016</td><td>2412</td><td>7.2344e+10</td><td>3</td><td>6769352639</td><td>< 2.2e-16</td></tr>
</table>

Table a3: box plots of features 1-8

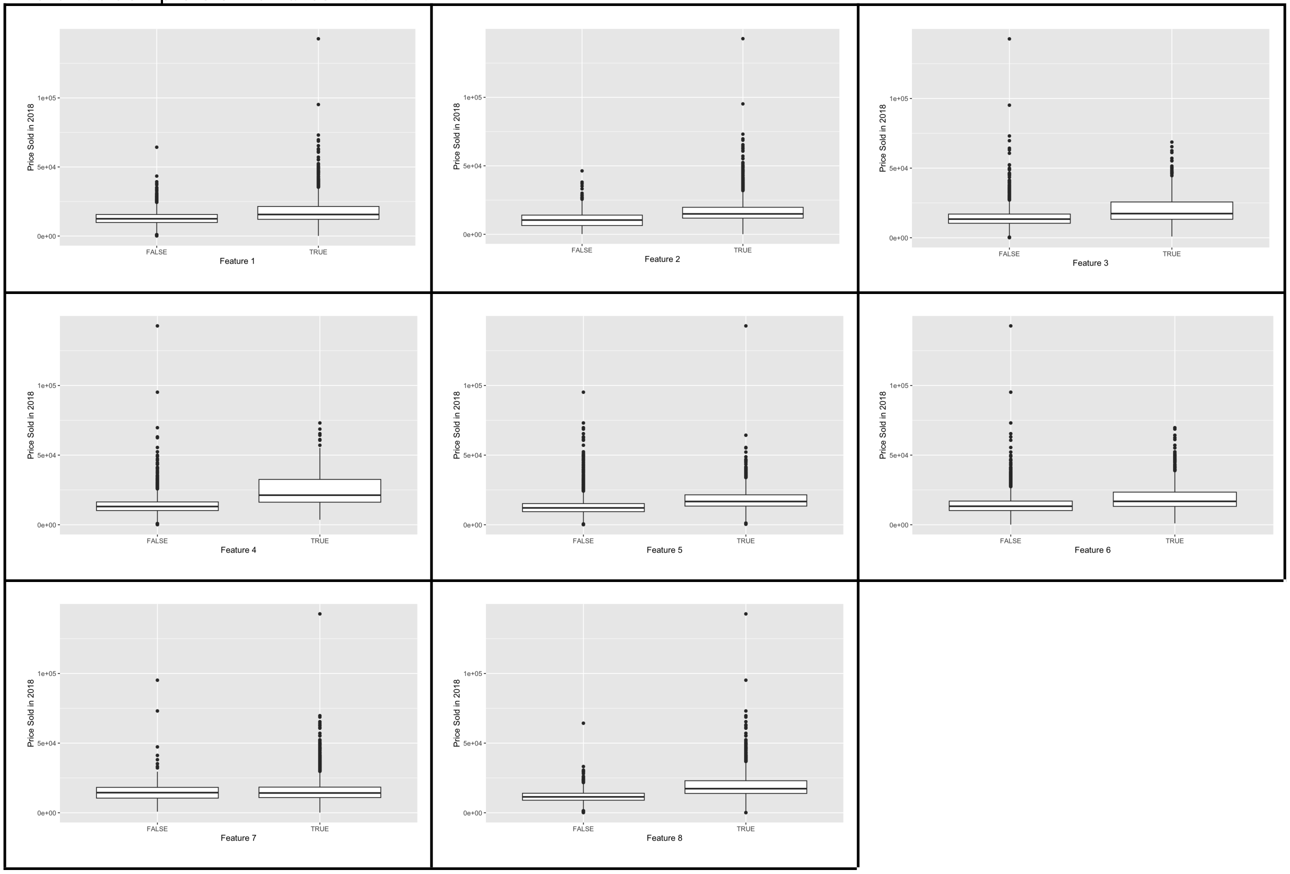

Table a4: summary table of full model

	Estimate	Std. Error	p-value		Estimate	Std. Error	p-value

(Intercept)	6.85E+04	3.07E+03	$10^{(-16)}$	model_key520	1.34E+03	2.97E+02	6.53E-06
mileage	-2.60E-02	1.54E-03	$10^{(-16)}$	model_keyX1	-9.76E+03	5.74E+02	<2E-16
engine_power	9.12E+01	3.04E+00	$10^{(-16)}$	model_keyX3	-7.68E+03	5.15E+02	<2E-16
log(age)	-8.37E+03	2.53E+02	$10^{(-16)}$	model_keyX5	-9.83E+02	5.35E+02	0.066286
car_typeestate	-1.85E+03	3.06E+02	1.70E-09	feature_1TRUE	1.02E+03	1.69E+02	1.49E-09
car_typesedan	4.10E+02	3.01E+02	0.172655	feature_2TRUE	5.45E+02	2.09E+02	0.00911
car_typesuv	8.66E+03	5.02E+02	$10^{(-16)}$	feature_3TRUE	7.93E+02	1.94E+02	4.62E-05
model_key116	-1.23E+03	3.71E+02	0.000924	feature_4TRUE	1.03E+03	2.43E+02	2.50E-05
model_key316	5.04E+02	4.50E+02	0.262591	feature_5TRUE	-2.04E+02	1.70E+02	0.230605
model_key318	1.07E+02	3.47E+02	0.758276	feature_6TRUE	9.61E+02	1.81E+02	1.17E-07
model_key320	-1.34E+03	2.85E+02	2.87E-06	feature_8TRUE	1.41E+03	1.79E+02	6.30E-15

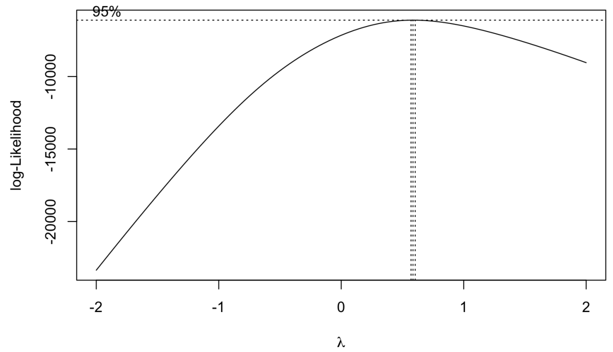

Figure a4: Boxcox function applied on reduced model

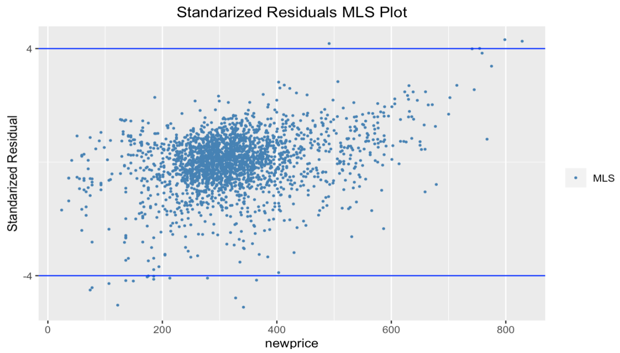

Figure a5: Standardized Residual plot of reduced model with transformed price

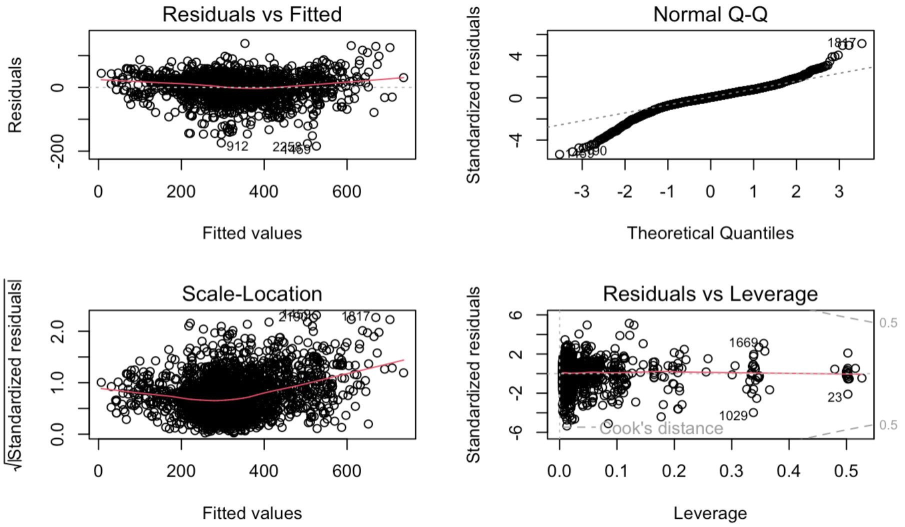

Figure a6: Diagnostic plots of W.L.S.

Table a5: summary table of LASSO model

	Estimate	**Std. Error**	**P-value**		**Estimate**	**Std. Error**	**P-value**
(Intercept)	1.026e+03	2.158e+01	<2E-16	feature_2TRUE	1.457e+01	2.53E+00	9.78E-09
mileage	-3.409e-04	1.78E-05	<2E-16	feature_3TRUE	9.040e+00	2.24E+00	0.0000544
engine_power	1.139e+00	2.914e-02	<2E-16	feature_4TRUE	1.488e+01	2.78E+00	9.18E-08
log(age)	-1.143e+02	2.93E+00	<2E-16	feature_5TRUE	2.636e+00	1.95E+00	0.17557
car_typeestate	-1.661e+01	2.479e+00	2.60E-11	feature_6TRUE	8.648e+00	2.08E+00	0.0000327
car_typesedan	1.151e+01	2.62E+00	0.0000119	feature_7TRUE	1.313e+01	4.14E+00	0.00154
car_typesuv	3.245e+01	3.021e+00	<2E-16	feature_8TRUE	2.339e+01	2.055e+00	<2E-16
feature_1TRUE	1.591e+01	1.92E+00	<2E-16				

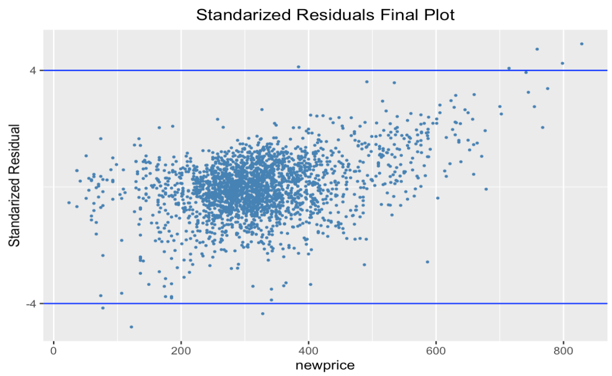

Figure a7: Standardized Residual plot of final model based on LASSO

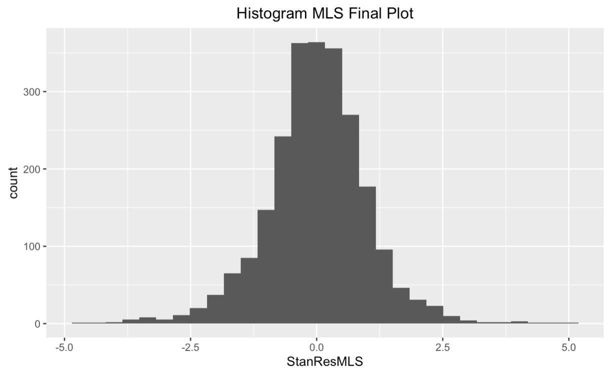

Figure a8: Histogram of standardized residuals for final model based on LASSO

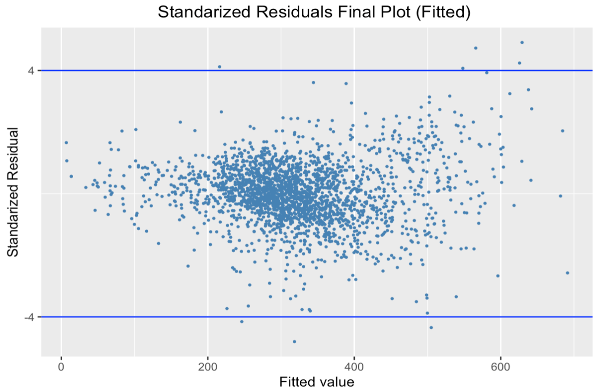

Figure a9: Standardized Residual vs. Fitted Values plot of final model based on LASSO

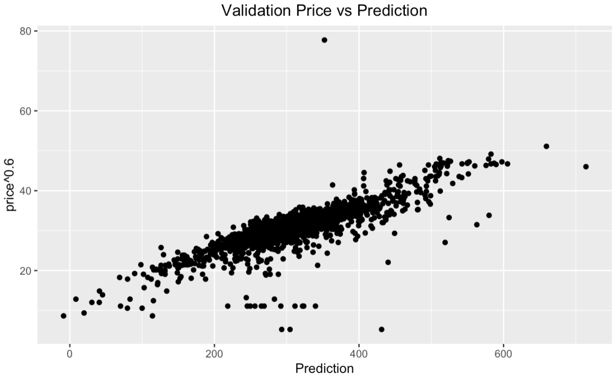

Figure a10: Plot of Validation Price^0.6 vs. Prediction for final model based on LASSO

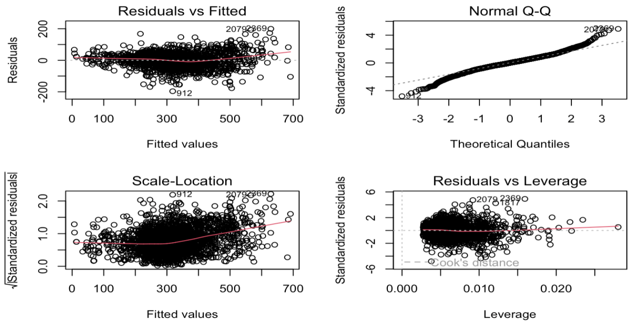

Figure a11: Diagnostic plots of final model

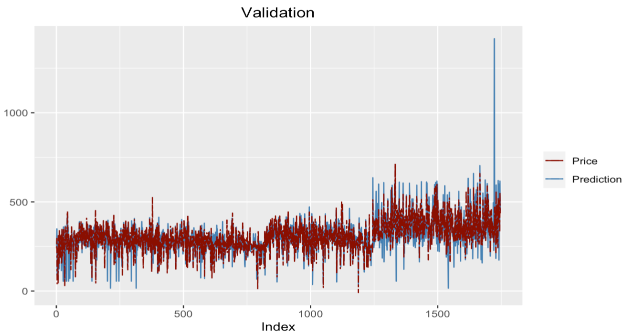

Figure a12: Validation Plot

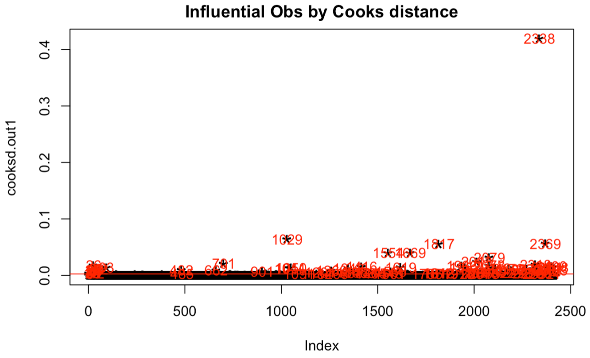

Figure a13: Cook’s Distance for Full Model Data

Omitted codes due to page limit (validation, graphs, etc,.)

```
bmw <- read.csv('~/Downloads/BMWpricing_trainingvalidation.csv',
as.is = T)
bmw$mileage <- as.numeric(bmw$mileage)
bmw$engine_power <- as.numeric(bmw$engine_power)
bmw$price <- as.numeric(bmw$price)
bmw <- bmw %>% filter(mileage >= 0, )
bmw <- bmw %>% filter(engine_power > 0)
bmw <- bmw %>% filter(paint_color != 'orange')
bmw.regis <- strptime(bmw$registration_date, "%m/%d/%Y")
bmw.sold <- strptime(bmw$sold_at, "%m/%d/%Y")
bmw$age <- difftime(bmw.sold, bmw.regis, units = 'days')
bmw$age <- as.numeric(bmw$age)
# Convert all the categorical variables to factors
categorical.colnames <- colnames(bmw)[c(2:3, 6:16)]
for (.col in categorical.colnames){
bmw[, .col] <- factor(bmw[, .col], ordered = F)}
DataSetTraining = bmw %>% filter(bmw$training == 1)
DataSetValidation = bmw %>% filter(bmw$validation == 1)
attach(DataSetTraining)
data <- data.frame(price, mileage, engine_power, age, paint_color,
car_type)
# remove orange color (only six observations)
data <- data %>% filter(paint_color != 'orange')
ggpairs(data, upper = list(continuous = wrap("points", alpha = 0.3, size
=0.15), combo = wrap("box", alpha = 0.8, size = 0.15, outlier.size=
0.05)), lower = list(continuous = wrap('cor', size = 2.75), combo =
wrap("facethist", bins = 40)), axis.text.y = element_text(size = 6))
ha <- list(others = c("coupe","subcompact","van", "convertible",
"hatchback"), estate  = c("estate"), sedan = c("sedan"), subcompact =
c("subcompact"), suv = c("suv"))
for (i in 1:length(ha))
levels(DataSetTraining$car_type)[levels(DataSetTraining$car_type)%i
n%ha[[i]]] <- names(ha)[i]
detach(DataSetTraining)
attach(DataSetTraining)
m.wo_paint <- lm(price ~ mileage + engine_power + age + car_type)
m.wt_paint <- lm(price ~ mileage + engine_power + age + car_type +
paint_color)
anova(m.wo_paint, m.wt_paint)
m.wo_cart <- lm(price ~ mileage + engine_power + age)
m.wt_cart <- lm(price ~ mileage + engine_power + age + car_type)
anova(m.wo_cart, m.wt_cart)
m.wo_model <- lm(price ~ mileage + engine_power + age + car_type)
m.wt_model <- lm(price ~ mileage + engine_power + age + car_type +
model_key)
anova(m.wo_model, m.wt_model)
levels(DataSetTraining$model_key)[levels(DataSetTraining$model_ke
y)%in%c("114", "118", "120", "123", "125", "135", "216 Active
Tourer", "216 Gran Tourer", "218", "218 Active Tourer", "218 Gran
Tourer", "220", "318 Gran Turismo", "320 Gran Turismo", "325", "325
Gran Turismo", "328", "330", "330 Gran Turismo", "335", "335 Gran
Turismo", "418 Gran Coupé", "420", "420 Gran Coupé", "425", "430",
"430 Gran Coupé", "435", "435 Gran Coupé", "518", "520 Gran
Turismo", "523", "525", "528", "530", "530 Gran Turismo", "535",
"535 Gran Turismo", "640", "640 Gran Coupé", "650", "730", "740",
"750","i3", "i8", "M135", "M235", "M3", "M4", "M550", "X4", "X5
M", "X5 M50", "X6", "X6 M", "Z4")] <- "other_key"
detach(DataSetTraining)
attach(DataSetTraining)
for(i in 1:8){
ft_variable <- paste("feature", i, sep = "_")
ggplot(DataSetTraining, aes(x = ft_variable, y = price)) +
geom_boxplot() + xlab("ft_variable") + ylab("Price Sold in 2018")}
formula_wo_ft <- 'price ~ mileage + engine_power + log(age) +
car_type + model_key'
m.wt_model <- lm(price ~ mileage + engine_power + log(age) +
car_type + model_key)
```

```
formula_mls <- paste(formula_wo_ft, paste(sapply(c(1:6,8),
function(x){paste0("feature_",x)}), collapse = " + "), sep = " + ")
m.mls <- lm(as.formula(formula_mls), data = DataSetTraining)
StanResMLS <- rstandard(m.mls)
dataMLS <- data.frame(price,StanResMLS)
ggplot() + geom_point(data=dataMLS, aes(x=price, y=StanResMLS,
color = "MLS"), size = 0.5) + geom_hline(yintercept=4,color='blue') +
geom_hline(yintercept=-4, color='blue') + scale_color_manual(name =
element_blank(), labels = c("MLS"), values = c("steelblue")) + labs(y =
"Standarized Residual") + ggtitle("Standarized Residuals MLS Plot") +
 scale_y_continuous(breaks = c(-4, 4, 10, 20))
# remove 1 outlier
out1 <- is.nan(StanResMLS) | abs(StanResMLS) > 20
sum(out1)
detach(DataSetTraining)
DataSetTraining <- DataSetTraining[!out1, ]
attach(DataSetTraining)
Fitted = fitted(m.mls)
dataMLSFitted <- data.frame(Fitted,StanResMLS)
# MLS Stan. Res. vs Fitted plot
ggplot() +
  geom_point(data=dataMLSFitted, aes(x=Fitted, y=StanResMLS,
color = "MLS"), size = 0.5) +
  geom_hline(yintercept=4,color='blue') + geom_hline(yintercept=-4,
color='blue') +
  scale_color_manual(name = element_blank(), labels = c("MLS"),
values = c("steelblue")) +
  labs(y = "Standarized Residual") + labs(x = "Fitted value") +
  ggtitle("Standarized Residuals MLS Plot (Fitted) ") +
  scale_y_continuous(breaks = c(-4, 4, 10, 20))
#normal qq plot
p<- ggplot(data.frame(StanResMLS), aes(sample = StanResMLS)) +
  ggtitle("QQ MLS Plot")
p + stat_qq() + stat_qq_line() + labs(y = "Observed Quantiles", x =
'Theoretical Quantiles')
# Histogram of MLS
ggplot(data = data.frame(StanResMLS), aes(x = StanResMLS)) +
geom_histogram(bins = 30) +
  ggtitle("Histogram MLS Plot")
boxcox(m.mls,plotit=TRUE)
# Transformation
formula_mls_trans <- gsub('price', '(price)^0.6', formula_mls)
m.trans <- lm(as.formula(formula_mls_trans), data = DataSetTraining)
summary(m.trans)
m.trans1 <- lm(formula_mls_trans, data = DataSetTraining, weights =
engine_power)
summary(m.trans1)
plot(m.trans1)
# LASSO Variable Selection:
attach(DataSetTraining)
lasso <- DataSetTraining[-c(1, 2, 3, 6, 7, 8, 18, 19, 20, 21)]
lm <- lm(price^0.6 ~ mileage + engine_power + log(age) + car_type +
feature_1 + feature_2 +
         feature_3 + feature_4 + feature_5 + feature_6 + feature_7 +
feature_8 , data=DataSetTraining)
summary(lm)
lasso.matrix <- model.matrix(lm)
training <- createDataPartition(price^0.6, p=0.8, list=F)
tX <- lasso.matrix[training, ]
vX <- lasso.matrix[-training,]
train_y <- as.matrix((price^0.6)[training])
lasso.model <- cv.glmnet(tX, train_y, alpha=1, nfolds=10)
lasso.coefs <- coef(lasso.model, s="lambda.min")
final <- lm(price^0.6 ~ mileage + engine_power + log(age) + car_type
+ feature_1 + feature_2 + feature_3 + feature_4 + feature_5 +
feature_6 + feature_7 + feature_8, data=DataSetTraining)
summary(final)
avPlots(final)
```