Project1

Yuanjian Zhou, Ziying Cheng, Yite Zhu

2019/11/11

## 1 Motivation of the Study

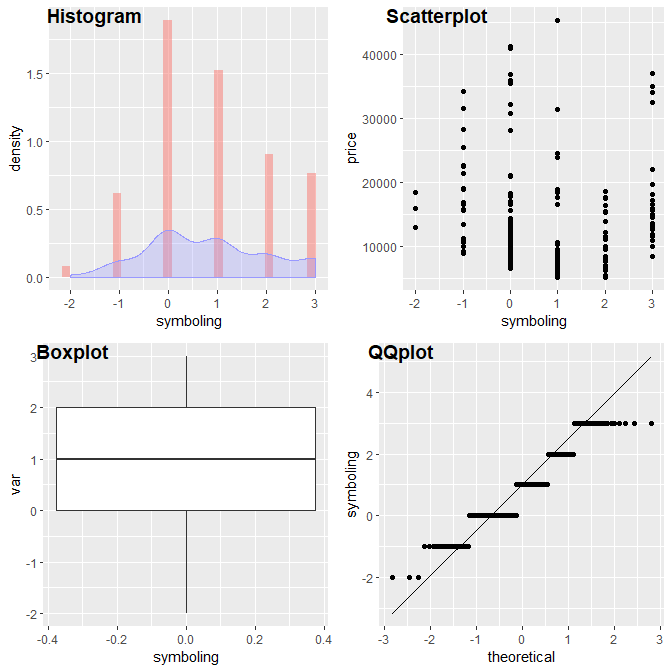
The report focuses on modeling of car prices based on a dataset surrounding the price information and all sorts of features of numerous brands of vehicles in US market. An accurate prediction of car prices is our main concern, because it greatly facilitates both the price-setting decisions of companies and the car-purchasing decisions of consumers. The idea is that cars, in theory, are priced fair according to some key factors. The dataset contains some essential elements consumers care about, together with some key parameters vehicle manufacturers put an emphasis on during producing and marketing, and should be able to provide relatively useful information about the price of any certain new car given these relevant information.

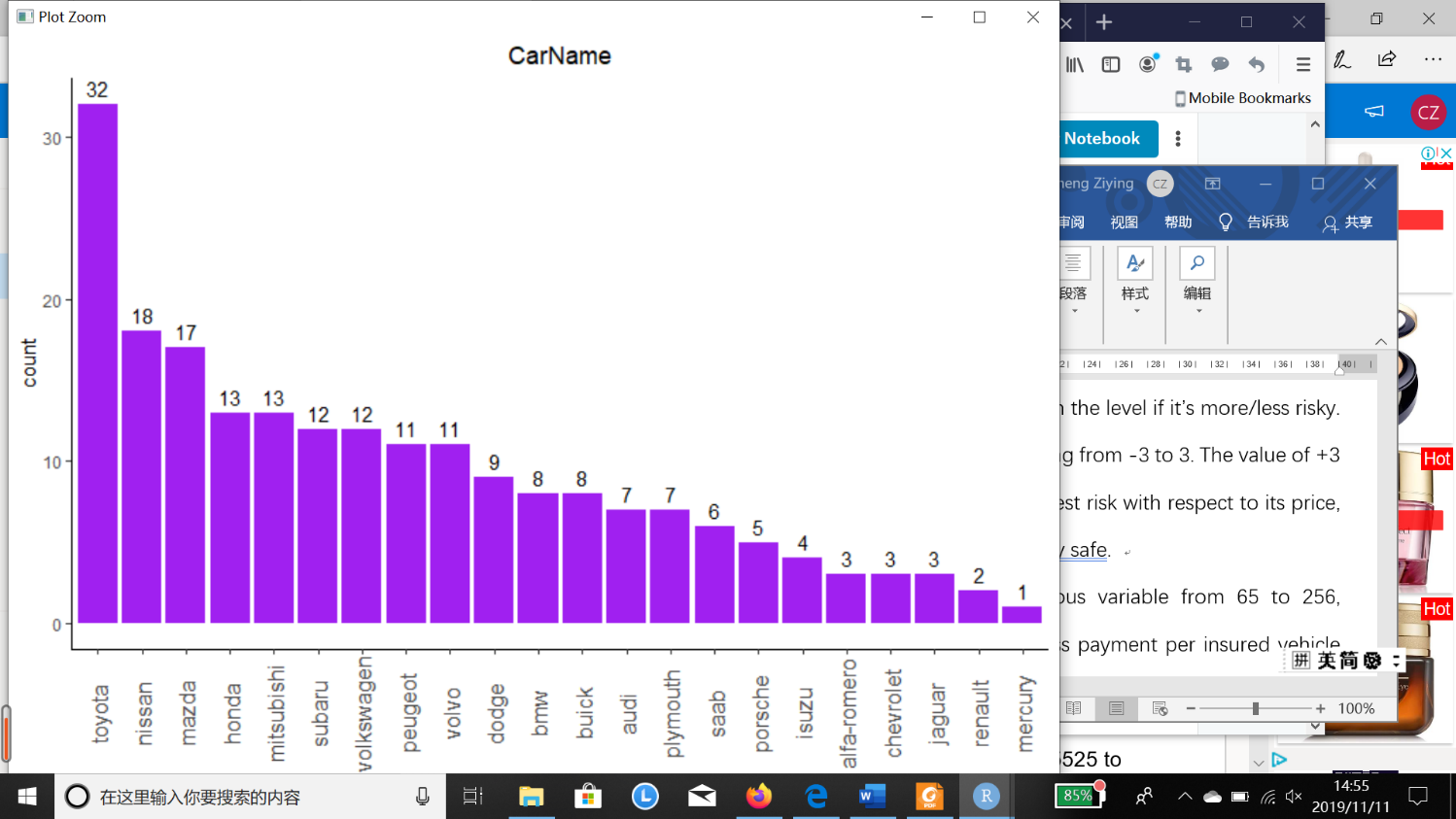
The estimation of marginal effects of these factors, should be interpretable to make better purchasing decisions, and should be able to make a comprehensive prediction of the final price. At the end of the day, consumer decisions collapsed greatly into comparing prices, considering financing and other elements concerning economic status of the family, according to Prieto and Caemmerer (2013). Thus, a certain degree of data transformation to satisfy linearity and normality is preferred, let alone that in the following section, tests suggest log transformation is good enough.

We expect a high considering the market of cars are relatively efficient nowadays since the information relating to the product is large enough to reflect major fluctuations in prices. Also, cars are liquid for selling. Studies have shown that second-hand car market reflects quality in product prices readily (GINTER, YOUNG and DICKSON, 1987). The arbitrage opportunities are rare and the prices should be fair in general. Therefore, given that appropriate independent variables are included in the model and forms of variables are chose to serve the prediction power, the unexplained uncertainty in the target variable should be comparably small.

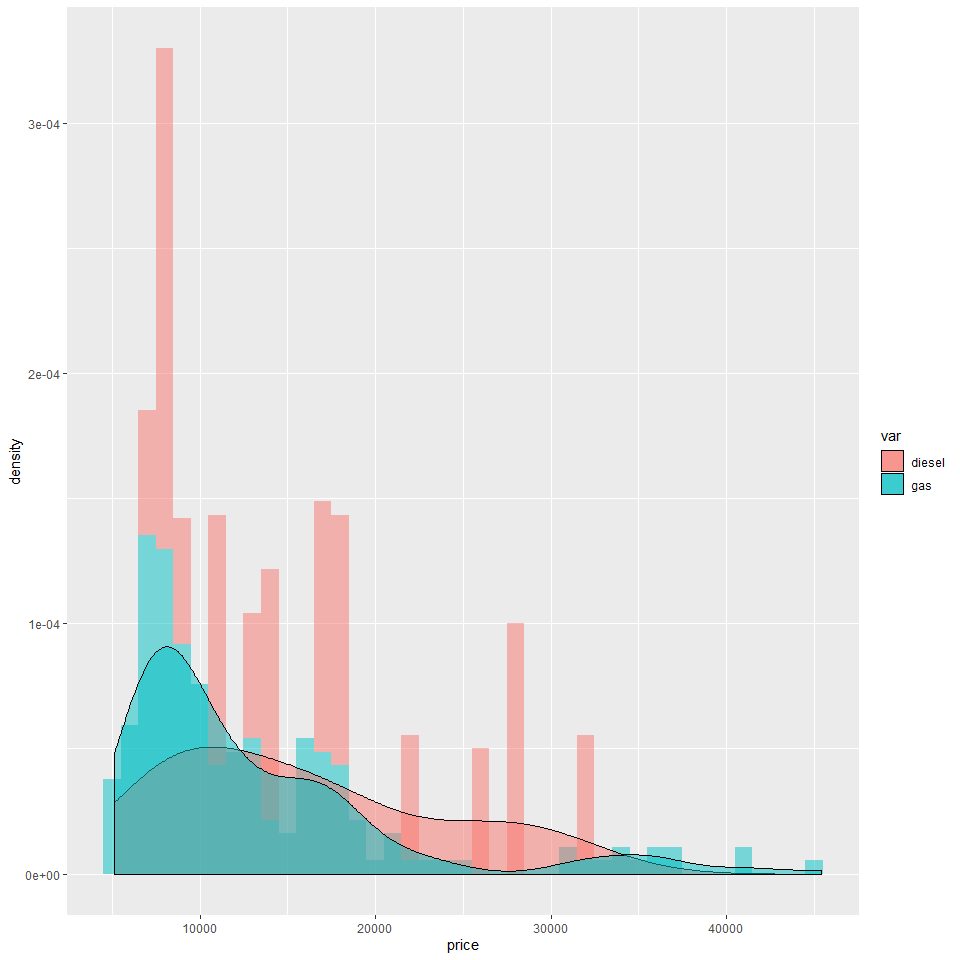
## 2. Data Description

(i) There are 26 variables in the dataset, including 15 continuous, 1 integer and 10 nominals. Regarding price as the target, we set all. The followings are the descriptive analysis of variables accordingly.

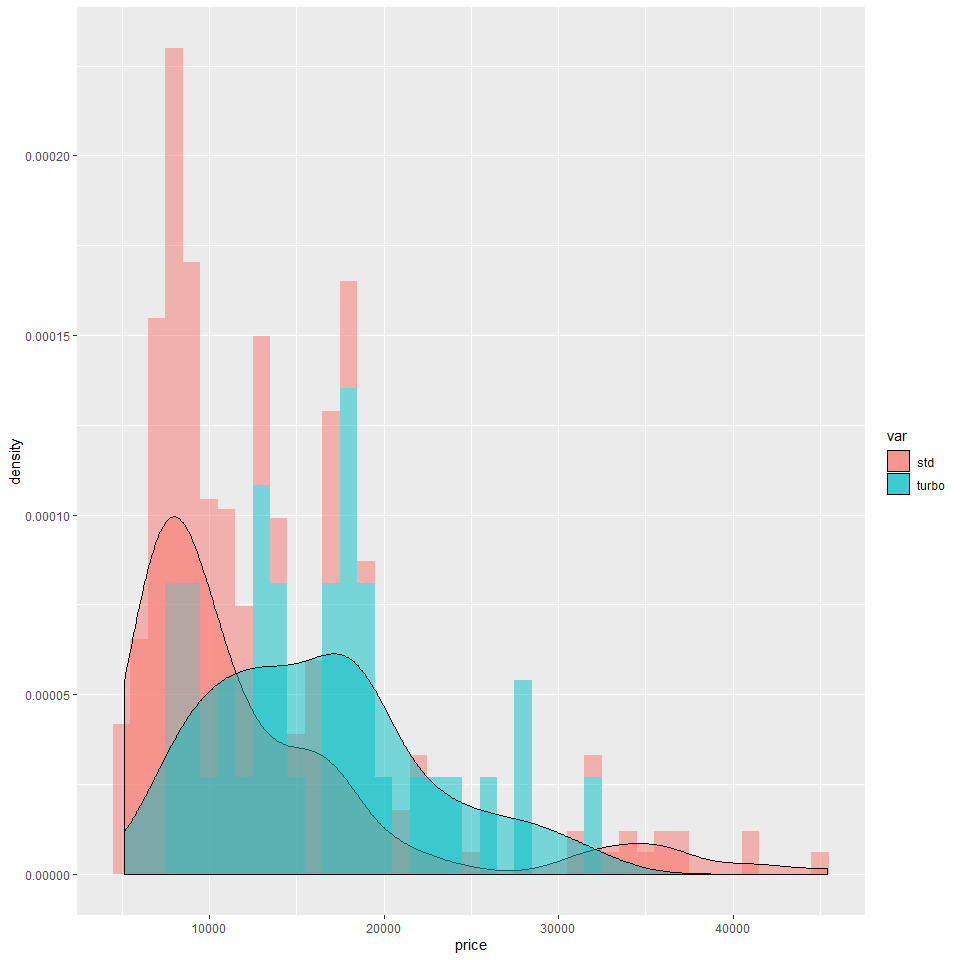
1. Car\_ID: The integer variable refers to the unique id of each observation, counting from 1-205, demonstrating the number of observations but not associated with any explaining power to the target. This is a variable that will be removed at first.
2. symboling: a variable who gets its name from the symboling process, which is named by actuaries. The variable corresponds to the degree to which the auto mobile is riskier than its price indicates. It means that cars are initially assigned a risk factor symbol associated with its price, which will be adjusted by moved up/down the level if it’s more/less risky. Thus, it is a categorical value, ranging from -3 to 3. The value of +3 demonstrated the car has the highest risk with respect to its price, which -3 indicates the auto is pretty safe. In the dataset, the most risky level is +3, the safest level is -2, and the median level is 1. The correlation with target “price” is -0.08, referring to no close relationship.
3. CarName: It is natural to suspect that, on average, prices differ with brands. It is later summarized, using brand average price, into low-brand, medium-brand, and top-brand, average prices of which drop sequentially. To be specific, we divide by price interval 0-10000, 10000-20000, and 20000-50000 respectively. There are 22 kinds of brand totally and their frequency in observations range from 1 to 32.



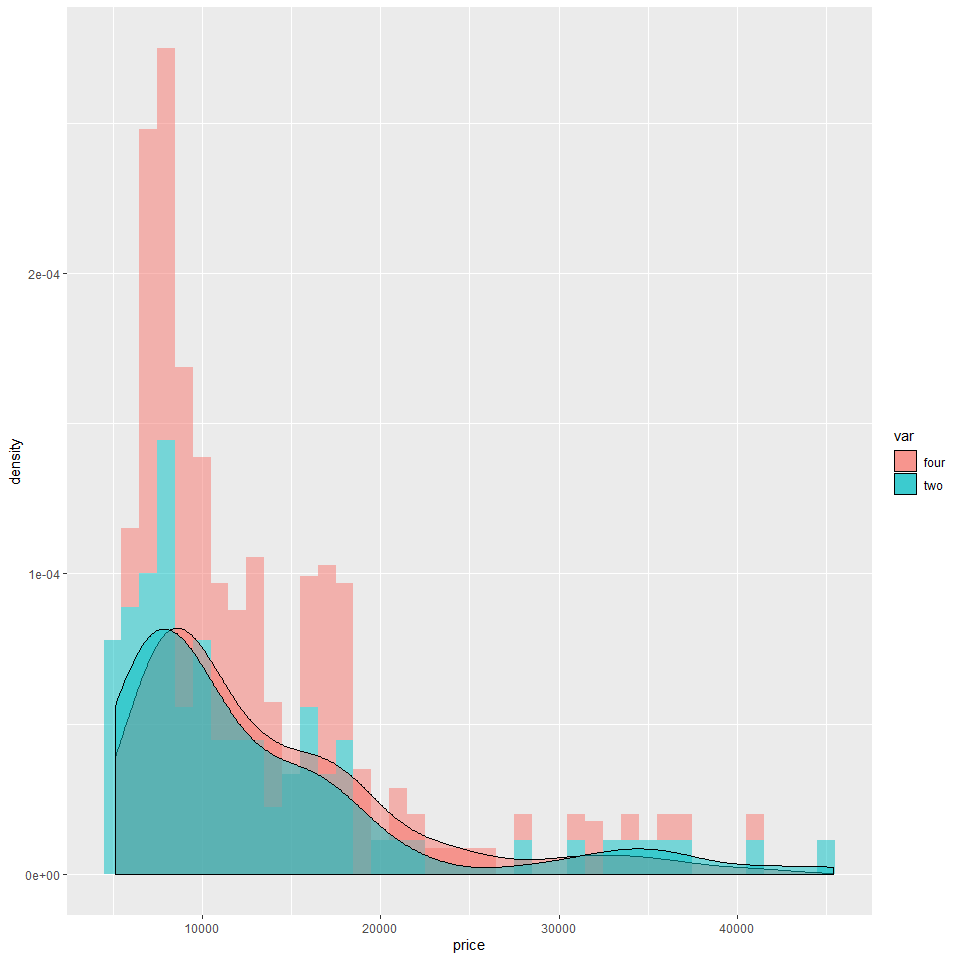
1. Fueltype: it’s a categorial variable, dividing the fuel type in two: gas or diesel. To buy a car is one thing, but to maintain a car is another thing. Consumers tend to have preference over types of energy the vehicle uses, creating a demand shock to change the overall prices. According to the graph below, where blue represents the price of gas cars, and red represents that of dissel-fueled cars, gas car has higher concentration at the lower price, while there are more diesel cars at mid-level price. There seems to be difference in the correlation to price between the types.



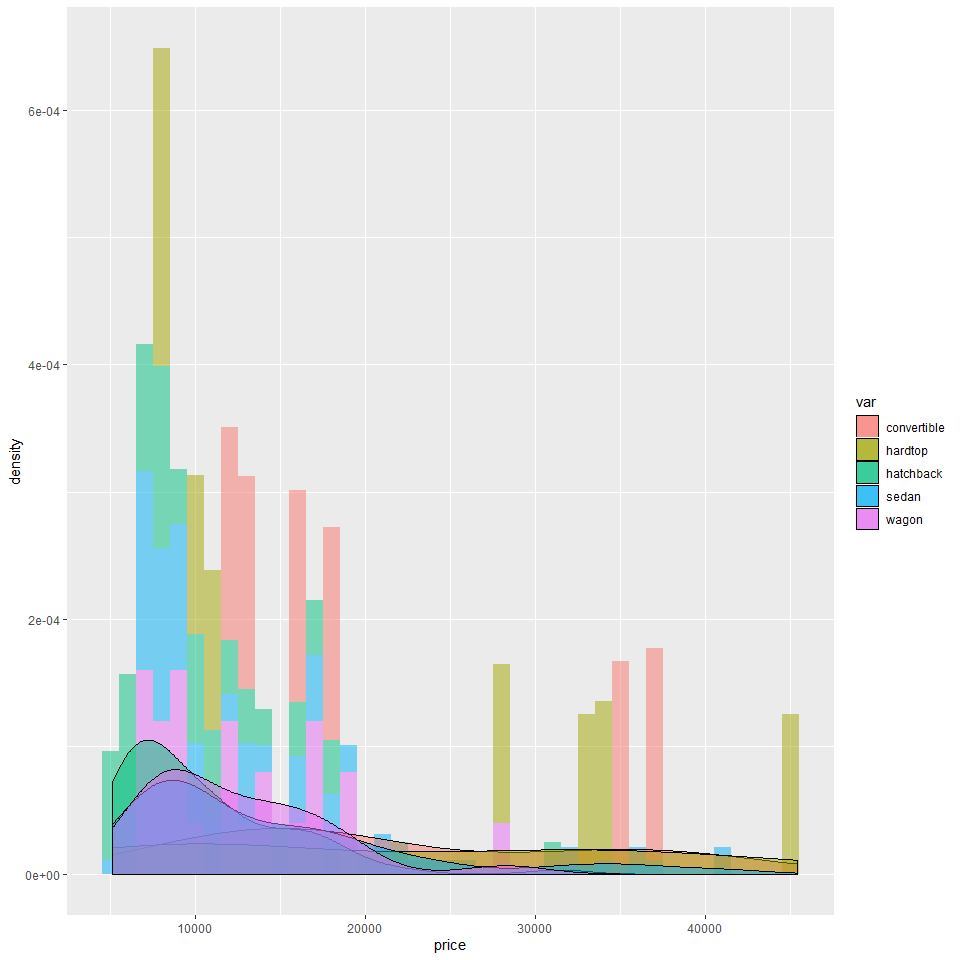
1. Aspiration: the categorical variable refers to the aspiration used in a car, std or turbo. Again, this is about preference and usability. If plotting a density distribution of aspiration with respect to price, we can see std (the red one) has right skewed distribution, and the mean of std cars is a round 7500 while the mean price of turbo car (the blue one) is larger than 17000, associated with a big difference.



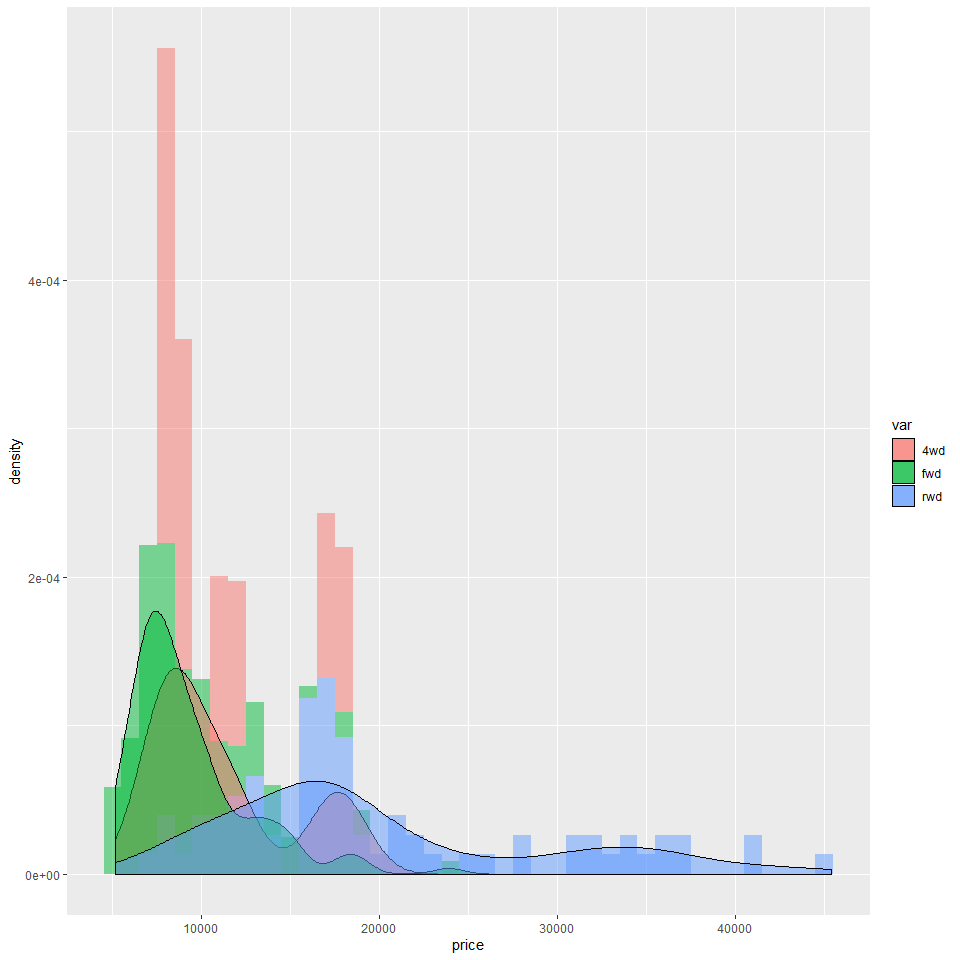
1. Doornumber: the categorical variable shows the number of doors in a car with four (the red one) or two (the blue one). According to the density plot, there seems to be no significance between the two classification.



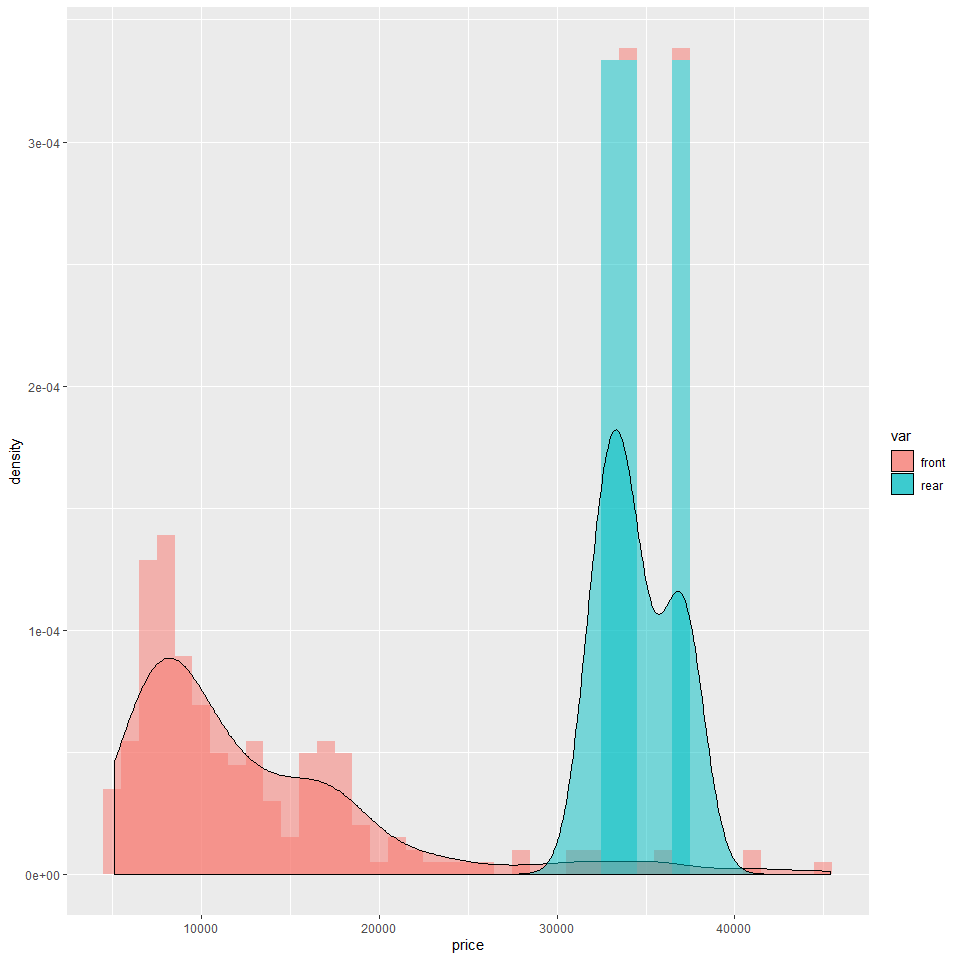
1. Carbody: the categorical variable describes the pattern of car body and the classifications are hardtop, wagon, sedan, hatchback, convertible. The convertible and hardtop have higher mean price than the others but the standard deviation is really large. So we may take a closer look by calculating accurately.



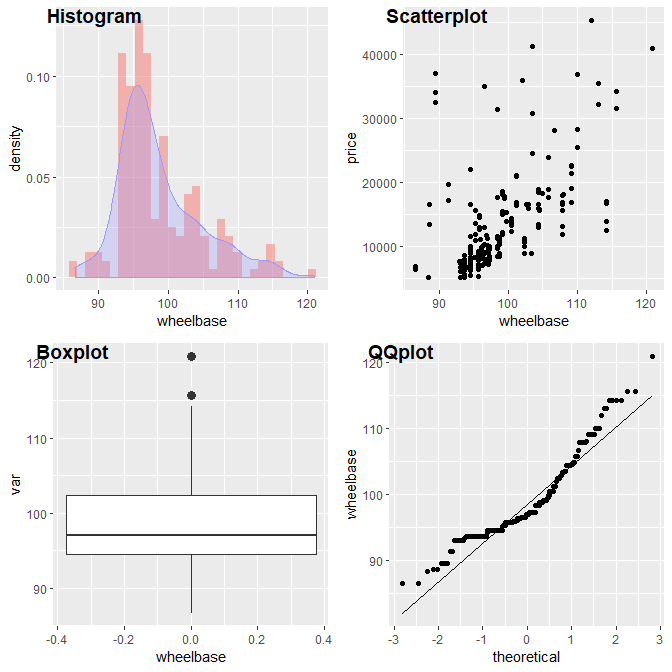
1. Drivewheel: the categorical variable demonstrates the type of drive wheel, 4wd, fwd, rwd. Type of drivewheel is concerned with functionality, which fascinates car fans, resulting in demand shocks. Among them, fwd car(green)has lowest mean price while the 4wd(red) has two peak and rwd car(blue) has the highest mean price and highest standard deviation.



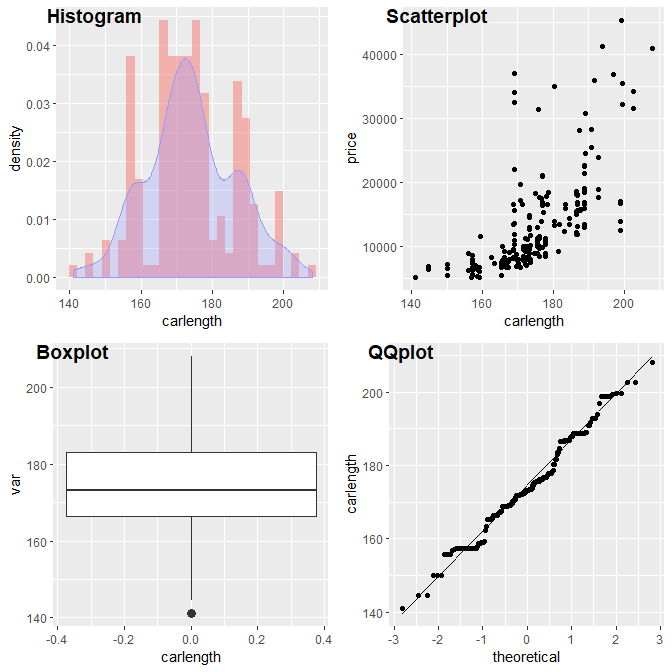
1. Enginelocation: There are two types of engine locations among the observations, front or rear. Location of car engine is another functionality of concern for some consumers. There seems to be significant difference between the price of two price. The front group(red) concentrates at the low price, with mean around 10000 while nearly all the rear car(blue) valued more than 30000.



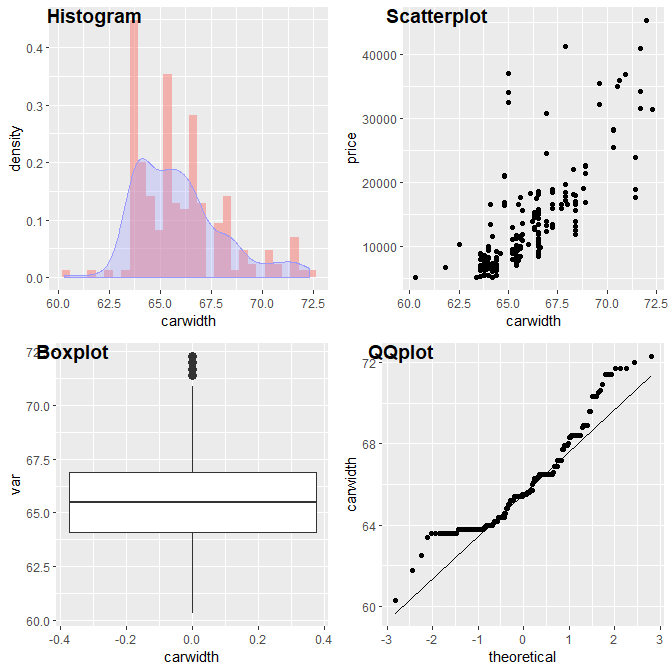
1. Wheelbase: It is a numeric variable describes the distance between a car front wheel to its back wheel. The minimum value is 86.80 and the maximum is 120.90, the mean is 98.76 and the mean is 98.76. According to the graphs below, we can see it is not strictly normal distributed, with several outliers and a trend of right skewed distribution. The correlation with price is 0.58 suggesting the close positive relationship with the dependent variable.



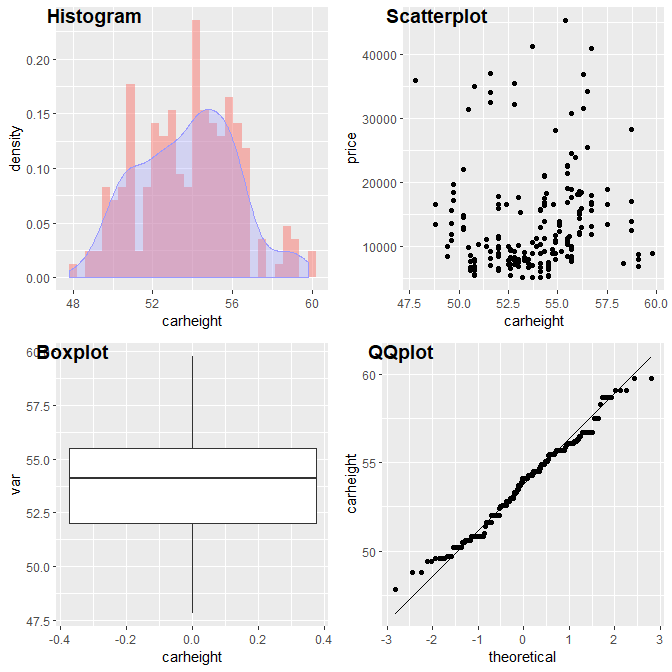
1. Carlength: The length of car ranges from 141.1 to 208.1, nearly normal distributed, concentrating at 174 and is strongly correlated with price.



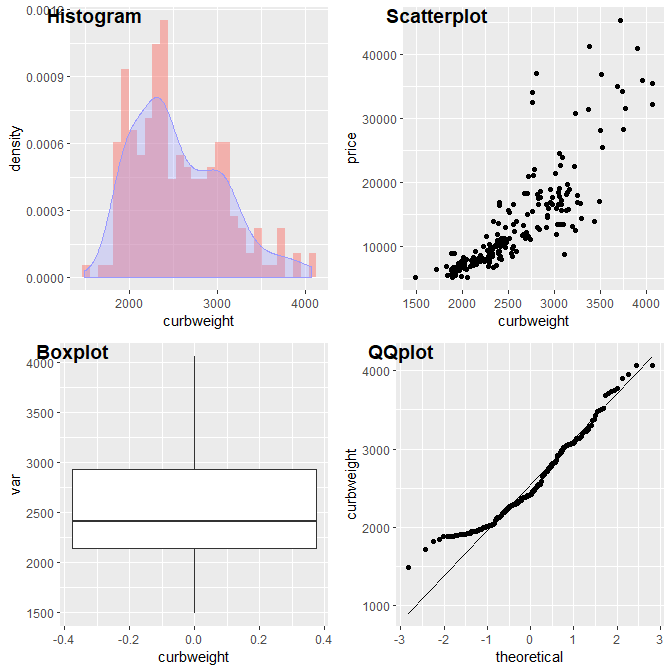
1. Carwidth: The width of car notes the width of the car, ranging from 141.1 to 208.1. The distribution is not bad in normality with mean 174.0 however it still tends to be kind of right skewed distribution with several outliers. The positive correlation with target is quite obvious according to the scatterplot.



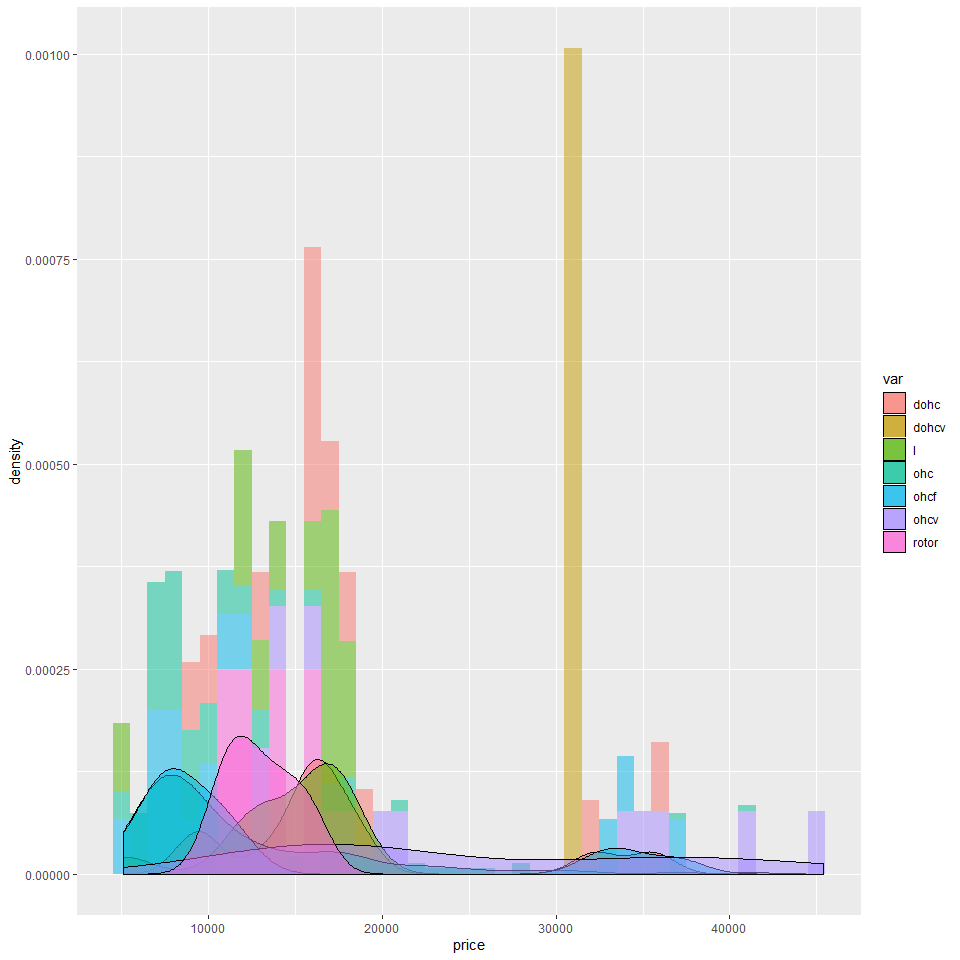
1. Carheight: The height of car ranges from 47.8 to 59.8 in the dataset. The distribution is like normal distribution with mean equal to 54.1. According to scatterplot, the correlation is not so obvious between the carheight and target.



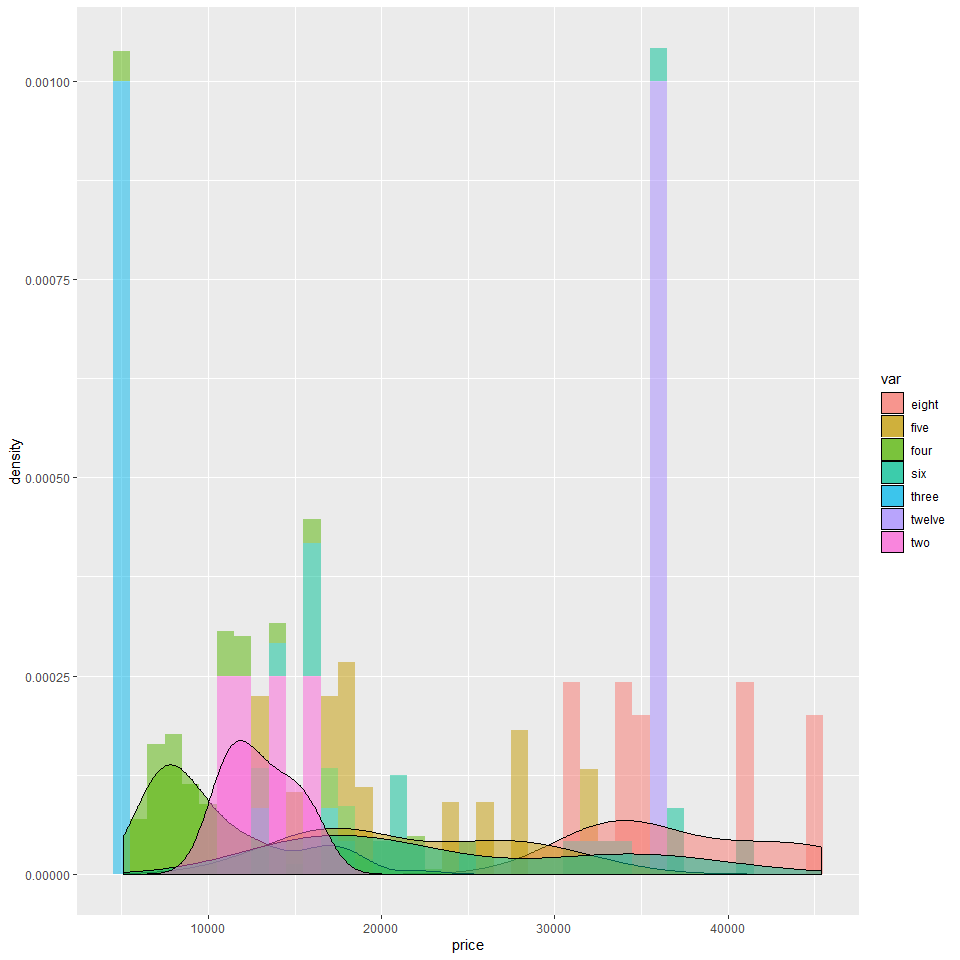
1. Curbweight: It is a numeric variable reporting the weight of a car without occupants or baggage. Ranging from 1488 to 4066, it is not normally distributed largely because of a thin tail. The scatterplot indicates a strong positive correlation between the variable and target.



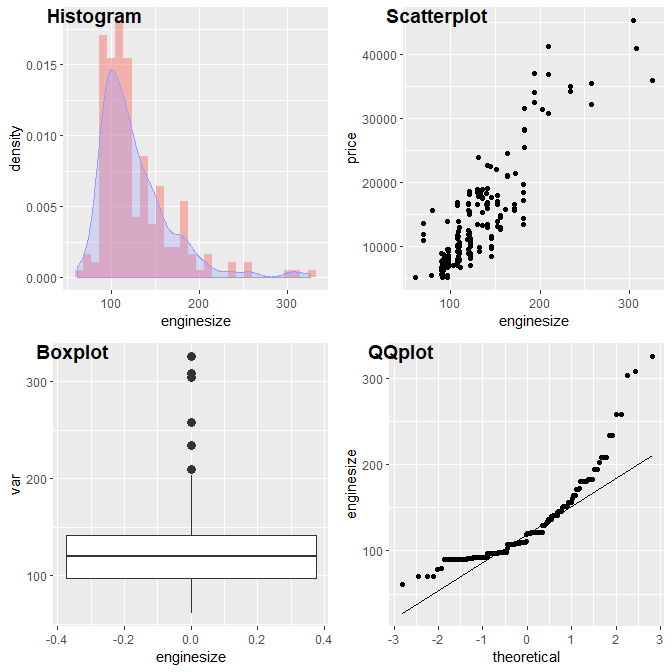
1. Enginetype: Type of engine includes dohc, dohcv, l, ohc, ohcf, ohcv, rotor. There are kind of difference of car price between them. Types of engine. The following two describes engine parameters as well. Though car features can be overwhelming for beginners, engines, horsepower are often straightforward for consumers.



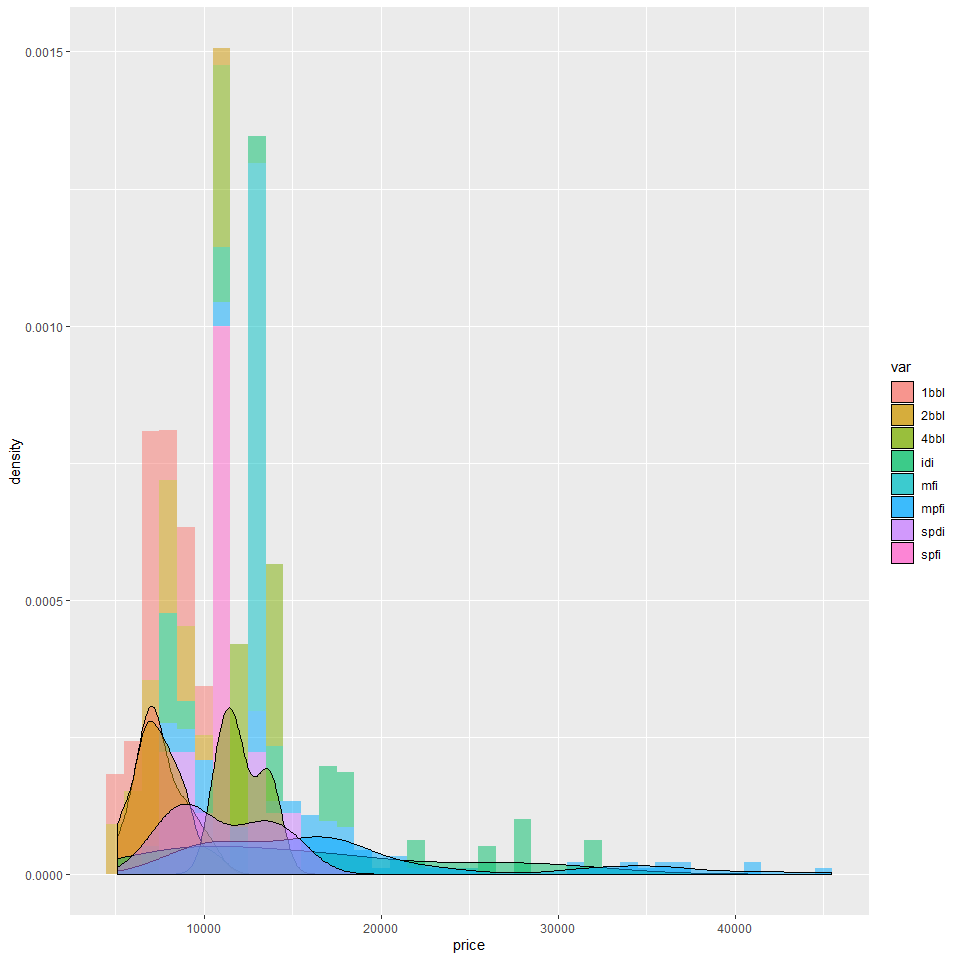
1. Cylindernumber：The categorical variable says the number of cylinders placed in the car, with possible value of two, three, four, five, six, eight, twelve. Roughly, It seems that the group price means obey the order: three< four < two <eight



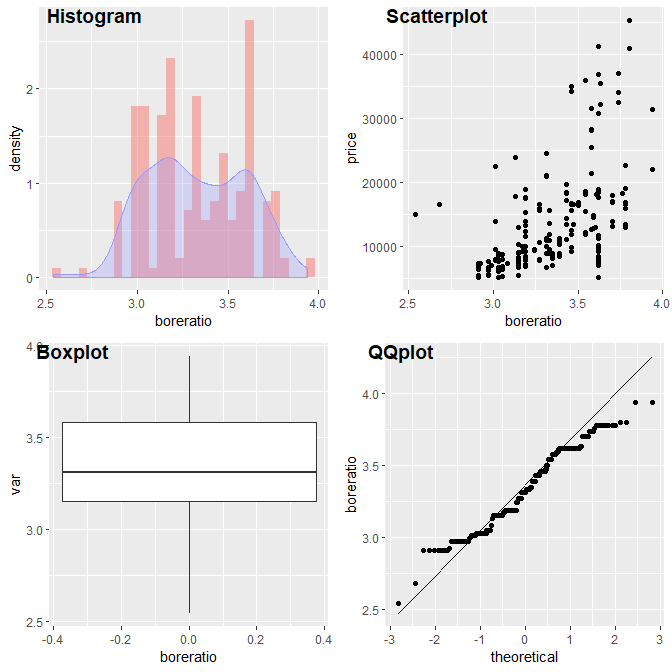
1. Enginesize：The size of car ranges from 61 to 326 with mean equal to. From the density plot and QQ-plot, we can see the distribution is far from normal distribution with right skewed distribution and several outliers. However, we can see it is closely correlated with the target following scatterplot.



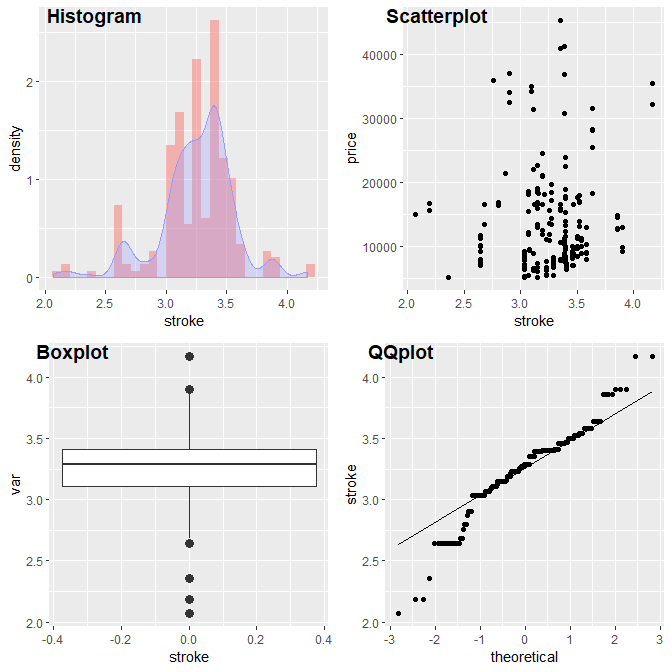
1. Fuelsystem: The categorical variable refers to the fuel system of the car, including 1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi. The difference of price among different group can not be easily pointed out at a glance of the graph.



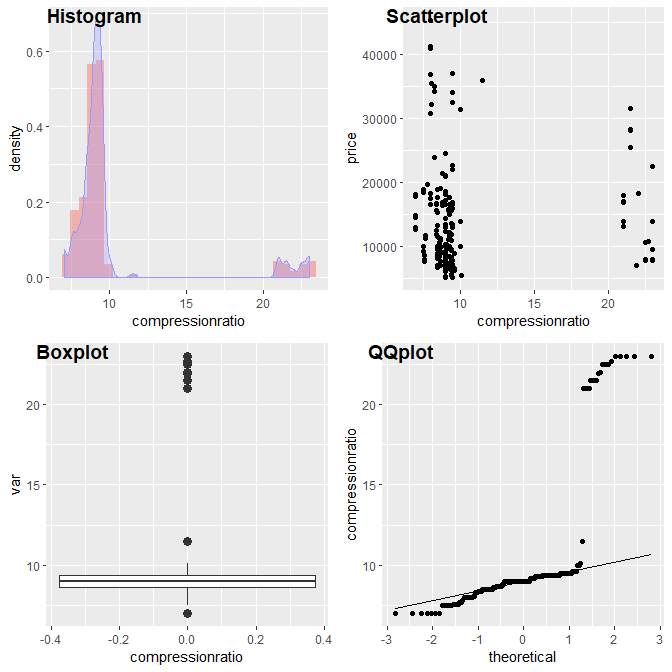
1. Boreratio: The stroke-boreratio of car is associated with the features of engine. It ranges from 2.54 to 3.94. From the density plot and QQplot, the thin tail of distribution may drive it away from strict normality.



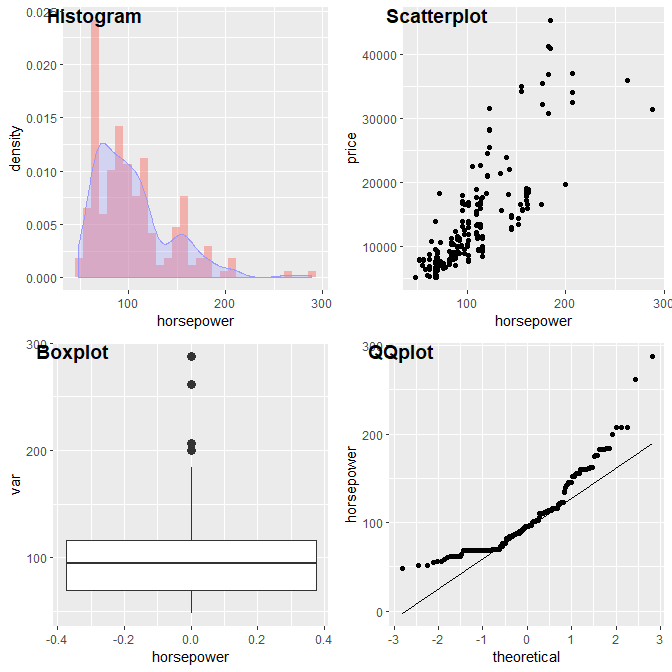
1. Stroke: The numeric variable describes stroke or volume inside the engine with range from 2.07 to 4.17. It can hardly have normal distribution because of outliers and a trend of left skewed distribution.



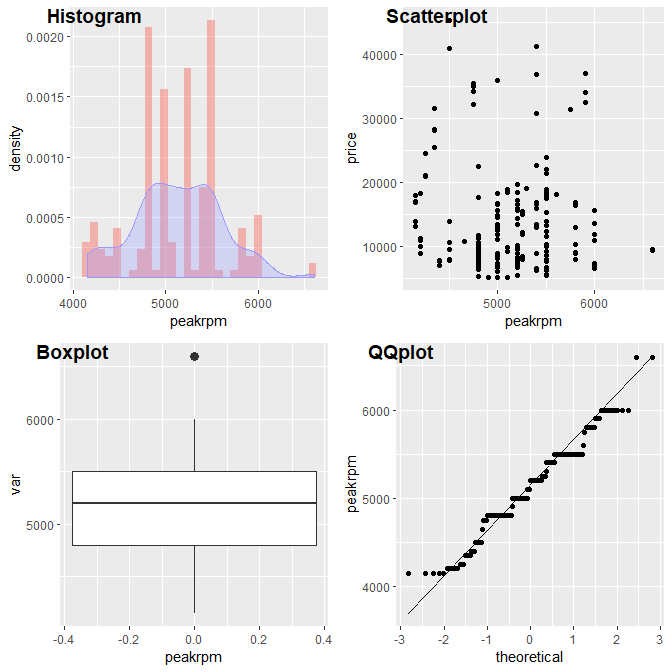
1. Compressionratio: compression ratio of car (Numeric) According to the density plot, it has leptokurtic and a number of outliers, tending to be right skewed distribution, which drives it away from normal distribution.



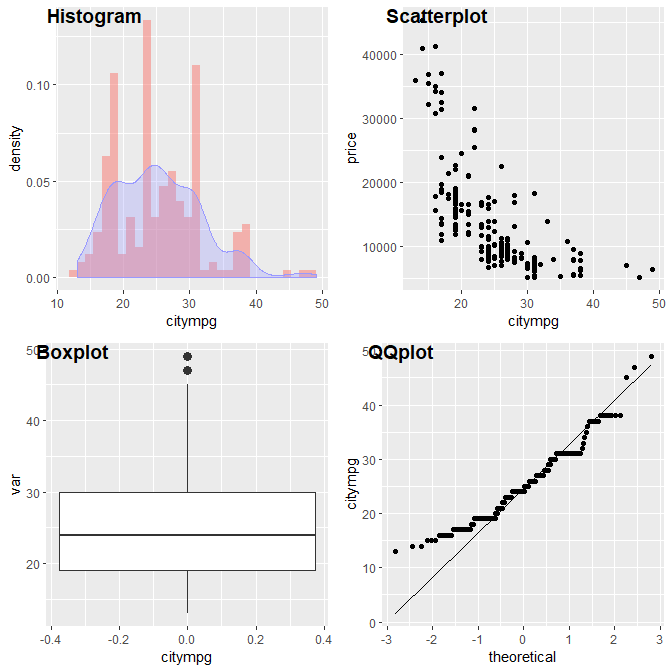
1. Horsepower: The numerical variable stands for the power of car engine, ranging from 48 to 288. Its mean is 104.1. We can see the clear trend of right skewed distribution with quantities of outliers, demonstrating that the transformation might be necessary. The positive relationship between the target and the variable is rather obvious.



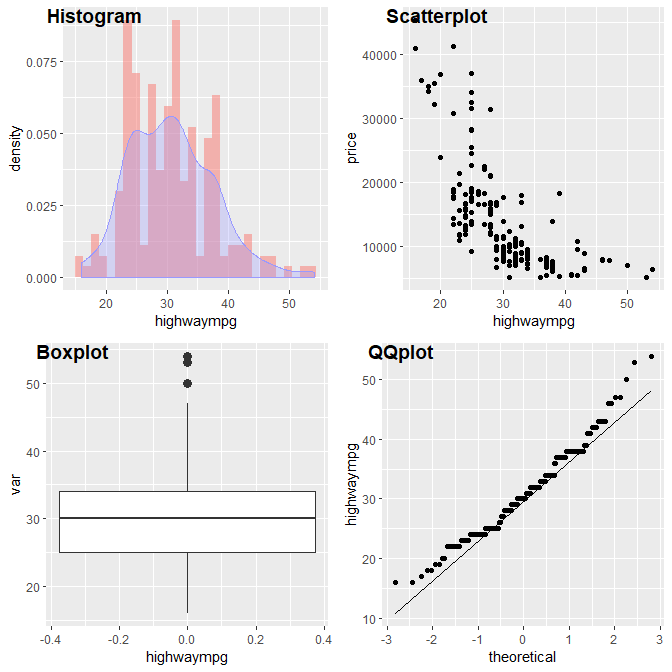
1. Peakrpm: The car peak rpm is a numeric variable, ranging from 4150 to 6600. Like the next variable, this number indicates the cost of keeping a car. The distribution is not bad in normality with few outliers. The scatterplot doesn’t generate obvious trend.



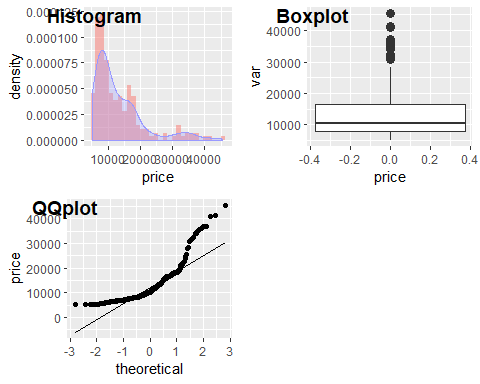
1. citympg: The numeric variable tells the number of miles that a car can travel using certain amount of fuel in city. It ranges from 13 to 49 with mean equal to 24. We can hardly say it is normal distributed because of thin tail and right skewed distribution. The negative correlation is quite obvious according to scatterplot.



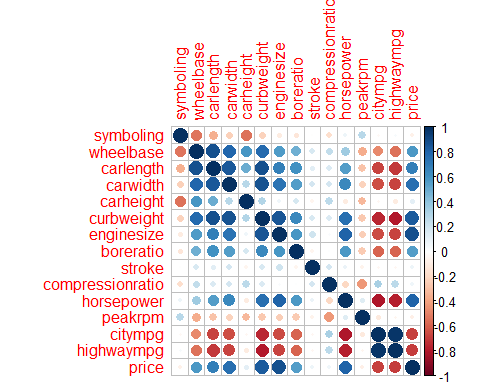
1. Highwaympg: The numeric variable tells the number of miles that a car can travel using certain amount of fuel on highway. It ranges from 16 to 54 with mean equal to 31. We can hardly say it is normal distributed because of thin tail and right skewed distribution. The negative correlation is quite obvious according to scatterplot.



1. Price (Dependent variable): The numeric variable reports the car price, ranging from 5,118 to 45,400 . The mean is 13,276 and standard deviation is 7,989. It has obvious right skewed distribution with many potential outliers.



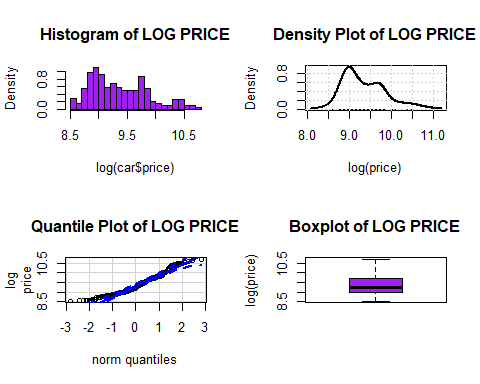
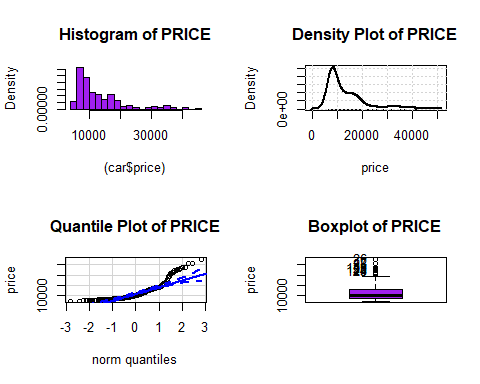
(ii) According to the correlation plot, we can easily find that enginesize, curbweight, carwidth, carlength have strong positive correlation with the dependent variable price, while the citympg and highwaympg is closely negatively associated with the price. The deep colors in the correlation matrix plot imply regression models with high explanatory power.



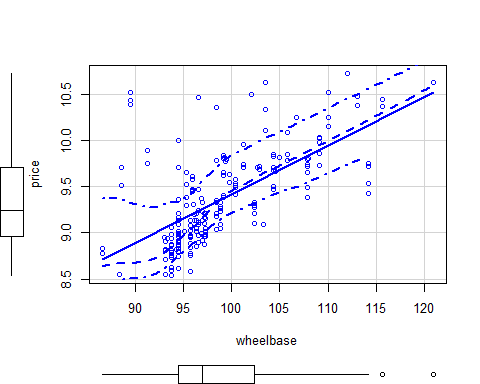
(iii) Tranformation

We do observe the similarities of certain variables that could cause serious multicollinearity and should be subject to further test. But before that, let us look into each variables to conduct power transformation to satisfy linearity assumptions.

To cope with nonlinearities, firstly, we log-transform our target variable, the price. According to the following boxplots and qqplots, we observe abnormal skewness and peaks in price. To solve the problem, we apply log-transform without costing the interpretability of our model. The result looks more normal and less skewed.

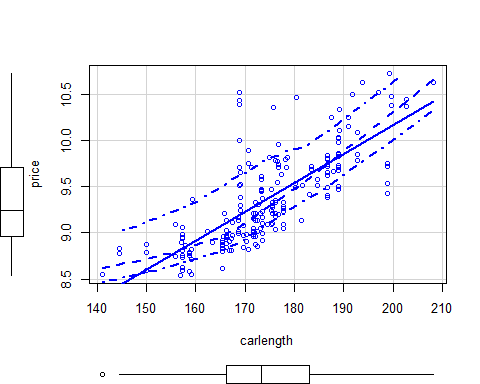
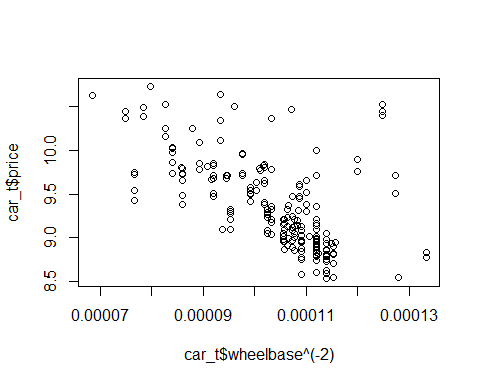


We can start transforming other variables to satisfy linearity assumptions now. Apply transformation methods to all continuous independent variables. From the following results, we pick out those not suitable for transformation and those suitable for transformation.

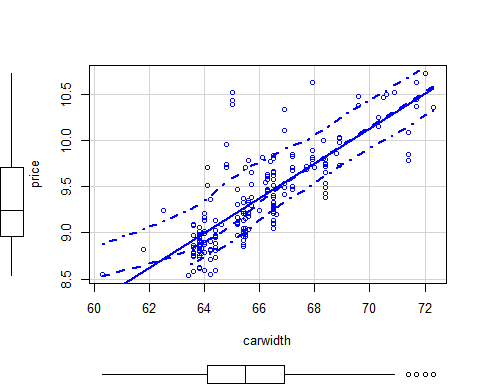


## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 -3.2248 -2 -5.0198 -1.4298  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 13.55647 1 0.00023149  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 23.47673 1 1.2643e-06

## LRT df pval  
## LR test, lambda = (-2) 1.915291 1 0.16638



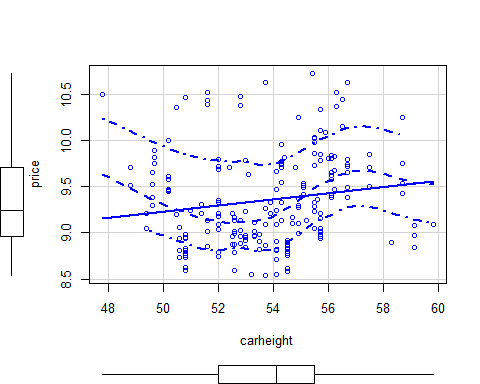
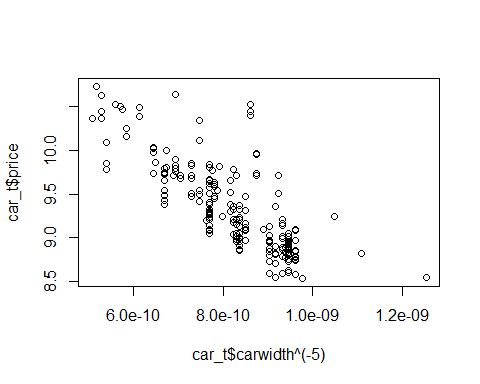
## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 1.1036 1 -0.2153 2.4225  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 2.698352 1 0.10045  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 0.02372912 1 0.87758



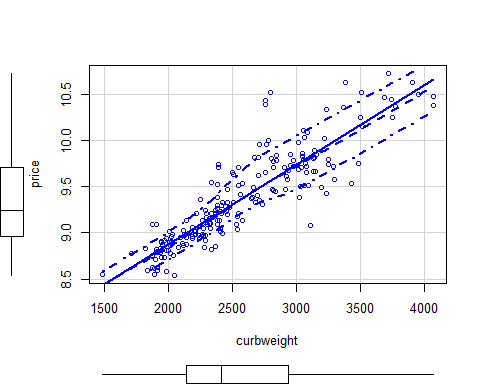
## Warning in estimateTransform.default(X, Y, weights, family, ...):  
## Convergence failure: return code = 52

## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 -5.2814 -5.28 -5.3399 -5.2229  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 14.12497 1 0.00017106  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 20.01518 1 7.683e-06

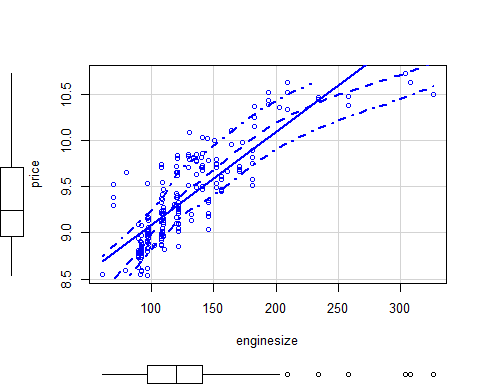
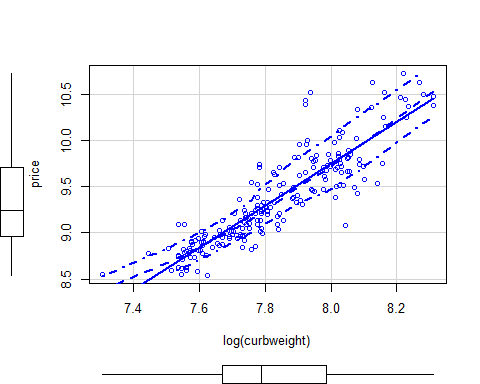
## LRT df pval  
## LR test, lambda = (-5) 0.03644398 1 0.8486



## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 0.6225 1 -2.0849 3.3299  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 0.2024757 1 0.65273  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 0.0748093 1 0.78446

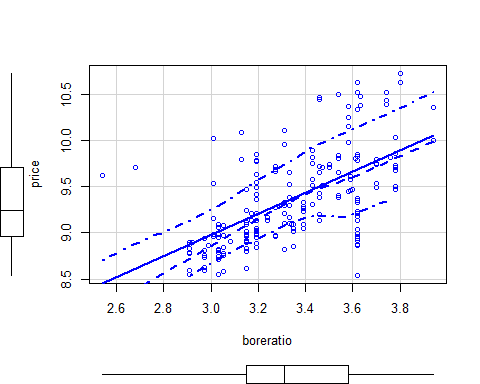
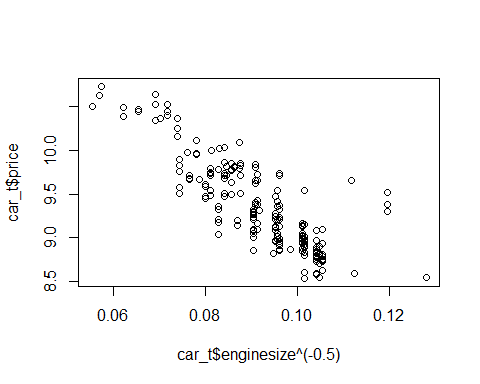


## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 0.0297 0 -0.4107 0.4702  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 0.01745683 1 0.89489  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 18.96786 1 1.3294e-05

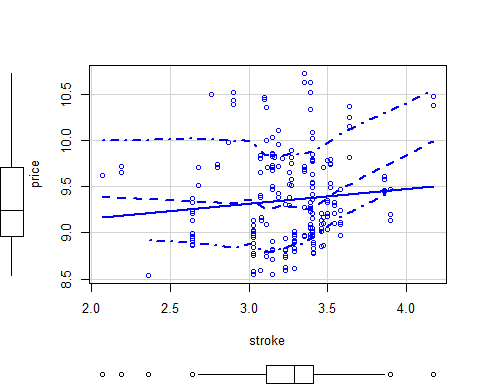


## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 -0.3327 -0.5 -0.6132 -0.0523  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 5.699271 1 0.016972  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 100.1357 1 < 2.22e-16

## LRT df pval  
## LR test, lambda = (-0.5) 1.327011 1 0.24934

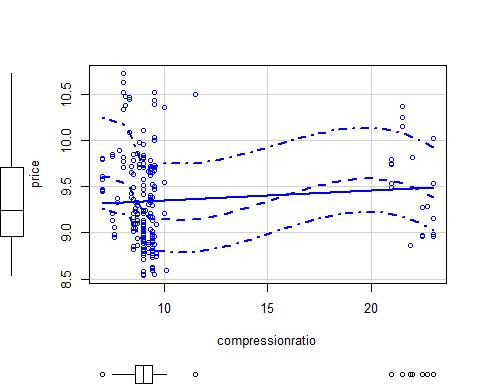
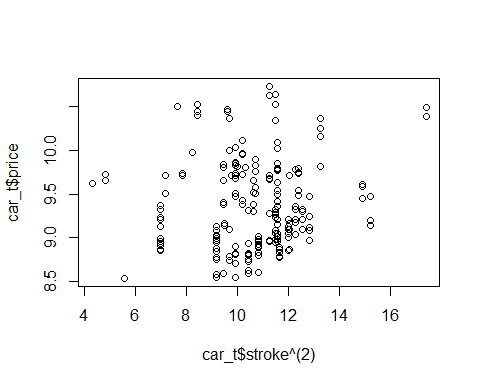


## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 1.3575 1 -0.107 2.822  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 3.399376 1 0.065221  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 0.2306629 1 0.63103



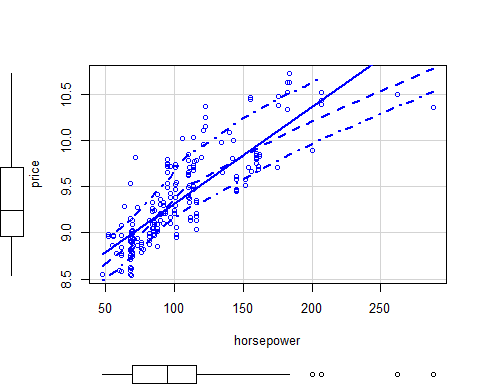
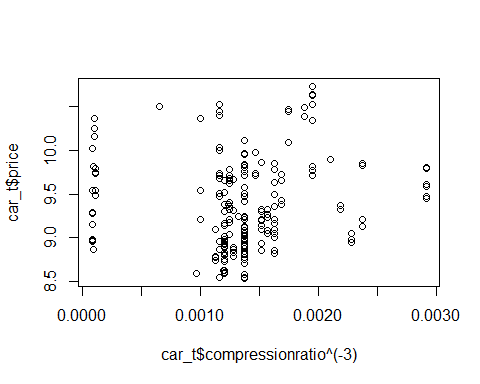
## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 2.5912 2 1.6763 3.506  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 33.18718 1 8.3701e-09  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 12.15847 1 0.00048865

## LRT df pval  
## LR test, lambda = (2) 1.630912 1 0.20158

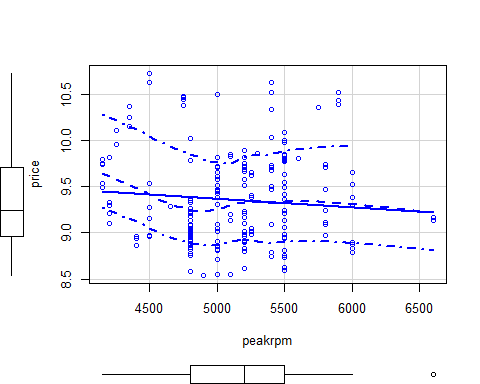
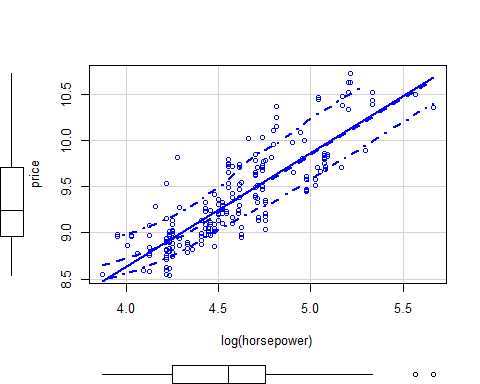


## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 -3.1017 -3.1 -3.5757 -2.6277  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 200.9263 1 < 2.22e-16  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 354.7097 1 < 2.22e-16

## LRT df pval  
## LR test, lambda = (-3) 0.1786258 1 0.67256

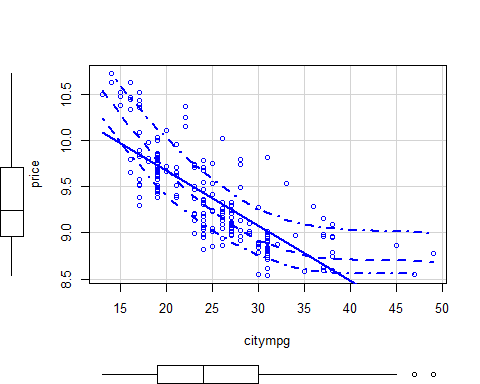
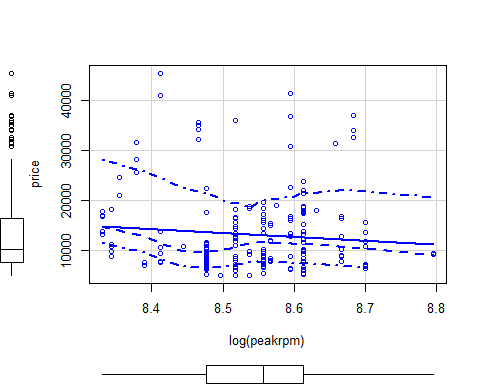


## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 -0.2552 0 -0.5339 0.0234  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 3.334221 1 0.067853  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 87.41857 1 < 2.22e-16



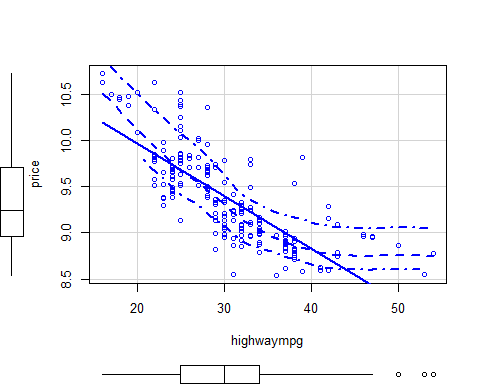
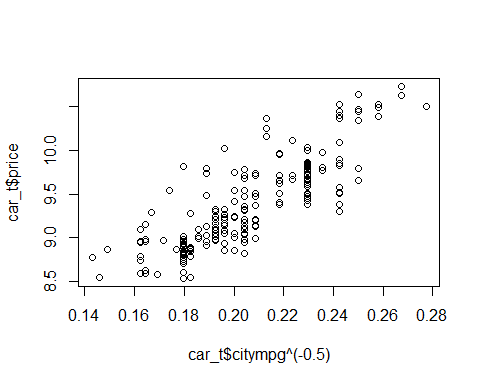
## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 0.6656 1 -0.5486 1.8799  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 1.14784 1 0.284  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 0.2921156 1 0.58887

## LRT df pval  
## LR test, lambda = (0) 1.14784 1 0.284

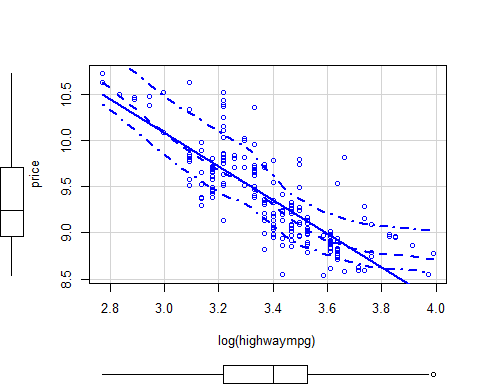


## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 -0.5474 -0.5 -0.9266 -0.1681  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 7.931598 1 0.0048579  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 61.09461 1 5.4401e-15

## LRT df pval  
## LR test, lambda = (-0.5) 0.05988531 1 0.80668



## bcPower Transformation to Normality   
## Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd  
## Y1 -0.3239 0 -0.7024 0.0546  
##   
## Likelihood ratio test that transformation parameter is equal to 0  
## (log transformation)  
## LRT df pval  
## LR test, lambda = (0) 2.749626 1 0.097277  
##   
## Likelihood ratio test that no transformation is needed  
## LRT df pval  
## LR test, lambda = (1) 42.16003 1 8.4102e-11



To sum up, applying log transform to the following continual variables proved to be useful in the later sections: curbweight, horsepower, peakrpm, highwaympg. The rest of the continuous variables are either better off left unchanged or should go through higher power or negative power transformation. Test results suggest that it is acceptable to apply log-transform to the four we selected and is good enough to restore linearity of the model. But transform other than log-transform would harm the interpretability of our model greatly. That is the reason why we keep other variables intact.

## [1] "Transform for curbweight:"

## LRT df pval  
## LR test, lambda = (0) 0.01745683 1 0.89489

## [1] "Transform for horsepower:"

## LRT df pval  
## LR test, lambda = (0) 3.334221 1 0.067853

## [1] "Transform for peakrpm:"

## LRT df pval  
## LR test, lambda = (0) 1.14784 1 0.284

## [1] "Transform for highwaympg:"

## LRT df pval  
## LR test, lambda = (0) 2.749626 1 0.097277

## 3. The Model

Fortunately, the data, after examination, contains no missing values.

Now we are ready for model building. To start with, we build a baseline model by including all main effects according to correlation plots, apply Mallows Cp, and select main effects again. Firstly, rule out the variables that forms perfect multicollinearity, and rule out the terms that has variation inflation factor (VIF) larger than 6 in a step-by-step fashion.

By ensuring a low VIF for each remaining variable, the model should be much more immune to multicollinearity.

## GVIF Df GVIF^(1/(2\*Df))  
## symboling 16.258358 5 1.321623  
## fueltype 3.809502 1 1.951795  
## aspiration 2.832243 1 1.682927  
## doornumber 3.459373 1 1.859939  
## carbody 9.388963 4 1.323053  
## drivewheel 6.891337 2 1.620227  
## enginelocation 3.277389 1 1.810356  
## wheelbase 13.496856 1 3.673807  
## carlength 14.251656 1 3.775137  
## carwidth 8.175947 1 2.859361  
## carheight 3.994561 1 1.998640  
## enginetype 87.737589 6 1.451885  
## boreratio 3.906396 1 1.976460  
## stroke 2.418687 1 1.555213  
## horsepower 15.251159 1 3.905273  
## peakrpm 2.431531 1 1.559337  
## highwaympg 11.092083 1 3.330478  
## brandlevel 5.889035 2 1.557798

From there, we use Mallows Cp to identify main effects to keep.

Thus, we have our first baseline model. But it is still subject to further improvement, so we apply the previous process on to this model to eliminate less relevant variables.

See the model results below:

##   
## Call:  
## lm(formula = price ~ symboling1 + symboling3 + fueltypegas +   
## carbody + drivewheelfwd + enginelocationrear + wheelbase +   
## carlength + carwidth + boreratio + horsepower + highwaympg +   
## brandlevel, data = car\_t)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.31710 -0.08370 -0.01076 0.07788 0.42540   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.886861 0.819468 4.743 4.16e-06 \*\*\*  
## symboling1 -0.066782 0.025251 -2.645 0.008870 \*\*   
## symboling3 0.045595 0.041347 1.103 0.271554   
## fueltypegas -0.135311 0.041307 -3.276 0.001256 \*\*   
## carbodyhardtop -0.263691 0.075715 -3.483 0.000618 \*\*\*  
## carbodyhatchback -0.258207 0.061506 -4.198 4.16e-05 \*\*\*  
## carbodysedan -0.210320 0.064901 -3.241 0.001412 \*\*   
## carbodywagon -0.254501 0.070089 -3.631 0.000364 \*\*\*  
## drivewheelfwd -0.109050 0.027157 -4.016 8.58e-05 \*\*\*  
## enginelocationrear 0.322417 0.100338 3.213 0.001545 \*\*   
## wheelbase 0.005507 0.004205 1.310 0.191966   
## carlength 0.004437 0.002322 1.911 0.057562 .   
## carwidth 0.042217 0.010436 4.045 7.64e-05 \*\*\*  
## boreratio -0.154530 0.051416 -3.005 0.003016 \*\*   
## horsepower 0.569067 0.066791 8.520 5.21e-15 \*\*\*  
## highwaympg -0.114940 0.106716 -1.077 0.282839   
## brandlevelmedium-grade 0.087776 0.023525 3.731 0.000253 \*\*\*  
## brandleveltop-grade 0.431109 0.042393 10.169 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1307 on 187 degrees of freedom  
## Multiple R-squared: 0.9383, Adjusted R-squared: 0.9326   
## F-statistic: 167.2 on 17 and 187 DF, p-value: < 2.2e-16

From here we go through again the iterated process of ruling out irrelevant variables. Just to clarify, ruling out dummy variable terms means to pool the non-present groups together, which is normal in car prices because some features simply does not make much difference for regular consumers.

The following is our second baseline model. We expect an improvement in performance.

And the model result is as follow:

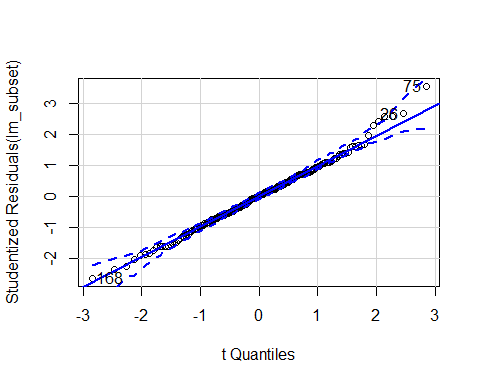
##   
## Call:  
## lm(formula = price ~ symboling1 + fueltypegas + carbody + drivewheelfwd +   
## enginelocationrear + carlength + carwidth + boreratio + horsepower +   
## brandlevel, data = car\_t)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.31243 -0.08347 0.00158 0.08384 0.43504   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.159673 0.430984 7.331 6.38e-12 \*\*\*  
## symboling1 -0.076960 0.024282 -3.169 0.001780 \*\*   
## fueltypegas -0.125105 0.038507 -3.249 0.001370 \*\*   
## carbodyhardtop -0.282575 0.072836 -3.880 0.000144 \*\*\*  
## carbodyhatchback -0.261702 0.059584 -4.392 1.86e-05 \*\*\*  
## carbodysedan -0.227102 0.060014 -3.784 0.000207 \*\*\*  
## carbodywagon -0.266225 0.065942 -4.037 7.84e-05 \*\*\*  
## drivewheelfwd -0.123908 0.025621 -4.836 2.73e-06 \*\*\*  
## enginelocationrear 0.316234 0.097986 3.227 0.001472 \*\*   
## carlength 0.006824 0.001852 3.685 0.000298 \*\*\*  
## carwidth 0.047867 0.009746 4.911 1.94e-06 \*\*\*  
## boreratio -0.166464 0.051168 -3.253 0.001350 \*\*   
## horsepower 0.602148 0.052614 11.445 < 2e-16 \*\*\*  
## brandlevelmedium-grade 0.087011 0.023139 3.760 0.000226 \*\*\*  
## brandleveltop-grade 0.433206 0.042532 10.185 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1313 on 190 degrees of freedom  
## Multiple R-squared: 0.9367, Adjusted R-squared: 0.9321   
## F-statistic: 201 on 14 and 190 DF, p-value: < 2.2e-16

And the improvement in AIC and BIC is direct and apparent.

## df AIC  
## lm\_subset0 19 -233.2154  
## lm\_subset 16 -234.2282

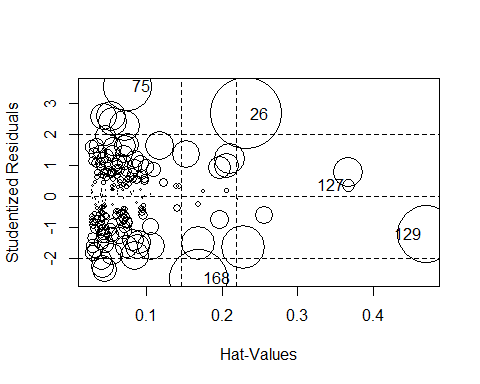
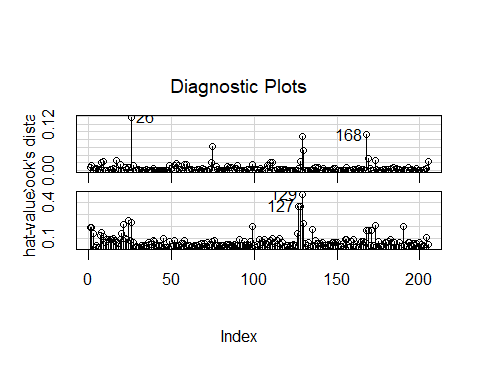
## df BIC  
## lm\_subset0 19 -170.0782  
## lm\_subset 16 -181.0600

To further improve the model, we looked into outliers, leverage points, and influential points and tried to remove them.



## [1] 26 75 168

## No Studentized residuals with Bonferroni p < 0.05  
## Largest |rstudent|:  
## rstudent unadjusted p-value Bonferroni p  
## 75 3.547212 0.00049099 0.10065



## StudRes Hat CookD  
## 26 2.6623685 0.23148896 0.137920041  
## 75 3.5472118 0.07429897 0.063459311  
## 127 0.3437046 0.36752615 0.004597753  
## 129 -1.2260583 0.47010462 0.088672027  
## 168 -2.6513894 0.16898624 0.092369860

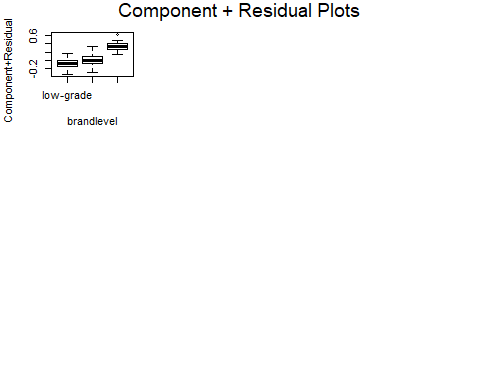
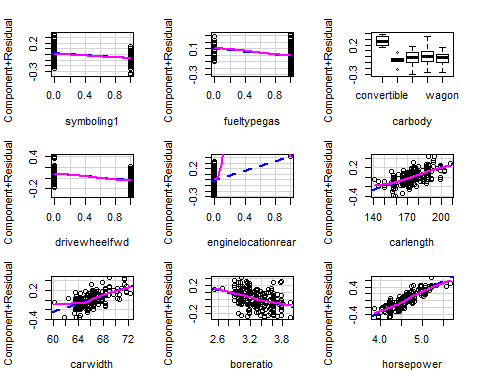
We have a new model removing all outliers, leverage points, and influential points from sample. The model result is as follow:

## Call: lm(formula = price ~ symboling1 + fueltypegas + carbody +  
## drivewheelfwd + enginelocationrear + carlength + carwidth + boreratio +  
## horsepower + brandlevel, data = car\_t, subset = -c(75, 26, 53, 127, 129,  
## 168))  
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -80458.17 6863.75 -11.722 < 2e-16 \*\*\*  
## symboling1 -567.28 389.28 -1.457 0.146743   
## fueltypegas -540.59 615.38 -0.878 0.380838   
## carbodyhardtop -5101.57 1403.33 -3.635 0.000360 \*\*\*  
## carbodyhatchback -4153.04 987.86 -4.204 4.09e-05 \*\*\*  
## carbodysedan -3826.12 998.87 -3.830 0.000175 \*\*\*  
## carbodywagon -4489.96 1088.12 -4.126 5.58e-05 \*\*\*  
## drivewheelfwd -1336.23 405.82 -3.293 0.001190 \*\*   
## enginelocationrear 9661.32 2428.92 3.978 1.00e-04 \*\*\*  
## carlength 76.42 29.33 2.606 0.009915 \*\*   
## carwidth 969.57 153.78 6.305 2.08e-09 \*\*\*  
## boreratio -2901.23 804.47 -3.606 0.000400 \*\*\*  
## horsepower 6515.63 828.21 7.867 3.00e-13 \*\*\*  
## brandlevelmedium-grade 562.20 366.10 1.536 0.126344   
## brandleveltop-grade 10641.74 673.33 15.805 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard deviation: 2061 on 184 degrees of freedom  
## Multiple R-squared: 0.9287  
## F-statistic: 171.2 on 14 and 184 DF, p-value: < 2.2e-16   
## AIC BIC   
## 3618.23 3670.93

Residual standard deviation: 0.122 on 184 degrees of freedom Multiple R-squared: 0.9418 F-statistic: 212.6 on 14 and 184 DF, p-value: < 2.2e-16 AIC BIC -256.19 -203.49 To see if multicollinearity still exists, we resort to VIF. The result is that very few to none were detected, which is a good thing for our model.

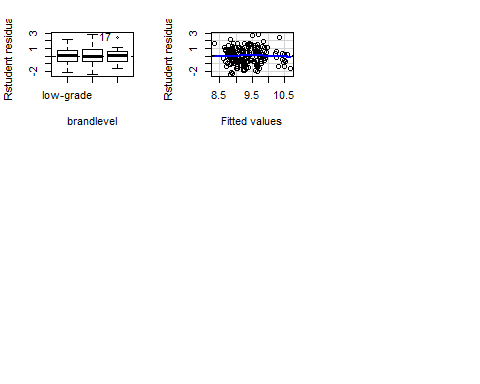
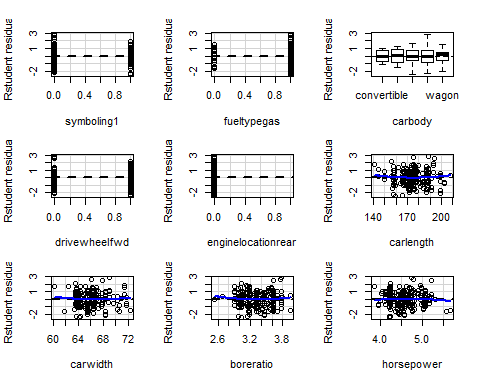
## GVIF Df GVIF^(1/(2\*Df))  
## symboling1 1.370506 1 1.170686  
## fueltypegas 1.532343 1 1.237879  
## carbody 2.444383 4 1.118204  
## drivewheelfwd 1.855041 1 1.361999  
## enginelocationrear 1.382089 1 1.175623  
## carlength 6.081559 1 2.466082  
## carwidth 4.980529 1 2.231710  
## boreratio 2.165745 1 1.471647  
## horsepower 3.632992 1 1.906041  
## brandlevel 2.145071 2 1.210209

To test the linearity of our model, we use component residual plots to detect potential non-linearity. The graph shows that all variables are virtually well-behaved.



## MLE of lambda Score Statistic (z) Pr(>|z|)  
## carwidth 4.06261 0.6564 0.5116  
## carlength 2.06920 0.4698 0.6385  
## boreratio -1.35136 0.6466 0.5179  
## horsepower 0.34582 -0.8043 0.4212  
##   
## iterations = 19

To perform valid hypothesis test, we require that residuals adhere to normal distribution. Probably due to proper model selection, Jarque-Bera test result suggests that the residuals are distributed normally. Our hypothesis tests can be referred due to this, since normality assumption is thus satisfied.



## Test stat Pr(>|Test stat|)   
## symboling1 0.8243 0.41086   
## fueltypegas -0.5238 0.60108   
## carbody   
## drivewheelfwd -1.9341 0.05464 .  
## enginelocationrear -1.8073 0.07235 .  
## carlength 0.8848 0.37743   
## carwidth 0.7564 0.45040   
## boreratio 0.8041 0.42236   
## horsepower -0.5370 0.59189   
## brandlevel   
## Tukey test -0.6117 0.54070   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##   
## Jarque Bera Test  
##   
## data: lm\_subset\_nounusal$residuals  
## X-squared = 0.32961, df = 2, p-value = 0.8481

We suspect that higher order terms can be of influence in this regard. RESET test results, however, suggest that there is no need for including higher-order terms.

##   
## RESET test  
##   
## data: lm\_subset  
## RESET = 0.28276, df1 = 8, df2 = 182, p-value = 0.971

##   
## RESET test  
##   
## data: lm\_subset  
## RESET = 0.28985, df1 = 8, df2 = 182, p-value = 0.9687

Lastly, we test for heteroskedasticity using Non-constant Variance Score test, Breusch-Pagan test, and Goldfeld-Quandt test. All test results suggest that there is no heteroskedasticity in our model, pointing to the efficiency of this model.

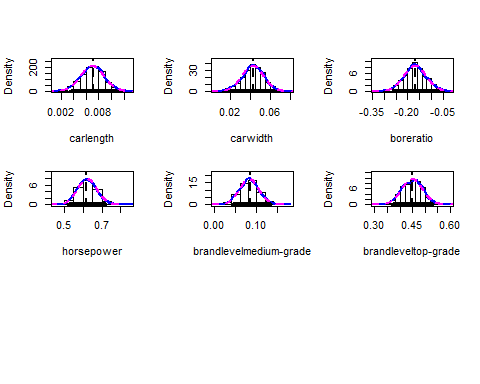
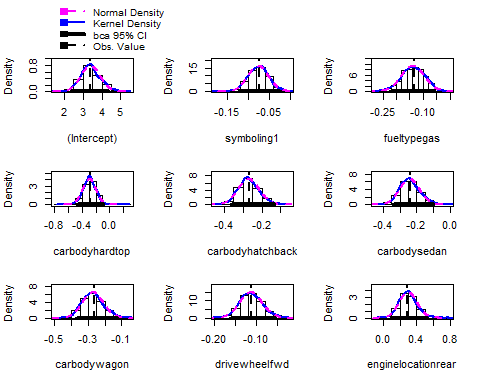
## Non-constant Variance Score Test   
## Variance formula: ~ fitted.values   
## Chisquare = 0.110135, Df = 1, p = 0.73999

##   
## studentized Breusch-Pagan test  
##   
## data: lm\_subset\_nounusal  
## BP = 18.171, df = 14, p-value = 0.1991

##   
## Goldfeld-Quandt test  
##   
## data: lm\_subset\_nounusal  
## GQ = 0.82942, df1 = 85, df2 = 84, p-value = 0.8043  
## alternative hypothesis: variance increases from segment 1 to 2

Usually, we can test the robustness of coefficients by bootstrapping the model. Bootstrap requires doing a resampling of our original dataset with replacement, and estimating the regression model for repetitive times, so that we obtain a measure of the sampling distribution for our parameter estimates.

##   
## Number of bootstrap replications R = 951   
## original bootBias bootSE bootMed  
## (Intercept) 3.3621425 3.6075e-02 0.5103700 3.3761679  
## symboling1 -0.0745749 -1.7761e-03 0.0245005 -0.0753284  
## fueltypegas -0.1339768 -4.3254e-03 0.0429820 -0.1374854  
## carbodyhardtop -0.2826829 -1.2077e-02 0.0969706 -0.2928791  
## carbodyhatchback -0.2649705 -5.4867e-03 0.0563157 -0.2735374  
## carbodysedan -0.2370114 -5.0306e-03 0.0596630 -0.2459307  
## carbodywagon -0.2668488 -5.2888e-03 0.0617745 -0.2751379  
## drivewheelfwd -0.1125383 -1.1534e-03 0.0266496 -0.1146732  
## enginelocationrear 0.2930621 6.0151e-03 0.1061649 0.2934894  
## carlength 0.0070669 -9.4409e-05 0.0018772 0.0070182  
## carwidth 0.0433013 -5.0269e-04 0.0108766 0.0430586  
## boreratio -0.1694659 3.6649e-04 0.0476647 -0.1700832  
## horsepower 0.6171413 5.0876e-03 0.0516117 0.6207233  
## brandlevelmedium-grade 0.0845242 -2.1891e-03 0.0230229 0.0821322  
## brandleveltop-grade 0.4498724 -6.5095e-04 0.0414428 0.4509183

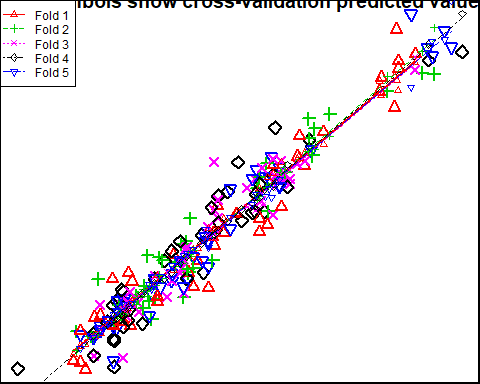


## (Intercept) symboling1 fueltypegas   
## 3.267480688 -0.090554352 -0.109447026   
## carbodyhardtop carbodyhatchback carbodysedan   
## -0.330329460 -0.283844418 -0.259158755   
## carbodywagon drivewheelfwd enginelocationrear   
## -0.295115831 -0.119192319 0.426192686   
## carlength carwidth boreratio   
## 0.007619823 0.044254315 -0.174737087   
## horsepower brandlevelmedium-grade brandleveltop-grade   
## 0.609314561 0.082522266 0.421736970

We can see by comparison that virtually all estimates fall within the 95% confidence interval of bootstrap distributions, indicating the robustness of our model.

From the histogram of the boot estimates, we can see that all estimates have a good normal distribution. This result resonates with Jarque-Bera Test results. Since asymptotic normality is satisfied, according to the construction of OLS estimation, sampling distribution of estimates should be normal.

Lastly, we should evaluate the performance of our model with some criterion like MSE. Since we do not have a test dataset, we can instead use the cross-validation method.



## Cross Validation MSE is: 0.02003463

From the results above, we can see that our model has a 0.02 MSE, which is quite good and all folds have similar MSE around this number. What’s more, the 5 fitted lines stay close to each other graphically, which also validate good stability and strong robustness.

## 4. The Conclusion

The following variables will affect the price in the following manners:

First, there is significant difference between the symboling1 and other levels with respect to the correlation with price. The symboling1 is associated with an estimation of 9.06% less in car price compared with others.

The car using gas as fuel is associated with about 10.94% less in price than the car using diesel.

Compared with the convertible- carbody car, the car with hardtop carbody is estimated to decrease the car price by 33.03%, the car with hatchback carbody is associated with a decrease of 28.38% in car price, the sedan-body tends to lower the car price by 25.92%, and the car with wagon body is 29.51% in price on average.

For the type of drive wheel, there is significant difference between fwd and others with respect to the correlation with price. The car with fwd wheel is related to a 11.92% decrease in price.

As for the engine location, the rear location will move up the car price by 42.62%.

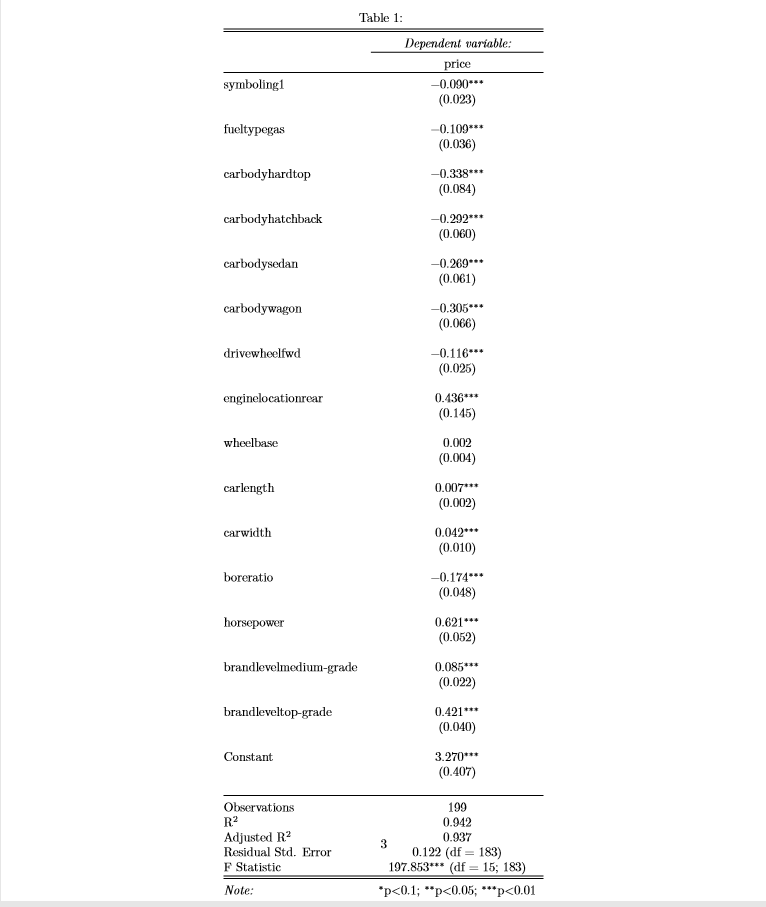
A unit increase in car length is associated with 0.76% increase in car price.

A unit increase in car width tends to move up the car price by 4.43%.

If boreration of the car increase 1 unit, the car price tends to decrease by 17.47%..

Since we have exerted log-transformation on horsepower, 1% increase in horsepower is associated with 0.61% increase in car price.

Refer to low-level brand as the baseline, the car of medium level brand is 8.25% higher in price and the car of top-level brand is 42.17% higher in price in estimation.



Overall our model is good at prediction, have extremely high , satisfies assumptions including linearity, non-multicollinearity, strict exogeneity, spherical error, and achieved normal error using a large v enough sample. The transformation is crucial in helping the model rule out heteroskedasticity and non-normal residuals, and VIF is the way to go when it comes to multicollinearity. Transformation also assists in maintaining a linear form of the formula.

## Appendix

### I. Code

#### 1.Descriptive analysis

set.seed(100)  
library(ggplot2)  
library(ggpubr)  
library(psych)  
library(tseries)  
library(multcomp)  
library(corrplot)  
library(broom)  
library(pastecs)  
library(broom)  
library(lmtest)  
library(DAAG)  
library(tidyverse)  
library(car)  
library(GGally)  
library(corrplot)  
library(mice)  
library(lmtest)  
library(leaps)  
library(MASS)  
library(boot)  
library(stargazer)

pro = read.csv("CarPrice\_Assignment.csv", header = TRUE)  
head(pro)

summary(pro)

str(pro)

attach(pro)  
pro = subset(pro, select = -car\_ID)

# no missing values  
md.pattern(pro)  
  
# Collapse car name into brands  
x = c("alfa-romero", "audi", "bmw", "buick", "chevrolet", "dodge", "honda",   
 "isuzu", "jaguar", "mazda", "mercury", "mitsubishi", "nissan", "peugeot",   
 "plymouth", "porsche", "renault", "saab", "subaru", "toyota", "volkswagen",   
 "volvo", "vw")  
  
brand = numeric(nrow(pro))  
for (i in seq(1, nrow(pro))) {  
 temp = unlist(strsplit(as.character(pro[i, 2]), " "))  
 for (name in x) {  
 if (name %in% temp) {  
 if (name != "alfa-romero") {  
 brand[i] = name  
 } else {  
 brand[i] = "AR"  
 }  
 }  
 }  
}  
brand = factor(brand)  
pro = cbind(pro, brand)  
pro$CarName <- NULL  
  
  
par(mfrow = c(5, 5))  
n = nrow(pro)  
k = 1 + log(n)  
nam = names(pro)  
  
  
# Histogram and scattermatrix for the continual variables  
fac = NULL  
num = NULL  
for (i in seq(1, length(nam))) {  
 if (class(pro[[i]]) != "factor") {  
 hist(pro[[i]], breaks = 20, col = "#99ccff", main = nam[i], xlab = colnames(pro)[i])  
 num = c(num, nam[i])  
 } else {  
 fac = c(fac, nam[i])  
 }  
}  
  
ind = which(colnames(pro) %in% fac)  
  
scat = pro[, -ind]  
scatterplotMatrix(scat, smooth = list(method = gamLine))

name <- strsplit(as.character(pro$CarName), " ")  
brands <- sapply(name, "[", 1)  
brands[which(brands == "maxda")] = "mazda"  
brands[which(brands == "porcshce")] = "porsche"  
brands[which(brands == "toyouta")] = "toyota"  
brands[which(brands == "vw" | brands == "vokswagen")] = "volkswagen"  
brands[which(brands == "Nissan")] = "nissan"  
unique(brands)  
  
pro = cbind(pro, brands)  
pro$brands <- factor(pro$brands, levels = names(sort(table(pro$brands), decreasing = T)))  
ggplot(pro, aes(x = brands)) + geom\_bar(fill = "purple") + geom\_text(stat = "count",   
 aes(label = ..count..), vjust = -0.5) + theme\_classic() + ggtitle("CarName") +   
 theme(plot.title = element\_text(hjust = 0.5), axis.text.x = element\_text(angle = 90,   
 size = 12, vjust = 0.5))

# no missing values  
md.pattern(pro)  
  
# Collapse car name into brands  
x = c("alfa-romero", "audi", "bmw", "buick", "chevrolet", "dodge", "honda",   
 "isuzu", "jaguar", "mazda", "mercury", "mitsubishi", "nissan", "peugeot",   
 "plymouth", "porsche", "renault", "saab", "subaru", "toyota", "volkswagen",   
 "volvo", "vw")  
  
brand = numeric(nrow(pro))  
for (i in seq(1, nrow(pro))) {  
 temp = unlist(strsplit(as.character(pro[i, 2]), " "))  
 for (name in x) {  
 if (name %in% temp) {  
 if (name != "alfa-romero") {  
 brand[i] = name  
 } else {  
 brand[i] = "AR"  
 }  
 }  
 }  
}  
brand = factor(brand)  
pro = cbind(pro, brand)  
pro$CarName <- NULL  
  
  
par(mfrow = c(5, 5))  
n = nrow(pro)  
k = 1 + log(n)  
nam = names(pro)  
  
  
# Histogram and scattermatrix for the continual variables  
fac = NULL  
num = NULL  
for (i in seq(1, length(nam))) {  
 if (class(pro[[i]]) != "factor") {  
 hist(pro[[i]], breaks = 20, col = "#99ccff", main = nam[i], xlab = colnames(pro)[i])  
 num = c(num, nam[i])  
 } else {  
 fac = c(fac, nam[i])  
 }  
}  
  
ind = which(colnames(pro) %in% fac)  
  
scat = pro[, -ind]  
scatterplotMatrix(scat, smooth = list(method = gamLine))

stat.desc(price)  
par(mfrow = c(2, 2))  
truehist(price, col = "#99ccff", ylab = "Counts", xlab = "Values")  
lines(density(price))  
qqPlot(~price, data = heart)  
Boxplot(~price, data = heart)  
jarque.bera.test(price)

gcp <- function(var, transp = 0.5, ind) {  
 # entire hist  
 g1 = ggplot(pro, aes(x = var)) + geom\_histogram(aes(y = ..density.., fill = "#99CCFF"),   
 alpha = transp, show.legend = FALSE) + geom\_density(alpha = transp -   
 0.2, color = "#9999ff", fill = "#9999ff") + labs(x = num[ind])  
   
 # group by target  
 g2 = ggplot(pro, aes(x = var, y = price)) + geom\_point() + labs(x = num[ind])  
   
 # boxplot  
 g3 = ggplot(pro, aes(y = var)) + geom\_boxplot(outlier.size = 3) + labs(x = num[ind])  
   
 # qqplot  
 g4 = ggplot(pro, aes(sample = var)) + geom\_qq() + geom\_qq\_line() + labs(y = num[ind])  
   
 g = ggarrange(g1, g2, g3, g4, ncol = 2, nrow = 2, labels = c("Histogram",   
 "Scatterplot", "Boxplot", "QQplot"))  
 return(g)  
}  
  
symbolgraph = gcp(symboling, ind = 1)  
symbolgraph  
  
# summary  
sumit <- function(var) {  
 print(describe(var))  
 print(summary(var))  
 cat("Range is:", "\n", range(var), "\n")  
 cat("Correlation with target is:", "\n", cor(var, price), "\n")  
 jarque.bera.test(var)  
}  
sumit(symboling)

gcpprice <- function(var, transp = 0.5, ind) {  
 # entire hist  
 g1 = ggplot(pro, aes(x = var)) + geom\_histogram(aes(y = ..density.., fill = "#99CCFF"),   
 alpha = transp, show.legend = FALSE) + geom\_density(alpha = transp -   
 0.2, color = "#9999ff", fill = "#9999ff") + labs(x = num[ind])  
   
 # boxplot  
 g3 = ggplot(pro, aes(y = var)) + geom\_boxplot(outlier.size = 3) + labs(x = num[ind])  
   
 # qqplot  
 g4 = ggplot(pro, aes(sample = var)) + geom\_qq() + geom\_qq\_line() + labs(y = num[ind])  
   
 g = ggarrange(g1, g3, g4, ncol = 2, nrow = 2, labels = c("Histogram", "Boxplot",   
 "QQplot"))  
 return(g)  
}  
pricegraph <- gcpprice(price, ind = 15)  
pricegraph

wheelbasegraph = gcp(wheelbase, ind = 2)  
wheelbasegraph  
sumit(wheelbase)

carlengthgraph = gcp(carlength, ind = 3)  
carlengthgraph  
sumit(carlength)

carwidthgraph = gcp(carwidth, ind = 4)  
carwidthgraph  
sumit(carlength)

carheightgraph = gcp(carheight, ind = 5)  
carheightgraph  
sumit(carheight)

curbweightgraph = gcp(curbweight, ind = 6)  
curbweightgraph  
sumit(curbweight)

enginesizegraph = gcp(enginesize, ind = 7)  
enginesizegraph  
sumit(enginesize)

boreratiograph = gcp(boreratio, ind = 8)  
boreratiograph  
sumit(boreratio)

strokegraph = gcp(stroke, ind = 9)  
strokegraph  
sumit(stroke)

compressionratiograph = gcp(compressionratio, ind = 10)  
compressionratiograph  
sumit(compressionratio)

horsepowergraph = gcp(horsepower, ind = 11)  
horsepowergraph  
sumit(horsepower)

peakrpmgraph = gcp(peakrpm, ind = 12)  
peakrpmgraph  
sumit(peakrpm)

citympggraph = gcp(citympg, ind = 13)  
citympggraph  
sumit(citympg)

highwaympggraph = gcp(highwaympg, ind = 14)  
highwaympggraph  
sumit(highwaympg)

# See distribution of the response according to categories, using factor  
# variables levels(CarName) shows that there is simply too many CarName to  
# graph this efficiently  
  
gfp <- function(var, transp = 0.5) {  
 g = ggplot(pro, aes(x = price, fill = var)) + geom\_histogram(aes(y = ..density..),   
 alpha = transp, binwidth = 1000) + geom\_density(alpha = transp)  
 return(g)  
}  
  
fueltypegraph = gfp(fueltype)  
aspirationgraph = gfp(aspiration)  
doornumbergraph = gfp(doornumber)  
carbodygraph = gfp(carbody)  
drivewheelgraph = gfp(drivewheel)  
enginelocationgraph = gfp(enginelocation)  
enginetypegraph = gfp(enginetype)  
cylindernumbergraph = gfp(cylindernumber)  
fuelsystemgraph = gfp(fuelsystem)  
  
fueltypegraph  
aspirationgraph  
doornumbergraph  
carbodygraph  
drivewheelgraph  
enginelocationgraph  
enginetypegraph  
cylindernumbergraph  
fuelsystemgraph

library(corrplot)  
corrplot(cor(scat))

#### 2.Modeling

library(tidyverse)  
library(car)  
library(GGally)  
library(corrplot)  
library(mice)  
library(lmtest)  
library(leaps)  
library(MASS)  
library(boot)  
library(DAAG)  
library(tseries)  
library(stargazer)

# input the data and have a glimpse of it.  
car = read.csv("car.csv", header = T)  
str(car)  
head(car)

# data wrangling  
  
## remove ID variable  
car = subset(car, select = -car\_ID)  
  
## get the brands of car in each observation and correct the typo.  
name = strsplit(as.character(car$CarName), " ")  
brands = sapply(name, "[", 1)  
brands[which(brands == "maxda")] = "mazda"  
brands[which(brands == "porcshce")] = "porsche"  
brands[which(brands == "toyouta")] = "toyota"  
brands[which(brands == "vw" | brands == "vokswagen")] = "volkswagen"  
brands[which(brands == "Nissan")] = "nissan"  
car = cbind(car, brands)  
  
## get the grade of cars calcualted by mean price  
mean\_price = car %>% group\_by(brands) %>% summarise(price\_mean = mean(price))  
mean\_price$brandlevel = cut(mean\_price$price\_mean, c(0, 10000, 20000, 40000),   
 labels = c("low-grade", "medium-grade", "top-grade"))  
car = merge(car, mean\_price, by = "brands")  
car = subset(car, select = -c(CarName, price\_mean, brands))  
  
## change symboling to factor according to the dictionary  
car$symboling = factor(car$symboling)

# price  
par(mfrow = c(2, 2))  
hist\_PRICE = hist((car$price), freq = F, breaks = 20, main = "Histogram of PRICE",   
 col = "purple")  
densityPlot(~price, data = car, main = "Density Plot of PRICE")  
qq\_PRICE = qqPlot(~price, data = car, main = "Quantile Plot of PRICE", id = F)  
box\_PRICE = Boxplot(~price, data = car, main = "Boxplot of PRICE", col = "purple")  
  
par(mfrow = c(2, 2))  
hist\_PRICE = hist(log(car$price), freq = F, breaks = 20, main = "Histogram of LOG PRICE",   
 col = "purple")  
densityPlot(~log(price), data = car, main = "Density Plot of LOG PRICE")  
qq\_PRICE = qqPlot(~log(price), data = car, main = "Quantile Plot of LOG PRICE",   
 id = F)  
box\_PRICE = Boxplot(~log(price), data = car, main = "Boxplot of LOG PRICE",   
 col = "purple")  
  
car\_t = car  
car\_t$price = log(car\_t$price)

# wheelbase  
scatterplot(price ~ wheelbase, data = car\_t)  
summary(t <- powerTransform(wheelbase ~ price, data = car\_t)) #-2  
testTransform(t, lambda = -2)  
plot(car\_t$wheelbase^(-2), car\_t$price)  
  
# carlength  
scatterplot(price ~ carlength, data = car\_t)  
summary(t <- powerTransform(carlength ~ price, data = car\_t)) #1  
  
# carwidth  
scatterplot(price ~ carwidth, data = car\_t)  
summary(t <- powerTransform(carwidth ~ price, data = car\_t)) # -5

## Warning in estimateTransform.default(X, Y, weights, family, ...):  
## Convergence failure: return code = 52

testTransform(t, lambda = -5)  
plot(car\_t$carwidth^(-5), car\_t$price)  
  
# carheight  
scatterplot(price ~ carheight, data = car\_t)  
summary(powerTransform(carheight ~ price, data = car\_t)) #1  
  
# cubrweight  
scatterplot(price ~ curbweight, data = car\_t)  
summary(t <- powerTransform(curbweight ~ price, data = car\_t)) #0  
scatterplot(price ~ log(curbweight), data = car\_t)  
  
# enginesize  
scatterplot(price ~ enginesize, data = car\_t)  
summary(t <- powerTransform(enginesize ~ price, data = car\_t)) #-0.5  
testTransform(t, -0.5)  
plot(car\_t$enginesize^(-0.5), car\_t$price)  
  
# boreratio  
scatterplot(price ~ boreratio, data = car\_t)  
summary(t <- powerTransform(boreratio ~ price, data = car\_t)) #1  
  
# stroke  
scatterplot(price ~ stroke, data = car\_t)  
summary(t <- powerTransform(stroke ~ price, data = car\_t)) #2  
testTransform(t, 2)  
plot(car\_t$stroke^(2), car\_t$price)  
  
# compressionratio  
scatterplot(price ~ compressionratio, data = car\_t)  
summary(t <- powerTransform(compressionratio ~ price, data = car\_t)) #-3  
testTransform(t, -3)  
plot(car\_t$compressionratio^(-3), car\_t$price)  
  
# horsepower  
scatterplot(price ~ horsepower, data = car\_t)  
summary(t <- powerTransform(horsepower ~ price, data = car\_t)) #0  
scatterplot(price ~ log(horsepower), data = car\_t)  
  
# peakrpm  
scatterplot(price ~ peakrpm, data = car\_t)  
summary(t <- powerTransform(peakrpm ~ price, data = car\_t)) #0  
testTransform(t, 0)  
scatterplot(price ~ log(peakrpm), data = car)  
  
# citympg  
scatterplot(price ~ citympg, data = car\_t)  
summary(t <- powerTransform(citympg ~ price, data = car\_t)) # -0.5  
testTransform(t, -0.5)  
plot(car\_t$citympg^(-0.5), car\_t$price)  
  
# highwaympg  
scatterplot(price ~ highwaympg, data = car\_t)  
summary(t <- powerTransform(highwaympg ~ price, data = car\_t)) #0  
scatterplot(price ~ log(highwaympg), data = car\_t)

# Evaluate the transformation  
p1 = powerTransform(curbweight ~ log(price), car, family = "bcPower")  
p2 = powerTransform(horsepower ~ log(price), car, family = "bcPower")  
p3 = powerTransform(peakrpm ~ log(price), car, family = "bcPower")  
p4 = powerTransform(highwaympg ~ log(price), car, family = "bcPower")  
  
  
print("Transform for curbweight:")  
testTransform(p1, 0)  
print("Transform for horsepower:")  
testTransform(p2, 0)  
print("Transform for peakrpm:")  
testTransform(p3, 0)  
print("Transform for highwaympg:")  
testTransform(p4, 0)  
  
# Conduct the transformation of independent variables according to part 1  
car\_t = car %>% mutate(curbweight = log(curbweight), horsepower = log(horsepower),   
 peakrpm = log(peakrpm), highwaympg = log(highwaympg))

# check missing values  
md.pattern(car)

# do the regression on all related variables according to the correlation  
# plot and scatter plot.  
lm\_basal = lm(price ~ . - compressionratio + stroke + carheight + peakrpm, data = car\_t)  
summary(lm\_basal)

# remove no-converge variables  
lm\_basal = lm\_basal %>% update(. ~ . - fuelsystem - cylindernumber, data = car\_t)  
summary(lm\_basal)  
car::vif(lm\_basal)

# remove VIF>5 step 1  
lm\_basal = lm\_basal %>% update(. ~ . - citympg, data = car\_t)  
car::vif(lm\_basal)

# remove VIF>5 step 2  
lm\_basal = lm\_basal %>% update(. ~ . - curbweight, data = car\_t)  
car::vif(lm\_basal)

# remove VIF>5 step 3  
lm\_basal = lm\_basal %>% update(. ~ . - enginesize, data = car\_t)  
car::vif(lm\_basal)

From there, we uses Mallows Cp to identify main effects to keep.

# Use Mallows Cp to choose main effect.  
subset = regsubsets(formula(lm\_basal), method = "forward", nbest = 1000, nvmax = 100,   
 data = car\_t)  
subset\_s = summary(subset)  
which.min(subset\_s$cp)  
coef(subset, which.min(subset\_s$cp))

$$ log(price)=\beta\_{1}+\beta\_{2}carlength+\beta\_{3}carwidth+\beta\_{4}boreratio+ \beta\_{5}log(horsepower)+\\ \alpha\_{1}symboling1+\alpha\_{2}symboling3+ \alpha\_{3}fueltypegas+\alpha\_{4}carbodyhardtop+\alpha\_{5}carbodyhatchback+\\ +\alpha\_{6}carbodysedan+\alpha\_{7}carbodywagon+\alpha\_{8}drivewheelfwd+ \alpha\_{9}enginelocationrear+\\ \gamma\_{1}enginetypedohcv+\gamma\_{2}enginetypeohc+\\ \gamma\_{3}brandlevelmediumgrade+\gamma\_{4}brandleveltopgrade+e $$

# do the regression using above predictors.  
car\_t = car\_t %>% mutate(symboling1 = ifelse(symboling == 1, 1, 0), symboling3 = ifelse(symboling ==   
 3, 1, 0), fueltypegas = ifelse(fueltype == "gas", 1, 0), drivewheelfwd = ifelse(drivewheel ==   
 "fwd", 1, 0), enginelocationrear = ifelse(enginelocation == "rear", 1, 0))  
  
lm\_subset0 = lm(price ~ symboling1 + symboling3 + fueltypegas + carbody + drivewheelfwd +   
 enginelocationrear + wheelbase + carlength + carwidth + boreratio + horsepower +   
 highwaympg + brandlevel, data = car\_t)  
summary(lm\_subset0)

# remove unsiginifcant variables step 1  
lm\_subset = lm\_subset0 %>% update(. ~ . - symboling3, data = car\_t)  
summary(lm\_subset)

# remove unsiginifcant variables step 2  
lm\_subset = lm\_subset %>% update(. ~ . - wheelbase, data = car\_t)  
summary(lm\_subset)

$$ log(price)=*{1}+*{2}carlength+*{3}carwidth+*{4}boreratio+ *{5}log(horsepower)+\* {1}symboling1+ *{3}fueltypegas+*{4}carbodyhardtop+*{5}carbodyhatchback+\ +*{6}carbodysedan+*{7}carbodywagon+*{8}drivewheelfwd+ *{9}enginelocationrear+\* {3}brandlevelmediumgrade+\_{4}brandleveltopgrade+e

$$

lm\_subset = lm\_subset %>% update(. ~ . - highwaympg, data = car\_t)  
summary(lm\_subset)

# Compare our 1st basic main effect model with the 2nd baseline model  
# removing some less relevant variables.  
AIC(lm\_subset0, lm\_subset)  
BIC(lm\_subset0, lm\_subset)

# check unusual observations  
qqPlot(lm\_subset, data = car\_t, id = list(n = 3))  
outlierTest(lm\_subset)  
influenceIndexPlot(lm\_subset, var = c("cook", "hat"))  
influencePlot(lm\_subset)

# Compare our basic main effect model with the model removing unusual  
# observations.  
lm\_subset\_nounusal = lm\_subset %>% update(subset = -c(75, 26, 53, 127, 129,   
 168), data = car\_t)  
S(lm\_subset\_nounusal)

# test the multi-collinearity  
car::vif(lm\_subset\_nounusal)

# test linearity  
car\_t\_sub = car\_t[-c(26, 53, 75, 127, 129, 168), ]  
crPlots(lm\_subset\_nounusal)  
boxTidwell(price ~ carwidth + carlength + boreratio + horsepower, ~symboling1 +   
 fueltypegas + carbody + drivewheelfwd + enginelocationrear + brandlevel,   
 data = car\_t\_sub)

## Warning in boxTidwell.default(y, X1, X2, max.iter = max.iter, tol = tol, :  
## maximum iterations exceeded

# Check residuals  
residualPlots(lm\_subset\_nounusal, type = "rstudent")  
jarque.bera.test(lm\_subset\_nounusal$residuals)

# Test model misspecification  
resettest(lm\_subset, 2, type = "regressor")  
resettest(lm\_subset, 3, type = "regressor")

# Check homoskedasticity  
ncvTest(lm\_subset\_nounusal)  
bptest(lm\_subset\_nounusal)  
gqtest(lm\_subset\_nounusal)

# Robustness of Coefficients  
coef\_boot = Boot(lm\_subset\_nounusal, R = 999)  
summary(coef\_boot)  
hist(coef\_boot)  
coef(lm\_subset\_nounusal)

# Cross-Validation to check model performance  
par(mar = c(1, 0, 2, 0))  
cv = CVlm(data = car\_t, form.lm = lm\_subset\_nounusal, m = 5, printit = F, plotit = T)  
# 0.03 0.02 0.02 0.02 0.02  
cat("Cross Validation MSE is:", attr(cv, "ms"))

# Final model  
S(lm\_subset\_nounusal)

## Bibliography

GINTER, J., YOUNG, M. and DICKSON, P. (1987). A Market Efficiency Study of Used Car Reliability and Prices. Journal of Consumer Affairs, 21(2), pp.258-276.

Prieto, M. and Caemmerer, B. (2013). An exploration of factors influencing car purchasing decisions. International Journal of Retail & Distribution Management, 41(10), pp.738-764.

Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.