Multi-Linear-Reg-Car-Proj

Pavlo Mysak

2023-11-05

Problem Statement

Geely Auto, a Chinese automobile company, seeks to expand its presence in the US market by establishing a manufacturing unit to produce cars locally. They aim to gain a competitive edge by accurately predicting the prices of cars in the American market.

The company has collected a comprehensive dataset on various car attributes in the American market. The goal is to develop a predictive model to forecast the prices of cars, enabling Geely Auto to anticipate and set competitive prices for their vehicles.

Business Goal

The objective is to build a robust predictive model that accurately estimates car prices based on a set of independent variables. This predictive tool will empower Geely Auto's management to anticipate market dynamics, allowing for informed decision-making in designing cars, devising business strategies, and adjusting pricing to meet specific targets. By leveraging the model's predictions, the company aims to proactively understand and adapt to the pricing dynamics of the American market, thereby enhancing their competitive positioning and strategic planning.

Data

Import the data, remove unnecessary columns, split the data into training and validation sets and view a quick summary.

```
library(car)
```

```
## Loading required package: carData
```

```
data <- read.csv('/Users/pavlomysak/Downloads/archive-2/CarPrice_Assignment.csv', str
ingsAsFactors = T)
dt <- subset(data, select = -c(car_ID, CarName))

train <- dt[1:130,]
valid <- dt[-(1:130),]

summary(train)
```

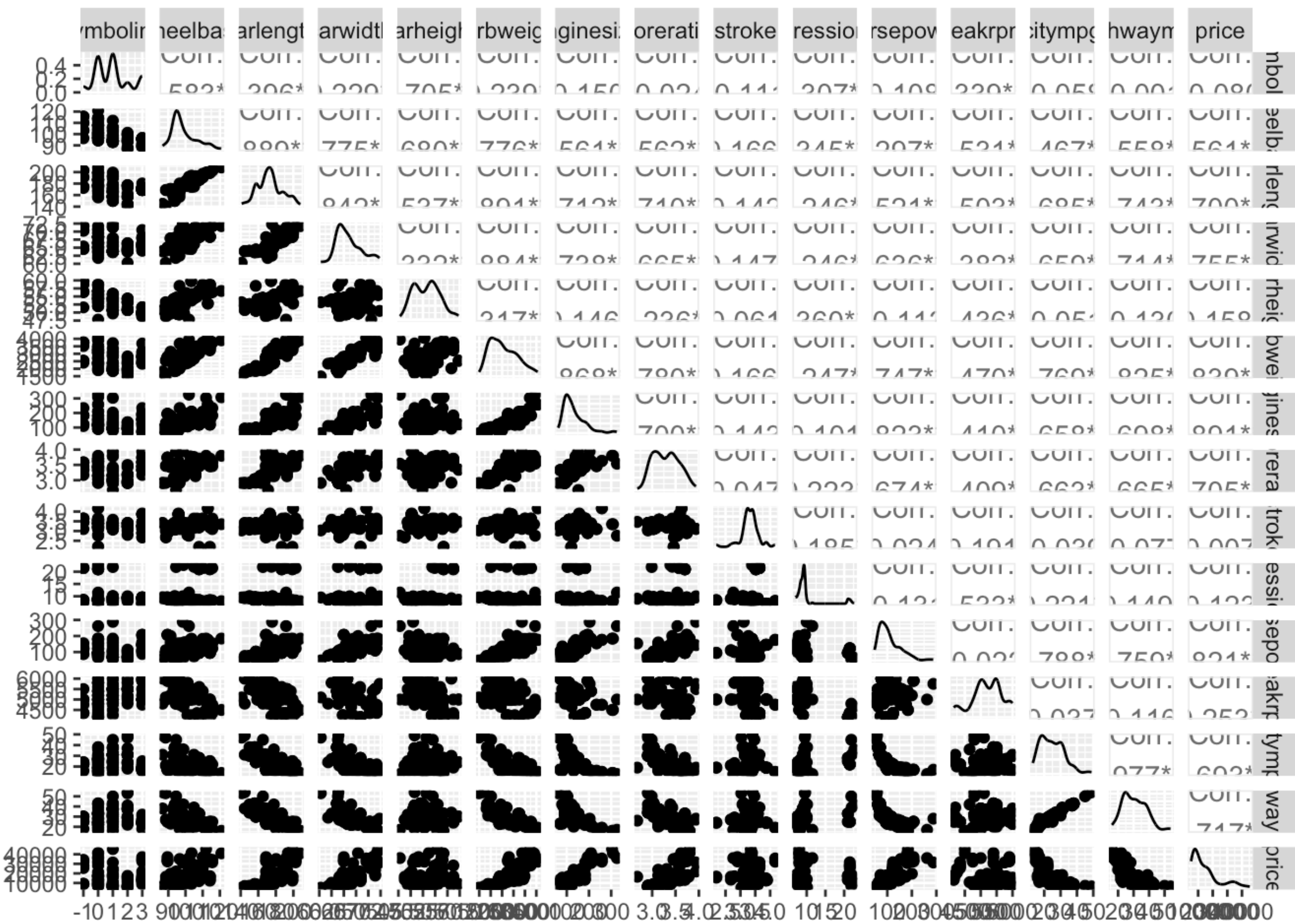
```
##    symboling         fueltype   aspiration  doornumber        carbody
##  Min.   :-1.0000   diesel: 12   std  :105   four:65    convertible: 4
##  1st Qu.: 0.0000   gas   :118   turbo: 25   two :65    hardtop    : 5
##  Median : 1.0000                                       hatchback  :48
##  Mean   : 0.9538                                       sedan      :61
##  3rd Qu.: 1.0000                                       wagon      :12
##  Max.   : 3.0000
##
##  drivewheel enginelocation   wheelbase        carlength        carwidth
##  4wd: 2     front:127      Min.   : 86.60   Min.   :141.1   Min.   :60.30
##  fwd:77     rear :  3      1st Qu.: 93.70   1st Qu.:165.3   1st Qu.:63.92
##  rwd:51                    Median : 96.30   Median :172.8   Median :65.40
##                            Mean   : 98.63   Mean   :173.3   Mean   :66.01
##                            3rd Qu.:101.80   3rd Qu.:178.5   3rd Qu.:67.90
##                            Max.   :120.90   Max.   :208.1   Max.   :72.30
##
##    carheight       curbweight    enginetype cylindernumber   enginesize
##  Min.   :47.80   Min.   :1488   dohc : 4   eight : 5      Min.   : 61.0
##  1st Qu.:50.80   1st Qu.:2012   dohcv: 1   five  :10      1st Qu.: 97.0
##  Median :53.50   Median :2408   l    :12   four  :91      Median :120.0
##  Mean   :53.28   Mean   :2576   ohc  :94   six   :18      Mean   :131.4
##  3rd Qu.:55.05   3rd Qu.:3054   ohcf : 3   three : 1      3rd Qu.:152.0
##  Max.   :59.80   Max.   :4066   ohcv :12   twelve: 1      Max.   :326.0
##                                 rotor: 4   two   : 4
##    fuelsystem   boreratio         stroke      compressionratio   horsepower
##  mpfi   :48   Min.   :2.680   Min.   :2.190   Min.   : 7.000   Min.   : 48.0
##  2bbl   :45   1st Qu.:3.030   1st Qu.:3.195   1st Qu.: 8.425   1st Qu.: 70.0
##  idi    :12   Median :3.330   Median :3.290   Median : 9.000   Median : 97.0
##  1bbl   :11   Mean   :3.301   Mean   :3.317   Mean   : 9.953   Mean   :107.8
##  spdi   : 9   3rd Qu.:3.470   3rd Qu.:3.460   3rd Qu.: 9.400   3rd Qu.:123.0
##  4bbl   : 3   Max.   :3.940   Max.   :4.170   Max.   :22.700   Max.   :288.0
##  (Other): 2
##     peakrpm        citympg        highwaympg        price
##  Min.   :4150   Min.   :13.00   Min.   :16.00   Min.   : 5151
##  1st Qu.:5000   1st Qu.:19.00   1st Qu.:25.00   1st Qu.: 7421
##  Median :5200   Median :24.00   Median :30.00   Median :10996
##  Mean   :5197   Mean   :24.85   Mean   :30.41   Mean   :14406
##  3rd Qu.:5500   3rd Qu.:31.00   3rd Qu.:37.00   3rd Qu.:17387
##  Max.   :6000   Max.   :49.00   Max.   :54.00   Max.   :45400
##
```

Correlation Matrix with Numeric/Integer Variables

Here we parse through the training dataframe to create a new dataframe including only numeric/int datatype columns. Using this new data frame, we can produce a correlation matrix to get a better idea of how our variables are related to one another. In the graphic below, we see quite a few linear relationships- it appears that a linear regression might be a good tool for this job!

```
## Loading required package: ggplot2
```

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



Building our Initial Models

To fit our linear regression, we will be using the Ordinary Least Squares method. The OLS method is an optimization algorithm that minimizes the sum of squared vertical distances between observed data points and responses predicted by the linear model. These distances are also known as our residuals.

In addition to using the OLS method to fit the model, we will also be using a bidirectional stepwise selection algorithm to construct an initial model. Stepwise selection approaches use a sequence of steps (forward, backward or both) to select the variables that maximize (or minimize) a certain model-fit criteria. The criteria we will be using is the Akaike Information Criterion or AIC. AIC is an estimator of prediction error and therefore suites the needs of our model application. It is important to note, however, that automated model selection algorithms can be seen as problematic because they are prone to over-fitting of data. In other words, the best AIC score does not always lead to the best model for real-world applications. Taking this into account, the stepwise selected model will only act as our initial model and further modifications will be made based on findings from exploratory data analysis (EDA) and intuition based off of domain knowledge.

Let’s build our initial model.

```
null_model <- lm(data = train, price~1)
full_model <- lm(data = train, price~.)

model_1 <- step(null_model, scope = list(lower=null_model, upper=full_model),
     direction = 'both', trace = F)

summary(model_1)
```

```
##
## Call:
## lm(formula = price ~ enginesize + enginetype + cylindernumber +
##     drivewheel + peakrpm + carwidth + carbody + stroke + boreratio +
##     horsepower + compressionratio + fueltype, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3926  -1054      0   1020   6714
##
## Coefficients: (1 not defined because of singularities)
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -5.929e+04  1.423e+04  -4.168 6.38e-05 ***
## enginesize            1.410e+02  2.818e+01   5.005 2.29e-06 ***
## enginetypedohcv      -1.359e+04  5.720e+03  -2.375 0.019359 *
## enginetypel           2.108e+03  1.833e+03   1.150 0.252909
## enginetypeohc         4.897e+03  1.592e+03   3.075 0.002688 **
## enginetypeohcf        2.348e+03  2.402e+03   0.978 0.330538
## enginetypeohcv       -7.119e+03  1.878e+03  -3.790 0.000253 ***
## enginetyperotor       1.241e+02  6.590e+03   0.019 0.985011
## cylindernumberfive   -7.730e+03  3.477e+03  -2.223 0.028369 *
## cylindernumberfour   -7.542e+03  4.289e+03  -1.758 0.081649 .
## cylindernumbersix    -5.175e+03  2.724e+03  -1.900 0.060246 .
## cylindernumberthree   2.174e+03  5.658e+03   0.384 0.701527
## cylindernumbertwelve -2.071e+04  3.990e+03  -5.191 1.04e-06 ***
## cylindernumbertwo            NA         NA      NA       NA
## drivewheelfwd        -1.355e+03  1.572e+03  -0.862 0.390642
## drivewheelrwd         2.066e+03  1.581e+03   1.307 0.194189
## peakrpm               2.058e+00  6.411e-01   3.210 0.001765 **
## carwidth              8.590e+02  1.967e+02   4.366 3.00e-05 ***
## carbodyhardtop       -1.691e+03  1.427e+03  -1.185 0.238645
## carbodyhatchback     -4.258e+03  1.471e+03  -2.893 0.004642 **
## carbodysedan         -3.117e+03  1.405e+03  -2.219 0.028663 *
## carbodywagon         -3.255e+03  1.510e+03  -2.155 0.033439 *
## stroke               -3.577e+03  1.061e+03  -3.373 0.001045 **
## boreratio            -5.545e+03  2.218e+03  -2.500 0.013983 *
## horsepower            6.056e+01  1.979e+01   3.059 0.002821 **
## compressionratio      9.748e+02  4.661e+02   2.092 0.038911 *
## fueltypegas           9.682e+03  6.192e+03   1.564 0.120945
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1697 on 104 degrees of freedom
## Multiple R-squared:  0.9733, Adjusted R-squared:  0.9669
## F-statistic: 151.8 on 25 and 104 DF,  p-value: < 2.2e-16
```

Model 1 Assesment

There are many variables in this model! The Adjusted R-Squared value looks good, but there are quite a few insignificant coefficients included in this model fit.

Let's clean up a few of these factor variables next. Namely, CylinderNumber, CarBody and EngineType. Our method to re-code these variables is to group them by Coefficient Estimate and Significance.

For example, CylinderNumber can be re-coded into two groups: Negative and Insignificant. This is because Four, Five, Six and Twelve cylinder engines each have a negative coefficient (negatively impact price) while being statistically significant. The other two factors in this variable, three and two cylinder engines, are insignificant and have non-negative coefficients.

We will repeat this process with the other aforementioned variables. Ideally, this will lower the complexity of our model and strengthen the predictive capabilities.

```
train$cylindernumber_recoded <- factor(with(train, ifelse(cylindernumber %in% c('five
', 'four', 'six', 'twelve'),
                                                          'neg','insig')))
valid$cylindernumber_recoded <- factor(with(valid, ifelse(cylindernumber %in% c('five
', 'four', 'six', 'twelve'),
                                                                'neg','insig')))

train$carbody_recoded <- factor(with(train, ifelse(carbody %in% c('hardtop','hatchbac
k','wagon'),
                                              'neg','insig')))
valid$carbody_recoded <- factor(with(valid, ifelse(carbody %in% c('hardtop','hatchbac
k','wagon'),
                                                         'neg','insig')))

train$enginetype_recoded <- factor(with(train, ifelse(enginetype %in% c('l','ohcf','r
otor'),'insig',
                ifelse(enginetype %in% c('dohcv','ohcv'), 'neg','pos'))))
valid$enginetype_recoded <- factor(with(valid, ifelse(enginetype %in% c('l','ohcf','r
otor'),'insig',
                                                       ifelse(enginetype %in% c('dohcv
','ohcv'), 'neg','pos'))))
```

Re-Running

Let's rerun model 1 with these re-coded values and assess our results.

```
model_2 <- lm(formula = price ~ enginesize + cylindernumber_recoded + enginetype_reco
ded +
     stroke + compressionratio + peakrpm + carbody_recoded + carwidth +
     enginelocation + curbweight + carlength + aspiration, data = train)
summary(model_2)
```

```
##
## Call:
## lm(formula = price ~ enginesize + cylindernumber_recoded + enginetype_recoded +
##     stroke + compressionratio + peakrpm + carbody_recoded + carwidth +
##     enginelocation + curbweight + carlength + aspiration, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6235.3 -1578.5  -277.6  1191.6 11648.9
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -3.199e+04  1.233e+04  -2.595  0.01068 *
## enginesize                1.119e+02  1.577e+01   7.098 1.08e-10 ***
## cylindernumber_recodedneg -8.546e+03  1.065e+03  -8.028 9.04e-13 ***
## enginetype_recodedneg    -4.167e+03  1.460e+03  -2.855  0.00511 **
## enginetype_recodedpos     4.763e+03  1.068e+03   4.458 1.92e-05 ***
## stroke                   -4.949e+03  9.631e+02  -5.138 1.13e-06 ***
## compressionratio          1.700e+02  7.656e+01   2.220  0.02834 *
## peakrpm                   2.355e+00  7.034e-01   3.348  0.00110 **
## carbody_recodedneg       -9.401e+02  4.948e+02  -1.900  0.05993 .
## carwidth                  4.656e+02  2.172e+02   2.144  0.03415 *
## enginelocationrear        1.241e+04  2.314e+03   5.365 4.20e-07 ***
## curbweight                7.450e+00  1.692e+00   4.404 2.38e-05 ***
## carlength                -6.220e+01  4.574e+01  -1.360  0.17655
## aspirationturbo          -4.800e+02  7.260e+02  -0.661  0.50979
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2432 on 116 degrees of freedom
## Multiple R-squared:  0.9389, Adjusted R-squared:  0.9321
## F-statistic: 137.2 on 13 and 116 DF,  p-value: < 2.2e-16
```

Model 2 Assessment

It appears that our Adjusted R-Squared Value decreased, but most of our variables are now significant! Let’s return back to our correlation matrix from earlier and see if there are any strong variables we are missing.

When consulting with the correlation matrix, strong linear relationships with price are seen with the following variables: wheelbase, carlength, carwidth, curbweight, enginesize, boreratio, horsepower, citympg, highwaympg

Stroke and CompressionRatio, both seen in our model, have very poor linear relationships with price. However, Stroke is quite statistically significant.

Let’s examine the Model 2 variables and their relationships to Price.

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:car':
##
##     recode
```

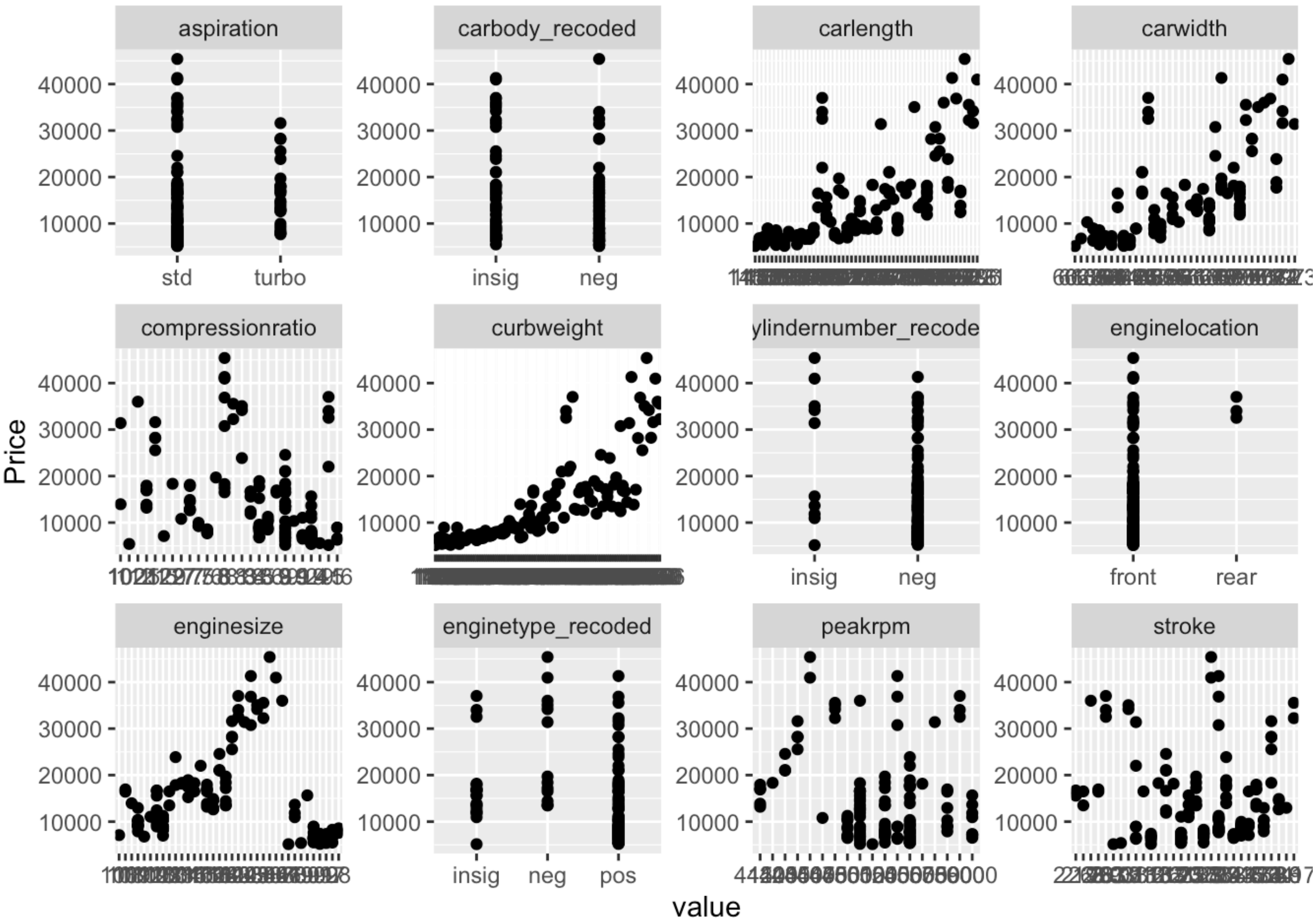
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
train_subset_long <- tidyr::gather(select(train, c('price', 'enginesize', 'cylindernu
mber_recoded', 'enginetype_recoded',
                                                   'stroke', 'compressionratio', 'pea
krpm', 'carbody_recoded',  'carwidth',
                                                   'enginelocation', 'curbweight', 'c
arlength', 'aspiration')), key = "variable", value = "value", -price)
```

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

```
# Plotting the scatterplots using ggplot and facet_wrap
ggplot(train_subset_long, aes(x = value, y = price)) +
  geom_point() +
  facet_wrap(~ variable, scales = 'free') +
  labs(y = "Price")
```



Let's construct a 3rd model using a collection of variables from Model 2 and the variables we've identified to be strong from the correlation matrix.

Model 3 formula = price ~ enginesize + cylindernumber_recoded + enginetype_recoded + boreratio + horsepower + carbody_recoded + carwidth + enginelocation + curbweight + stroke

```
model_3 <- lm(formula = price ~ enginesize + cylindernumber_recoded + enginetype_reco
ded +
    boreratio + horsepower + carbody_recoded + carwidth +
    enginelocation + curbweight + stroke, data = train)
summary(model_3)
```

```
##
## Call:
## lm(formula = price ~ enginesize + cylindernumber_recoded + enginetype_recoded +
##     boreratio + horsepower + carbody_recoded + carwidth + enginelocation +
##     curbweight + stroke, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6523.7 -1543.3  -102.7  1258.3 11648.8
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -13165.489  12451.908  -1.057 0.292531
## enginesize                    100.548     14.503   6.933 2.35e-10 ***
## cylindernumber_recodedneg   -9031.909   1033.545  -8.739 1.88e-14 ***
## enginetype_recodedneg       -5772.440   1556.082  -3.710 0.000318 ***
## enginetype_recodedpos        4042.262   1069.459   3.780 0.000248 ***
## boreratio                   -4781.879   1452.273  -3.293 0.001310 **
## horsepower                     33.665     10.565   3.186 0.001844 **
## carbody_recodedneg           -963.219    459.447  -2.096 0.038177 *
## carwidth                      499.808    198.311   2.520 0.013061 *
## enginelocationrear          13551.318   2317.067   5.848 4.52e-08 ***
## curbweight                      6.039      1.471   4.106 7.46e-05 ***
## stroke                      -4825.964    896.289  -5.384 3.76e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2415 on 118 degrees of freedom
## Multiple R-squared:  0.9387, Adjusted R-squared:  0.933
## F-statistic: 164.3 on 11 and 118 DF,  p-value: < 2.2e-16
```

After some trial and error of this third model, we have a formula that includes all statistically significant variables (excluding the intercept coefficient) and an Adjusted R-Squared value of 0.933.

Interpretation of Model 3

Generally, simple linear models (including only main effects) follow the formula:

Y = B0 + B1*X1* + *B2*X2 + … + Error

where Y is our Dependent Variable, B0 is our Y-Intercept, B(n) are our Slope Coefficients and X(n) are our Independent Variables. This formula will help us with fitting predictions to this model.

In our model:

1 unit increase in engine size increases the price of a vehicle by $100.

1 unit increase in bore ratio decreases the price of a vehicle by $4782.

1 unit increase in horese power increases the price of a vehicle by $34.

1 unit increase in car width increases the price of a vehicle by $500.

1 unit increase in curb weight increases the price of a vehicle by $6.

1 unit increase in stroke decreases the price of a vehicle by $4826.

If a vehicle has a four, five, six or twelve cylinder engine, the price is decreased by $9032

If a vehicle has a DOHCV or OHCV engine, the price is decreased by $5772

If a vehicle has a DOHC, OHC engine, the price is increased by $4042

If a vehicle has a Hardtop, Hatchback or Wagon body, the price is decreased by $963

If a vehicle has its engine located in the rear, the price is increased by $13551

Evaluating the Validity and Performance of our Model

Adjusted R-Squared

As seen above, the Adj. R-Squared value for our final model is 0.933. This means that our model can account for about 93.3% of the observations within our data. While keeping in mind that 0.7 is the industry standard for linear models, this is pretty good! However, Adj. R-Squared values do not tell the whole story and can be misleading.

RMSE

Root-Mean-Squared Error is another method of evaluating regression model performance. It measures the difference between the model's predicted values and our actual (observed) values. RMSE can also be seen as the standard deviation of the residuals associated with our model or the average error of predictions. The lower the RMSE, the better our model fits the data.

Let's calculate the in-sample and out-of-sample RMSE, respectively. Our out-of-sample RMSE is higher than our in-sample, which is to be expected, but only by 577 Units (dollars). Because the difference is fairly small, only some over-fitting is present in our model. Overall, our out-of-sample predictions may be about $2877.56 off from the real price.

```
sqrt(mean((train$price-predict(model_3, train))^2)) # in-sample
```

```
## [1] 2300.629
```

```
sqrt(mean((valid$price-predict(model_3, valid))^2)) # out-of-sample
```

```
## [1] 2877.506
```

Predictions

Geely Auto has envisioned a revolutionary sedan that they've determined would be popular in the American auto market. The company's aspiration is to craft a car that encapsulates superior performance, innovative design, and competitive pricing.

With this ambition in mind, Geely Auto has meticulously designed a prototype featuring specific attributes:

Enginesize: 150

Cylindernumber: four

Enginetype: ohcv

Boreratio: 3.4

Horsepower: 195

Carbody: sedan

Carwidth: 65

Enginelocation: front

Curbweight: 2515

Stroke: 3.2

According to our model, Geely Auto should price this new sedan at $9652 to adequately compete in the American auto market. This data, in conjunction with the costs associated with producing the vehicle will be crucial in determining the final consumer price of the vehicle.

```
new_car <- data.frame(enginesize = 150,
                      cylindernumber_recoded = 'neg',
                      enginetype_recoded = 'neg',
                      boreratio = 3.4,
                      horsepower = 195,
                      carbody_recoded = 'insig',
                      carwidth = 65,
                      enginelocation = 'front',
                      curbweight = 2515,
                      stroke = 3.2)

predict(model_3, newdata = new_car)
```

```
##        1
## 9652.262
```

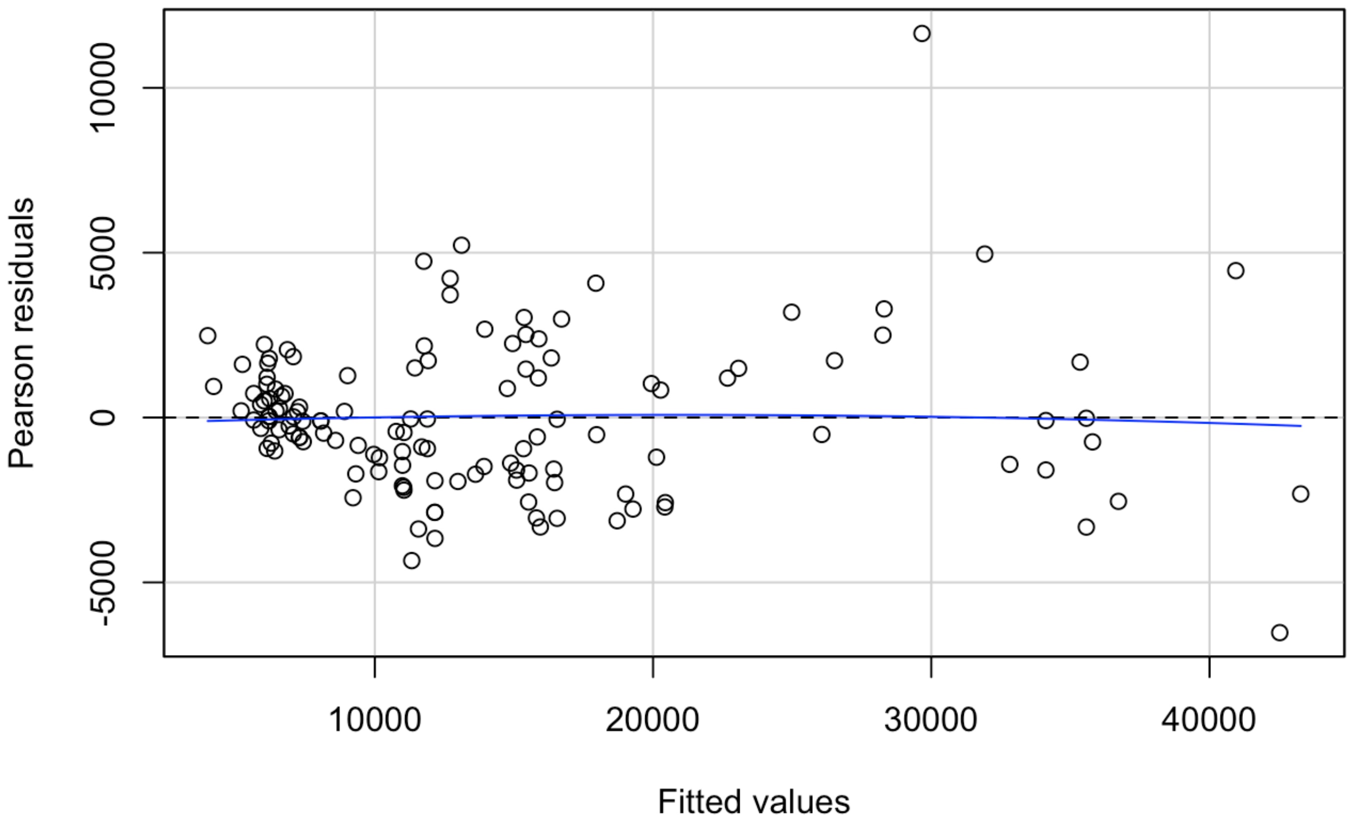
Residual Analysis

the mean of residuals is approximately 0, however, the residuals do not appear to be randomly distributed around the horizontal axis. This suggests that a non-linear model may be a more appropriate fit for our data.

```
mean(model_3$residuals)
```

```
## [1] -8.04553e-14
```

```
residualPlot(model_3)
```

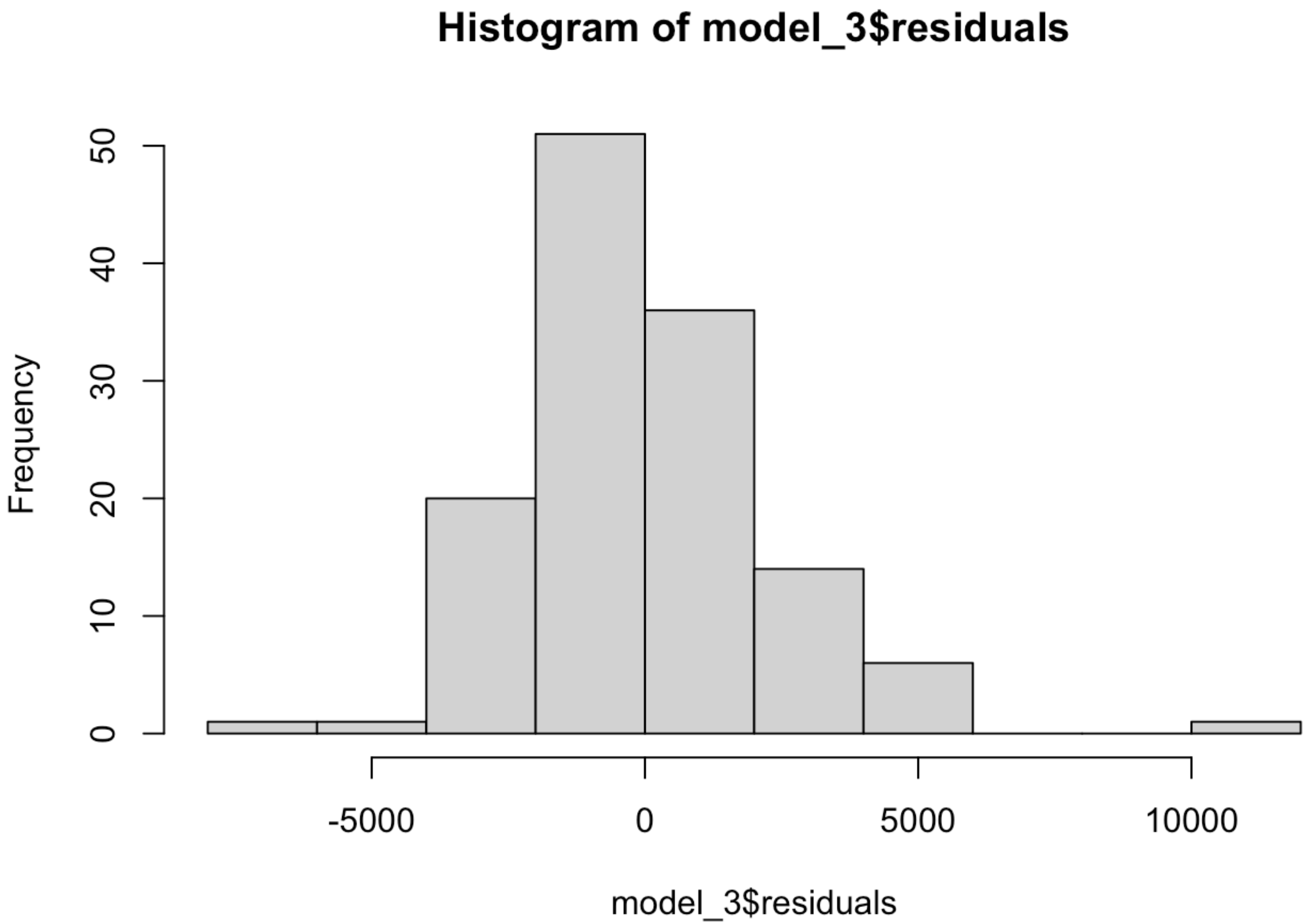


Do our residuals follow a normal distribution? According to the Shapiro-Wilk test, they do not follow a normal distribution, but if we generate a histogram, they appear to be fairly normal, aside from the outliers on the right tail. If we remove these extreme values, we see that the residuals do indeed follow a normal distribution according to the Shapiro-Wilk test for normality.

```
shapiro.test(model_3$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model_3$residuals
## W = 0.94518, p-value = 4.92e-05
```

```
hist(model_3$residuals)
```




```
shapiro.test(model_3$residuals[model_3$residuals < quantile(model_3$residuals, 0.999)
]) # removing upper outliers
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model_3$residuals[model_3$residuals < quantile(model_3$residuals, 0.999)]
## W = 0.98987, p-value = 0.4681
```

Heteroscedasticity

Heteroscedasticity is an issue that occurs when the variance of the predicted variable changes over different values of the independent variable. The existence of heteroscedasticity is a major concern in regression analysis and the analysis of variance, as it invalidates statistical tests of significance that assume that the modelling errors all have the same variance. We can check for Heteroscedasticity with the Breusch-Pagen Test (NCV Test).

According to our test, Heteroscedasticity is present in our model, however, because we are using the OLS method in fitting of our model, our predictions will remain unbiased and consistent. (Although, they will no longer qualify to be the Best Linear Unbiased Estimators because they are no longer efficient).

```
ncvTest(model_3)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 40.83443, Df = 1, p = 1.6569e-10
```

Multicollinearity

there seems to be multicollinearity present in a few variables within our model. To measure multicollinearity, we use the Variance Inflation Factor Test. When running a VIF Test, we are essentially making each variable a dependent variable and regressing it against every other variable.

At a threshold GVIF of 5, our problem variables are: EngineSize, EngineType, CarWidth,Curbweight

Generally, multicollinearity means that our coefficients are not uniquely established in our model. This can be an issue when the purpose of your model is to explain how your independent variables interact with your dependent variable, but for predictive purposes, it is not much of an issue.

```
vif(model_3)
```

```
##                            GVIF Df GVIF^(1/(2*Df))
## enginesize            11.227953  1        3.350814
## cylindernumber_recoded  1.690993  1        1.300382
## enginetype_recoded     8.449354  2        1.704927
## boreratio              3.435743  1        1.853576
## horsepower             4.700562  1        2.168078
## carbody_recoded        1.176523  1        1.084677
## carwidth               5.255788  1        2.292551
## enginelocation         2.698388  1        1.642677
## curbweight            17.137167  1        4.139706
## stroke                 1.406755  1        1.186067
```

Conclusions

In our final linear model, the variables used to predict car prices are engine size, cylinder number, engine type, bore ratio, horsepower, car body type, car width, engine location, curb weigh and stroke.

There are some issues found in our model evaluation process that degrade the validity of our model. These issues include multicollinearity, heteroscedasticity and a not perfectly random residual plot. While these complications do deflate our confidence in the explanatory power of our model and some statistical significance of coefficients, they do not have a fatal impact on the predictive power of our model.

Referencesn used

https://cran.r-project.org/web/packages/olsrr/vignettes/heteroskedasticity (https://cran.r-project.org/web/packages/olsrr/vignettes/heteroskedasticity)

https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/WIKIPEDI/W120529O.pdf (https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/WIKIPEDI/W120529O.pdf)

https://bookdown.org/max/FES/greedy-stepwise-selection.html (https://bookdown.org/max/FES/greedy-stepwise-selection.html)

https://medium.com/geekculture/akaike-information-criterion-model-selection-c47df96ee9a8 (https://medium.com/geekculture/akaike-information-criterion-model-selection-c47df96ee9a8)