Analysis

Justin Chan and Isaac Plotkin

3/3/2022

Analysis: Final Project

Research Question and Modeling Objective

The research question we have is how to best predict car prices based on various car features in our car dataset. Therefore, our modeling objective is to create the best possible linear model from our set of features in our car dataset and use this model to make predictions on car prices for new cars.

Description of Data and Response Variable

Data

The observations of the dataset are cars where each row is a car with the columns being various features of the car. The dataset includes 26 columns. One column is an observation index and another column is car price which is the variable we are trying to predict. We have 24 car features and 205 cars that we can use for our linear regression model.

The data was originally collected from various market surveys of different types of cars across the United States market around 1987 to learn how to price cars in China depending on the American market. There is an assumption that the cars in the dataset have been randomly chosen from the set of cars in the various market surveys. Link to the dataset: https://www.kaggle.com/hellbuoy/car-price-prediction.The car dataset from Kaggle and necessary packages are downloaded in the following lines of code.

```
car_data <- read_csv("data/CarPrice_Assignment.csv")
```

```
## Rows: 205 Columns: 26
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): CarName, fueltype, aspiration, doornumber, carbody, drivewheel, en...
## dbl (16): car_ID, symboling, wheelbase, carlength, carwidth, carheight, curb...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(car_data)
```

```
## Rows: 205
## Columns: 26
## $ car_ID           <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16~
## $ symboling        <dbl> 3, 3, 1, 2, 2, 2, 1, 1, 1, 0, 2, 0, 0, 0, 1, 0, 0, 0,~
## $ CarName          <chr> "alfa-romero giulia", "alfa-romero stelvio", "alfa-ro~
## $ fueltype         <chr> "gas", "gas", "gas", "gas", "gas", "gas", "gas", "gas~
## $ aspiration       <chr> "std", "std", "std", "std", "std", "std", "std", "std~
## $ doornumber       <chr> "two", "two", "two", "four", "four", "two", "four", "~
## $ carbody          <chr> "convertible", "convertible", "hatchback", "sedan", "~
## $ drivewheel       <chr> "rwd", "rwd", "rwd", "fwd", "4wd", "fwd", "fwd", "fwd~
## $ enginelocation   <chr> "front", "front", "front", "front", "front", "front",~
## $ wheelbase        <dbl> 88.6, 88.6, 94.5, 99.8, 99.4, 99.8, 105.8, 105.8, 105~
## $ carlength        <dbl> 168.8, 168.8, 171.2, 176.6, 176.6, 177.3, 192.7, 192.~
## $ carwidth         <dbl> 64.1, 64.1, 65.5, 66.2, 66.4, 66.3, 71.4, 71.4, 71.4,~
## $ carheight        <dbl> 48.8, 48.8, 52.4, 54.3, 54.3, 53.1, 55.7, 55.7, 55.9,~
## $ curbweight       <dbl> 2548, 2548, 2823, 2337, 2824, 2507, 2844, 2954, 3086,~
## $ enginetype       <chr> "dohc", "dohc", "ohcv", "ohc", "ohc", "ohc", "ohc", "~
## $ cylindernumber   <chr> "four", "four", "six", "four", "five", "five", "five"~
## $ enginesize       <dbl> 130, 130, 152, 109, 136, 136, 136, 136, 131, 131, 108~
## $ fuelsystem       <chr> "mpfi", "mpfi", "mpfi", "mpfi", "mpfi", "mpfi", "mpfi~
## $ boreratio        <dbl> 3.47, 3.47, 2.68, 3.19, 3.19, 3.19, 3.19, 3.19, 3.13,~
## $ stroke           <dbl> 2.68, 2.68, 3.47, 3.40, 3.40, 3.40, 3.40, 3.40, 3.40,~
## $ compressionratio <dbl> 9.00, 9.00, 9.00, 10.00, 8.00, 8.50, 8.50, 8.50, 8.30~
## $ horsepower       <dbl> 111, 111, 154, 102, 115, 110, 110, 110, 140, 160, 101~
## $ peakrpm          <dbl> 5000, 5000, 5000, 5500, 5500, 5500, 5500, 5500, 5500,~
## $ citympg          <dbl> 21, 21, 19, 24, 18, 19, 19, 19, 17, 16, 23, 23, 21, 2~
## $ highwaympg       <dbl> 27, 27, 26, 30, 22, 25, 25, 25, 20, 22, 29, 29, 28, 2~
## $ price            <dbl> 13495.00, 16500.00, 16500.00, 13950.00, 17450.00, 152~
```

General Description of Variables

The following is the data dictionary of our dataset that gives a clear, general description of our variables/covariates that can be used in the model. - symboling: Its assigned insurance risk rating (Categroical) - carCompany: Name of car campany (Categorical) - fueltype: Car fuel type i.e gas or diesel (Categorical) - aspiration: Aspiration used in a car (Categorical) - doornumber: Number of doors in a car (Categorical) - carbody: Body of car (Categorical) - drivewheel: Type of drive wheel(Categorical) - enginelocation: Location of car engine (Categorical) - wheelbase: Wheelbase of car (Numeric) - carlength: Length of car (Numeric) - carwidth: Width of car (Numeric) - carheight: Height of car (Numeric) - curbweight: The weight of a car withoput occupants or baggage (Numeric) - enginetype: Type of engine (Categorical) - cylindernumber: Cylinder placed in car (Categorical) - enginesize: Size of car (Numeric) - fuelsystem: Fuel Sytem of car (Categorical) - boreratio: Boreration of car (Numeric) - stroke: Stroke or volume inside the engine (Numeric) - compressionratio: compression ratio of car (Numeric) - horsepower: Horsepower (Numeric) - peakrpm: car peak rpm (Numeric) - citympg: mileage in city (Numeric) - highwaympg: milaege on highway (Numeric) - price: price of car (Numeric)

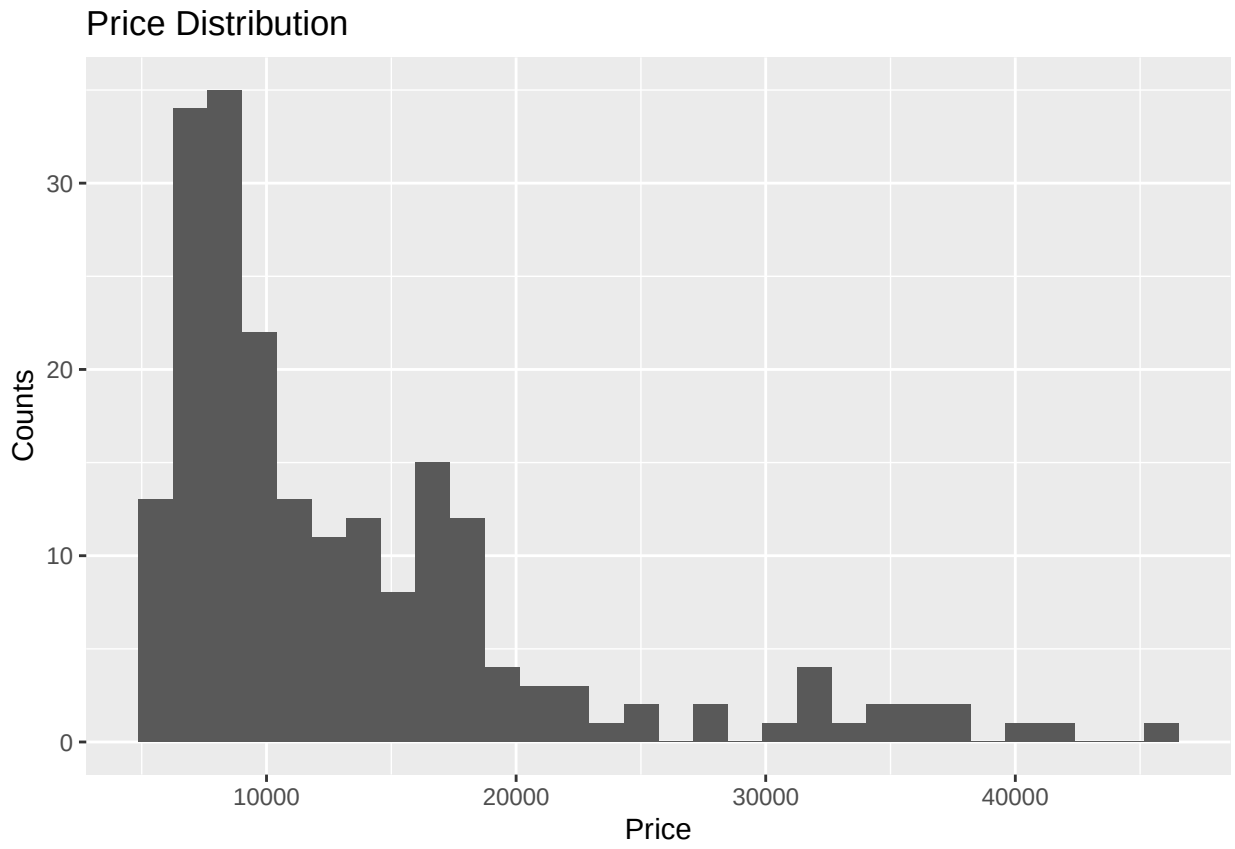
Response Variable: Price

The response variable, price, is the price of the car in our dataset. In order to be able to predict price, we performed some initial univariate analysis of price to observe its spread in the dataset.

```
ggplot(data = car_data, aes(x = price)) +
  geom_histogram() +
```

```
  labs(x = "Price",
       y = "Counts",
       title = "Price Distribution")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The response variable, price, seems unimodal meaning that there is one peak. It also seems to be skewed to the right where there are many datapoints that have price around 5,000-10,000 but there are a few outliers that have price over than 25,000 dollars. To follow up with our analysis, we also created summary statistics for price to see if the statistics reflected the graph we observed.

```
car_data %>%
  summarise(min = min(price),
            q1 = quantile(price, probs = c(0.25)),
            median = median(price),
            q3 = quantile(price, probs = c(0.75)),
            max = max(price),
            iqr = IQR(price),
            mean = mean(price),
            std_dev = sd(price)
            )
```

```
## # A tibble: 1 x 8
##     min    q1 median    q3   max   iqr   mean std_dev
##   <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl>  <dbl>   <dbl>
## 1  5118  7788  10295 16503 45400  8715 13277.   7989.
```

It seems like our summary statistics further support the graph where the quantiles q1 and q2 are much smaller due to the concentration of points based on the distance of min, q1, and median compared to the distance of median, q3, and max.

EDA

Univariate

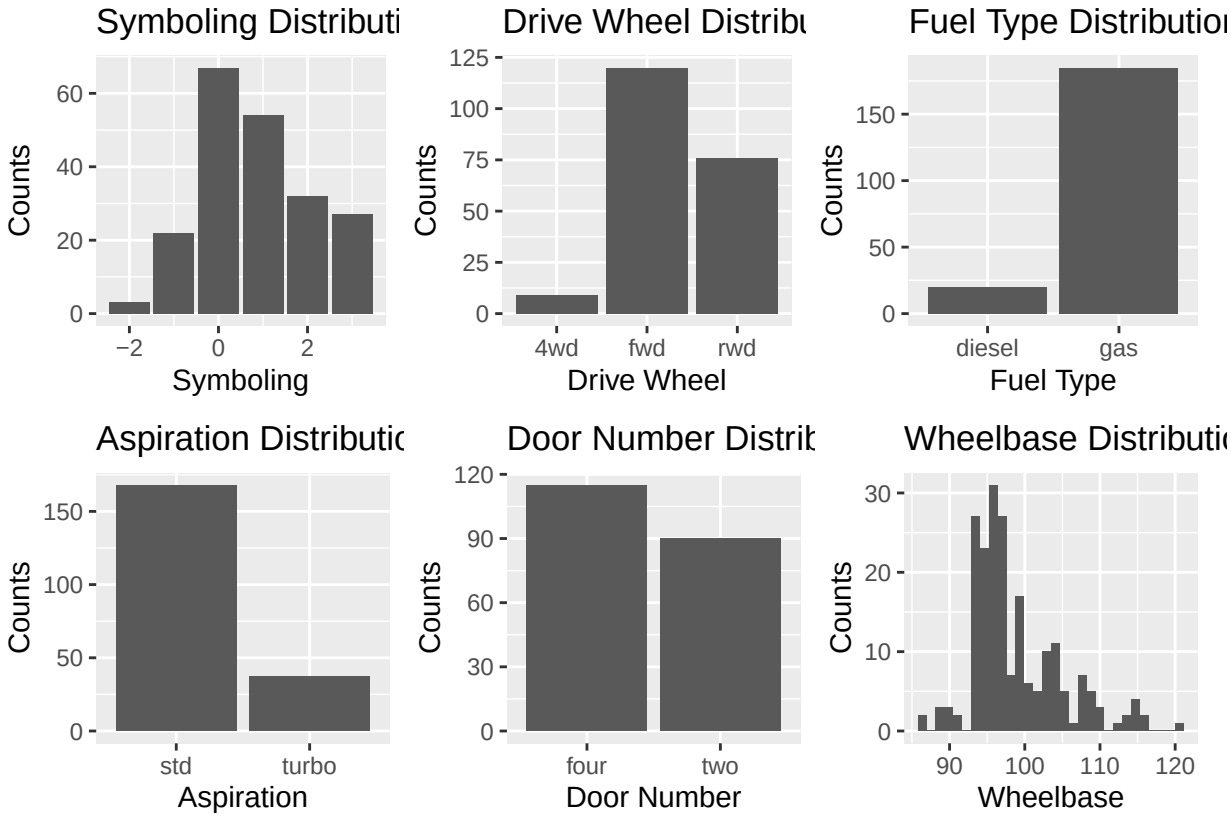
In the following code block, we plotted the 23 covariates/possible predictor variables to do a simple univariate analysis. We used bar graphs for categorical variables and histograms for continuous variables. We formatted the graphs to be able to optimize for space on the pdf and still be able to see the visualization analysis for each variable.

```
p1 <- ggplot(data = car_data, aes(x = symboling)) +
  geom_bar() +
  labs(x = "Symboling",
       y = "Counts",
       title = "Symboling Distribution")

p2 <- ggplot(data = car_data, aes(x = drivewheel)) +
  geom_bar() +
  labs(x = "Drive Wheel",
       y = "Counts",
       title = "Drive Wheel Distribution")

p3 <- ggplot(data = car_data, aes(x = fueltype)) +
  geom_bar() +
  labs(x = "Fuel Type",
       y = "Counts",
       title = "Fuel Type Distribution")

p4 <- ggplot(data = car_data, aes(x = aspiration)) +
  geom_bar() +
  labs(x = "Aspiration",
       y = "Counts",
       title = "Aspiration Distribution")

p5 <- ggplot(data = car_data, aes(x = doornumber)) +
  geom_bar() +
  labs(x = "Door Number",
       y = "Counts",
       title = "Door Number Distribution")

p6 <- ggplot(data = car_data, aes(x = carbody)) +
  geom_bar() +
  labs(x = "Car Body",
       y = "Counts",
       title = "Car Body Distribution")

p7 <- ggplot(data = car_data, aes(x = CarName)) +
  geom_bar() +
  labs(x = "Car Name",
       y = "Counts",
```

```
       title = "Car Name Distribution")

p8 <- ggplot(data = car_data, aes(x = wheelbase)) +
  geom_histogram() +
  labs(x = "Wheelbase",
       y = "Counts",
       title = "Wheelbase Distribution")

p9 <- ggplot(data = car_data, aes(x = carlength)) +
  geom_histogram() +
  labs(x = "Car Length",
       y = "Counts",
       title = "Car Length Distribution")

p10 <- ggplot(data = car_data, aes(x = carwidth)) +
  geom_histogram() +
  labs(x = "Car Width",
       y = "Counts",
       title = "Car Width Distribution")

p11 <- ggplot(data = car_data, aes(x = carheight)) +
  geom_histogram() +
  labs(x = "Car Height",
       y = "Counts",
       title = "Car Height Distribution")

p12 <- ggplot(data = car_data, aes(x = curbweight)) +
  geom_histogram() +
  labs(x = "Curb Weight",
       y = "Counts",
       title = "Curb Weight Distribution")

p13 <- ggplot(data = car_data, aes(x = enginetype)) +
  geom_bar() +
  labs(x = "Engine Type",
       y = "Counts",
       title = "Engine Type Distribution")

p14 <- ggplot(data = car_data, aes(x = cylindernumber)) +
  geom_bar() +
  labs(x = "Cylinder Number",
       y = "Counts",
       title = "Cylinder Number Distribution")

p15 <- ggplot(data = car_data, aes(x = enginesize)) +
  geom_histogram() +
  labs(x = "Engine Size",
       y = "Counts",
       title = "Engine Size Distribution")

p16 <- ggplot(data = car_data, aes(x = fuelsystem)) +
  geom_bar() +
  labs(x = "Fuel System",
```

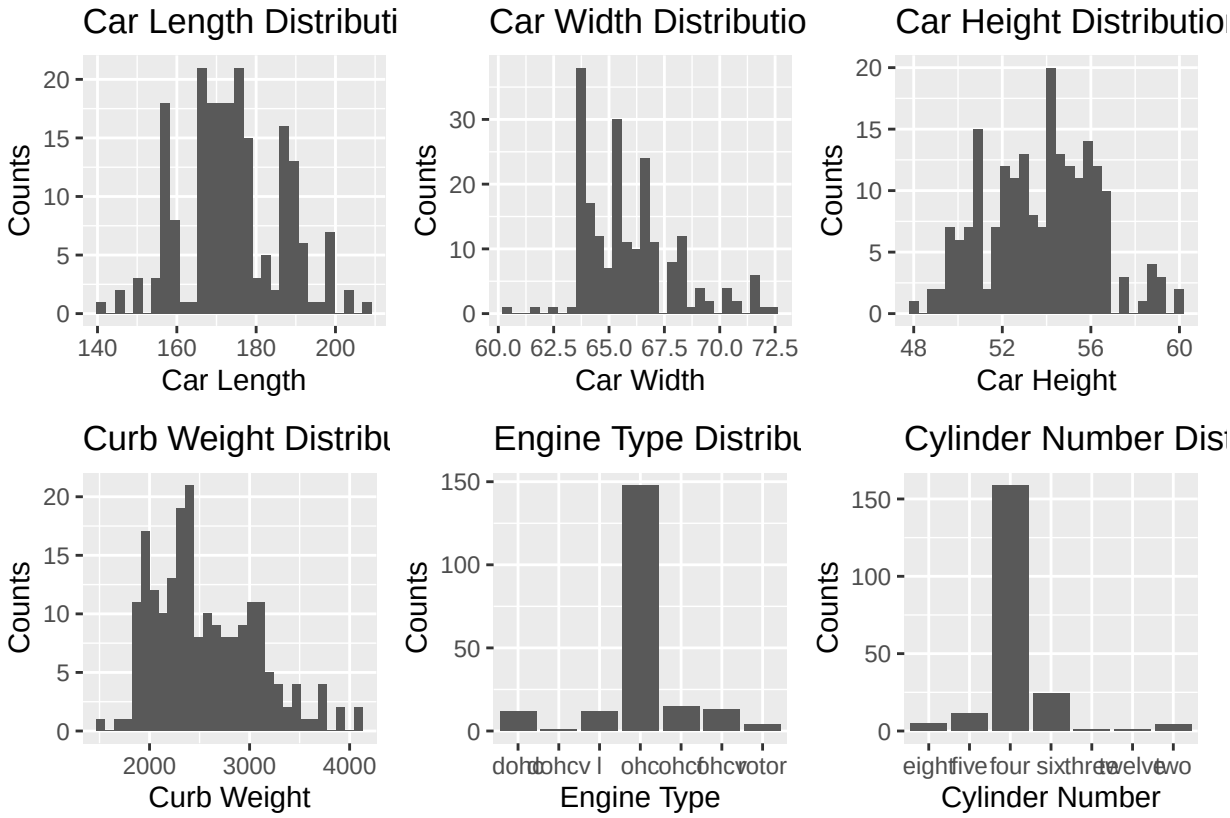
```
       y = "Counts",
       title = "Fuel System Distribution")

p17 <- ggplot(data = car_data, aes(x = boreratio)) +
  geom_histogram() +
  labs(x = "Bore Ratio",
       y = "Counts",
       title = "Bore Ratio Distribution")

p18 <- ggplot(data = car_data, aes(x = stroke)) +
  geom_histogram() +
  labs(x = "Stroke",
       y = "Counts",
       title = "Stroke Distribution")

p19 <- ggplot(data = car_data, aes(x = compressionratio)) +
  geom_histogram() +
  labs(x = "Compression Ratio",
       y = "Counts",
       title = "Compression Ratio Distribution")

p20 <- ggplot(data = car_data, aes(x = horsepower)) +
  geom_histogram() +
  labs(x = "Horsepower",
       y = "Counts",
       title = "Horsepower Distribution")

p21 <- ggplot(data = car_data, aes(x = peakrpm)) +
  geom_histogram() +
  labs(x = "Peak RPM",
       y = "Counts",
       title = "Peak RPM Distribution")

p22 <- ggplot(data = car_data, aes(x = citympg)) +
  geom_histogram() +
  labs(x = "City MPG",
       y = "Counts",
       title = "City MPG Distribution")

p23 <- ggplot(data = car_data, aes(x = highwaympg)) +
  geom_histogram() +
  labs(x = "Highway MPG",
       y = "Counts",
       title = "Highway MPG Distribution")

(p1+p2+p3)/(p4+p5+p8)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



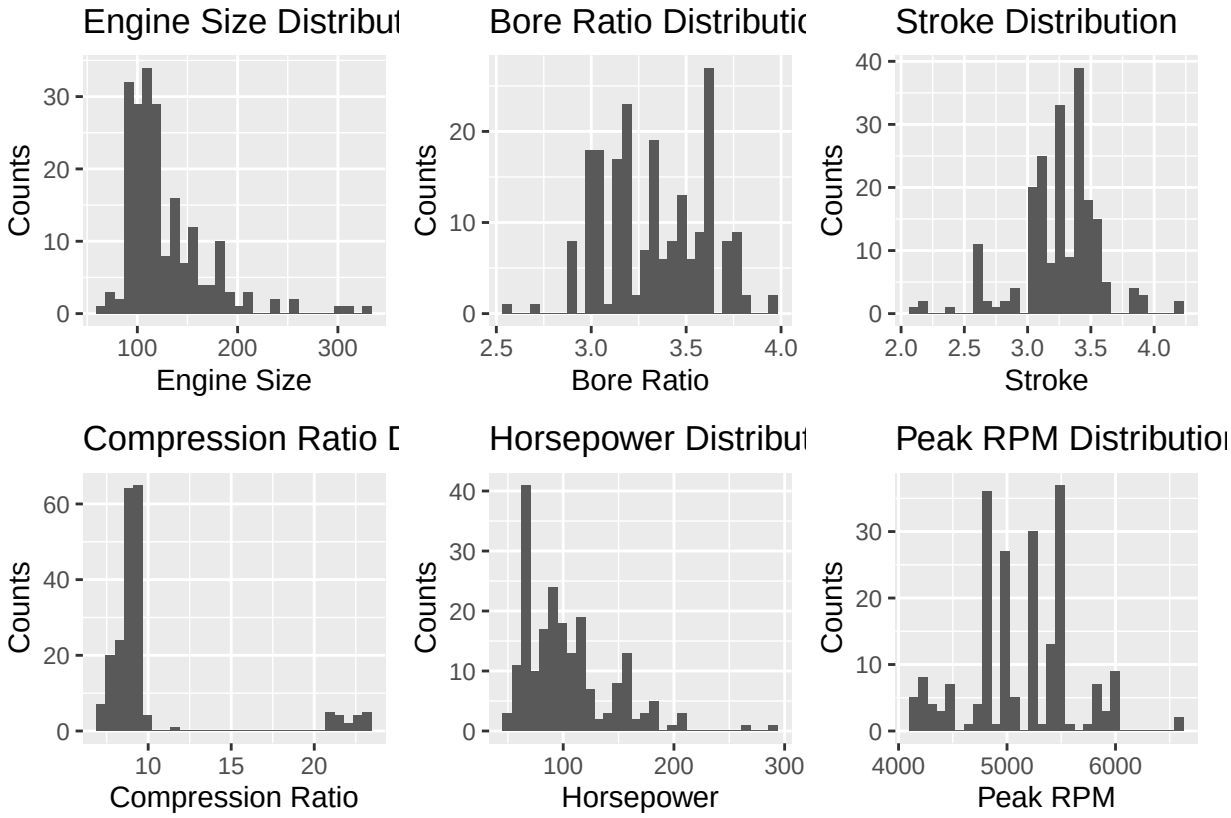

```
(p9+p10+p11)/(p12+p13+p14)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



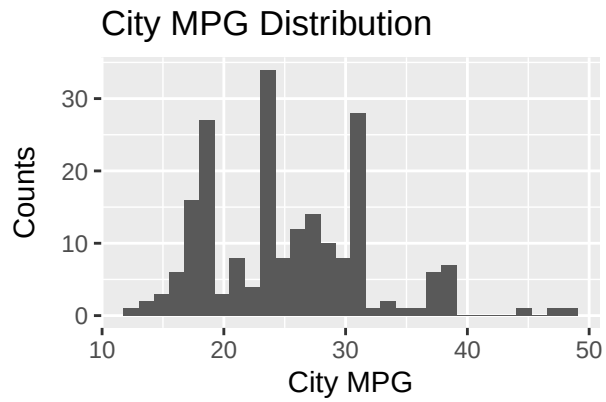

```
(p15+p17+p18)/(p19+p20+p21)
```

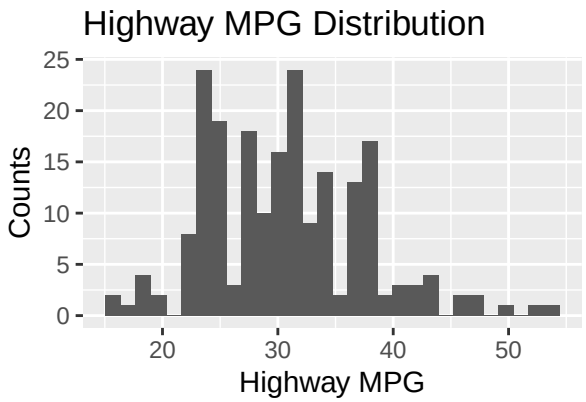
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

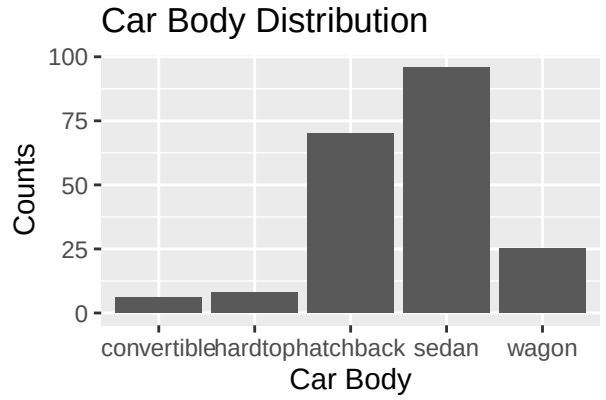


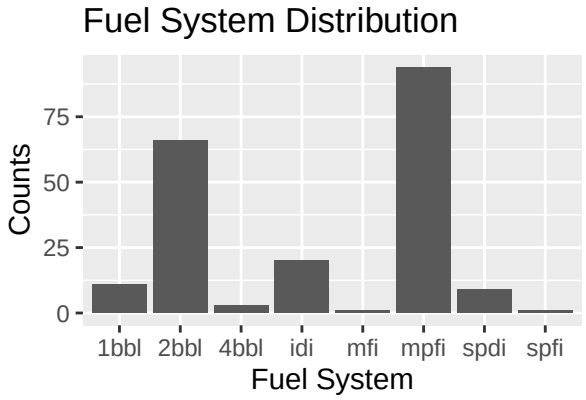

```
(p22+p23)/(p6 + p16)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

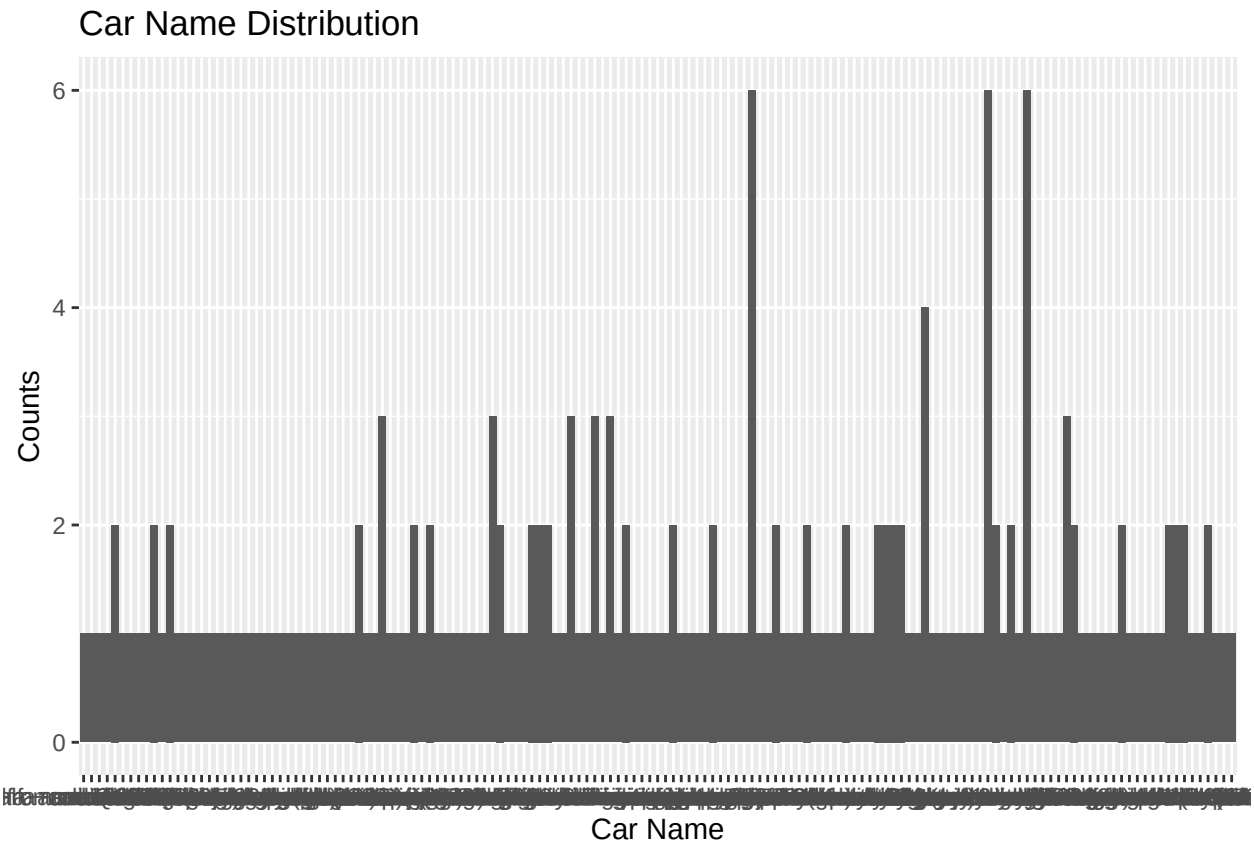

City MPG Distribution
Counts
City MPG
0
10
20
30
10
20
30
40
50


Highway MPG Distribution
Counts
Highway MPG
0
5
10
15
20
25
20
30
40
50


Car Body Distribution
Counts
Car Body
0
25
50
75
100
convertible
hardtop
hatchback
sedan
wagon


Fuel System Distribution
Counts
Fuel System
0
25
50
75
1bbl
2bbl
4bbl
idi
mfi
mpfi
spdi
spfi

p7



From the first six graphs, most of the bar graphs have uneven distribution of observations across all categories. For the histograms, they all seemed to be unimodal, but some seemed a bit skewed to the right with outliers. Moving onto the next six graphs, the two bar graphs seems very skewed where one category in enginetype and cylindernumber have most of the observations. In the histograms, most of the graphs seem either unimodal or bimodal and most of the graphs are skewed to the right. The next six graphs are all histograms where most graphs have smaller peaks. A third of the histograms seem to have peak in the middle. Another third seem to have a peak on the left and are skewed right. The last third have a peak a bit to the right and are a bit skewed to the left. For the next four graphs, the histograms seem to be trimodal where the middle peak is generally the highest and the bar graphs have two categories that have most of the observations. The next last graph has too many categories that have a count of 1 with a few of the categories having a count of 6.

After looking at all the graphs, we wanted to see the summary statistics of the univariate variables so we ran the summary method to see the individual statistics of each of our possible covariates.

```
summary(car_data)
```

```
##      car_ID      symboling         CarName            fueltype
##  Min.   :  1   Min.   :-2.0000   Length:205         Length:205
##  1st Qu.: 52   1st Qu.: 0.0000   Class :character   Class :character
##  Median :103   Median : 1.0000   Mode  :character   Mode  :character
##  Mean   :103   Mean   : 0.8341
##  3rd Qu.:154   3rd Qu.: 2.0000
##  Max.   :205   Max.   : 3.0000
##   aspiration         doornumber          carbody           drivewheel
##  Length:205         Length:205         Length:205         Length:205
##  Class :character   Class :character   Class :character   Class :character
```

```
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  enginelocation       wheelbase        carlength        carwidth
##  Length:205         Min.   : 86.60   Min.   :141.1   Min.   :60.30
##  Class :character   1st Qu.: 94.50   1st Qu.:166.3   1st Qu.:64.10
##  Mode  :character   Median : 97.00   Median :173.2   Median :65.50
##                     Mean   : 98.76   Mean   :174.0   Mean   :65.91
##                     3rd Qu.:102.40   3rd Qu.:183.1   3rd Qu.:66.90
##                     Max.   :120.90   Max.   :208.1   Max.   :72.30
##    carheight       curbweight    enginetype        cylindernumber
##  Min.   :47.80   Min.   :1488   Length:205         Length:205
##  1st Qu.:52.00   1st Qu.:2145   Class :character   Class :character
##  Median :54.10   Median :2414   Mode  :character   Mode  :character
##  Mean   :53.72   Mean   :2556
##  3rd Qu.:55.50   3rd Qu.:2935
##  Max.   :59.80   Max.   :4066
##    enginesize     fuelsystem          boreratio        stroke
##  Min.   : 61.0   Length:205         Min.   :2.54   Min.   :2.070
##  1st Qu.: 97.0   Class :character   1st Qu.:3.15   1st Qu.:3.110
##  Median :120.0   Mode  :character   Median :3.31   Median :3.290
##  Mean   :126.9                      Mean   :3.33   Mean   :3.255
##  3rd Qu.:141.0                      3rd Qu.:3.58   3rd Qu.:3.410
##  Max.   :326.0                      Max.   :3.94   Max.   :4.170
##  compressionratio   horsepower       peakrpm        citympg
##  Min.   : 7.00    Min.   : 48.0   Min.   :4150   Min.   :13.00
##  1st Qu.: 8.60    1st Qu.: 70.0   1st Qu.:4800   1st Qu.:19.00
##  Median : 9.00    Median : 95.0   Median :5200   Median :24.00
##  Mean   :10.14    Mean   :104.1   Mean   :5125   Mean   :25.22
##  3rd Qu.: 9.40    3rd Qu.:116.0   3rd Qu.:5500   3rd Qu.:30.00
##  Max.   :23.00    Max.   :288.0   Max.   :6600   Max.   :49.00
##    highwaympg        price
##  Min.   :16.00   Min.   : 5118
##  1st Qu.:25.00   1st Qu.: 7788
##  Median :30.00   Median :10295
##  Mean   :30.75   Mean   :13277
##  3rd Qu.:34.00   3rd Qu.:16503
##  Max.   :54.00   Max.   :45400
```

From these visualizations and statistics, we found the general distributions of each individual covariate which is always good to know before modeling. As we move onto bivariate analysis, we want to see how these distributions change when including price values to plot against them.

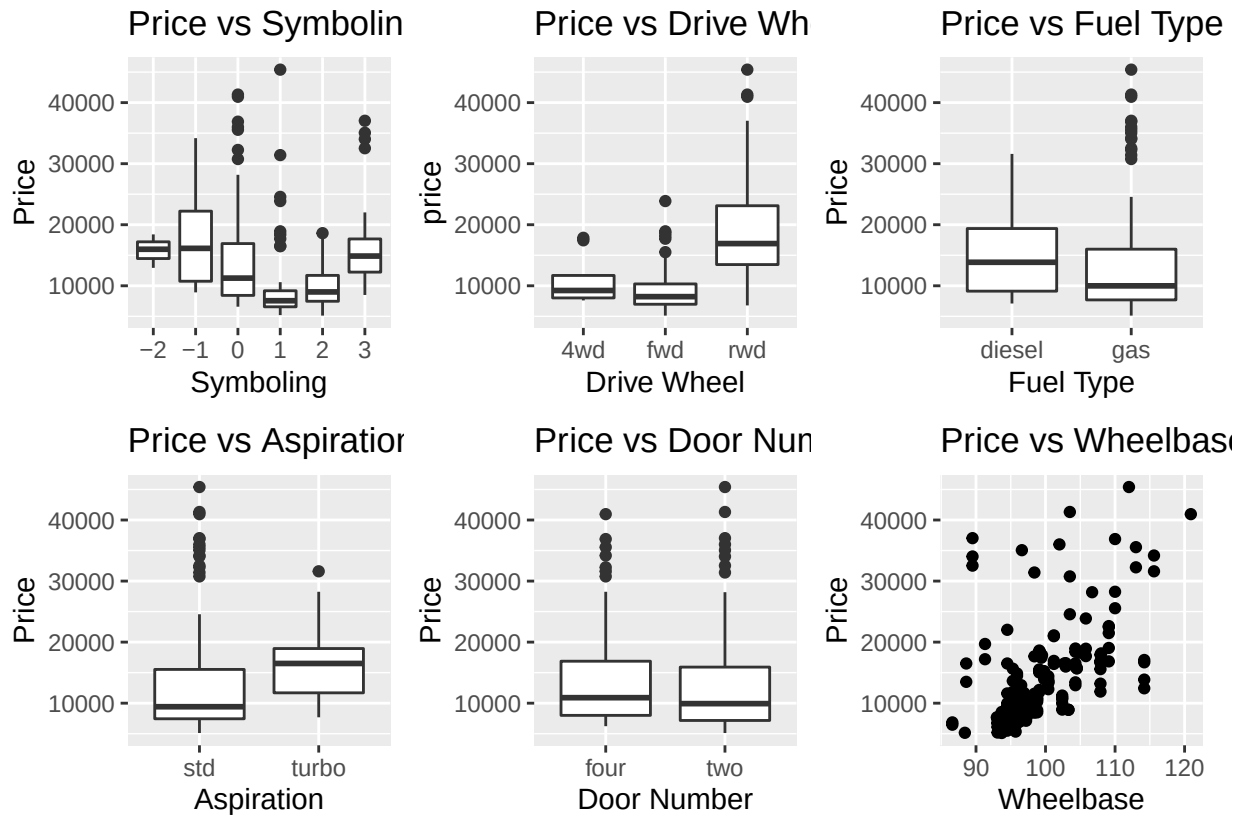
Bivariate

For bivariate analysis, we wanted to analyze each covariate vs price to see the relationship between each one. We want to be able to see first if a covariate could be used to distinguish price values for cars and see if there is a linear relationship between the predictor variable and our response variable. The following block of code gives us price vs each individual covariate using box plots and scatterplots for categorical and continuous variables.

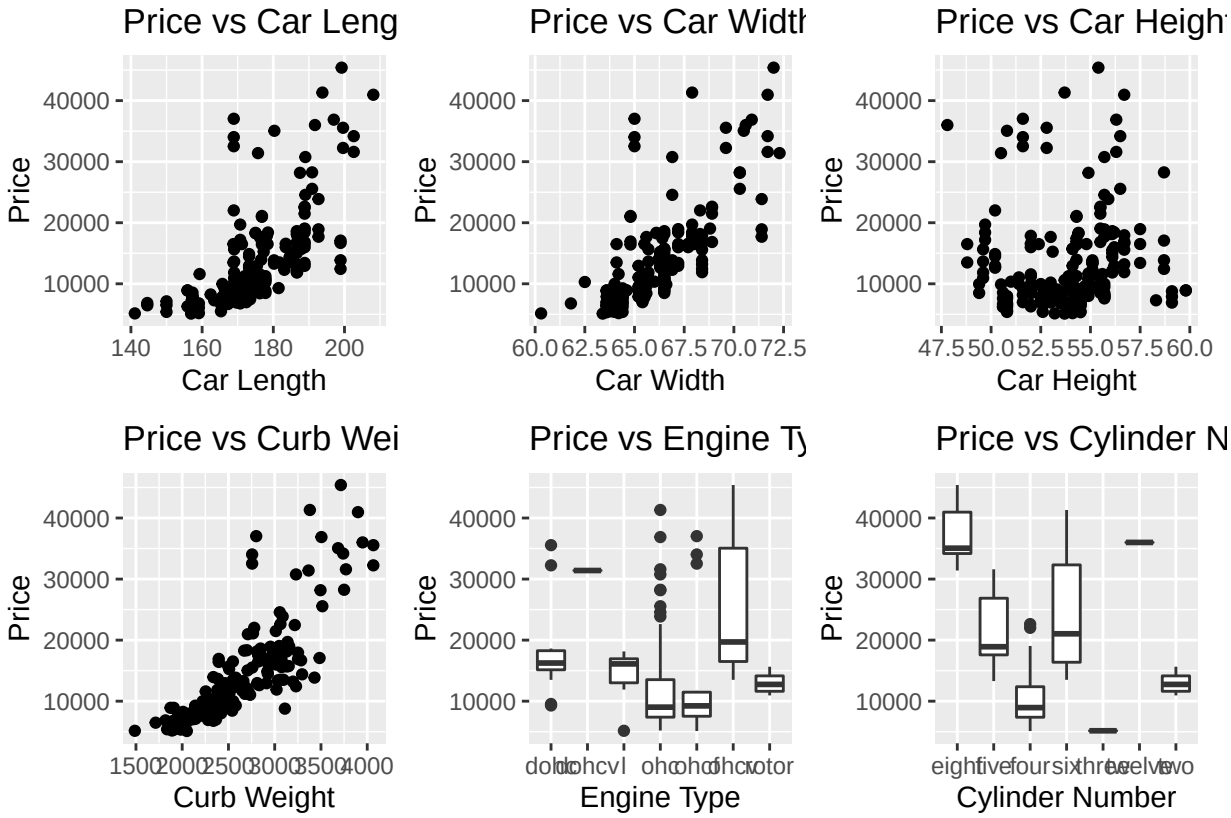
```r
b1 <- ggplot(data = car_data, aes(x = as.factor(symboling), y = price)) +
  geom_boxplot() +
  labs(x = "Symboling",
       y = "Price",
       title = "Price vs Symboling")

b2 <- ggplot(data = car_data, aes(x = drivewheel, y = price)) +
  geom_boxplot() +
  labs(x = "Drive Wheel",
       y = "price",
       title = "Price vs Drive Wheel")

b3 <- ggplot(data = car_data, aes(x = fueltype, y = price)) +
  geom_boxplot() +
  labs(x = "Fuel Type",
       y = "Price",
       title = "Price vs Fuel Type")

b4 <- ggplot(data = car_data, aes(x = aspiration, y = price)) +
  geom_boxplot() +
  labs(x = "Aspiration",
       y = "Price",
       title = "Price vs Aspiration")

b5 <- ggplot(data = car_data, aes(x = doornumber, y = price)) +
  geom_boxplot() +
  labs(x = "Door Number",
       y = "Price",
       title = "Price vs Door Number")

b6 <- ggplot(data = car_data, aes(x = carbody, y = price)) +
  geom_boxplot() +
  labs(x = "Car Body",
       y = "Price",
       title = "Price vs Car Body")

b7 <- ggplot(data = car_data, aes(x = CarName, y = price)) +
  geom_boxplot() +
  labs(x = "Car Name",
       y = "Price",
       title = "Price vs Car Name")

b8 <- ggplot(data = car_data, aes(x = wheelbase, y = price)) +
  geom_point() +
  labs(x = "Wheelbase",
       y = "Price",
       title = "Price vs Wheelbase")

b9 <- ggplot(data = car_data, aes(x = carlength, y = price)) +
  geom_point() +
  labs(x = "Car Length",
       y = "Price",
       title = "Price vs Car Length")
```

```r
b10 <- ggplot(data = car_data, aes(x = carwidth, y = price)) +
  geom_point() +
  labs(x = "Car Width",
       y = "Price",
       title = "Price vs Car Width")

b11 <- ggplot(data = car_data, aes(x = carheight, y = price)) +
  geom_point() +
  labs(x = "Car Height",
       y = "Price",
       title = "Price vs Car Height")

b12 <- ggplot(data = car_data, aes(x = curbweight, y = price)) +
  geom_point() +
  labs(x = "Curb Weight",
       y = "Price",
       title = "Price vs Curb Weight")

b13 <- ggplot(data = car_data, aes(x = enginetype, y = price)) +
  geom_boxplot() +
  labs(x = "Engine Type",
       y = "Price",
       title = "Price vs Engine Type")

b14 <- ggplot(data = car_data, aes(x = cylindernumber, y = price)) +
  geom_boxplot() +
  labs(x = "Cylinder Number",
       y = "Price",
       title = "Price vs Cylinder Number")

b15 <- ggplot(data = car_data, aes(x = enginesize, y = price)) +
  geom_point() +
  labs(x = "Engine Size",
       y = "Price",
       title = "Price vs Engine Size")

b16 <- ggplot(data = car_data, aes(x = fuelsystem, y = price)) +
  geom_boxplot() +
  labs(x = "Fuel System",
       y = "Price",
       title = "Price vs Fuel System")

b17 <- ggplot(data = car_data, aes(x = boreratio, y = price)) +
  geom_point() +
  labs(x = "Bore Ratio",
       y = "Price",
       title = "Price vs Bore Ratio")

b18 <- ggplot(data = car_data, aes(x = stroke, y = price)) +
  geom_point() +
  labs(x = "Stroke",
       y = "Price",
       title = "Price vs Stroke")
```

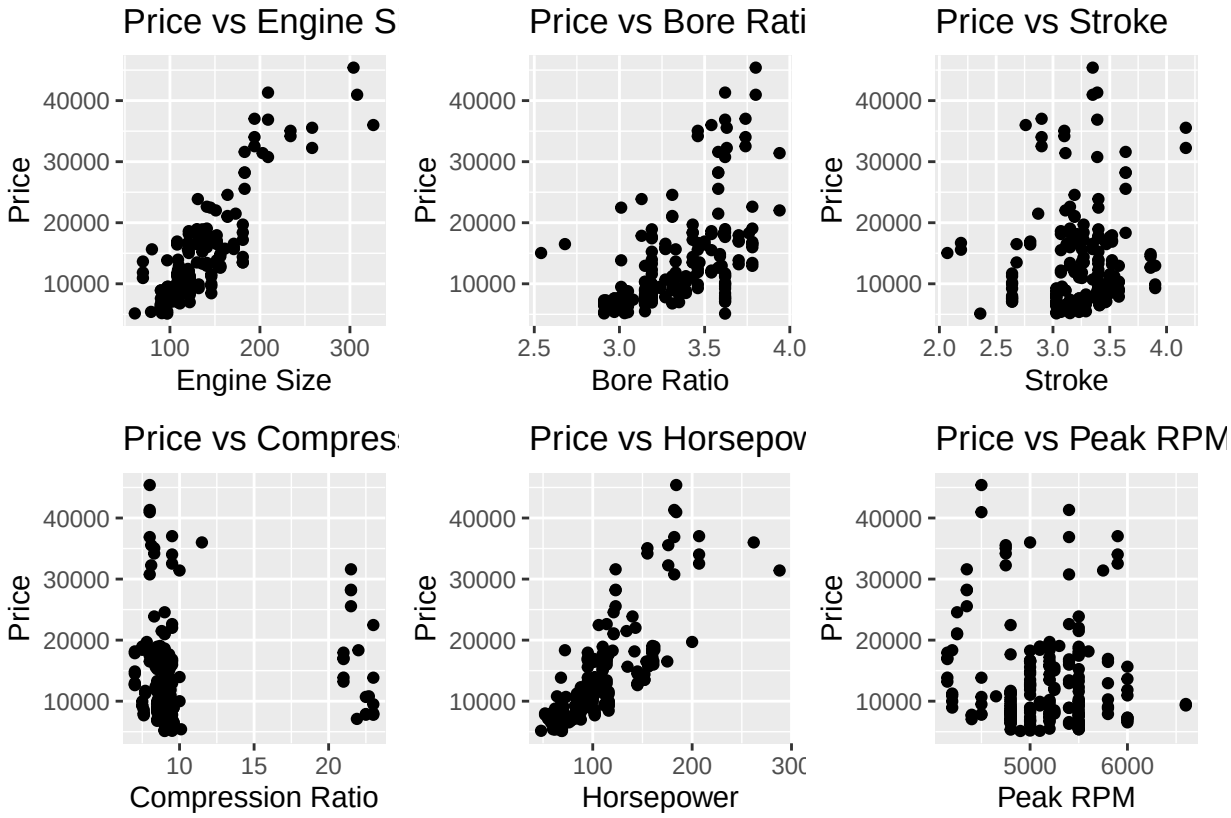
```r
b19 <- ggplot(data = car_data, aes(x = compressionratio, y = price)) +
  geom_point() +
  labs(x = "Compression Ratio",
       y = "Price",
       title = "Price vs Compression Ratio")

b20 <- ggplot(data = car_data, aes(x = horsepower, y = price)) +
  geom_point() +
  labs(x = "Horsepower",
       y = "Price",
       title = "Price vs Horsepower")

b21 <- ggplot(data = car_data, aes(x = peakrpm, y = price)) +
  geom_point() +
  labs(x = "Peak RPM",
       y = "Price",
       title = "Price vs Peak RPM")

b22 <- ggplot(data = car_data, aes(x = citympg, y = price)) +
  geom_point() +
  labs(x = "City MPG",
       y = "Price",
       title = "Price vs City MPG")

b23 <- ggplot(data = car_data, aes(x = highwaympg, y = price)) +
  geom_point() +
  labs(x = "Highway MPG",
       y = "Price",
       title = "Price vs Highway MPG")

(b1+b2+b3)/(b4+b5+b8) # Wheelbase, symboling, drive wheel, aspiration
```



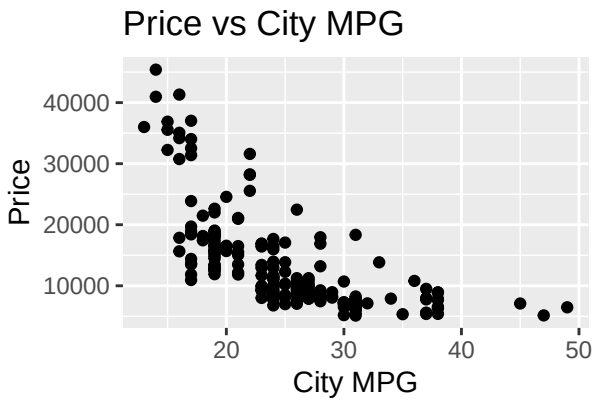

```
(b9+b10+b11)/(b12+b13+b14) # Cylinder num, curb weight, car length, car width
```

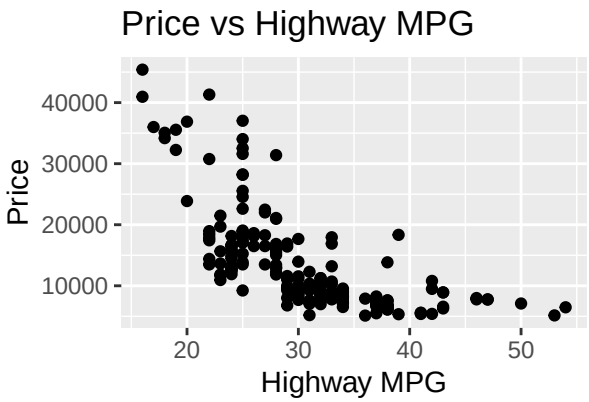


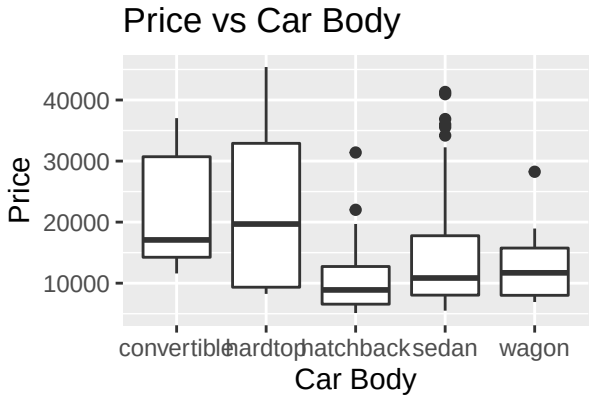

```
(b15+b17+b18)/(b19+b20+b21) # Engine size, horsepower, bore ration
```

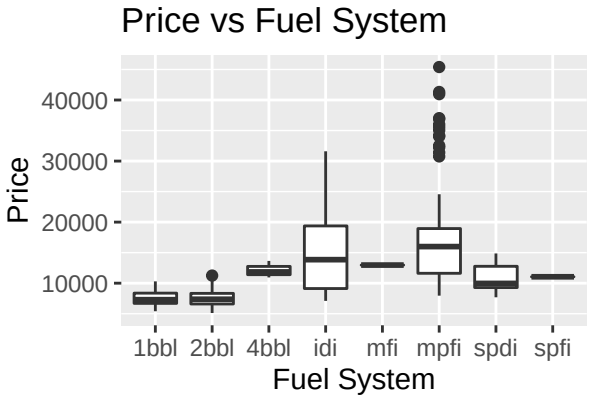



```
(b22+b23)/(b6 + b16) # fuel system, city mpg, highway mpg
```

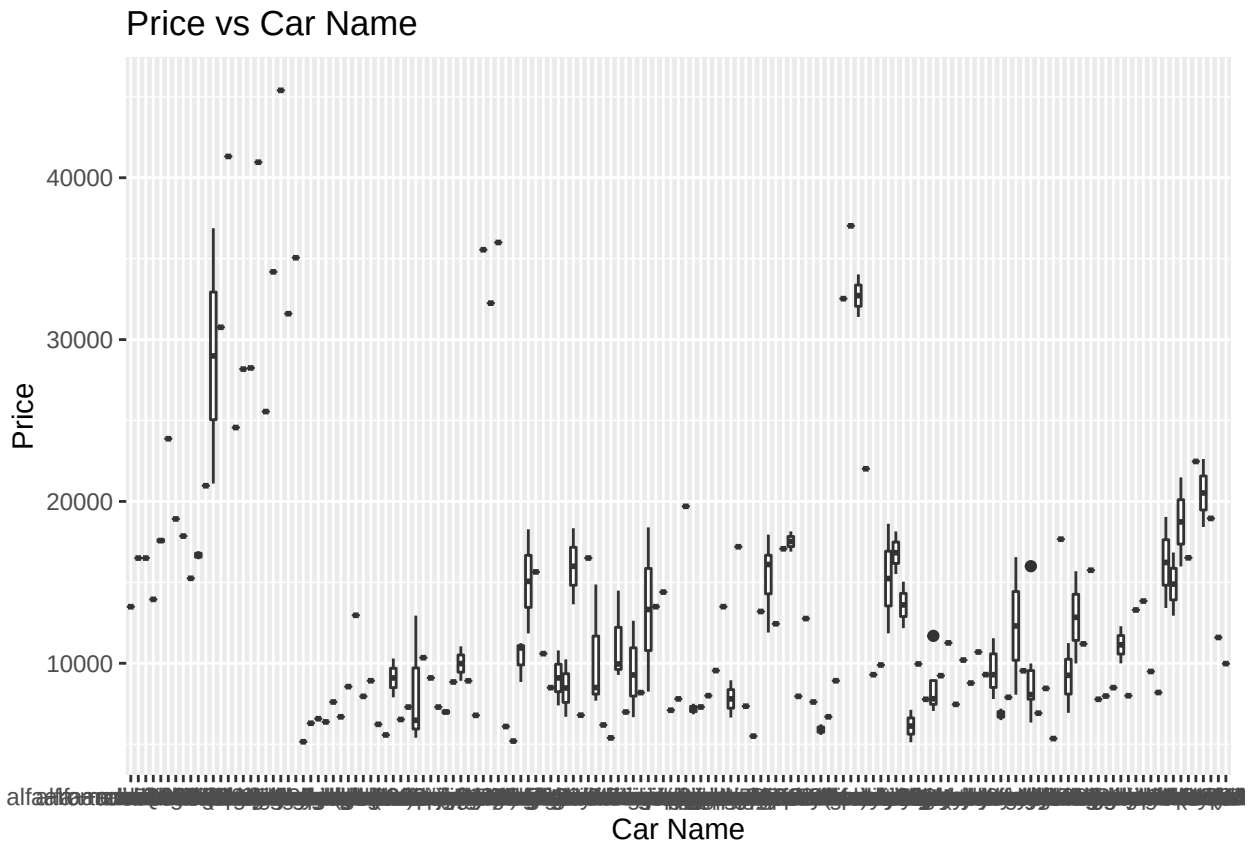

Price vs City MPG
Price
40000
30000
20000
10000
20
30
40
50
City MPG


Price vs Highway MPG
Price
40000
30000
20000
10000
20
30
40
50
Highway MPG


Price vs Car Body
Price
40000
30000
20000
10000
convertible
hardtop
hatchback
sedan
wagon
Car Body


Price vs Fuel System
Price
40000
30000
20000
10000
1bbl
2bbl
4bbl
idi
mfi
mpfi
spdi
spfi
Fuel System

```
b7
```

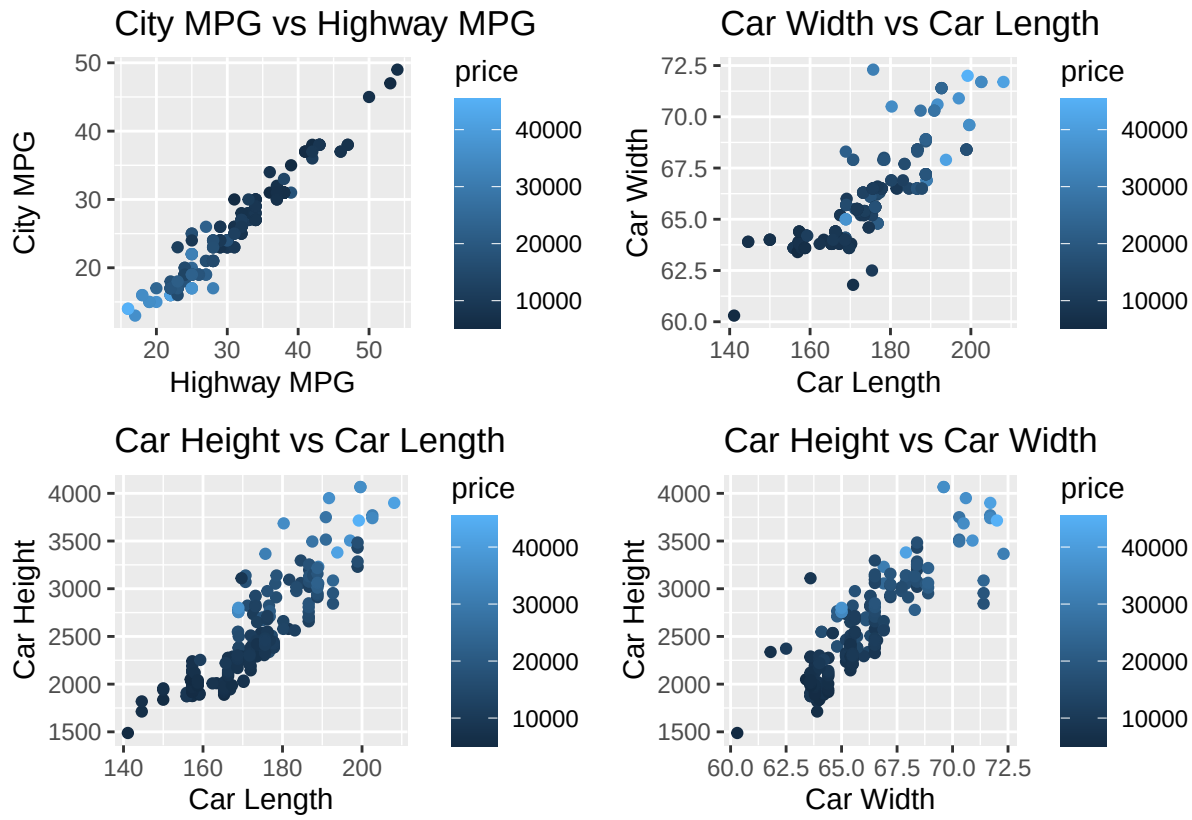


When choosing initial variables to choose for our linear model, we wanted to find scatterplots that had a clear linear relationship and boxplots where categories had different median values and price ranges. From our first six graphs, symboling and drivewheel had categories that had different distributions for price, so they are possible covariates. Aspiration and wheelbase seemed pretty reasonable to be covariates so they were also allowed to continue onto the next stage. Fueltype and doornumber were left out due to the minimal difference between their categories in boxplots. Out of the next six graphs analyzed, carlength, carwidth, and curb weight had the most clear linear relationship from the scatterplots. Car height seemed to have a weak linear relationship and was eliminated. From the boxplots, it seemed like the cylindernumber had each of its field to have their unique distributions for price which allowed it to continue as a possible covariate. Engine type seemed to have much more outliers that caused more overlapping of price ranges across categories so we decided to leave it out of further analysis. For the next six graphs, the scatterplots that looked like they had a linear relationship were enginesize, boreratio, and horsepower. The other three predictor variables, compressionratio, stroke, and peakrpm did not seem to have any linear relationship with price so they were excluded as candidates as covariates. Of the next four graphs presented, citympg and highwaympg have a decreasing linear relationship between price so they are allowed to be included for further analysis. Fuelsystem also seemed that it could be possible to be used as a covariate based on a bit of the spread across the categories. On the other hand, carbody categories had to have quite a bit of overlapping across categories which caused the elimination of further analysis. The last graph, carName, had both make and model of each car which only had a count of one car in each bin which would seem to perform well but, it seemed that there would be a too large spread to be effective to estimate price. Essentially, there were too many categories to be able to predict price so to avoid overfitting carname was excluded from further analysis.

Multivariate

Mutlivariate analysis will be used to determine if there was interactions between particular covariates. Specifically the ones we believed that would be used in the model and seemed to be associated with each other. The multivariate analysis we did was against highwaympg and citympg since there were miles per gallon variables and would probably be associated with each other. In addition to these two covariates, it was believed that carlength, carwidth, and curbweight would also have interactions since the bigger the car, generally the bigger the width, length and weight of the car. The following blocks of code plots the covariates in scatterplots against each other since they are continuous variables which price colored, with lighter blue representing high car price and dark blue being low car prices.

```
m1 <- ggplot(data = car_data, aes(x = highwaympg, y = citympg, color = price)) +
  geom_point() +
  labs(x = "Highway MPG",
       y = "City MPG",
       title = "City MPG vs Highway MPG")


m2 <- ggplot(data = car_data, aes(x = carlength, y = carwidth, color = price)) +
  geom_point() +
  labs(x = "Car Length",
       y = "Car Width",
       title = "Car Width vs Car Length")

m3 <- ggplot(data = car_data, aes(x = carlength, y = curbweight, color = price)) +
  geom_point() +
  labs(x = "Car Length",
       y = "Car Height",
       title = "Car Height vs Car Length")

m4 <- ggplot(data = car_data, aes(x = carwidth, y = curbweight, color = price)) +
  geom_point() +
  labs(x = "Car Width",
       y = "Car Height",
       title = "Car Height vs Car Width")

(m1+m2)/(m3+m4)
```



Due to the clear relationships between citympg and highwaympg, car height and car length, and car height and car width, these interactions may be something to consider when creating the linear model. Car width and car length seemed to have the weakest relationship but, should still be considered when making the model.

Modeling Approach

The modeling approach we decided would be best is to use the fourteen covariates that seemed to have a linear relationship between car price from our bivariate analysis for our model. A multivariate linear model will be used because our outcome of car price is continuous so it makes sense to have a linear model to be able to predict car price with multiple covariates. We wanted to use the AIC and BIC of each model to determine our model selection of which variables would be best to be used in the linear model since we do not want to overfit our model. We wanted to search all possible models so we used the method dredge() from the MuMIn package to be able to create all the possible models from our covariartes and order them by AIC and BIC values.

We wanted to use all the 14 covariates in our dataset that passed our initial bivariate analysis in a grid search using AIC to see which would be the best model based on the lowest AIC value. In the block of code below, a model with the fourteen covariates are made and each combination of the model with the covariates are created and ordered by AIC in ascending order. We selected the top 5 best models and the covariates associated with them.

```
full.model <- lm(price ~ wheelbase+symboling+drivewheel+aspiration+cylindernumber+curbweight*carlength*
dredge(full.model, rank = "AICc")
```

```
## Fixed term is "(Intercept)"
```

```
...
## Global model call: lm(formula = price ~ wheelbase + symboling + drivewheel + aspiration +
##     cylindernumber + curbweight * carlength * carwidth + enginesize +
##     horsepower + boreratio + fuelsystem + citympg * highwaympg,
##     data = car_data, na.action = "na.fail")
## ---
## Model selection table
##             (Int) asp        brr        crl        crw        cty        crb
## 19933   5.118e+05               -2.949e+03 -8.036e+03 -3.018e+02
## 28125   5.765e+05               -3.284e+03 -8.936e+03 -2.743e+02
## 151517  5.834e+05               -3.197e+03 -8.778e+03 -8.473e+02
## 28637   7.065e+05               -3.914e+03 -1.073e+04 -4.660e+02
## 28109   5.338e+05               -3.029e+03 -8.304e+03
...
```

```
# AIC
# 1 carlength, curbweight, citympg, cylindernumber, drivewheel, enginesize, highwaympg, horsepower, car
# 2 carlength, curbweight, citympg, cylindernumber, drivewheel, enginesize, highwaympg, horsepower, whe
# 3 carlength, curbweight, citympg, cylindernumber, drivewheel, enginesize, fuelsystem, highwaympg, hor
# 4 carlength, curbweight, citympg, cylindernumber, drivewheel, enginesize, fuelsystem, highwaympg, hor
# 5 carlength, curbweight, cylindernumber, drivewheel, enginesize, highwaympg, horsepower, wheelbase, c
```

As we mentioned it above, we also wanted to evaluate based on BIC as a indicator of a good model. We repeated the same process by using the same model with all the covariates and doing a grid search using BIC which was ordered in ascending order. Similar to the AIC model, we took the 5 top models for BIC and recorded the variables associated with each model.

```
dredge(full.model, rank = "BIC")
```

```
## Fixed term is "(Intercept)"
```

```
...
## Global model call: lm(formula = price ~ wheelbase + symboling + drivewheel + aspiration +
##     cylindernumber + curbweight * carlength * carwidth + enginesize +
##     horsepower + boreratio + fuelsystem + citympg * highwaympg,
##     data = car_data, na.action = "na.fail")
## ---
## Model selection table
##             (Int) asp        brr        crl        crw        cty        crb
## 18829   4.481e+05               -2.730e+03 -6.930e+03
## 27021   5.309e+05               -3.136e+03 -8.070e+03
## 22925   4.586e+05               -2.785e+03 -7.124e+03
## 18831   4.266e+05     -1.217e+03 -2.590e+03 -6.561e+03
## 19869   5.160e+05               -3.127e+03 -7.969e+03 -4.002e+02
...
```

```
# BIC
# 1 carlength, curbweight, drivewheel, enginesize, horsepower, carlength:curbweight
# 2 carlength, curbweight, drivewheel, enginesize, horsepower, wheelbase, carlength:curbweight
# 3 carlength, curbweight, drivewheel, enginesize, horsepower, symboling, carlength:curbweight
# 4 boreratio, carlength, curbweight, drivewheel, enginesize, horsepower, carlength:curbweight
# 5 carlength, curbweight, citympg, drivewheel, enginesize, highwaympg, horsepower, carlength:curbweigh
```

After observing the top 5 multivariate linear models given by AIC and BIC, we looked to see the overlaps between the top models for each to determine the best model that has both a low AIC and BIC. The AIC top 5 models included much more features than the BIC models so, the model selected was the BIC model that most closely resembled the AIC models. That model was number 5 on the BIC model which contained variables such as citympg, drivewheel, enginesize, and horsepower that were prevalent in the AIC models. The model will also include the interaction between car length and curbwight which was in every AIC and BIC model.

Output of Final Model

It was decided that number 5 for our BIC model was selected where are ANOVA table is displayed on the coefficients is displayed below of the final model selected.

```
final.model <- lm(price ~ carlength*curbweight+citympg+drivewheel+enginesize+highwaympg+horsepower, car
tidy(final.model, conf.int = TRUE) %>%
  kable(format = "markdown", digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	60903.482	17906.787	3.401	0.001	25587.643	96219.320
carlength	-380.101	98.530	-3.858	0.000	-574.423	-185.779
curbweight	-29.539	6.990	-4.226	0.000	-43.326	-15.753
citympg	-387.043	185.084	-2.091	0.038	-752.066	-22.019
drivewheelfwd	-1800.403	1268.908	-1.419	0.158	-4302.948	702.143
drivewheelrwd	423.462	1251.586	0.338	0.735	-2044.921	2891.845
enginesize	72.936	13.482	5.410	0.000	46.347	99.525
highwaympg	348.714	169.546	2.057	0.041	14.335	683.093
horsepower	54.240	14.074	3.854	0.000	26.483	81.996
carlength:curbweight	0.176	0.037	4.711	0.000	0.102	0.249

```
summary(final.model)
```

```
##
## Call:
## lm(formula = price ~ carlength * curbweight + citympg + drivewheel +
##     enginesize + highwaympg + horsepower, data = car_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7564.1 -1334.6  -255.8  1297.7 12093.0
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           6.090e+04  1.791e+04   3.401 0.000814 ***
## carlength            -3.801e+02  9.853e+01  -3.858 0.000155 ***
## curbweight           -2.954e+01  6.990e+00  -4.226 3.66e-05 ***
## citympg              -3.870e+02  1.851e+02  -2.091 0.037808 *
## drivewheelfwd        -1.800e+03  1.269e+03  -1.419 0.157536
## drivewheelrwd         4.235e+02  1.252e+03   0.338 0.735471
## enginesize            7.294e+01  1.348e+01   5.410 1.84e-07 ***
## highwaympg            3.487e+02  1.695e+02   2.057 0.041041 *
```

```
## horsepower            5.424e+01  1.407e+01   3.854 0.000158 ***
## carlength:curbweight  1.755e-01  3.725e-02   4.711 4.67e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3262 on 195 degrees of freedom
## Multiple R-squared:  0.8406, Adjusted R-squared:  0.8332
## F-statistic: 114.3 on 9 and 195 DF,  p-value: < 2.2e-16
```

Assumptions

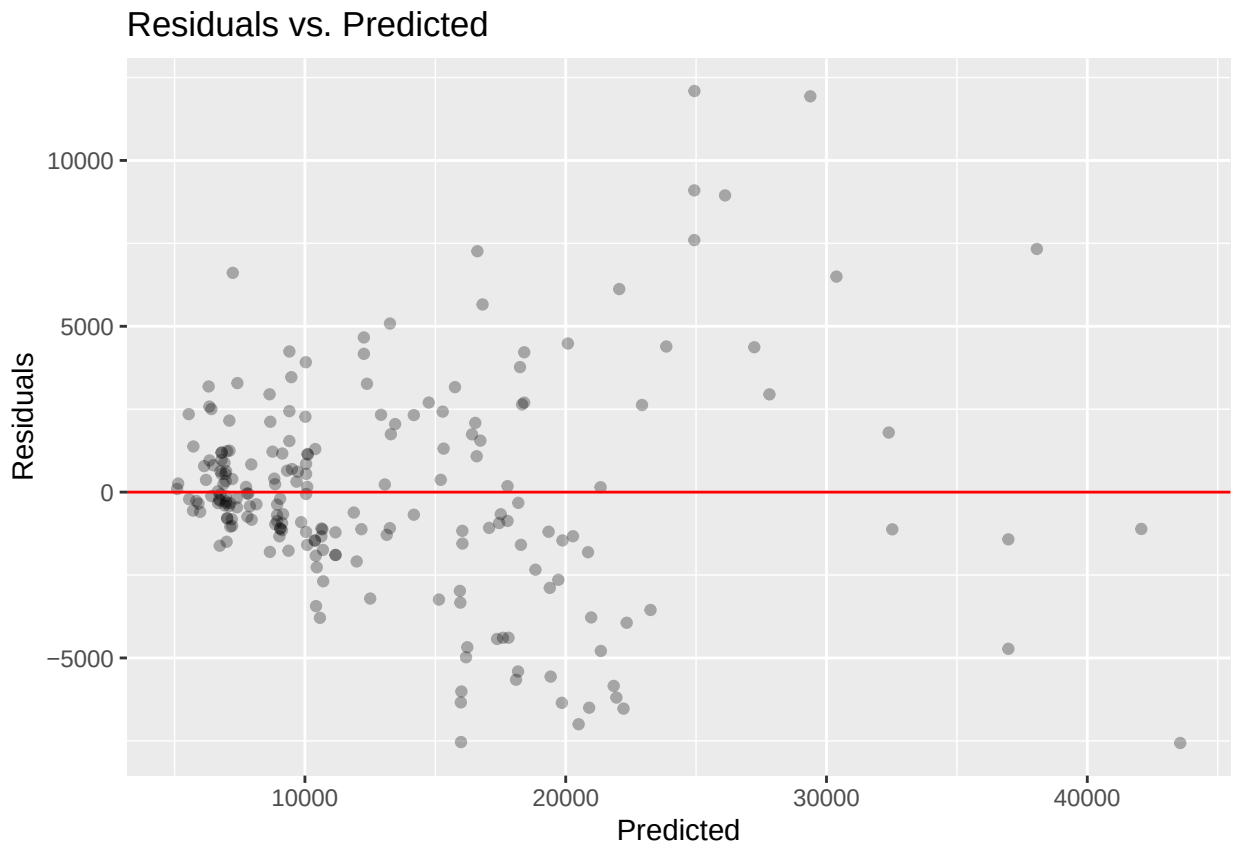
There are some assumptions that should be checked when using a multivariable linear model. These following assumptions should be checked:

- Linearity: Response variable has a linear relationship with predictor variables. There should be no pattern in the plots unless in the case for interaction terms.
  - Residuals vs Predicted Values
  - Residuals vs Every Predictor Variables
- Constant Variance: The regression is the same for all predictor variables. The height cloud of points should be constant across the x-axis or across all predictor variable values.
  - Residuals vs Predicted Values
- Normality: Response variable follows a Normal distribution around its mean for every predictor variable. The histogram should be approximately unimodal and symmetric and the points on the QQ Plot should follow on the diagonal line.
  - Histogram of Residuals
  - Normal QQ-Plot of Residuals
- Independence: All observations are independent. There should not be pattern in residuals across the order of observations.
  - Residuals vs Observation Number

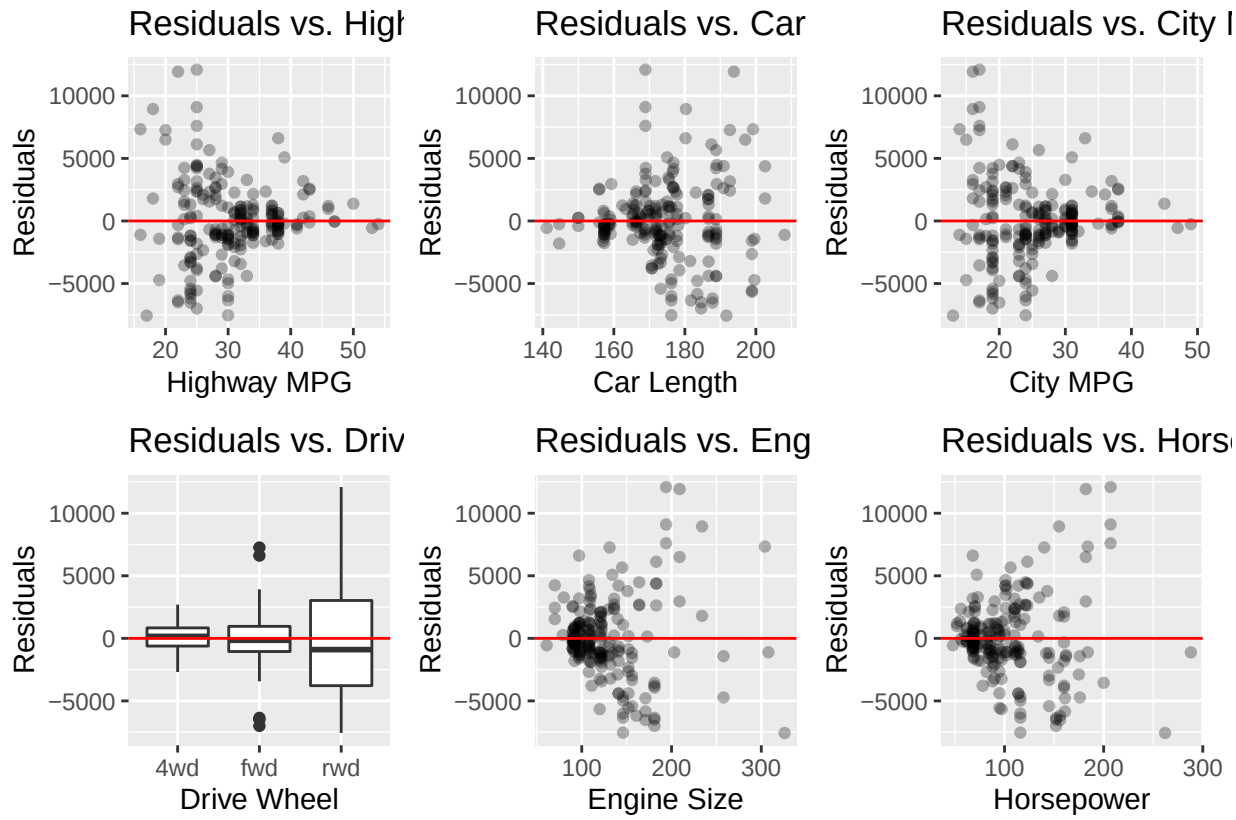
Linearity

```
car_aug <- augment(final.model)
l1 = ggplot(data = car_aug, aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Predicted", y = "Residuals", title = "Residuals vs. Predicted")

l2 = ggplot(data = car_aug, aes(x = carlength, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Car Length", y = "Residuals", title = "Residuals vs. Car Length")

l3 = ggplot(data = car_aug, aes(x = citympg, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "City MPG", y = "Residuals", title = "Residuals vs. City MPG")
```

```
l4 = ggplot(data = car_aug, aes(x = drivewheel, y = .resid)) +
  geom_boxplot() +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Drive Wheel", y = "Residuals", title = "Residuals vs. Drive Wheel")

l5 = ggplot(data = car_aug, aes(x = enginesize, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Engine Size", y = "Residuals", title = "Residuals vs. Engine Size")

l6 = ggplot(data = car_aug, aes(x = horsepower, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Horsepower", y = "Residuals", title = "Residuals vs. Horsepower")

l7 = ggplot(data = car_aug, aes(x = highwaympg, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Highway MPG", y = "Residuals", title = "Residuals vs. Highway MPG")

(l1)
```



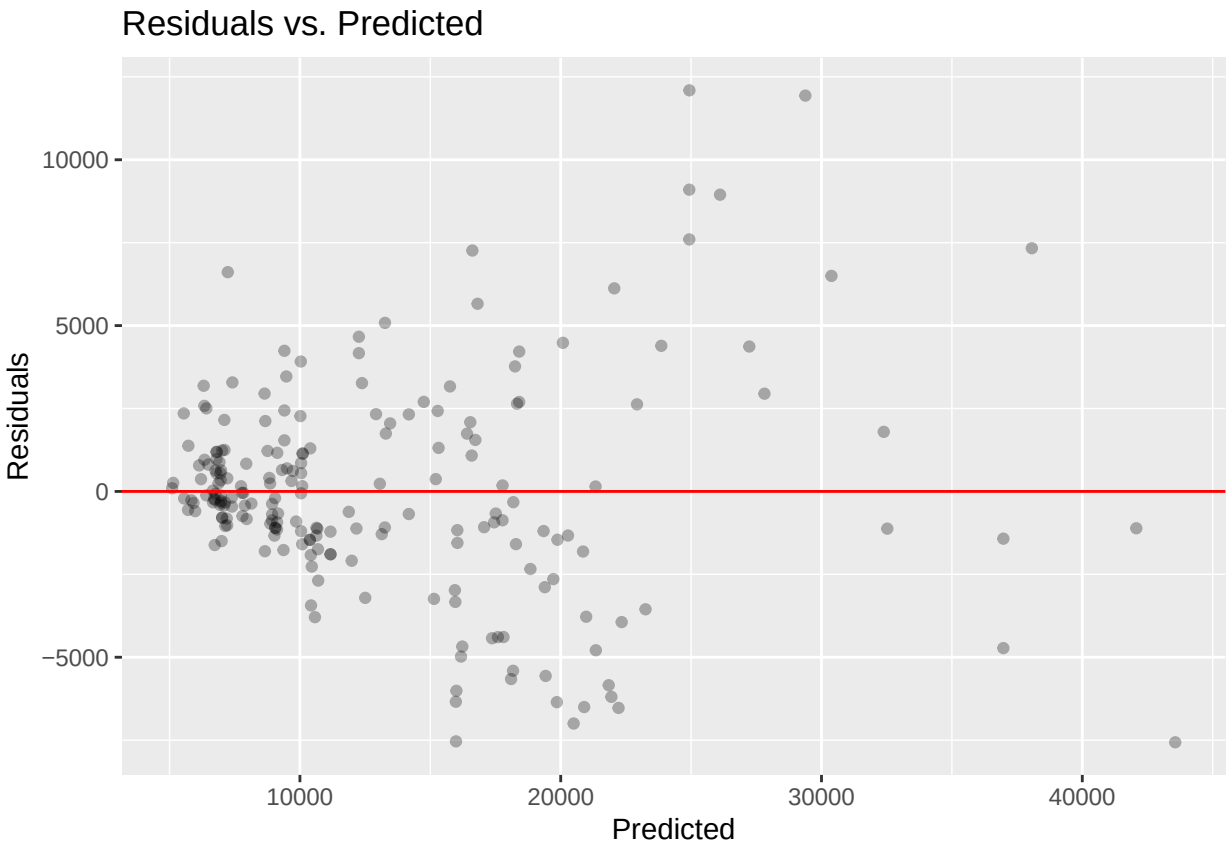

```
(l7+l2+l3)/(l4+l5+l6)
```



Based on the plots for checking the linearity assumption, it seems that the model does not fulfill the linearity assumption since there is a fan pattern in almost every predictor variable. Residuals tend to either increase or decrease as a predictor variable increases so it would seem the linearity assumption is not fulfilled. This also goes for the residuals versus predicted plots where there is also a fan pattern.

Constant Variance

```
ggplot(data = car_aug, aes(x = .fitted, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Predicted", y = "Residuals", title = "Residuals vs. Predicted")
```

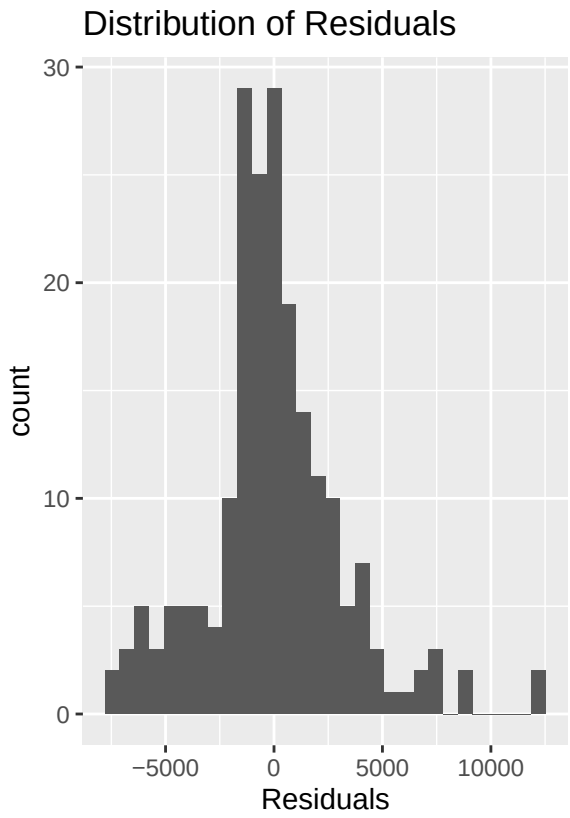


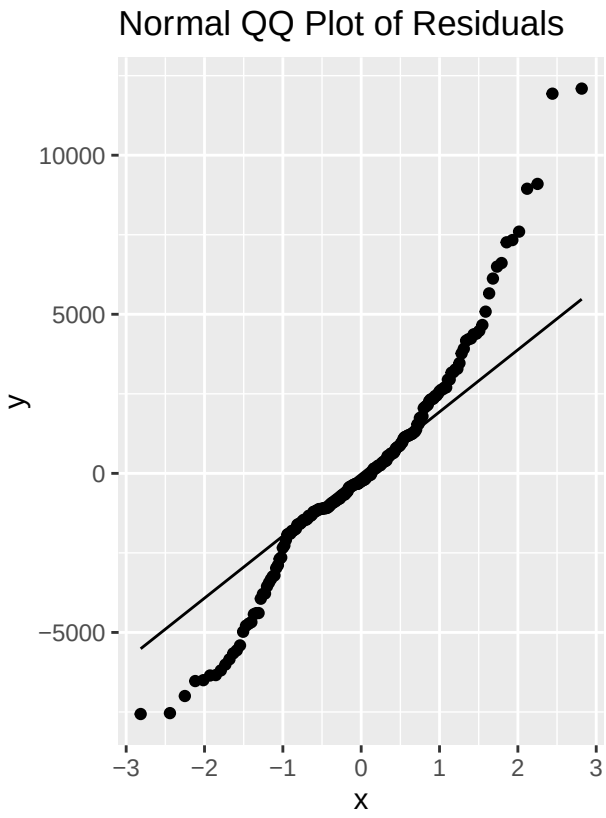
The model does not meet the constant variance assumption since the graph does not follow a constant variance across all predicted variables. The variance of the residuals tend to increase as the predicted variables increase so it has a fanning pattern.

Normal Condition

```
n1 = ggplot(data = car_aug, aes(x = .resid)) +
  geom_histogram() +
  labs(x = "Residuals", title = "Distribution of Residuals")

n2 = ggplot(data = car_aug, aes(sample = .resid)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "Normal QQ Plot of Residuals")

n1+n2
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

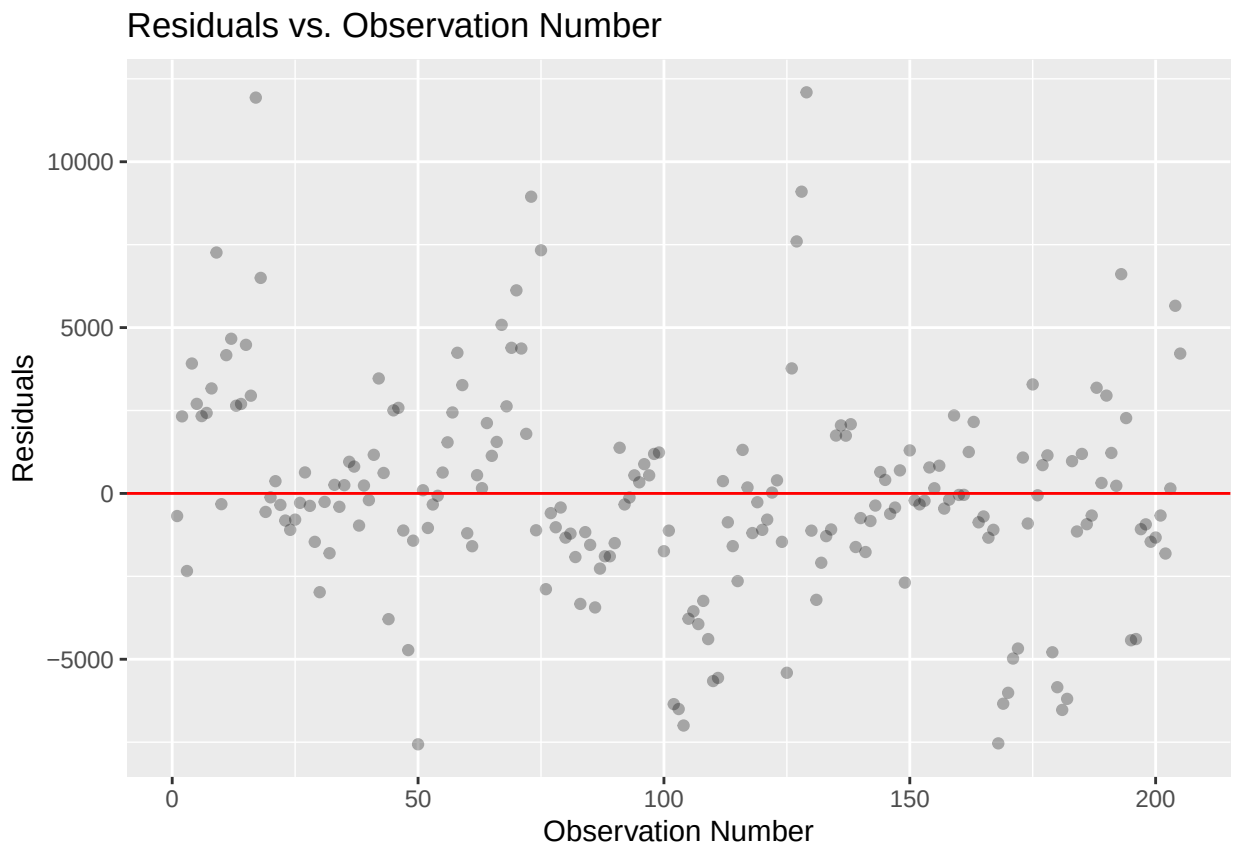




The normal condition is not met since the QQ Plot is not following a completely diagonal line. The tails of the QQ plot are not on the diagonal line. Since the histogram of the residual is not Normal and the QQ Plot is not following the diagonal on the tails, therefore the normal condition is not met for the model.

Independence

```
car_aug <- car_aug %>%
  mutate(obs_num = 1:nrow(car_aug))
ggplot(data = car_aug, aes(x = obs_num, y = .resid)) +
  geom_point(alpha = 0.3) +
  geom_hline(yintercept = 0, color = "red") +
  labs(x = "Observation Number",
       y = "Residuals",
       title = "Residuals vs. Observation Number")
```



The independence condition is met in our case since across the order of observations, there is constant variance meaning that it probably means that the observations are independent from one another.

Interpretations of Model Coefficients

The next part in our analysis is going to refer back to the coefficients in our model. We can display those numbers using the following lines of code.

```
tidy(final.model, conf.int = TRUE) %>%
  kable(format = "markdown", digits = 3)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	60903.482	17906.787	3.401	0.001	25587.643	96219.320
carlength	-380.101	98.530	-3.858	0.000	-574.423	-185.779
curbweight	-29.539	6.990	-4.226	0.000	-43.326	-15.753
citympg	-387.043	185.084	-2.091	0.038	-752.066	-22.019
drivewheelfwd	-1800.403	1268.908	-1.419	0.158	-4302.948	702.143
drivewheelrwd	423.462	1251.586	0.338	0.735	-2044.921	2891.845
enginesize	72.936	13.482	5.410	0.000	46.347	99.525
highwaympg	348.714	169.546	2.057	0.041	14.335	683.093
horsepower	54.240	14.074	3.854	0.000	26.483	81.996
carlength:curbweight	0.176	0.037	4.711	0.000	0.102	0.249

Based on the table of the linear model above, the estimate intercept is telling us the price of a car with a car length of 0, curb weight of 0, citympg of 0, engine size of 0, highwaympg of 0, and four wheel drive which is not realistic at all. The intercept estimate does not tell us anything meaningful relative towards the data model. The coefficient -380.101 for carbody means that the longer the car is by one unit, the price of the car will go down by 380.101 dollars. The coefficient, -29.539 for curb weight means that for every unit the weight goes up by 1, then the car price will generally decrease by 29.54 dollars. For the coefficient -387.043 for citympg means that for every increase in miles per gallon in the city for a car, then the price will generally drop 387.04 dollars. For the coefficients, -1800.40 for fwd in drivewheel and 423.462 for rwd in drivewheel means that relative to the intercept of 4wd then these would be added to the new intercept for observations of fwd and rwd. The coefficient 72.936 for engine size means that the for one unit increase for the enginesize means that there is a general increase of 72.936 in car price. For the coefficient 348.714 for highwaympg generally means that for every increase in mile per gallon a car has on the highway, there would be an increase of an average of 348.72 dollars in car price. For coefficient of horsepower of 54.240, this means that for every increase in horsepower a car has, in general that there is an increase of about 54.24 in car price.