Car Price Prediction

Linear Regression

Index

- Reading and Understanding data
- Data cleaning and Preparation
- Visualizing the data
- Deriving new features
- Bivariate analysis
- Creating dummy variable
- Splitting the train-test and Rescaling the features
- Building the model
- Residual analysis on train set
- Prediction and evaluation

Step 1: Reading and Understanding data

```
In [83]: df.columns

Out[83]: Index(['car_ID', 'symboling', 'CarName', 'fueltype', 'aspiration',
               'doornumber', 'carbody', 'drivewheel', 'enginelocation', 'wheelbase',
               'carlength', 'carwidth', 'carheight', 'curbweight', 'enginetype',
               'cylindernumber', 'enginesize', 'fuelsystem', 'boreratio', 'stroke',
               'compressionratio', 'horsepower', 'peakrpm', 'citympg', 'highwaympg',
               'price'],
              dtype='object')

In [84]: df.shape

Out[84]: (205, 26)
```

Reading the csv file and creating data Frame (df)

Step 2: Data cleaning and Preparation

In [87]:
```
# splitting company name from CarName
# dropping CarName column

df['Company'] = df['CarName'].apply(lambda x: x.split()[0])
df.drop(['CarName'], axis=1, inplace=True)
df.head()
```

Out[87]:

	car_ID	symboling	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	carlength	...	fuelsystem	boreratio	stroke	compres
0	1	3	gas	std	two	convertible	rwd	front	88.6	168.8	...	mpfi	3.47	2.68	
1	2	3	gas	std	two	convertible	rwd	front	88.6	168.8	...	mpfi	3.47	2.68	
2	3	1	gas	std	two	hatchback	rwd	front	94.5	171.2	...	mpfi	2.68	3.47	
3	4	2	gas	std	four	sedan	fwd	front	99.8	176.6	...	mpfi	3.19	3.40	
4	5	2	gas	std	four	sedan	4wd	front	99.4	176.6	...	mpfi	3.19	3.40	

5 rows × 26 columns

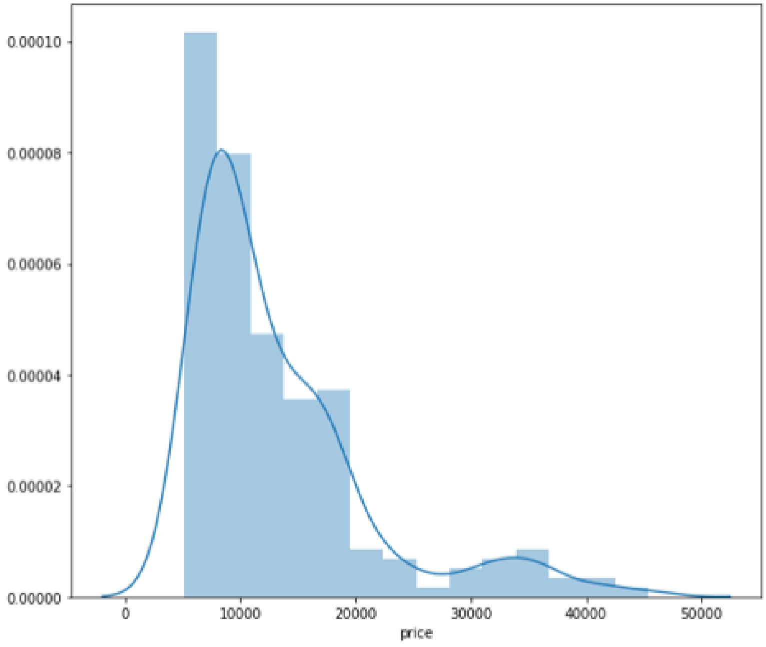
Splitting the Company from CarName variable

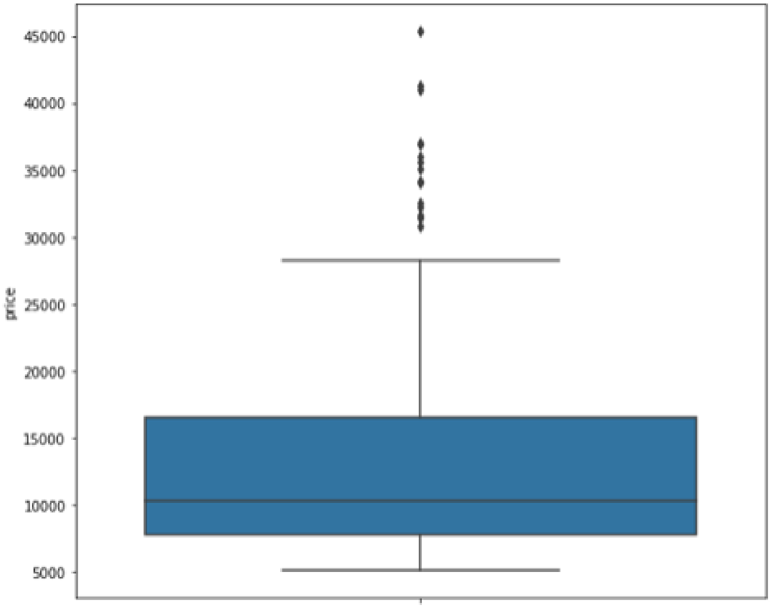
Spelling mistake correction

```
# correcting spelling mistakes of the Company

df.Company = df.Company.str.lower()

def replace_name(a,b):
    df.Company.replace(a,b,inplace=True)

replace_name('maxda','mazda')
replace_name('porcshce','porsche')
replace_name('toyouta','toyota')
replace_name('vokswagen','volkswagen')
replace_name('vw','volkswagen')

df.Company.unique()
```

```
array(['alfa-romero', 'audi', 'bmw', 'chevrolet', 'dodge', 'honda',
       'isuzu', 'jaguar', 'mazda', 'buick', 'mercury', 'mitsubishi',
       'nissan', 'peugeot', 'plymouth', 'porsche', 'renault', 'saab',
       'subaru', 'toyota', 'volkswagen', 'volvo'], dtype=object)
```

Step 3: Visualizing the data

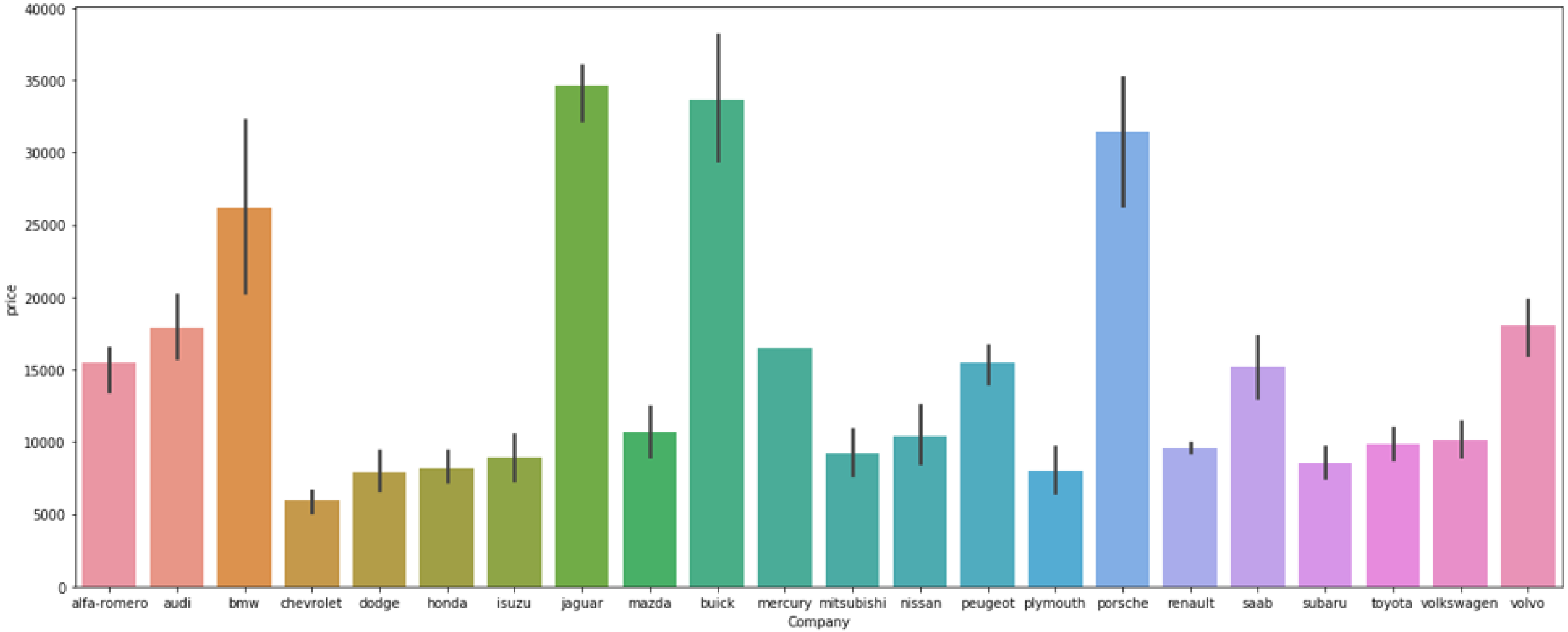




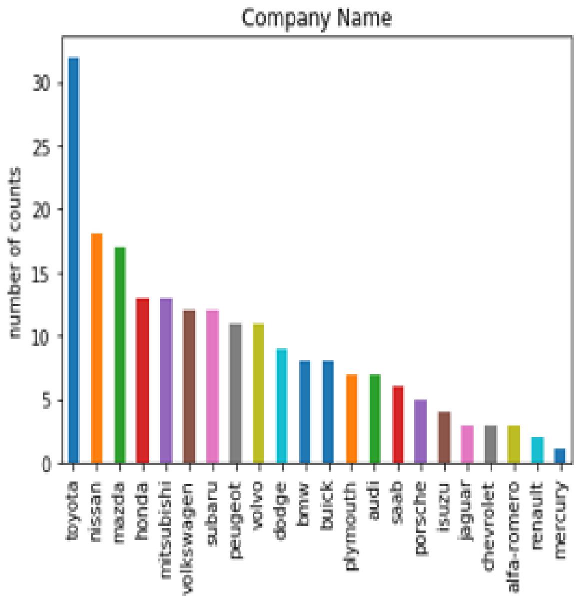
The graph is right skewed and there are not much outliers present in the price variable

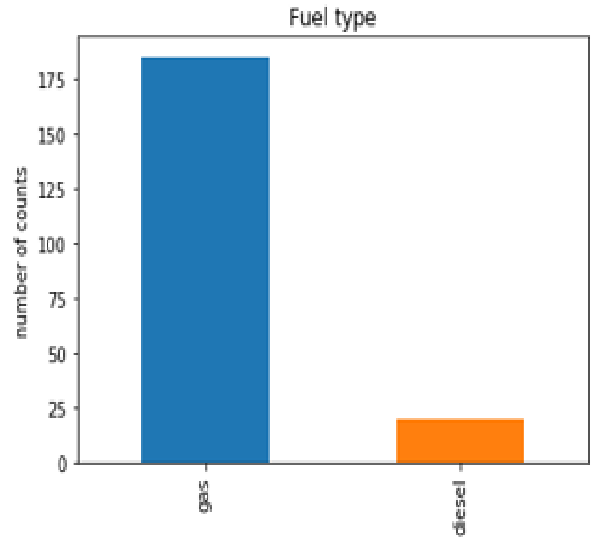
- there is difference between mean and median of the price
  - mean price = 13276.710571
  - median price = 10295.000000
- the distribution of the price is rightly skewed
- the data points have high variance, 85% of the data has price below 18500 and 15% of the data has price between 22563 to 45400

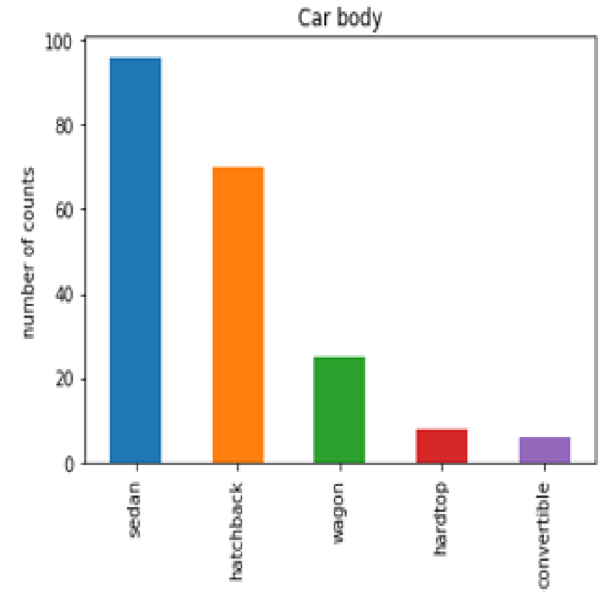
Visualizing categorical data



- Jaguar, Buick and porsche seem to be top 3 high price cars

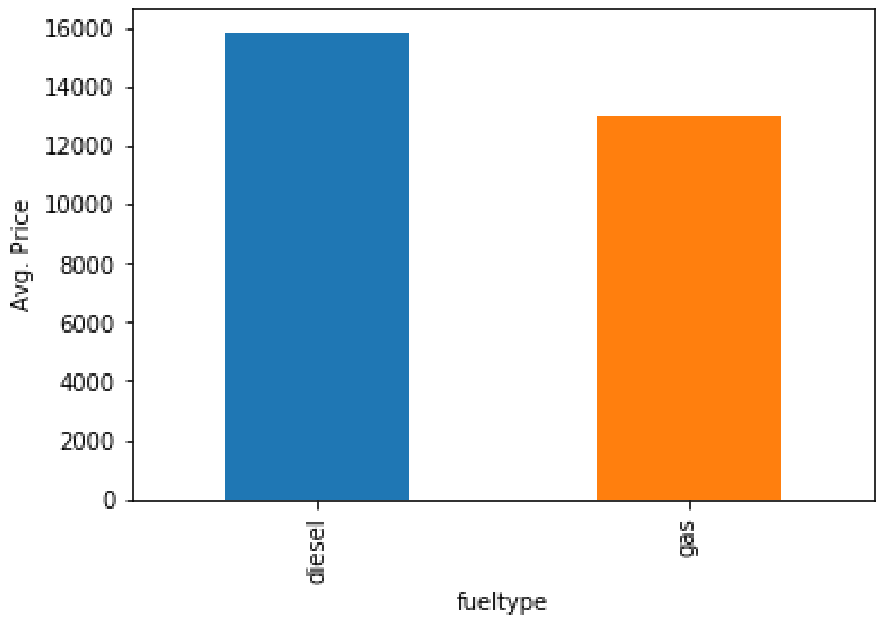


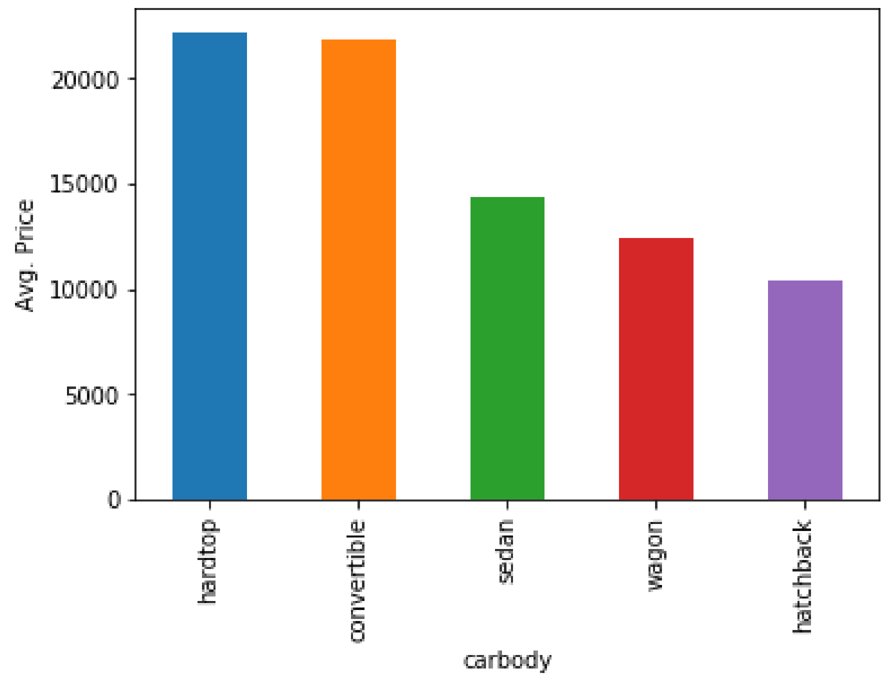




inference

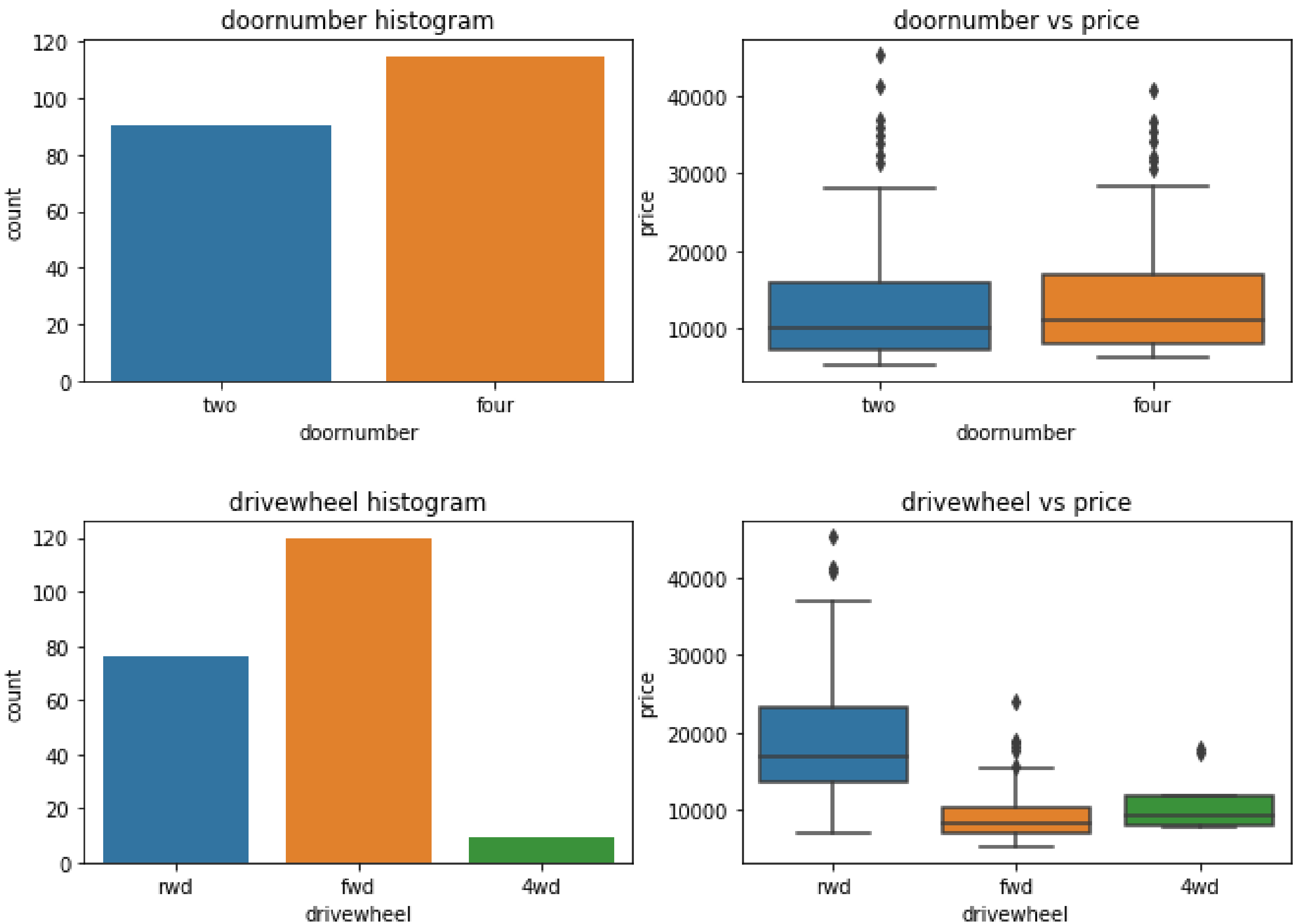
- toyota seem to be favoured company
- number of gas fuel type is greater than diesel fuel type car
- sedan is highly preferred

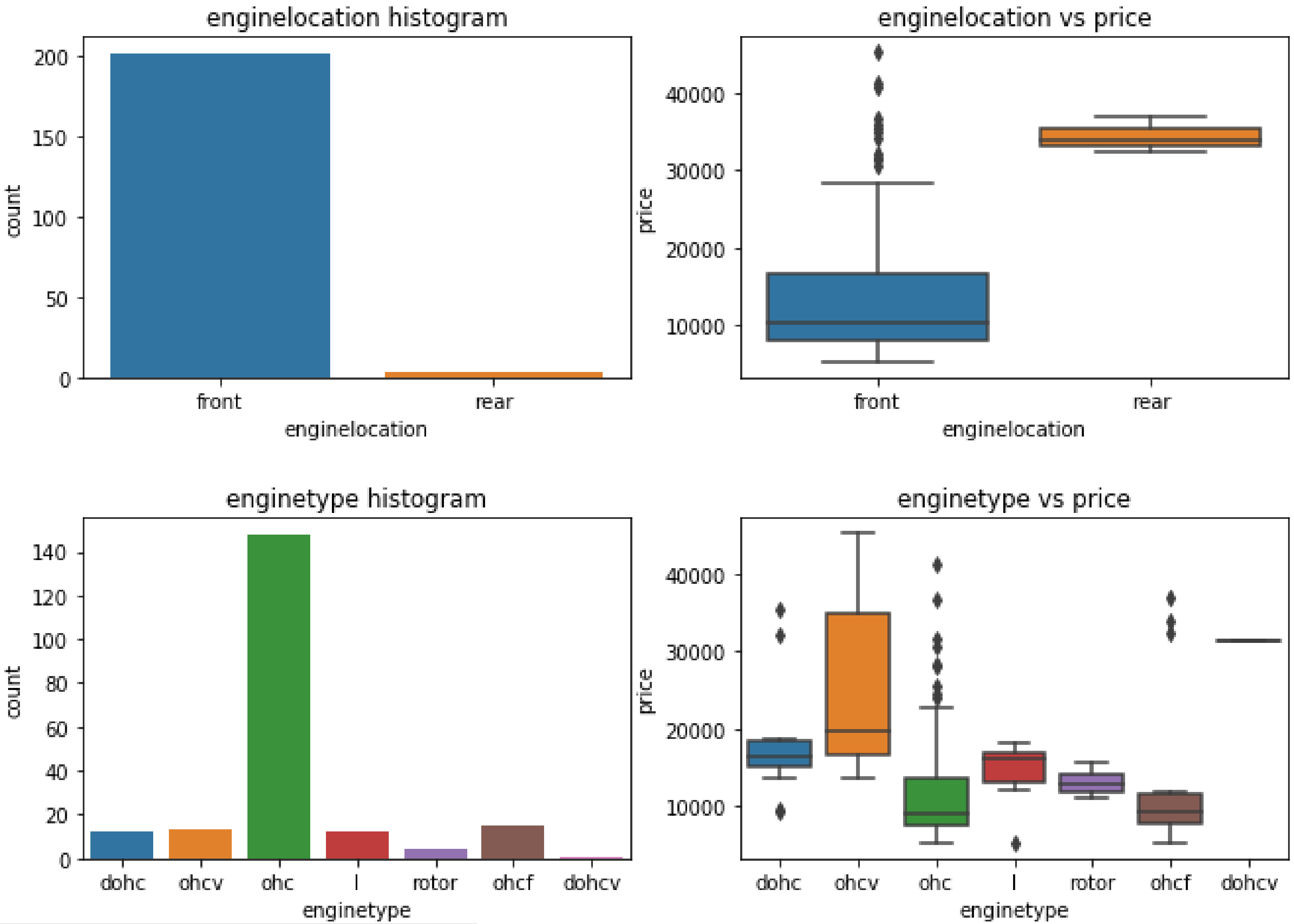


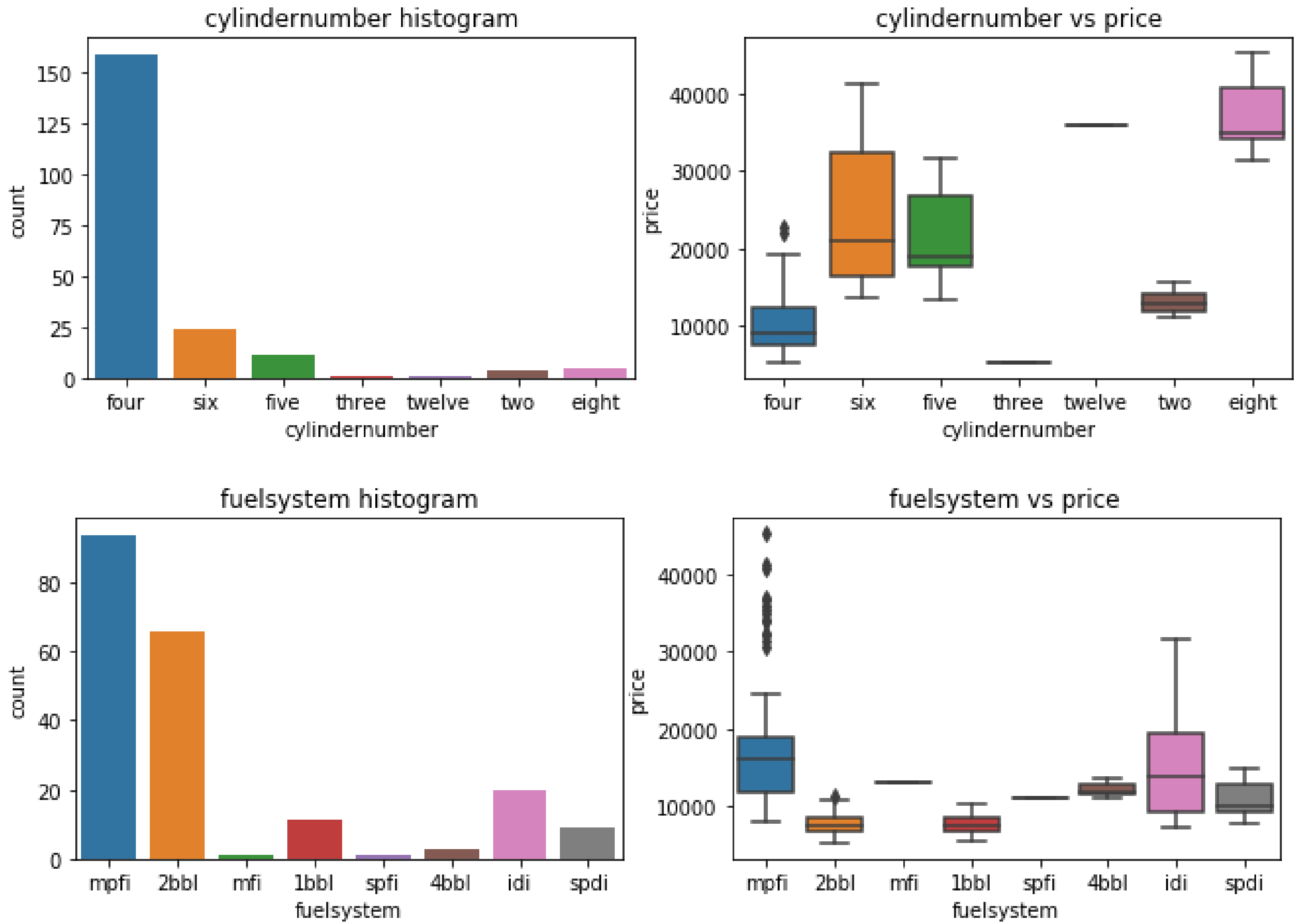


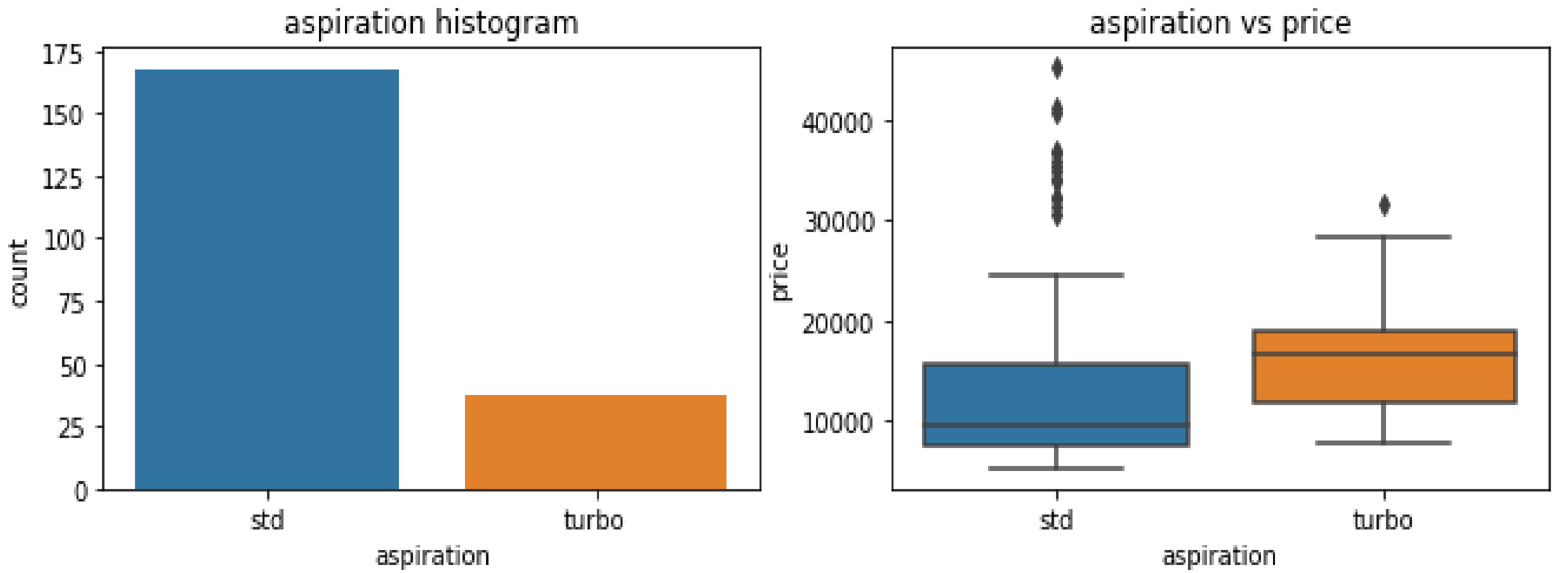
inference

- diesel car is more expensive than gas fuel type car
- hardtop and convertible cars are more expensive than the rest


doornumber histogram
count
doornumber
two
four
doornumber vs price
price
drivewheel histogram
drivewheel
rwd
fwd
4wd
drivewheel vs price

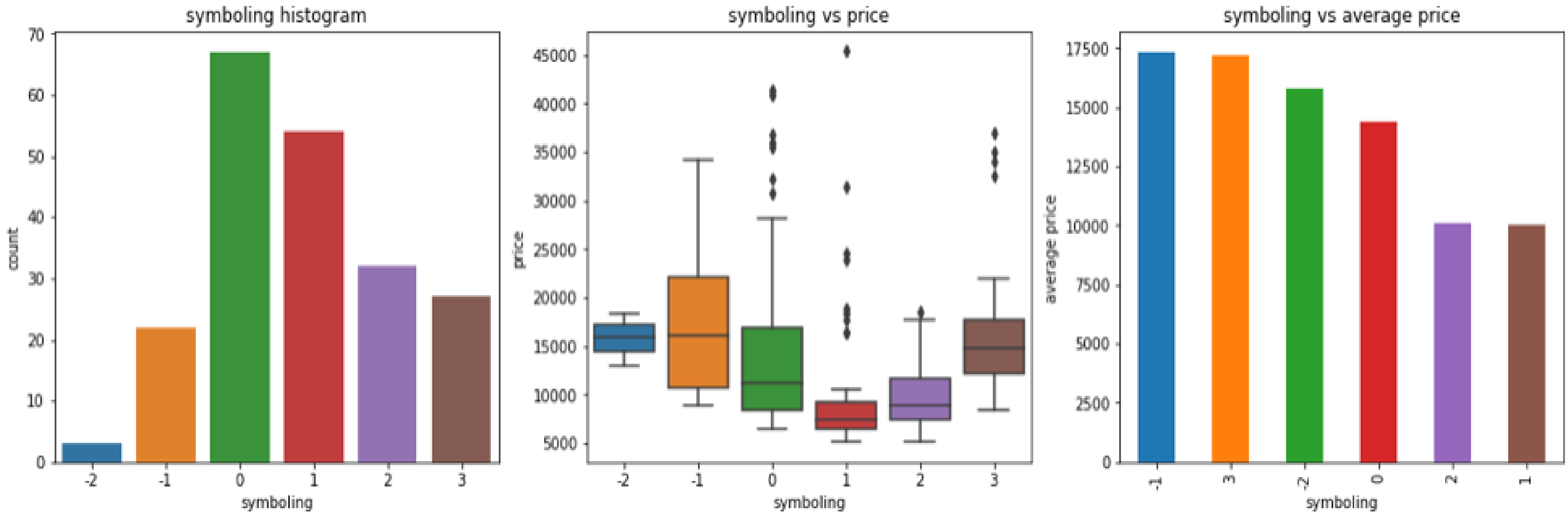

enginelocation histogram
count
0
50
100
150
200
front
rear
enginelocation
enginelocation vs price
price
10000
20000
30000
40000
front
rear
enginelocation
enginetype histogram
count
0
20
40
60
80
100
120
140
dohc
ohcv
ohc
l
rotor
ohcf
dohcv
enginetype
enginetype vs price
price
10000
20000
30000
40000
dohc
ohcv
ohc
l
rotor
ohcf
dohcv
enginetype


cylindernumber histogram
count
150
125
100
75
50
25
0
four
six
five
three
twelve
two
eight
cylindernumber
cylindernumber vs price
price
40000
30000
20000
10000
four
six
five
three
twelve
two
eight
cylindernumber
fuelsystem histogram
count
80
60
40
20
0
mpfi
2bbl
mfi
1bbl
spfi
4bbl
idi
spdi
fuelsystem
fuelsystem vs price
price
40000
30000
20000
10000
mpfi
2bbl
mfi
1bbl
spfi
4bbl
idi
spdi
fuelsystem



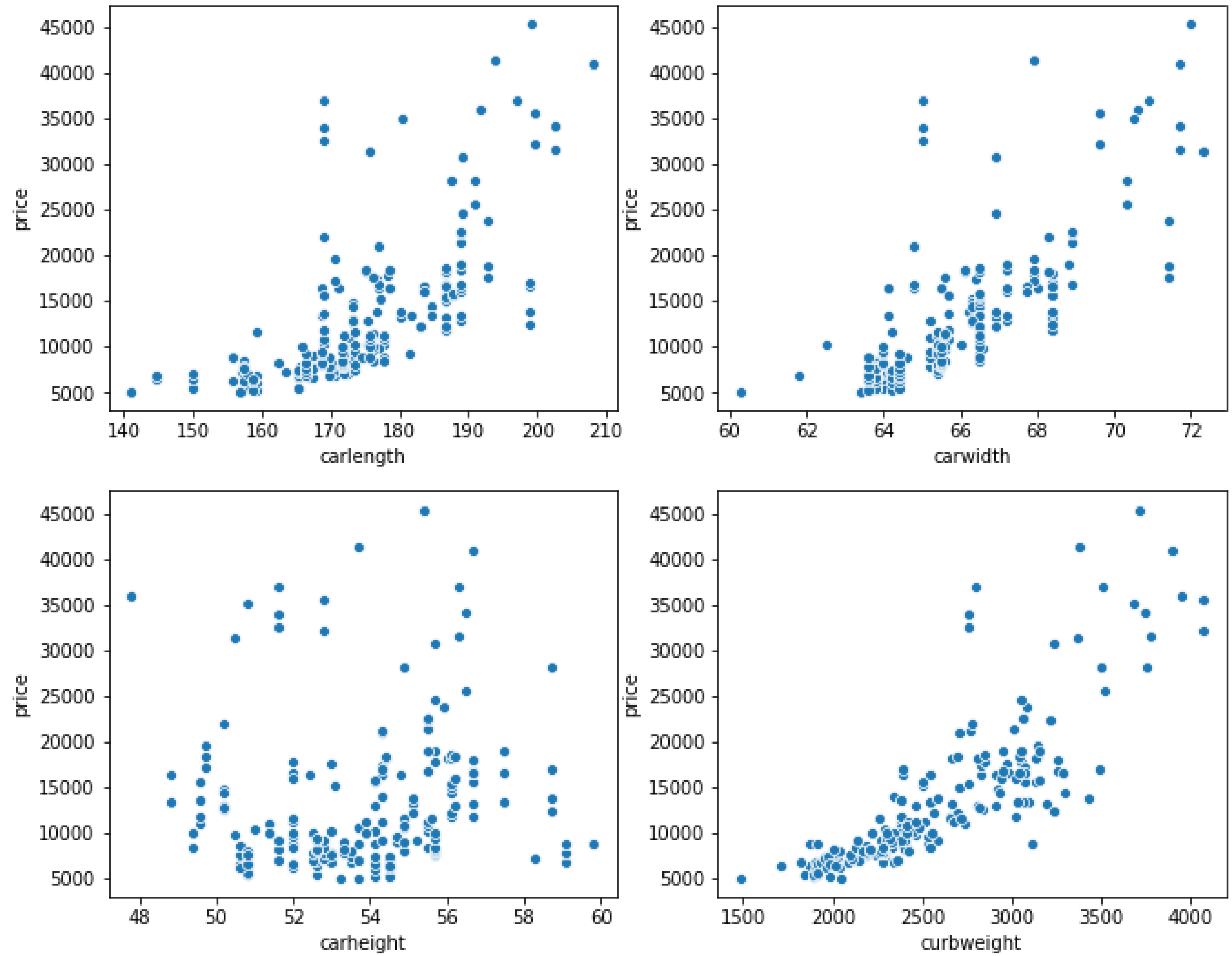
inference

- four door cars are highly preferred although they are more expensive than two door cars
- almost all the cars have front engine location and are cheaper than rear located engine cars
- most of the cars have ohc engine type but have lowest price
- dohcv are highly expensive cars
- most of the cars have 4 cylinders



- Its assigned insurance risk rating
- A value of +3 indicates that the auto is risky, -3 that it is probably pretty safe.

Visualising numerical data

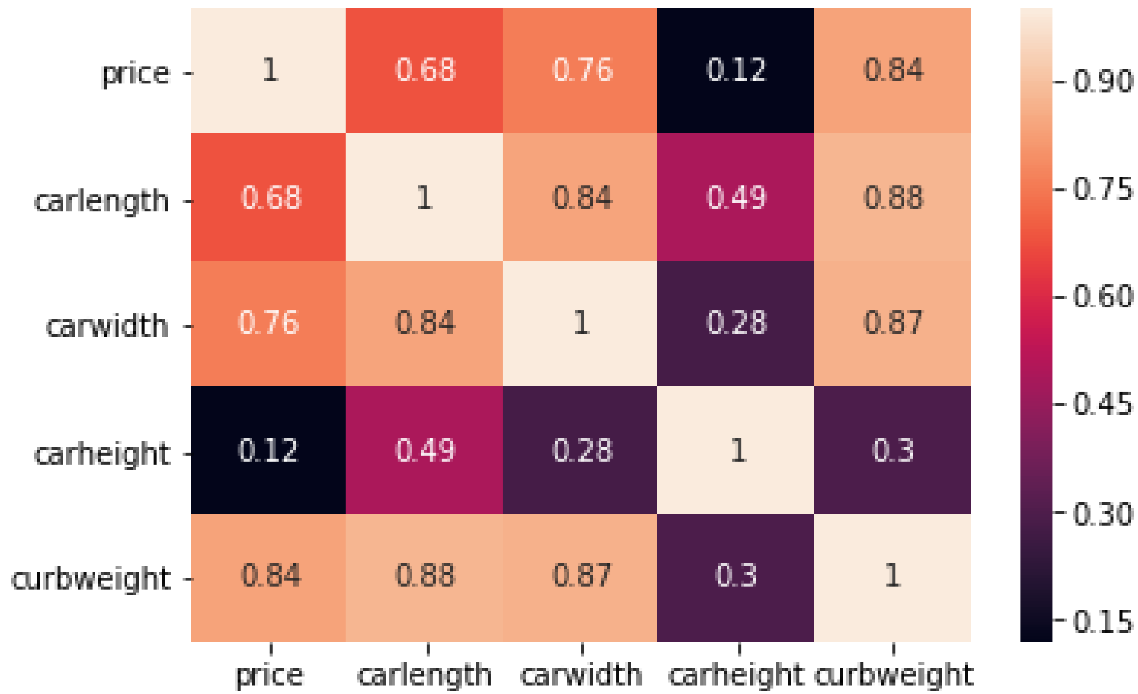


inference

- carlength, carwidth and curbweight seem to have correlation with price
- carheigth does not show any significant relation with the price

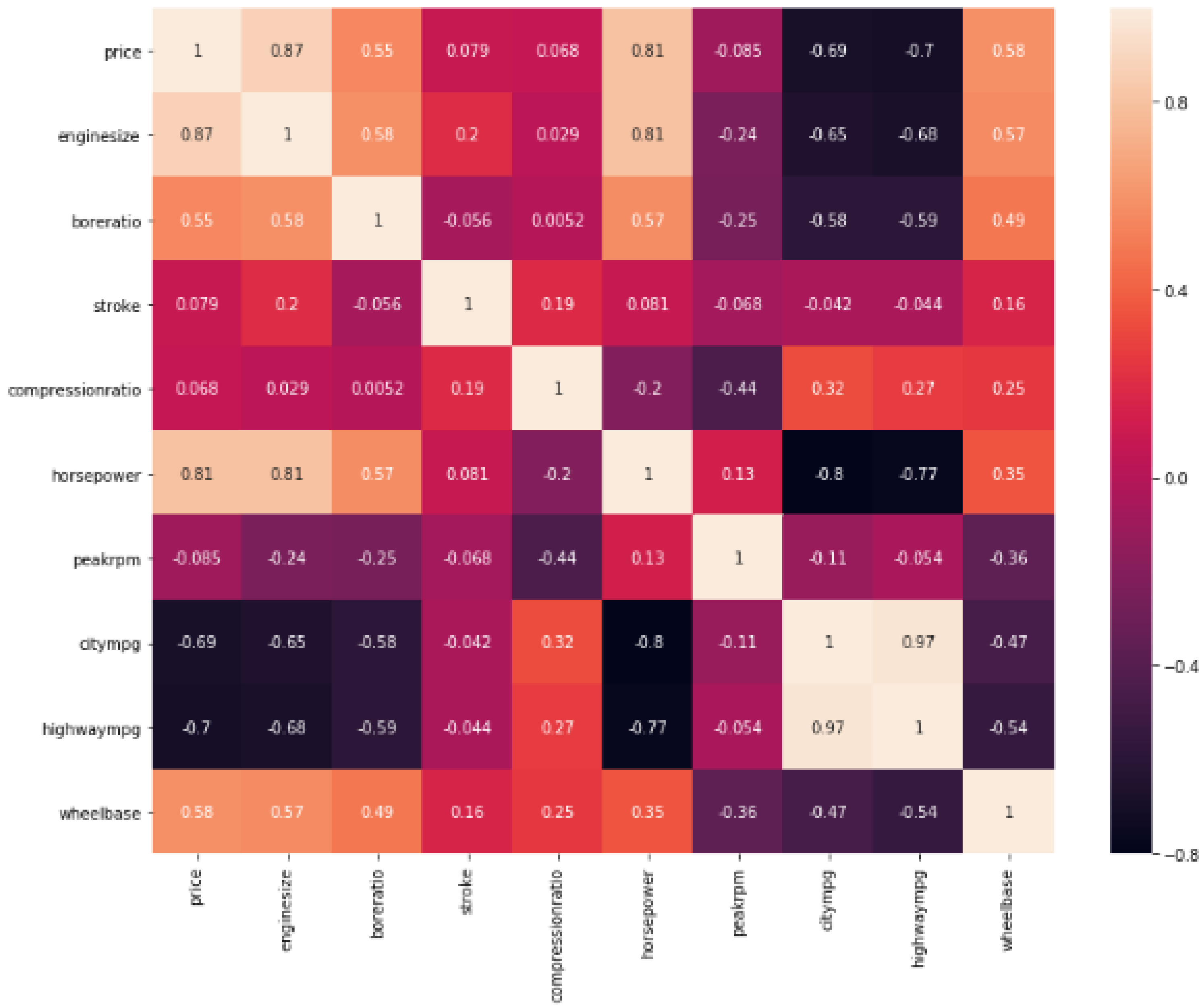
```python
sns.heatmap(df_temp.corr(), annot = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x19033b60128>
```



inference

- price is highly correlated to curbweight



price
enginesize
boreratio
stroke
compressionratio
horsepower
peakrpm
citympg
highwaympg
wheelbase
0.8
0.4
0.0
−0.4
−0.8

Step 4: Deriving new features

```
#Fuel economy
df['fuel_economy'] = (0.55 * df['citympg']) + (0.45 * df['highwaympg'])
```

```
# Binning the company based on average price of the car
df['price'] = df['price'].astype('int')
temp = df.copy()
table = temp.groupby(['Company'])['price'].mean()
temp = temp.merge(table.reset_index(), how='left',on='Company')
bins = [0,10000,20000,40000]
cars_bin=['Budget','Medium','Expensive']
df['carsrange'] = pd.cut(temp['price_y'],bins,right=False,labels=cars_bin)
df.head()
```

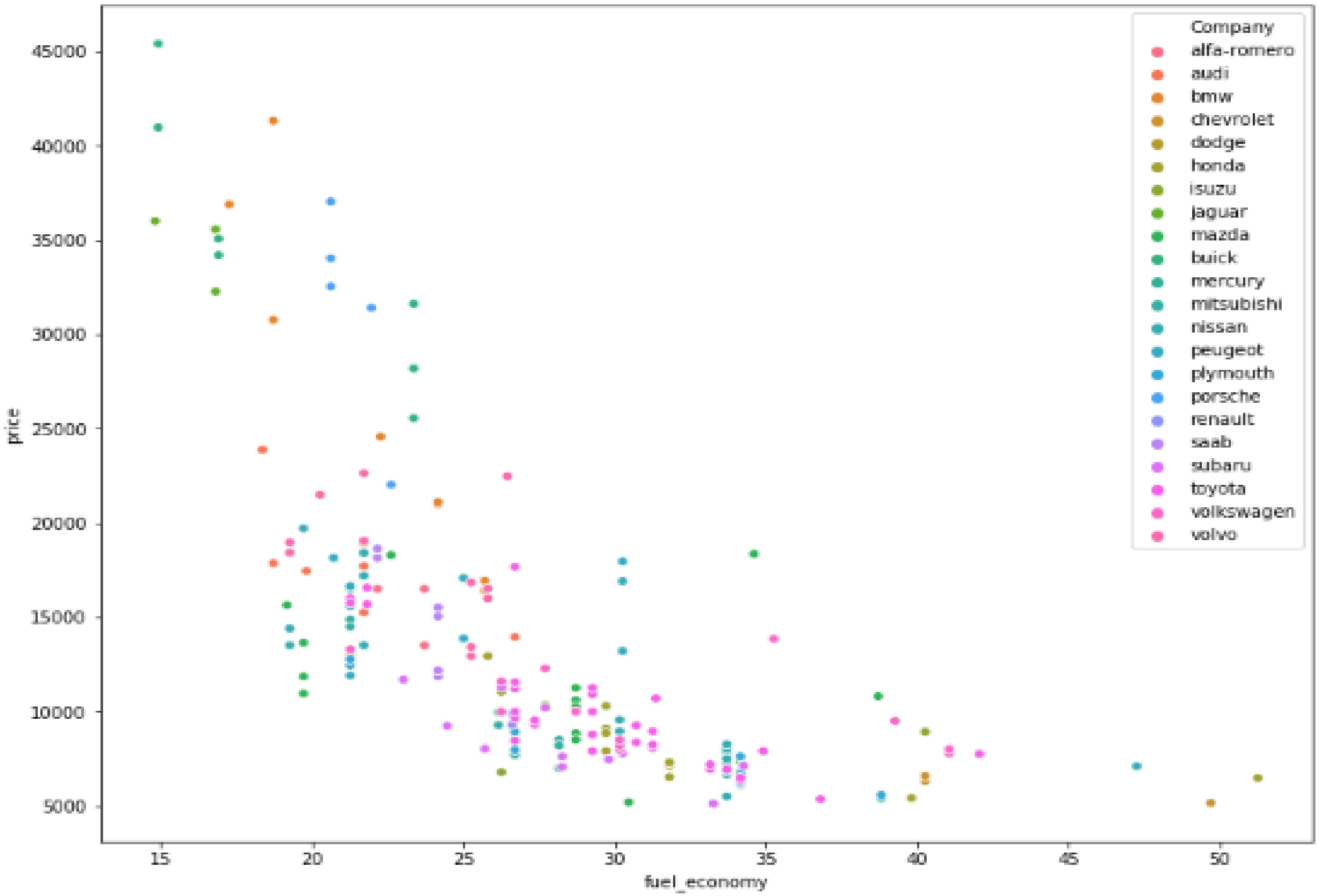
el	enginelocation	wheelbase	carlength	...	stroke	compressionratio	horsepower	peakrpm	citympg	highwaympg	price	Company	fuel_economy	carsrange
vd	front	88.6	168.8	...	2.68	9.0	111	5000	21	27	13495	alfa-romero	23.70	Medium
vd	front	88.6	168.8	...	2.68	9.0	111	5000	21	27	16500	alfa-romero	23.70	Medium
vd	front	94.5	171.2	...	3.47	9.0	154	5000	19	26	16500	alfa-romero	22.15	Medium
vd	front	99.8	176.6	...	3.40	10.0	102	5500	24	30	13950	audi	26.70	Medium
vd	front	99.4	176.6	...	3.40	8.0	115	5500	18	22	17450	audi	19.80	Medium

Calculating fuel_economy using citympg and highwaympg

Binning the company based on price

Dividing the car into ‘Budget’, ‘Medium’ and ‘Expensive’ carsrange

Step 5: Bivariate analysis



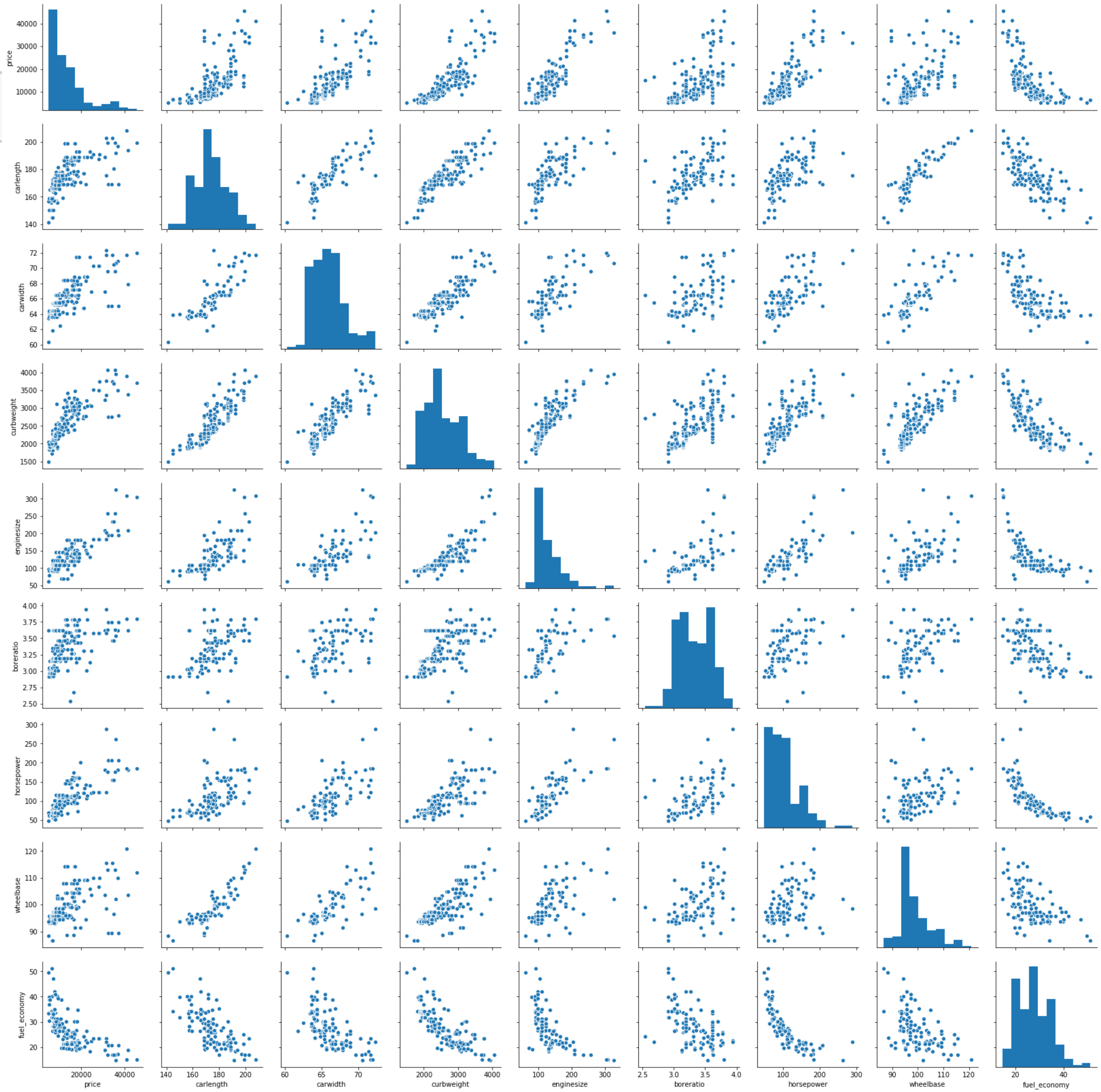
- fuel economy is negatively correlated with price

List of significant variables after analysis

- fueltype
- carbody
- drivewheel
- enginelocation
- enginetype
- cylindernumber
- aspiration
- carlength
- carwidth
- curbweight
- enginesize
- boreratio
- horsepower
- wheelbase
- fuel_economy
- carsrange

```
car.shape
```

(205, 17)



Step 6: Creating dummy variable

```
def dummies(x,df):
    temp = pd.get_dummies(df[x], drop_first = True)
    df = pd.concat([df, temp], axis = 1)
    df.drop([x], axis = 1, inplace = True)
    return df

car = dummies('carsrange',car)
car = dummies('fueltype',car)
car = dummies('carbody',car)
car = dummies('drivewheel',car)
car = dummies('enginelocation',car)
car = dummies('enginetype',car)
car = dummies('cylindernumber',car)
car = dummies('aspiration',car)
```

Creating dummy variables for the above mentioned categorical features

Step 7: Splitting the train-test data and Rescaling the features

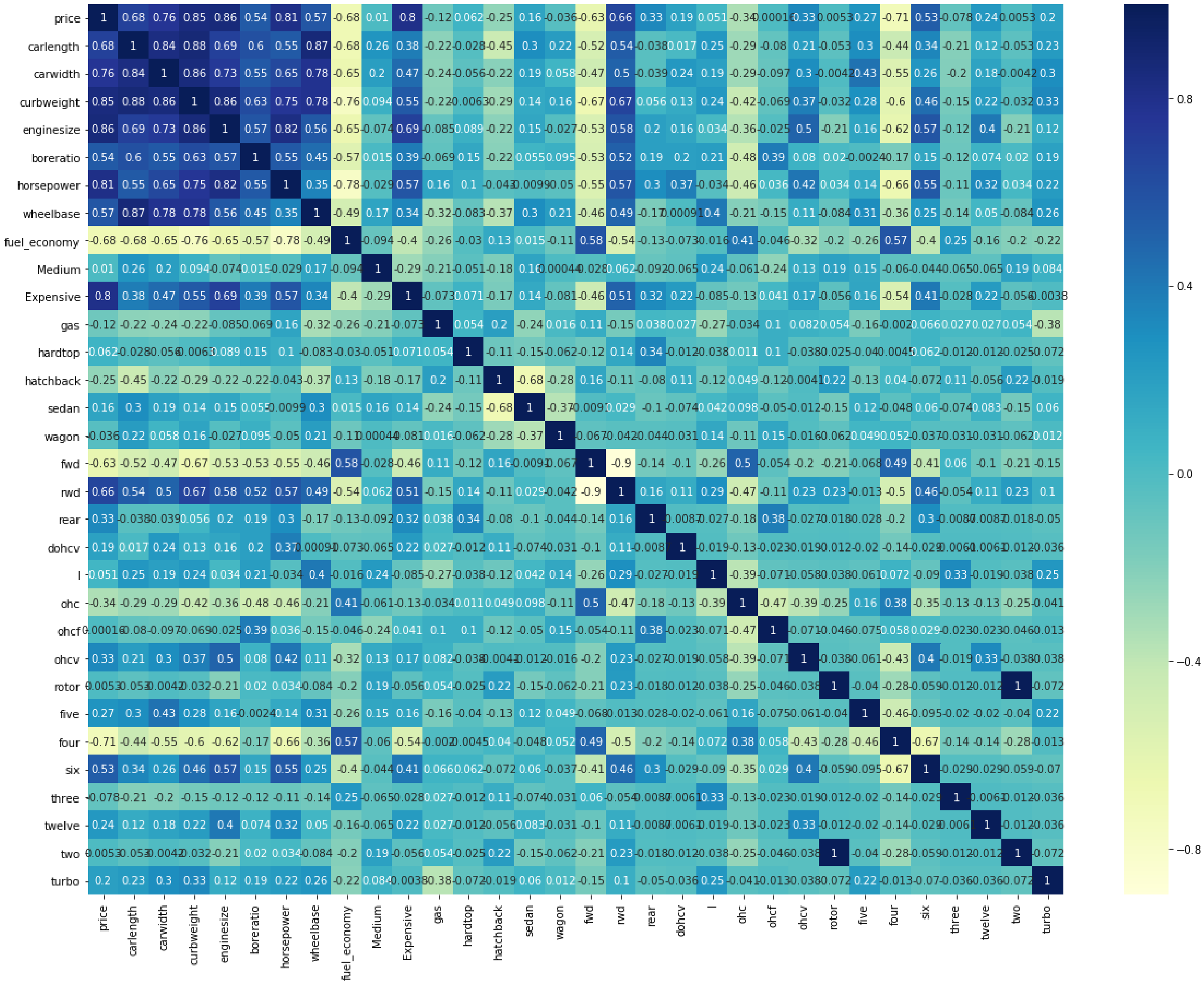
```
import sklearn
from sklearn.model_selection import train_test_split

car_train, car_test = train_test_split(car, train_size = 0.8, random_state = 100)
print(car_train.shape)
print(car_test.shape)
```

```
(164, 32)
(41, 32)
```

Splitting the car dataFrame into car_train (80%) and car_test (20%)

```
from sklearn.preprocessing import MinMaxScaler

# 1. instantiate an object
scaler = MinMaxScaler()

# create list of numeric vars
num_vars = ['carlength', 'carwidth', 'curbweight', 'enginesize', 'boreratio', 'horsepower', 'wheelbase', 'fuel_economy']

# 2. fit on data
car_train[num_vars] = scaler.fit_transform(car_train[num_vars])
car_train.head()
```

Rescaling the car_test numerical data using normalization


price
carlength
carwidth
curbweight
enginesize
boreratio
horsepower
wheelbase
fuel_economy
Medium
Expensive
gas
hardtop
hatchback
sedan
wagon
fwd
rwd
rear
dohcv
l
ohc
ohcf
ohcv
rotor
five
four
six
three
twelve
two
turbo
0.8
0.4
0.0
−0.4
−0.8

Step 8: Building the model

```
# dividing car_train into X and y variable
y_train = car_train.pop('price')
X_train = car_train
```

Using RFE(Recursive Feature Elimination) creating model and finding variables which are selected and are significant for the model

```
[('carlength', False, 12),
 ('carwidth', True, 1),
 ('curbweight', True, 1),
 ('enginesize', False, 21),
 ('boreratio', False, 11),
 ('horsepower', True, 1),
 ('wheelbase', False, 2),
 ('fuel_economy', False, 3),
 ('Medium', False, 13),
 ('Expensive', True, 1),
 ('gas', False, 20),
 ('hardtop', True, 1),
 ('hatchback', True, 1),
 ('sedan', True, 1),
 ('wagon', True, 1),
 ('fwd', False, 19),
 ('rwd', False, 15),
 ('rear', True, 1),
 ('dohcv', True, 1),
 ('l', False, 17),
 ('ohc', False, 8),
 ('ohcf', False, 9),
 ('ohcv', False, 10),
 ('rotor', False, 14),
 ('five', False, 7),
 ('four', False, 4),
 ('six', False, 6),
 ('three', False, 16),
 ('twelve', False, 5),
 ('two', False, 18),
 ('turbo', False, 22)]
```

Building model using statsmodel, for detailed statistics

Model 1

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.934
Model:                            OLS   Adj. R-squared:                  0.930
Method:                 Least Squares   F-statistic:                     216.8
Date:                Mon, 15 Jul 2019   Prob (F-statistic):           5.55e-85
Time:                        16:10:03   Log-Likelihood:                -1476.7
No. Observations:                 164   AIC:                             2975.
Df Residuals:                     153   BIC:                             3009.
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       4350.1729   1065.491      4.083      0.000    2245.200    6455.146
carwidth    9525.1787   1877.783      5.073      0.000    5815.450    1.32e+04
curbweight  1.009e+04   2196.046      4.596      0.000    5753.520    1.44e+04
horsepower  1.077e+04   1967.349      5.475      0.000    6885.493    1.47e+04
Expensive   9133.5870    682.846     13.376      0.000    7784.563    1.05e+04
hardtop    -4119.0959   1376.708     -2.992      0.003   -6838.908   -1399.284
hatchback  -3784.5405   1029.136     -3.677      0.000   -5817.691   -1751.390
sedan      -3023.9526   1010.659     -2.992      0.003   -5020.600   -1027.305
wagon      -4304.5389   1080.327     -3.984      0.000   -6438.822   -2170.256
rear        8186.6135   1861.162      4.399      0.000    4509.720    1.19e+04
dohcv      -5948.3076   2457.501     -2.420      0.017   -1.08e+04   -1093.292
==============================================================================
Omnibus:                       71.562   Durbin-Watson:                   1.899
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              362.728
Skew:                           1.533   Prob(JB):                     1.72e-79
Kurtosis:                       9.609   Cond. No.                         28.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Model 2

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.930
Model:                            OLS   Adj. R-squared:                  0.926
Method:                 Least Squares   F-statistic:                     228.1
Date:                Mon, 15 Jul 2019   Prob (F-statistic):           2.72e-84
Time:                        16:10:04   Log-Likelihood:                -1481.3
No. Observations:                 164   AIC:                             2983.
Df Residuals:                     154   BIC:                             3014.
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       2557.2278    903.465      2.830      0.005     772.443    4342.013
carwidth    9248.6200   1923.319      4.809      0.000    5449.126     1.3e+04
curbweight  1.049e+04   2247.943      4.665      0.000    6046.897    1.49e+04
horsepower  1.061e+04   2016.756      5.262      0.000    6627.942    1.46e+04
Expensive   9236.6506    699.362     13.207      0.000    7855.068    1.06e+04
hatchback  -1976.6593    854.312     -2.314      0.022   -3664.343    -288.975
sedan      -1245.5402    838.230     -1.486      0.139   -2901.454     410.373
wagon      -2542.0795    928.689     -2.737      0.007   -4376.693    -707.466
rear        7833.3739   1904.765      4.113      0.000    4070.534    1.16e+04
dohcv      -5917.8372   2520.129     -2.348      0.020   -1.09e+04    -939.352
==============================================================================
Omnibus:                       60.823   Durbin-Watson:                   1.818
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              260.864
Skew:                           1.324   Prob(JB):                     2.26e-57
Kurtosis:                       8.582   Cond. No.                         28.0
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Calculating VIF

	Features	VIF
0	const	30.64
2	curbweight	7.72
6	sedan	6.57
5	hatchback	6.16
1	carwidth	4.47
3	horsepower	4.08
7	wagon	3.76
4	Expensive	1.79
8	rear	1.64
9	dohcv	1.44

VIF > 5 should not be ignored
therefore dropping feature "curbweight" as it is showing high multicollinearity

Model 3

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.920
Model:                            OLS   Adj. R-squared:                  0.916
Method:                 Least Squares   F-statistic:                     223.9
Date:                Mon, 15 Jul 2019   Prob (F-statistic):           4.46e-81
Time:                        16:10:04   Log-Likelihood:                -1492.2
No. Observations:                 164   AIC:                             3002.
Df Residuals:                     155   BIC:                             3030.
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       2880.4237    959.253      3.003      0.003     985.527    4775.320
carwidth    1.557e+04   1454.293     10.704      0.000    1.27e+04    1.84e+04
horsepower  1.613e+04   1738.977      9.277      0.000    1.27e+04    1.96e+04
Expensive   1.009e+04    718.903     14.033      0.000    8668.525    1.15e+04
hatchback  -2617.7641    897.898     -2.915      0.004   -4391.460    -844.069
sedan      -1525.3648    890.330     -1.713      0.089   -3284.111     233.381
wagon      -1988.7744    980.849     -2.028      0.044   -3926.330     -51.219
rear        5773.5472   1973.113      2.926      0.004    1875.886    9671.209
dohcv      -1.065e+04   2456.529     -4.336      0.000   -1.55e+04   -5798.987
==============================================================================
Omnibus:                       50.533   Durbin-Watson:                   1.967
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              163.455
Skew:                           1.177   Prob(JB):                     3.21e-36
Kurtosis:                       7.287   Cond. No.                         19.5
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Model 4

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.916
Model:                            OLS   Adj. R-squared:                  0.912
Method:                 Least Squares   F-statistic:                     242.9
Date:                Mon, 15 Jul 2019   Prob (F-statistic):           1.75e-80
Time:                        16:10:04   Log-Likelihood:                -1496.6
No. Observations:                 164   AIC:                             3009.
Df Residuals:                     156   BIC:                             3034.
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       4026.4943    896.622      4.491      0.000    2255.409    5797.580
carwidth    1.429e+04   1420.152     10.060      0.000    1.15e+04    1.71e+04
horsepower  1.762e+04   1703.238     10.343      0.000    1.43e+04     2.1e+04
Expensive   1.056e+04    717.319     14.723      0.000    9144.129     1.2e+04
hatchback  -3565.6485    857.484     -4.158      0.000   -5259.425   -1871.872
sedan      -2442.3525    853.314     -2.862      0.005   -4127.892    -756.813
wagon      -2837.5598    959.416     -2.958      0.004   -4732.682    -942.437
dohcv      -1.153e+04   2496.688     -4.616      0.000   -1.65e+04   -6594.222
==============================================================================
Omnibus:                       38.175   Durbin-Watson:                   2.023
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              101.397
Skew:                           0.938   Prob(JB):                     9.59e-23
Kurtosis:                       6.365   Cond. No.                         18.7
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
#Calculating the Variance Inflation Factor
checkVIF(X_train_new)
```

	Features	VIF
0	const	25.38
5	sedan	5.73
4	hatchback	5.22
6	wagon	3.38
2	horsepower	2.45
1	carwidth	2.05
3	Expensive	1.59
7	dohcv	1.19

VIF > 5 should not be ignored
therefore dropping feature "sedan" as it is showing high multicollinearity

Model 5

OLS Regression Results

Dep. Variable:	price	R-squared:	0.912
Model:	OLS	Adj. R-squared:	0.908
Method:	Least Squares	F-statistic:	269.7
Date:	Mon, 15 Jul 2019	Prob (F-statistic):	5.48e-80
Time:	16:10:05	Log-Likelihood:	-1500.8
No. Observations:	164	AIC:	3016.
Df Residuals:	157	BIC:	3037.
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2063.0393	590.433	3.494	0.001	896.823	3229.256
carwidth	1.313e+04	1392.418	9.431	0.000	1.04e+04	1.59e+04
horsepower	1.894e+04	1676.599	11.296	0.000	1.56e+04	2.23e+04
Expensive	1.072e+04	731.419	14.654	0.000	9273.261	1.22e+04
hatchback	-1414.9357	422.449	-3.349	0.001	-2249.352	-580.519
wagon	-590.0016	563.712	-1.047	0.297	-1703.440	523.436
dohcv	-1.204e+04	2546.715	-4.726	0.000	-1.71e+04	-7005.974

Omnibus:	32.476	Durbin-Watson:	2.034
Prob(Omnibus):	0.000	Jarque-Bera (JB):	66.183
Skew:	0.896	Prob(JB):	4.25e-15
Kurtosis:	5.545	Cond. No.	17.1

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Model 6

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.911
Model:                            OLS   Adj. R-squared:                  0.908
Method:                 Least Squares   F-statistic:                     323.2
Date:                Mon, 15 Jul 2019   Prob (F-statistic):           4.98e-81
Time:                        16:10:05   Log-Likelihood:                -1501.3
No. Observations:                 164   AIC:                             3015.
Df Residuals:                     158   BIC:                             3033.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1962.6110    582.760      3.368      0.001     811.607    3113.615
carwidth    1.304e+04   1389.831      9.379      0.000    1.03e+04    1.58e+04
horsepower  1.899e+04   1676.369     11.328      0.000    1.57e+04    2.23e+04
Expensive   1.082e+04    725.477     14.910      0.000    9384.228    1.22e+04
hatchback  -1289.2380    405.141     -3.182      0.002   -2089.428    -489.048
dohcv      -1.212e+04   2546.316     -4.759      0.000   -1.71e+04   -7087.720
==============================================================================
Omnibus:                       34.628   Durbin-Watson:                   2.031
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               71.778
Skew:                           0.947   Prob(JB):                     2.59e-16
Kurtosis:                       5.630   Cond. No.                         17.0
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

	Features	VIF
0	const	10.25
2	horsepower	2.27
1	carwidth	1.88
3	Expensive	1.55
5	dohcv	1.19
4	hatchback	1.11

Model looks fine as all the values are less than 5

dropping "hatchback" just to check the statistics of the model

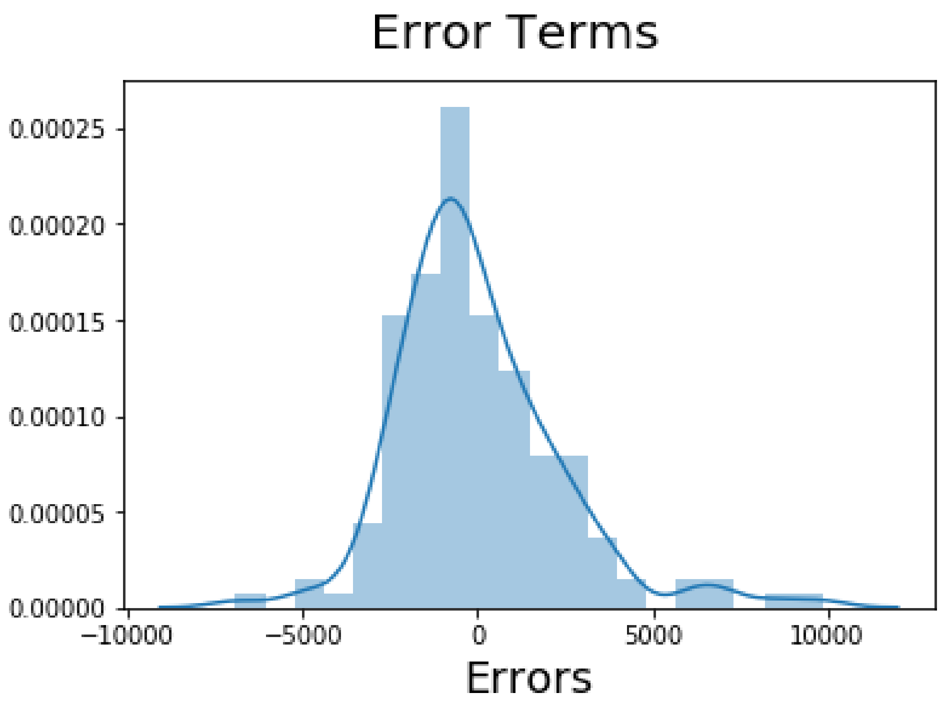
Model 7

	Features	VIF
0	const	8.49
2	horsepower	2.22
1	carwidth	1.78
3	Expensive	1.52
4	dohcv	1.16

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.905
Model:                            OLS   Adj. R-squared:                  0.903
Method:                 Least Squares   F-statistic:                     379.7
Date:                Mon, 15 Jul 2019   Prob (F-statistic):           3.22e-80
Time:                        16:10:05   Log-Likelihood:                -1506.4
No. Observations:                 164   AIC:                             3023.
Df Residuals:                     159   BIC:                             3038.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1193.7288    545.317      2.189      0.030     116.730    2270.728
carwidth    1.405e+04   1391.088     10.100      0.000    1.13e+04    1.68e+04
horsepower  1.825e+04   1707.143     10.691      0.000    1.49e+04    2.16e+04
Expensive   1.116e+04    737.543     15.137      0.000    9707.228    1.26e+04
dohcv      -1.326e+04   2592.276     -5.114      0.000   -1.84e+04   -8138.382
==============================================================================
Omnibus:                       39.069   Durbin-Watson:                   2.010
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               84.029
Skew:                           1.052   Prob(JB):                     5.67e-19
Kurtosis:                       5.806   Cond. No.                         16.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Final model as the p-value is less than 0.05 and VIF is also less than 5

Step 9: Residual analysis on train set



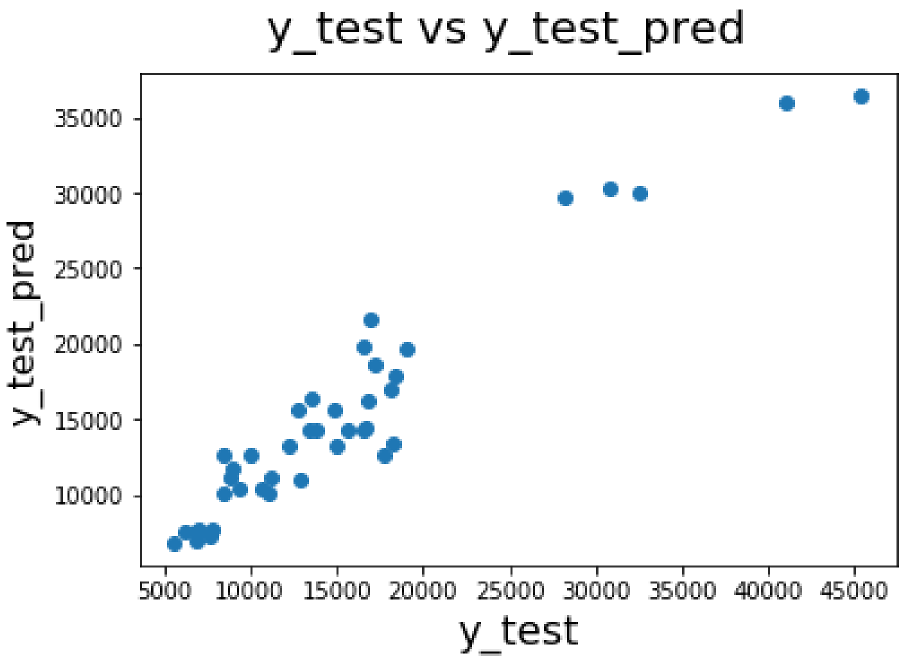
- errors seem to be normally distributed with mean = 0

Step 10: Prediction and evaluation

```
# making prediction
y_test_pred = lm.predict(X_test_new)
```

```
# evaluation
from sklearn.metrics import r2_score
r2_score(y_test, y_test_pred)
```

```
0.9094330812753356
```



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.905
Model:                            OLS   Adj. R-squared:                  0.903
Method:                 Least Squares   F-statistic:                     379.7
Date:                Mon, 15 Jul 2019   Prob (F-statistic):           3.22e-80
Time:                        16:10:07   Log-Likelihood:                -1506.4
No. Observations:                 164   AIC:                             3023.
Df Residuals:                     159   BIC:                             3038.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1193.7288    545.317      2.189      0.030     116.730    2270.728
carwidth    1.405e+04   1391.088     10.100      0.000    1.13e+04    1.68e+04
horsepower  1.825e+04   1707.143     10.691      0.000    1.49e+04    2.16e+04
Expensive   1.116e+04    737.543     15.137      0.000    9707.228    1.26e+04
dohcv      -1.326e+04   2592.276     -5.114      0.000   -1.84e+04   -8138.382
==============================================================================
Omnibus:                       39.069   Durbin-Watson:                   2.010
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               84.029
Skew:                           1.052   Prob(JB):                     5.67e-19
Kurtosis:                       5.806   Cond. No.                         16.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

since R-squared and adjusted R-squared are 0.912 and 0.909, hence 90% variance explained by the model
all the p-values are less than 0.05, therefore we can say the coefficients are statistically significant