COSC 3337: Data Science I

Homework 1 (Regression)

Dr. Rizk

[100 points]

About The Data

The data we'll be using comes from kaggle, and contains the following car attributes:

```
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
```

Problem Statement

You are required to model the price of cars with the available independent variables. It will be used by your management team to understand how exactly the prices vary with the independent variables. They can accordingly manipulate the design of the cars, the business

strategy etc. to meet certain price levels. Further, the model will be a good way for management to understand the pricing dynamics of a new market.

In general, your company would like for you to answer the following:

```
Which variables are significant in predicting the price of a car
How well those variables describe the price of a car
```

Part 1: Reading and Understanding the Data

[8 points]

Begin by importing some necessary libraries that you'll be using to explore the data.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import rcParams
```

In [2]:
```
rcParams['figure.figsize'] = 8, 5
sns.set_style('darkgrid')
```

1.1 Import the data using pandas and save into a variable named cars_df. Then display the first 5 rows.

[3 Points]

In [3]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
cars_df = pd.read_csv('car_data.csv')
cars_df.head()
```

Out[3]:

	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginel
0	1	3	alfa-romero giulia	gas	std	two	convertible	rwd	
1	2	3	alfa-romero stelvio	gas	std	two	convertible	rwd	
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	
3	4	2	audi 100 ls	gas	std	four	sedan	fwd	
4	5	2	audi 100ls	gas	std	four	sedan	4wd	

5 rows × 26 columns

1.2 Print some basic statistics of your data.

[3 Points]

In [4]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
cars_df.describe()
```

Out[4]:

	car_ID	symboling	wheelbase	carlength	carwidth	carheight	curbweight	enginesiz
count	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.000000	205.00000
mean	103.000000	0.834146	98.756585	174.049268	65.907805	53.724878	2555.565854	126.90731
std	59.322565	1.245307	6.021776	12.337289	2.145204	2.443522	520.680204	41.64269
min	1.000000	-2.000000	86.600000	141.100000	60.300000	47.800000	1488.000000	61.00000
25%	52.000000	0.000000	94.500000	166.300000	64.100000	52.000000	2145.000000	97.00000
50%	103.000000	1.000000	97.000000	173.200000	65.500000	54.100000	2414.000000	120.00000
75%	154.000000	2.000000	102.400000	183.100000	66.900000	55.500000	2935.000000	141.00000
max	205.000000	3.000000	120.900000	208.100000	72.300000	59.800000	4066.000000	326.00000

1.3 Print some general information about your data using pandas.

[2 Points]

In [5]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
cars_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   car_ID            205 non-null    int64
 1   symboling         205 non-null    int64
 2   CarName           205 non-null    object
 3   fueltype          205 non-null    object
 4   aspiration        205 non-null    object
 5   doornumber        205 non-null    object
 6   carbody           205 non-null    object
 7   drivewheel        205 non-null    object
 8   enginelocation    205 non-null    object
 9   wheelbase         205 non-null    float64
 10  carlength         205 non-null    float64
 11  carwidth          205 non-null    float64
 12  carheight         205 non-null    float64
 13  curbweight        205 non-null    int64
 14  enginetype        205 non-null    object
 15  cylindernumber    205 non-null    object
 16  enginesize        205 non-null    int64
 17  fuelsystem        205 non-null    object
 18  boreratio         205 non-null    float64
 19  stroke            205 non-null    float64
 20  compressionratio  205 non-null    float64
 21  horsepower        205 non-null    int64
 22  peakrpm           205 non-null    int64
 23  citympg           205 non-null    int64
 24  highwaympg        205 non-null    int64
 25  price             205 non-null    float64
dtypes: float64(8), int64(8), object(10)
memory usage: 41.8+ KB
```

By looking at our previous output, are there any missing values in this dataset? How do we know?

TYPE YOUR ANSWER HERE

There appreas to be no missing data as there are 205 entries across all columns in the dataset. We can tell if there are any missing values or not based on the out given by .info() which lists out all the columns that exist, with their total count of non-null values, and their datatypes. For this Dataset we can see that there are a total of 25 columns with 205 entries. And each columns has 205 non-null values meaning that there are no missing values.

Part 2: Data Cleaning and Preparation

[7 Points]

2.1 Instead of using car names, let's extract company names to see how the companies are distributed. Use the CarName column to create a new column in cars_df called 'car_company' that tells us which company the car belongs to. Once you've accomplished this, display all of the unique company names in our dataset.

Hint: Every carName value has the car company name placed in front of it, so you can parse it out. Also lowercase every company name to ensure that we don't double count something like bmw with BMW.

[3 Points]

In [6]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
cars_df['car_company'] = cars_df['CarName'].str.split(" ").str[0].str.lower()
print("The uinque companies in the dataset are:")
print(cars_df['car_company'].unique())
```

```
The uinque companies in the dataset are:
['alfa-romero' 'audi' 'bmw' 'chevrolet' 'dodge' 'honda' 'isuzu' 'jaguar'
 'maxda' 'mazda' 'buick' 'mercury' 'mitsubishi' 'nissan' 'peugeot'
 'plymouth' 'porsche' 'porcshce' 'renault' 'saab' 'subaru' 'toyota'
 'toyouta' 'vokswagen' 'volkswagen' 'vw' 'volvo']
```

2.2 Notice how there are some typos in the data. 'toyouta' should be 'toyota', 'porcshce' should be 'porsche', 'maxda' should be 'mazda', 'vokswagen' should be 'volkswagen', and 'vw' should be 'volkswagen'. Use Pandas to make these name corrections and display the unique company names again.

[4 Points]

In [7]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
correct_names = {
    'toyouta':'toyota',
    'porcshce':'porsche',
    'maxda':'mazda',
    'vokswagen':'volkswagen',
    'vw':'volkswagen'
}
cars_df['car_company'] = cars_df['car_company'].replace(correct_names)
print("The uinque companies in the dataset are:")
print(cars_df['car_company'].unique())
```

```
The uinque companies in the dataset are:
['alfa-romero' 'audi' 'bmw' 'chevrolet' 'dodge' 'honda' 'isuzu' 'jaguar'
 'mazda' 'buick' 'mercury' 'mitsubishi' 'nissan' 'peugeot' 'plymouth'
 'porsche' 'renault' 'saab' 'subaru' 'toyota' 'volkswagen' 'volvo']
```

Part 3: Visualising Categorical Data

[50 points]

CompanyName
Symboling
fueltype
enginetype
carbody
doornumber
enginelocation

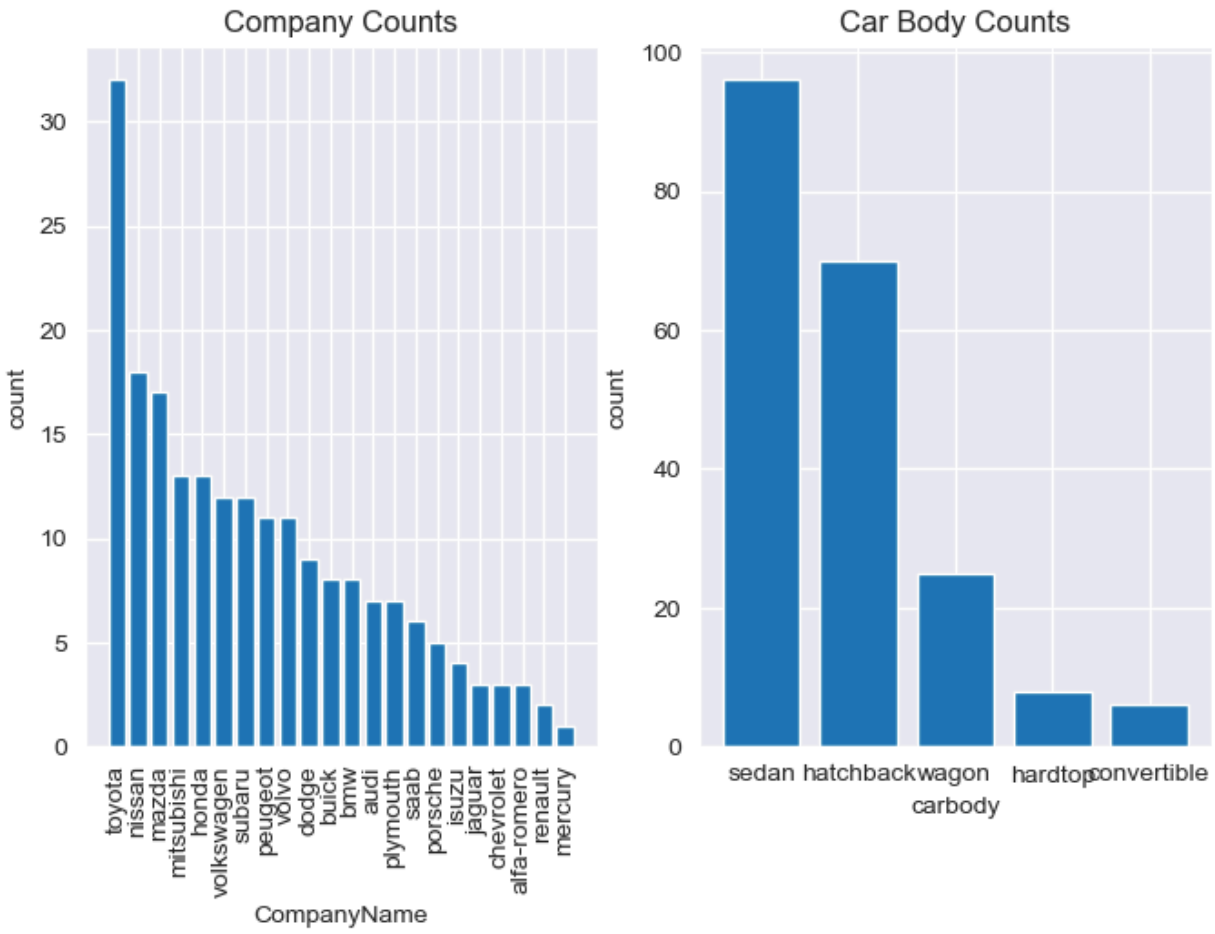
fuelsystem
cylindernumber
aspiration
drivewheel

3.1 Create the following plots

1. A plot of the unique company names on the x-axis, and the value counts on the y-axis.
2. A plot of the unique car bodys on the x-axis and value counts on the y-axis.

[4 Points]

In [8]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
x0 = cars_df['car_company'].value_counts().index
y0 = cars_df['car_company'].value_counts().values
x1 = cars_df['carbody'].value_counts().index
y1 = cars_df['carbody'].value_counts().values

fig, axes = plt.subplots(nrows=1, ncols=2)

axes[0].bar(x0, y0)
axes[0].set(title='Company Counts', xlabel='CompanyName', ylabel='count')
axes[0].tick_params(axis='x', rotation=90)

axes[1].bar(x1, y1)
axes[1].set(title='Car Body Counts', xlabel='carbody', ylabel='count')

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

The 2 graphs of unique car company counts, and unique car body counts helps us visually see the distributions and spread. Based on the results of the 2 graphs we can infere that the car brand Toyota is the most popular. The company name with less counts is mercury. And with the car body type sedan is most common follwed by hatchback type.

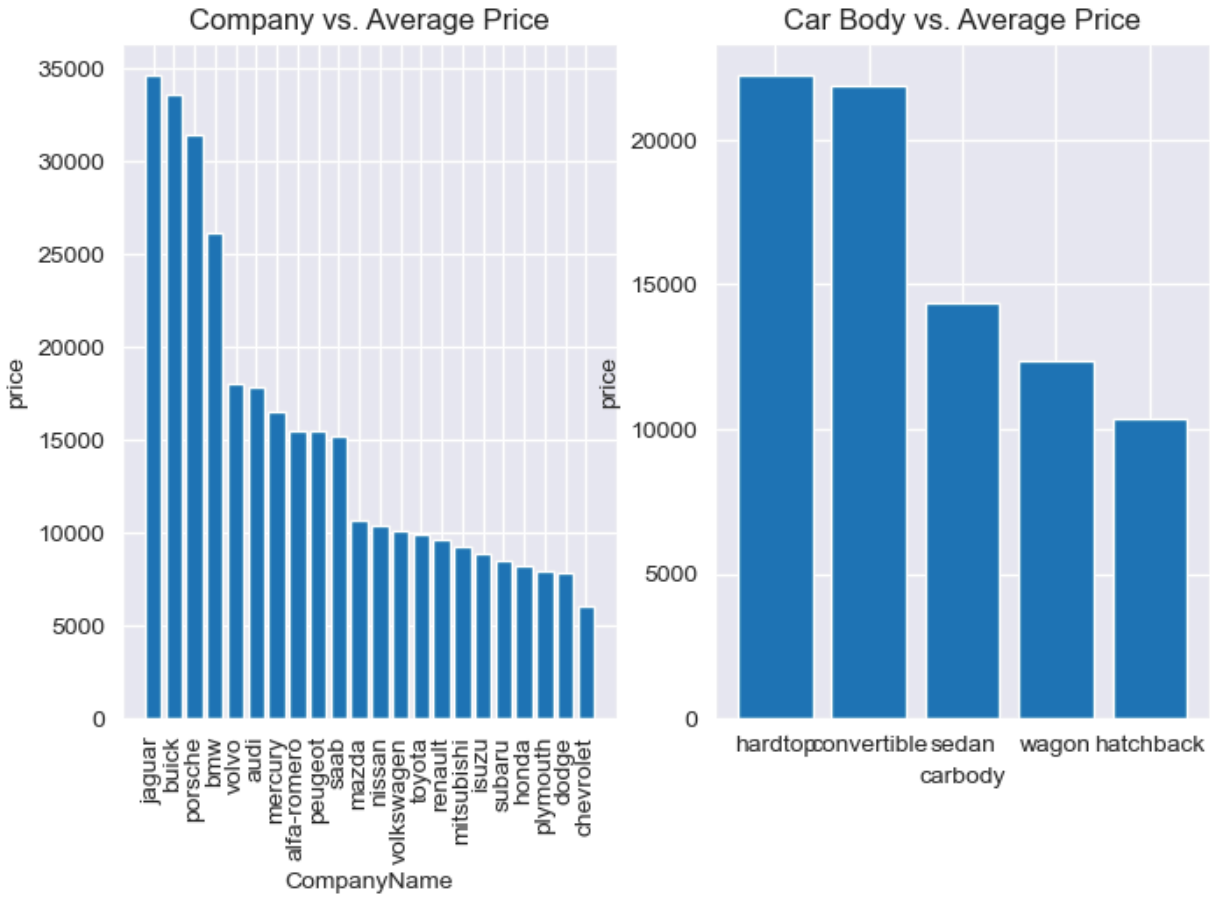
3.2 Create the following plots

```
1. A plot of the unique company names on the x-axis, and that
companies average price on the y-axis.
2. A plot of the unique car bodys on the x-axis and that car body's
average price on the y-axis.
```

[4 Points]

In [9]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
company_avg_price = cars_df.groupby('car_company')['price'].mean().sort_values(ascendi
carbody_avg_price = cars_df.groupby('carbody')['price'].mean().sort_values(ascending=F

x0 = company_avg_price.index
```

```
y0 = company_avg_price.values
x1 = carbody_avg_price.index
y1 = carbody_avg_price.values

fig, axes = plt.subplots(nrows=1, ncols=2)

axes[0].bar(x0, y0)
axes[0].set(title='Company vs. Average Price', xlabel='CompanyName', ylabel='price')
axes[0].tick_params(axis='x', rotation=90)

axes[1].bar(x1, y1)
axes[1].set(title='Car Body vs. Average Price', xlabel='carbody', ylabel='price')

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

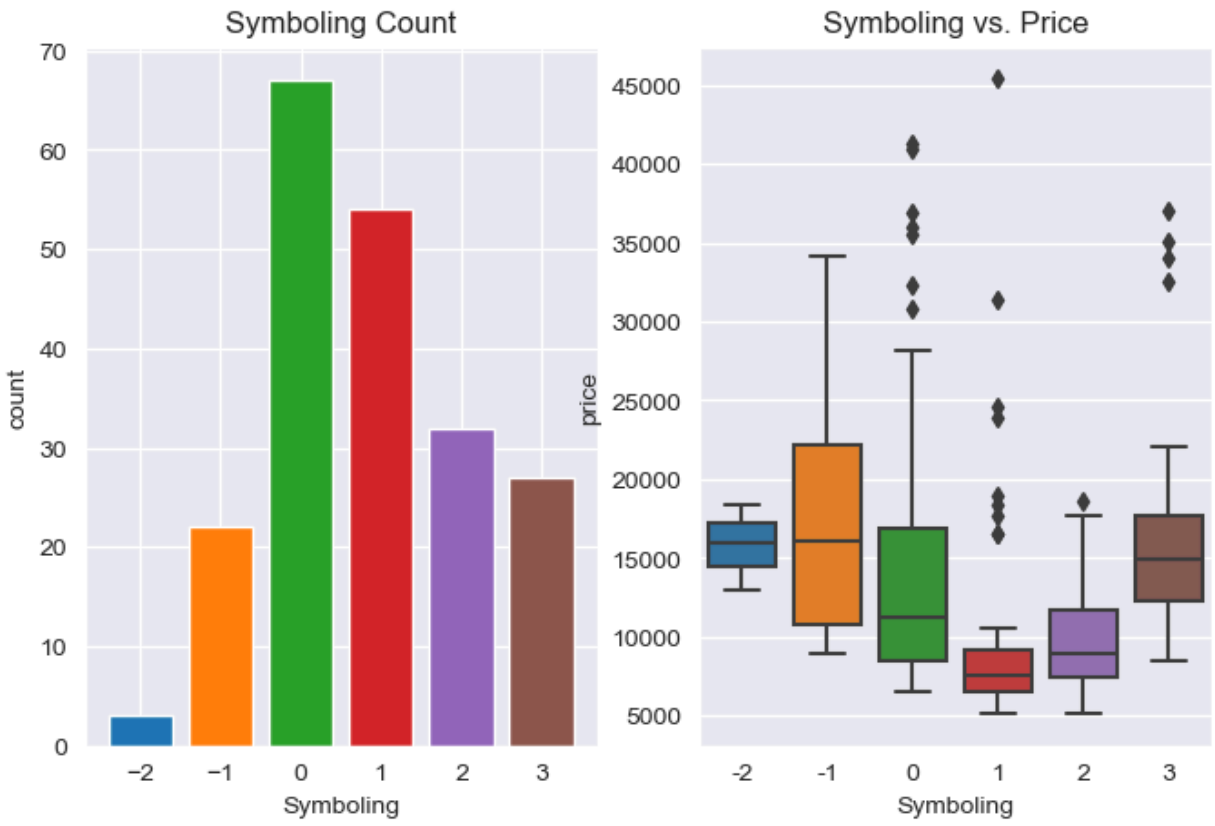
The 2 graphs show us the spread of car company and car body based on average prices. Looking at the first graph we can see that jaguar is most expensive while chevrolet is the least expensive. With the car body type hatchback is least expensive whereas hardtop is the most expensive.

3.3 Create the following plots

1. A plot of the unique symboling values on the x-axis, and the value counts on the y-axis.
2. A box plot of the unique symboling values on the x-axis and price on the y-axis.

[4 Points]

In [10]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
symboling_counts = cars_df['symboling'].value_counts().sort_index()
x0 = cars_df['symboling'].value_counts().sort_index().index
y0 = cars_df['symboling'].value_counts().sort_index().values

x1 = cars_df['symboling']
y1 = cars_df['price']

fig, axes = plt.subplots(nrows=1, ncols=2)
colors = plt.cm.tab10.colors[:len(symboling_counts)]

axes[0].bar(x0, y0, color=colors)
axes[0].set(title='Symboling Count', xlabel='Symboling', ylabel='count')
axes[0].set_xticks(symboling_counts.index)

sns.boxplot(x=x1, y=y1, ax=axes[1], palette=colors, order=symboling_counts.index)
axes[1].set(title='Symboling vs. Price', xlabel='Symboling', ylabel='price')

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

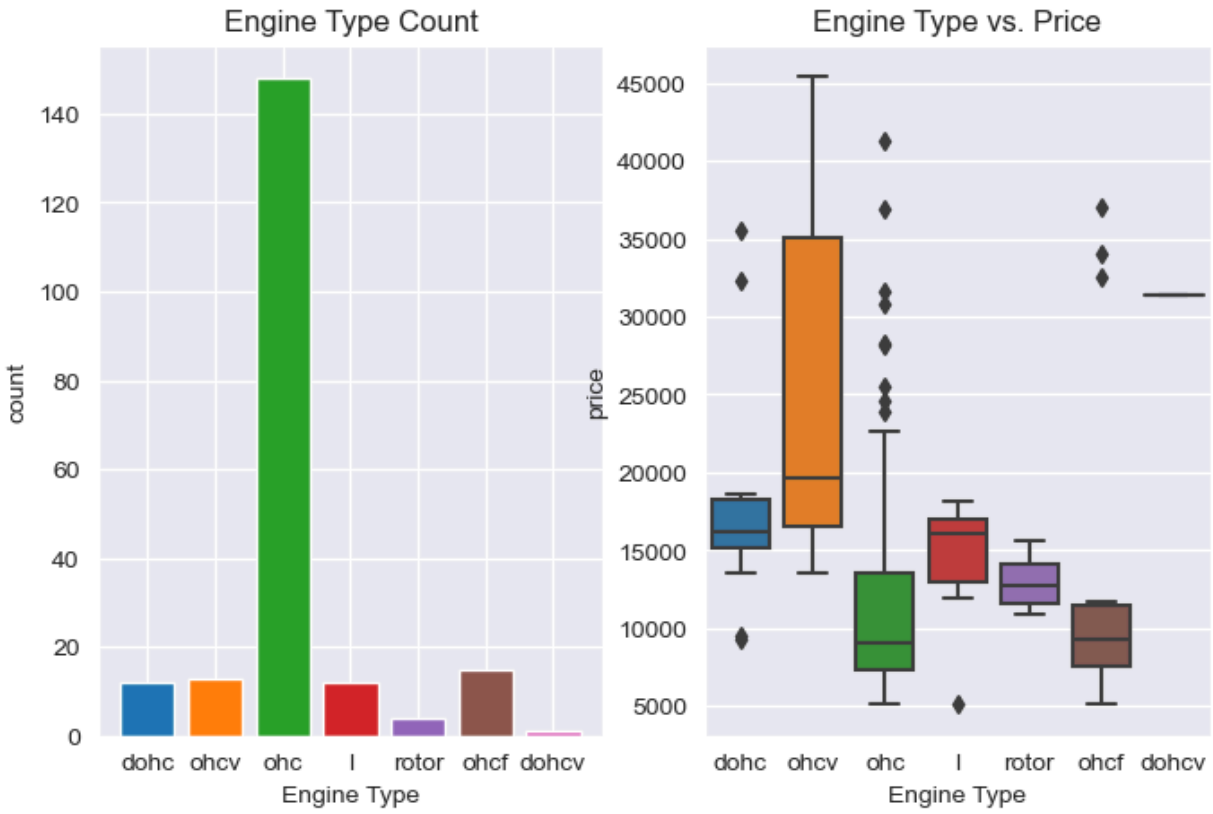
The first plot shows the count of how many times symboling values appears. The peak is at zero and the graph appears to be skewed left. The second graph is box plot of symboling versus car price, which helps visualize how prices vary across different symboling, which is risk factor. With some exception to outliers, there is a general trend of low symboling values is higher the price.

3.4 Create the following plots

1. A plot of enginetype on the x-axis, and the value counts on the y-axis.
2. A box plot of enginetype on the x-axis and price on the y-axis.

[4 Points]

In [11]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
enginetype_order = cars_df['enginetype'].unique()
enginetype_counts = cars_df['enginetype'].value_counts()
enginetype_counts = enginetype_counts[enginetype_order]
x0 = enginetype_counts.index
y0 = enginetype_counts.values

x1 = cars_df['enginetype']
y1 = cars_df['price']

fig, axes = plt.subplots(nrows=1, ncols=2)
colors = plt.cm.tab10.colors[:len(enginetype_counts)]

axes[0].bar(x0, y0, color=colors)
axes[0].set(title='Engine Type Count', xlabel='Engine Type', ylabel='count')
axes[0].set_xticks(enginetype_order)
axes[0].set_xticklabels(enginetype_order)

sns.boxplot(x=x1, y=y1, ax=axes[1], palette=colors, order=enginetype_order)
axes[1].set(title='Engine Type vs. Price', xlabel='Engine Type', ylabel='price')

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

The first graph shows the distrubition of different engine types. We can infere that ohc engine type is the most common. The box plot shows the distrubition of price and engine type. Based on the plot, it appears that ohcv engine type is more expensive while median of ohc engine type is the lowest price of all.

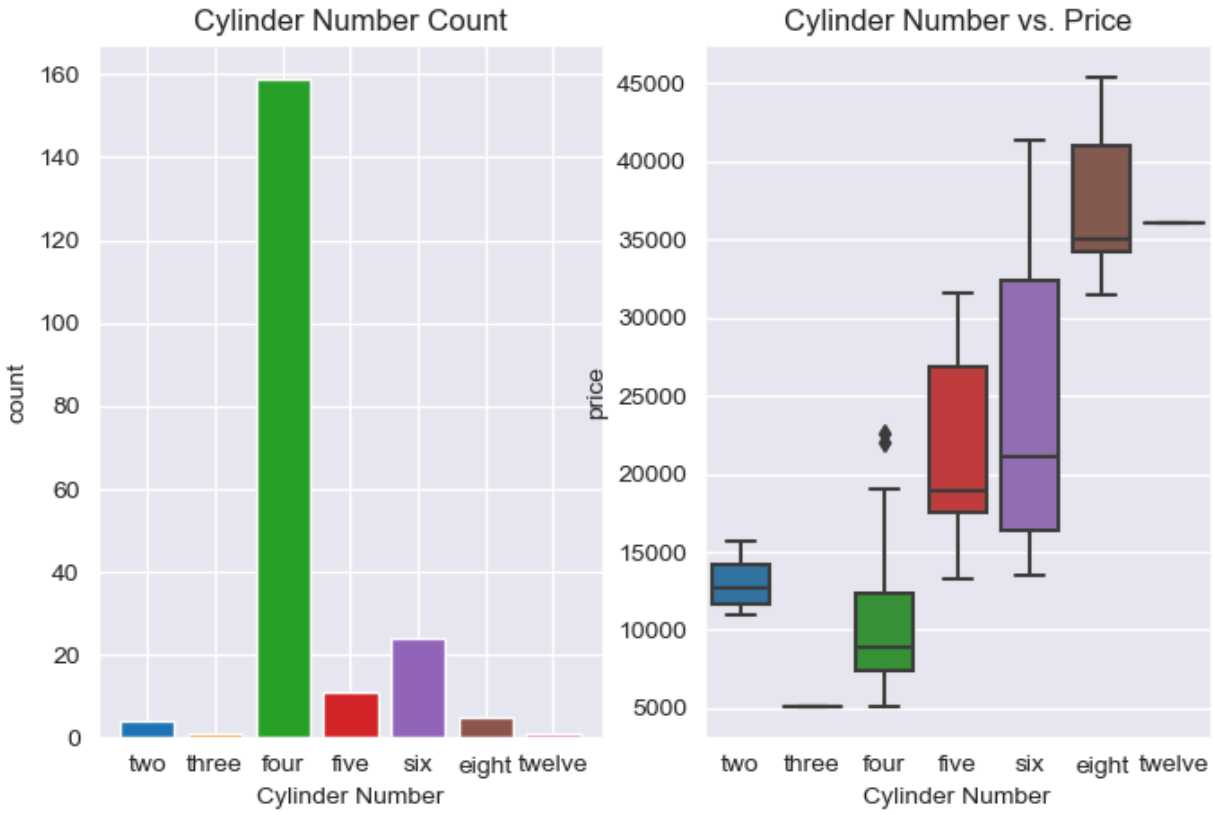
3.5 Create the following plots

1. A plot of cylindernumber on the x-axis, and the value counts on the y-axis.
2. A box plot of cylindernumber on the x-axis and price on the y-axis.

[4 Points]

In [12]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
# manually setting the order of x-axis
cylindernumber_order = ['two','three', 'four','five','six', 'eight','twelve']
cylindernumber_counts = cars_df['cylindernumber'].value_counts()

x1 = cars_df['cylindernumber']
y1 = cars_df['price']

fig, axes = plt.subplots(nrows=1, ncols=2)
```

```
colors = plt.cm.tab10.colors[:len(cylindernumber_counts)]

axes[0].bar(cylindernumber_order, [cylindernumber_counts[c] for c in cylindernumber_or
axes[0].set(title='Cylinder Number Count', xlabel='Cylinder Number', ylabel='count')

sns.boxplot(x=x1, y=y1, ax=axes[1], palette=colors, order=cylindernumber_order)
axes[1].set(title='Cylinder Number vs. Price', xlabel='Cylinder Number', ylabel='price

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

The first graphs shows the four cylinders are more common with the highest count. Three and twelve cylinder count is the lowest. The median price of four cylinder number is the lowest of all.

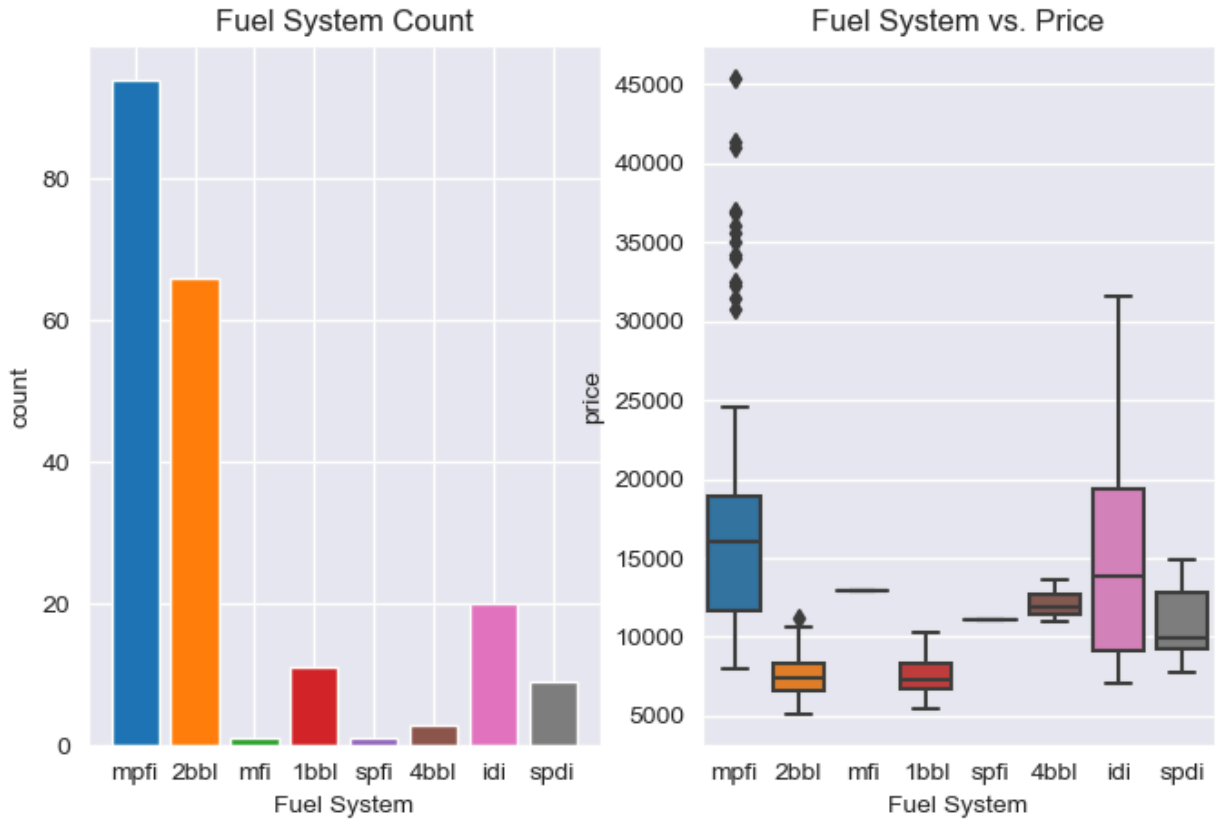
3.6 Create the following plots

1. A plot of fuelsystem on the x-axis, and the value counts on the y-axis.
2. A box plot of fuelsystem on the x-axis and price on the y-axis.

[5 Points]

In [13]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
fuelsystem_order = cars_df['fuelsystem'].unique()
fuelsystem_counts = cars_df['fuelsystem'].value_counts()
```

```
fuelsystem_counts = fuelsystem_counts[fuelsystem_order]
x0 = fuelsystem_counts.index
y0 = fuelsystem_counts.values

x1 = cars_df['fuelsystem']
y1 = cars_df['price']

fig, axes = plt.subplots(nrows=1, ncols=2)
colors = plt.cm.tab10.colors[:len(fuelsystem_counts)]

axes[0].bar(x0, y0, color=colors)
axes[0].set(title='Fuel System Count', xlabel='Fuel System', ylabel='count')
axes[0].set_xticks(fuelsystem_order)
axes[0].set_xticklabels(fuelsystem_order)

sns.boxplot(x=x1, y=y1, ax=axes[1], palette=colors, order=fuelsystem_order)
axes[1].set(title='Fuel System vs. Price', xlabel='Fuel System', ylabel='price')

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

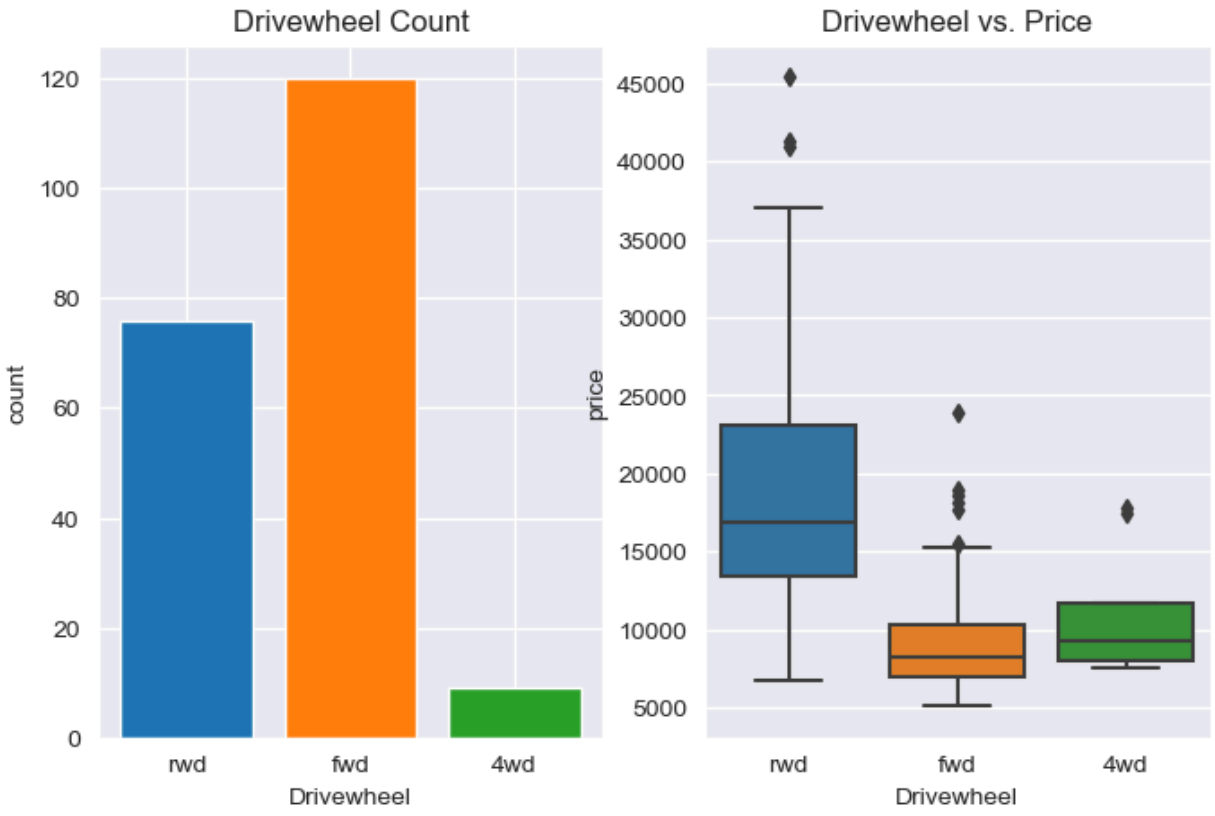
The first graph shows the distrubition of fuel systems and mpfi is the most common with the highest count followed by 2bbl. The median cost of 2bbl and 1bbl is about the same whereas the median price of mpfi fuel system is rather on the higher price range.

3.7 Create the following plots

1. A plot of drivewheel on the x-axis, and the value counts on the y-axis.
2. A box plot of drivewheel on the x-axis and price on the y-axis.

[5 Points]

In [14]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
drivewheel_order = cars_df['drivewheel'].unique()
drivewheel_counts = cars_df['drivewheel'].value_counts()
drivewheel_counts = drivewheel_counts[drivewheel_order]
x0 = drivewheel_counts.index
y0 = drivewheel_counts.values

x1 = cars_df['drivewheel']
y1 = cars_df['price']

fig, axes = plt.subplots(nrows=1, ncols=2)
colors = plt.cm.tab10.colors[:len(drivewheel_counts)]

axes[0].bar(x0, y0, color=colors)
axes[0].set(title='Drivewheel Count', xlabel='Drivewheel', ylabel='count')
axes[0].set_xticks(drivewheel_order)
axes[0].set_xticklabels(drivewheel_order)

sns.boxplot(x=x1, y=y1, ax=axes[1], palette=colors, order=drivewheel_order)
axes[1].set(title='Drivewheel vs. Price', xlabel='Drivewheel', ylabel='price')

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

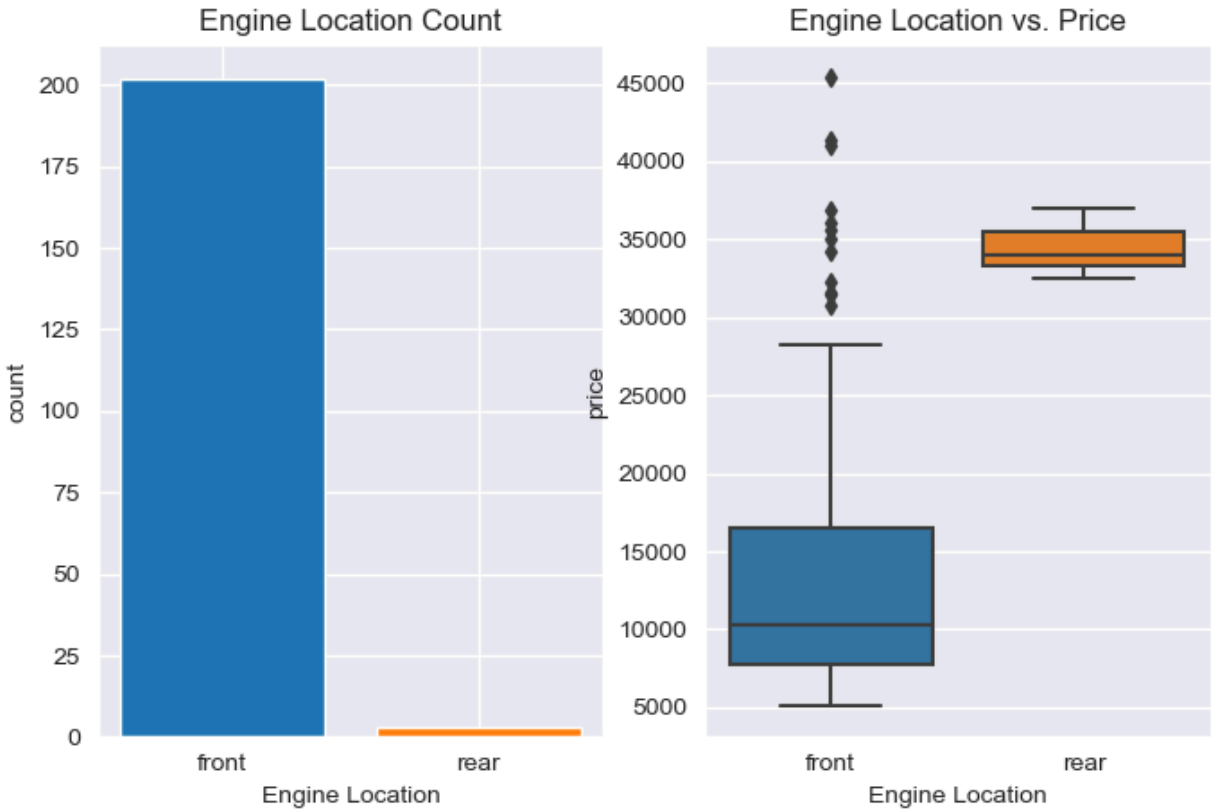
Based on the graph, front drive wheel is most common whereas 4 wheel drive is least common. The starting price of front drive wheels is lower than that of rear and 4 drive wheels.

3.8 Create the following plots

```
1. A plot of enginelocation on the x-axis, and the value counts on
the y-axis.
2. A box plot of enginelocation on the x-axis and price on the y-
axis.
```

[5 Points]

In [15]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
enginelocation_counts = cars_df['enginelocation'].value_counts()
x0 = enginelocation_counts.index
y0 = enginelocation_counts.values

x1 = cars_df['enginelocation']
y1 = cars_df['price']

fig, axes = plt.subplots(nrows=1, ncols=2)
colors = plt.cm.tab10.colors[:len(enginelocation_counts)]

axes[0].bar(x0, y0, color=colors)
axes[0].set(title='Engine Location Count', xlabel='Engine Location', ylabel='count')

sns.boxplot(x=x1, y=y1, ax=axes[1], palette=colors)
axes[1].set(title='Engine Location vs. Price', xlabel='Engine Location', ylabel='price

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

The most common engine location is the front which is also less expensive than that of rear engine location which is also very rare.

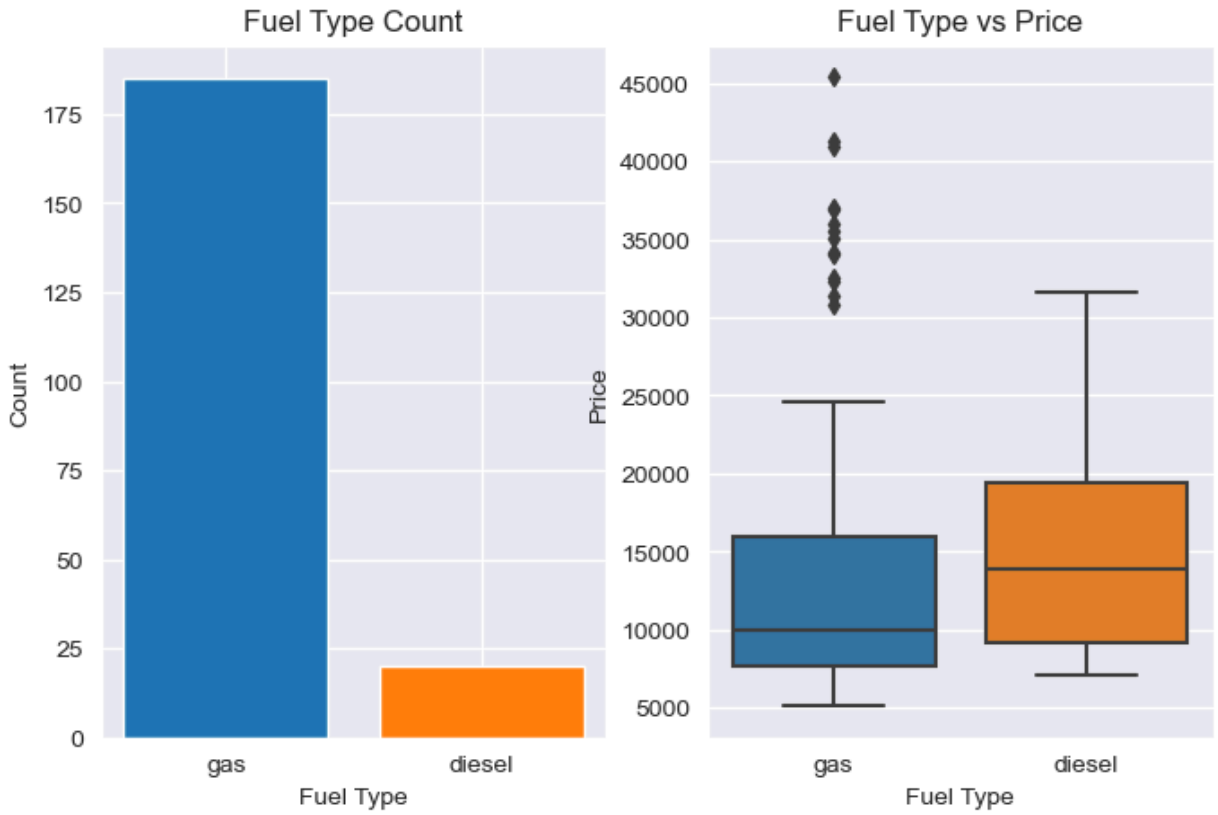
3.9 Create the following plots

1. A plot of fueltype on the x-axis, and the value counts on the y-axis.
2. A box plot of fueltype on the x-axis and price on the y-axis.

[5 Points]

In [16]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
fueltype_count = cars_df['fueltype'].value_counts()
x0 = fueltype_count.index
y0 = fueltype_count.values

x1 = cars_df['fueltype']
y1 = cars_df['price']

fig, axes = plt.subplots(nrows = 1, ncols = 2)
colors = plt.cm.tab10.colors[:len(fueltype_count)]

axes[0].bar(x0, y0, color = colors)
axes[0].set(title ='Fuel Type Count', xlabel ='Fuel Type', ylabel = 'Count')
```

```
sns.boxplot(x=x1, y=y1, palette = colors)
axes[1].set(title='Fuel Type vs Price', xlabel ='Fuel Type', ylabel='Price')

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

The most common fuel type is gas. But based on the box plot the prices of gas and diesel are not that much different. Though the prices of gas is slighty lower than that of diesel.

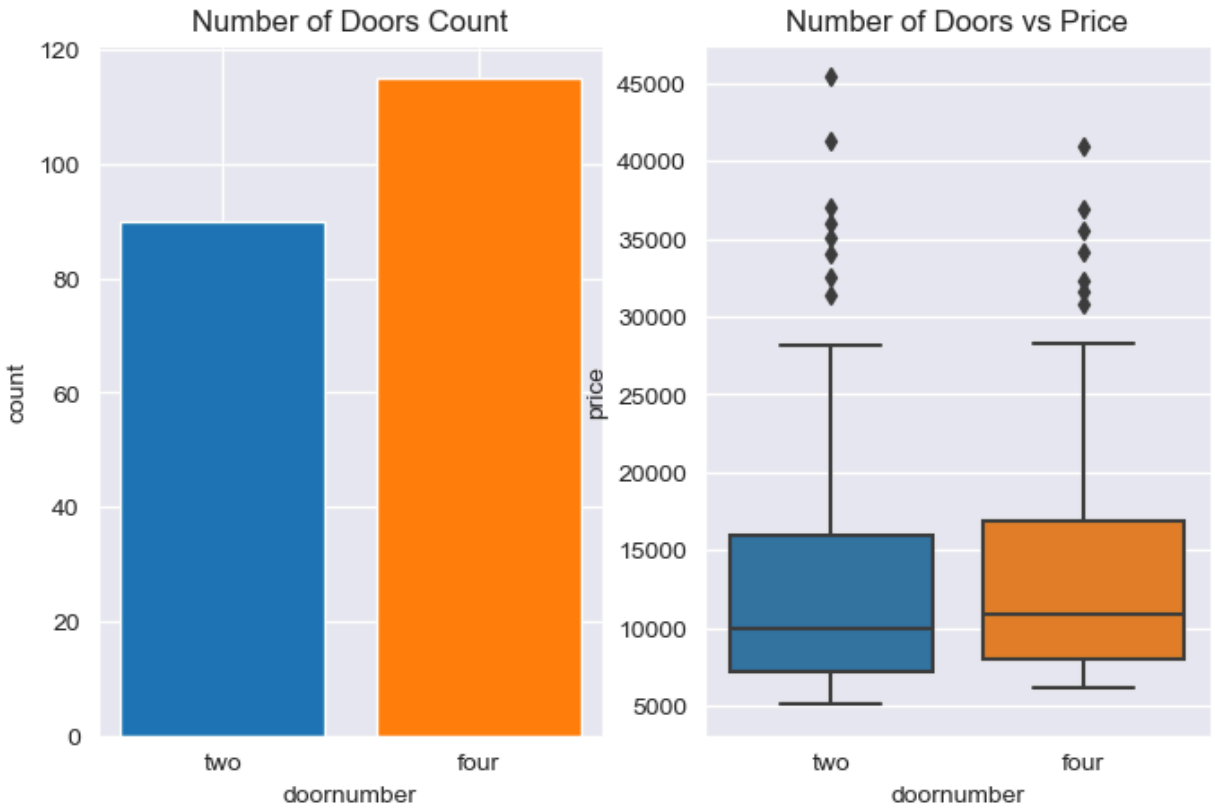
3.10 Create the following plots

1. A plot of doornumber on the x-axis, and the value counts on the y-axis.
2. A box plot of doornumber on the x-axis and price on the y-axis.

[5 Points]

In [17]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
doornumber_order = ['two', 'four']
doornumber_counts = cars_df['doornumber'].value_counts()

x1 = cars_df['doornumber']
y1 = cars_df['price']
```

```
fig, axes = plt.subplots(nrows = 1, ncols = 2)
colors = plt.cm.tab10.colors[:len(doornumber_counts)]

axes[0].bar(doornumber_order, [doornumber_counts[c] for c in doornumber_order], color=
axes[0].set(title='Number of Doors Count', xlabel='doornumber',ylabel='count')

sns.boxplot(x=x1, y=y1, palette = colors)
axes[1].set(title='Number of Doors vs Price', xlabel='doornumber',ylabel='price')

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

The first graph indicates that four doors are more common but the number of two doors cars are also not that far behind. As for the prices, they look almost the same.

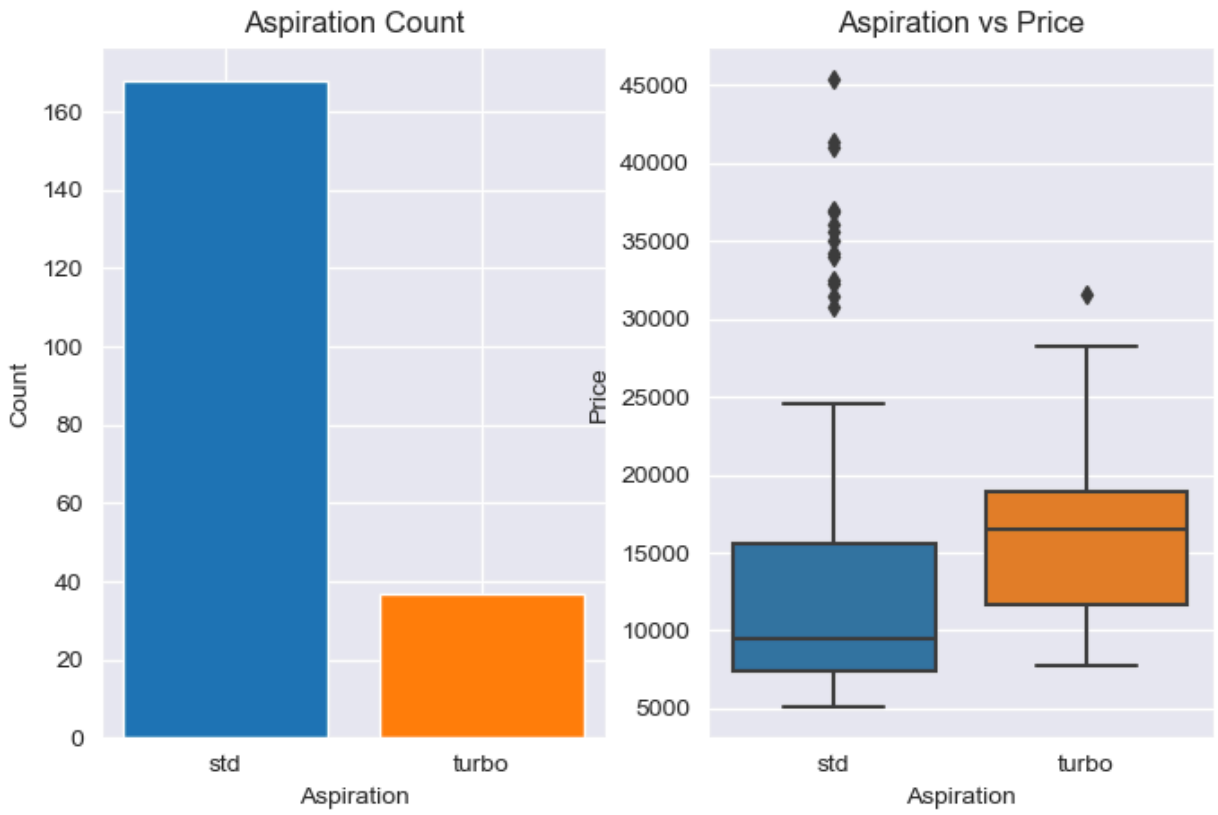
3.11 Create the following plots

1. A plot of aspiration on the x-axis, and the value counts on the y-axis.
2. A box plot of aspiration on the x-axis and price on the y-axis.

[5 Points]

In [18]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
aspiration_count = cars_df['aspiration'].value_counts()
```

```
x0 = aspiration_count.index
y0 = aspiration_count.values

x1 = cars_df['aspiration']
y1 = cars_df['price']

fig, axes = plt.subplots(nrows = 1, ncols = 2)
colors = plt.cm.tab10.colors[:len(aspiration_count)]

axes[0].bar(x0, y0, color = colors)
axes[0].set(title ='Aspiration Count', xlabel ='Aspiration', ylabel = 'Count')

sns.boxplot(x=x1, y=y1, palette = colors)
axes[1].set(title='Aspiration vs Price', xlabel ='Aspiration', ylabel='Price')

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

The first plot shows the distribution of standard vs. turbocharged engines. The standard is more common based on the graph. The second one shows that the prices of standard start at lower price point than that of a turbocharged one.

Part 4: Visualizing Numerical Data

[15 Points]

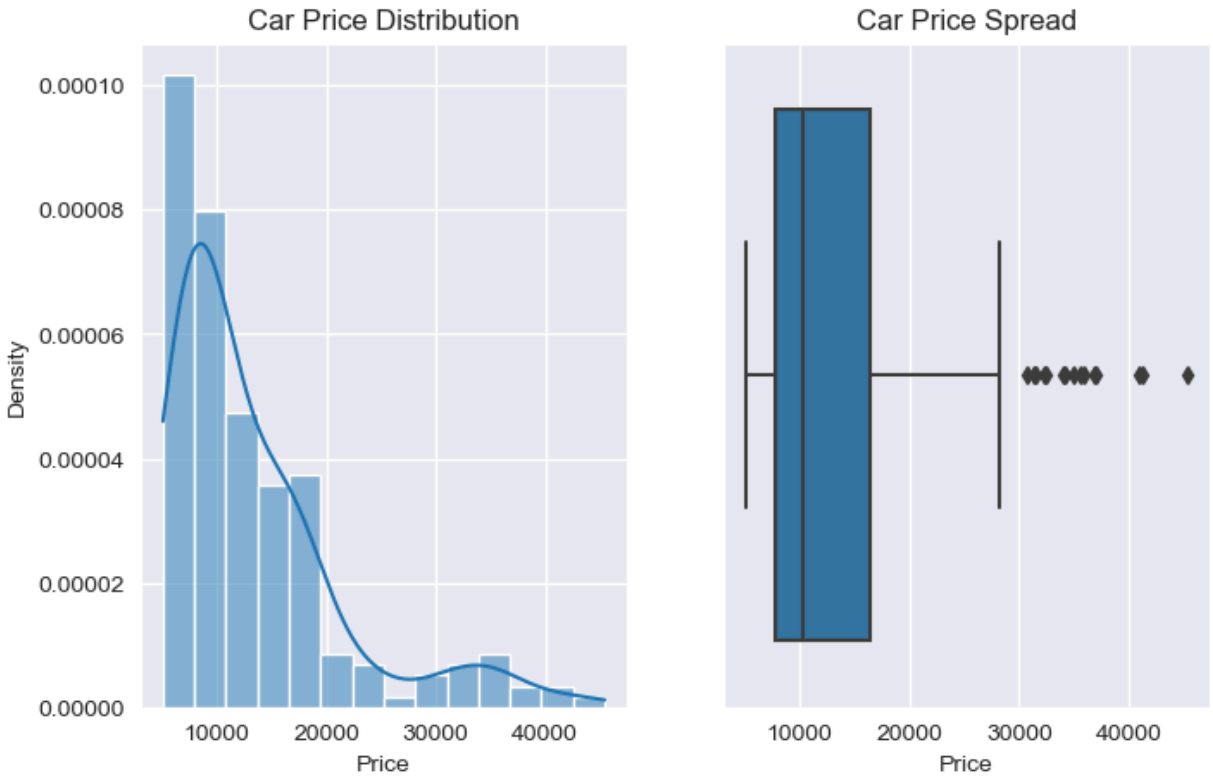
price
carlength
carwidth
carheight
curbweight
enginesize
boreratio
stroke
compressionratio
horsepower
peakrpm
wheelbase
citympg
highwaympg

4.1 Create the following plots

1. A plot showing the price distribution
2. A box plot of price

[3 Points]

In [19]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###

fig, axes = plt.subplots(nrows=1, ncols =2)

sns.histplot(cars_df['price'], kde=True, stat='density', ax=axes[0])
axes[0].set(title='Car Price Distribution', xlabel ='Price')

sns.boxplot(x=cars_df['price'], ax=axes[1])
axes[1].set(title='Car Price Spread', xlabel ='Price')

plt.show()
```



After creating these 2 plots, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

The price distribution plot is skewed right. This means that more cars are on the lower price point range. This can be further be supported by the price distribution seen in the box plot which shows the upper quartile under $20,000.

Let's try and get an idea of how the car attributes related to a car's size are related to price.

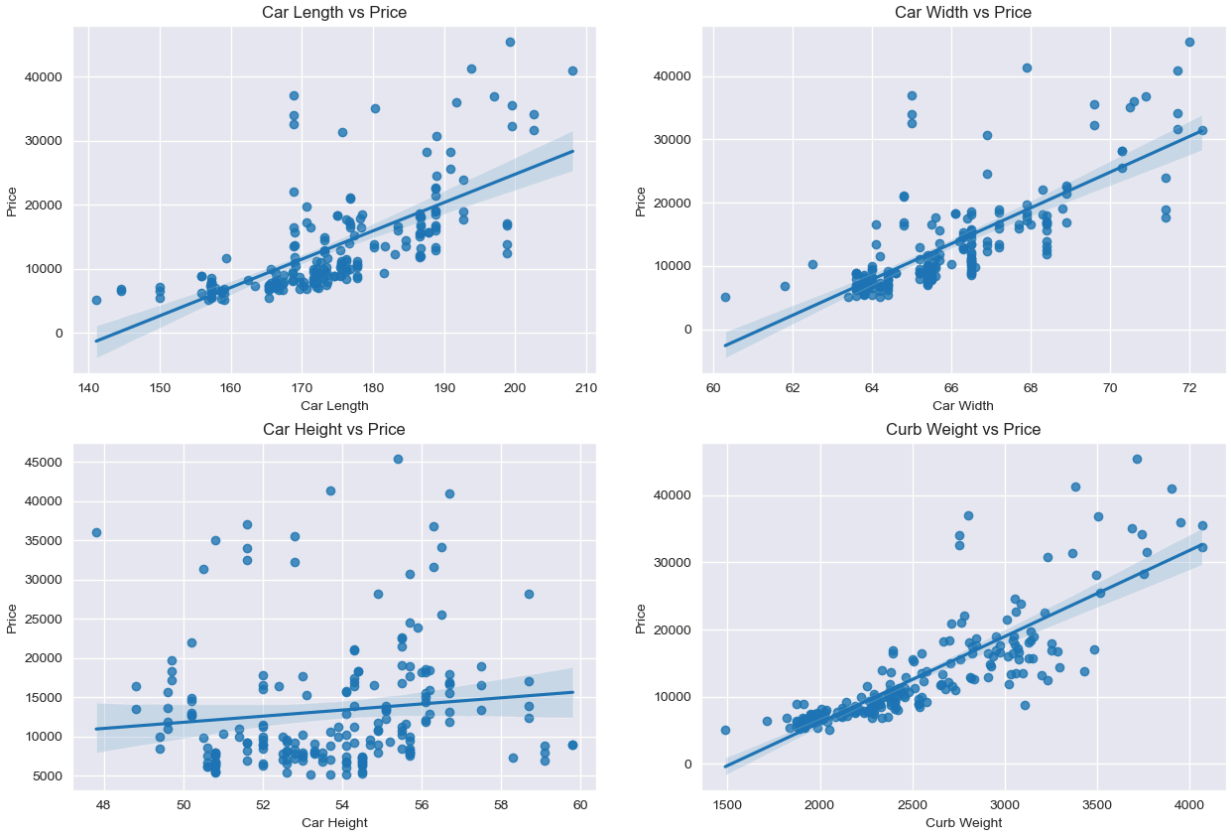
4.2 Create the following plots

1. A scatter plot of carlength vs price.
2. A scatter plot of carwidth vs price.
3. A scatter plot of carheight vs price.
4. A scatter plot of carweight vs price.

[3 Points]

In [20]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(15,10))
y_var = cars_df['price']
sns.regplot(x=cars_df['carlength'], y=y_var, ax=axes[0][0])
sns.regplot(x=cars_df['carwidth'], y=y_var, ax=axes[0][1])
sns.regplot(x=cars_df['carheight'], y=y_var, ax=axes[1][0])
sns.regplot(x=cars_df['curbweight'], y=y_var, ax=axes[1][1])

axes[0][0].set(title ='Car Length vs Price', xlabel ='Car Length', ylabel = 'Price')
```

```
axes[0][1].set(title ='Car Width vs Price', xlabel ='Car Width', ylabel = 'Price')
axes[1][0].set(title ='Car Height vs Price', xlabel ='Car Height', ylabel = 'Price')
axes[1][1].set(title ='Curb Weight vs Price', xlabel ='Curb Weight', ylabel = 'Price')

plt.show()
```



After creating the 4 plots above, describe what we can conclude from them.

TYPE YOUR ANSWER HERE

Based on the four scatter plots, there seems to be a positive linear coreelation between price and car length, width, and weight. So the higher the values of these three attributes, the more likely higher price.

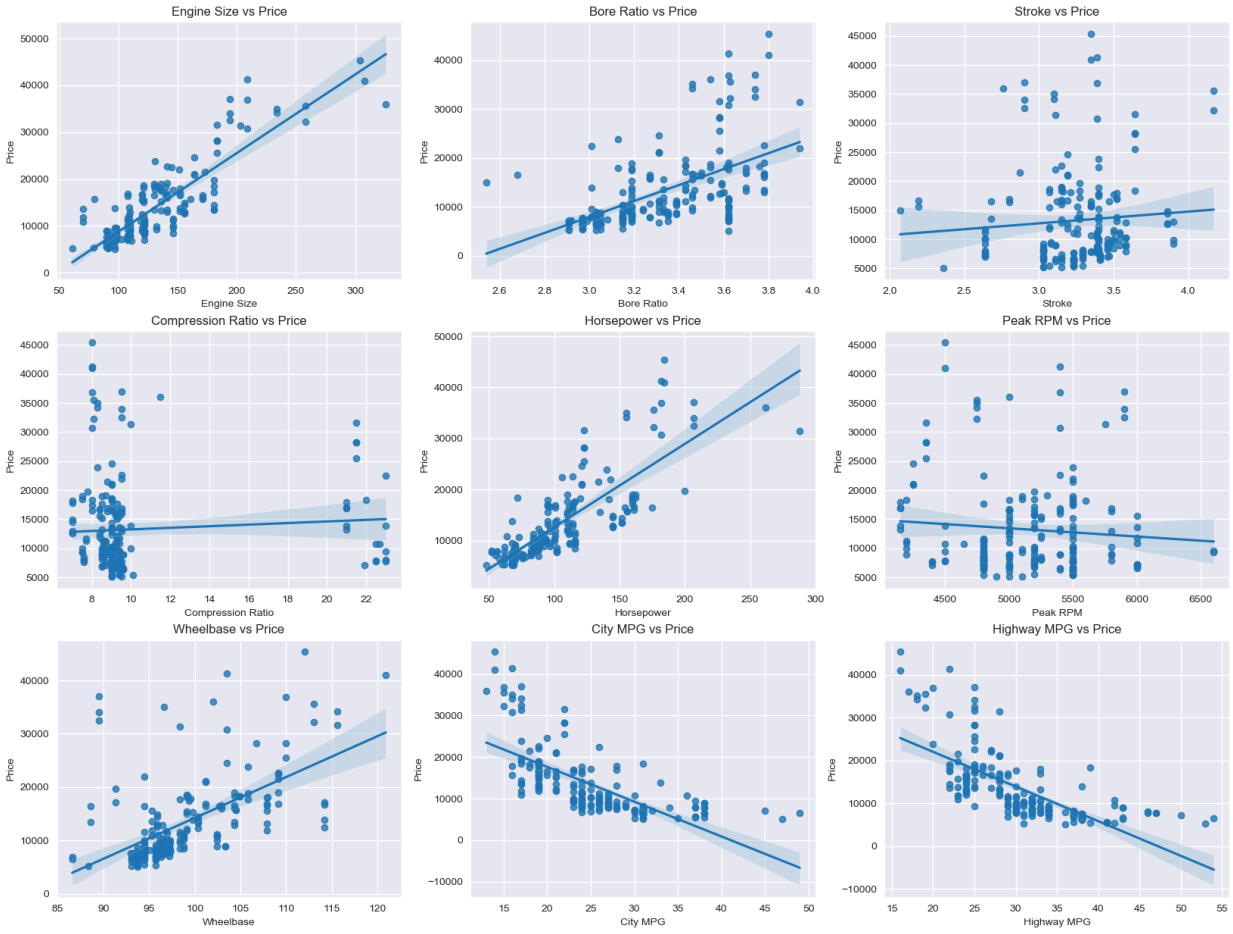
4.3 Create scatter plots of the remaining numerical variables to see their relationship with price.

Create the following plots

1. A scatter plot of enginesize vs price.
2. A scatter plot of boreratio vs price.
3. A scatter plot of stroke vs price.
4. A scatter plot of compressionratio vs price.
5. A scatter plot of horsepower vs price.
6. A scatter plot of peakrpm vs price.
7. A scatter plot of wheelbase vs price.
8. A scatter plot of citympg vs price.
9. A scatter plot of highwaympg vs price.

[3 Points]

In [21]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
fig, axes = plt.subplots(nrows=3, ncols=3, figsize=(20,15))
y_var = cars_df['price']
sns.regplot(x=cars_df['enginesize'], y=y_var, ax=axes[0][0])
sns.regplot(x=cars_df['boreratio'], y=y_var, ax=axes[0][1])
sns.regplot(x=cars_df['stroke'], y=y_var, ax=axes[0][2])
sns.regplot(x=cars_df['compressionratio'], y=y_var, ax=axes[1][0])
sns.regplot(x=cars_df['horsepower'], y=y_var, ax=axes[1][1])
sns.regplot(x=cars_df['peakrpm'], y=y_var, ax=axes[1][2])
sns.regplot(x=cars_df['wheelbase'], y=y_var, ax=axes[2][0])
sns.regplot(x=cars_df['citympg'], y=y_var, ax=axes[2][1])
sns.regplot(x=cars_df['highwaympg'], y=y_var, ax=axes[2][2])

axes[0][0].set(title ='Engine Size vs Price', xlabel ='Engine Size', ylabel = 'Price')
axes[0][1].set(title ='Bore Ratio vs Price', xlabel ='Bore Ratio', ylabel = 'Price')
axes[0][2].set(title ='Stroke vs Price', xlabel ='Stroke', ylabel = 'Price')
axes[1][0].set(title ='Compression Ratio vs Price', xlabel ='Compression Ratio', ylabe
axes[1][1].set(title ='Horsepower vs Price', xlabel ='Horsepower', ylabel = 'Price')
axes[1][2].set(title ='Peak RPM vs Price', xlabel ='Peak RPM', ylabel = 'Price')
axes[2][0].set(title ='Wheelbase vs Price', xlabel ='Wheelbase', ylabel = 'Price')
axes[2][1].set(title ='City MPG vs Price', xlabel ='City MPG', ylabel = 'Price')
axes[2][2].set(title ='Highway MPG vs Price', xlabel ='Highway MPG', ylabel = 'Price')

plt.show()
```



After creating the plots above, describe what we can conclude from them.

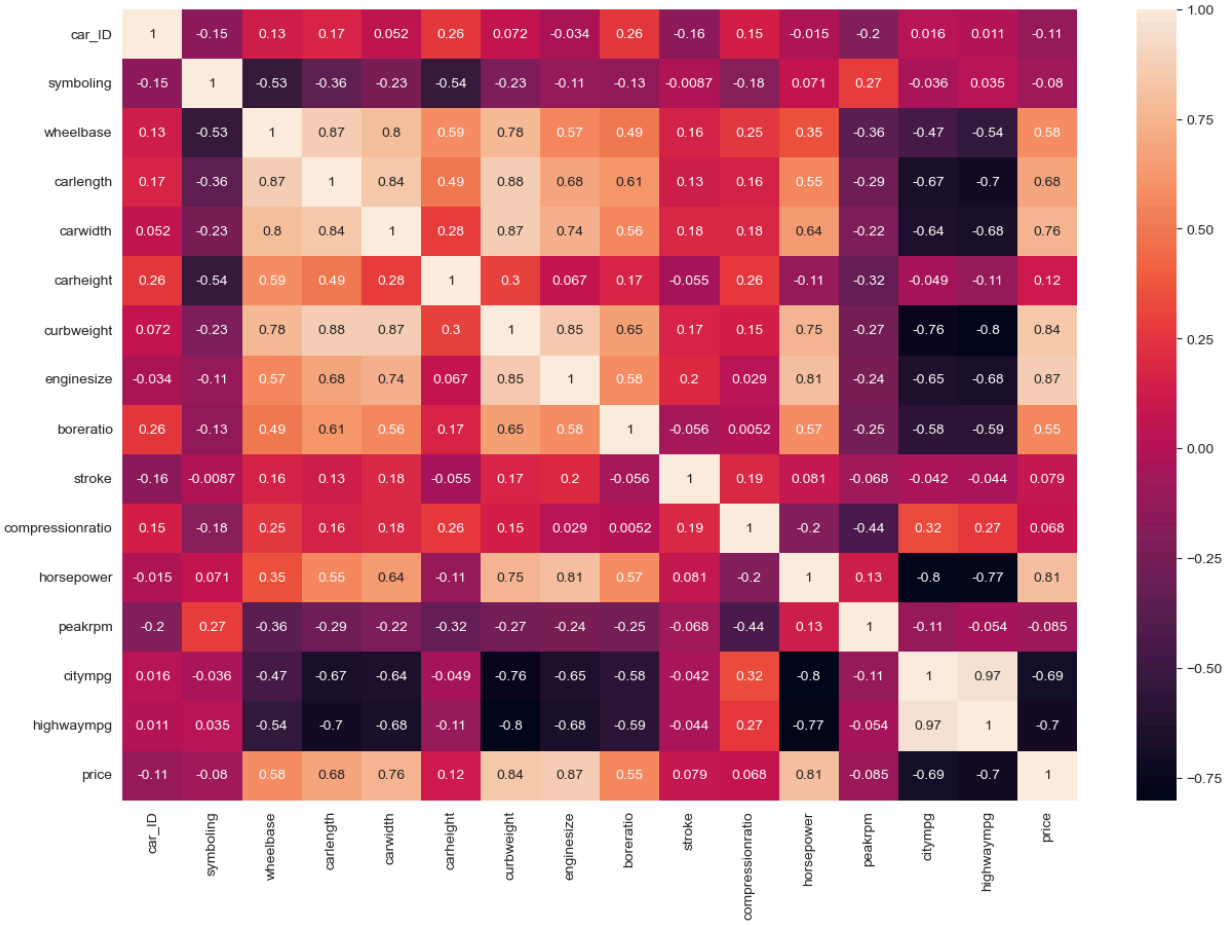
TYPE YOUR ANSWER HERE

The plot with engine size, horsepower, wheelbase, and bore ratio vs price showed positive linear correlation. The plots with city mpg, highway mpg vs price showed a negative linear correlation. The graphs with stroke, compression, and peak rpm vs shows are more constant based on the scatter plots above. But they do have huge clusters that are more concentrated around a small range like in the compression ratio vs price plot.

4.4 Create a heatmap or correlation matrix to inspect the correlations in our dataset.

[3 Points]

In [22]:
```
## WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###

# print(cars_df.dtypes)
selected_data = cars_df.select_dtypes(include=['int64', 'float64'])
plt.figure(figsize=(15,10))
sns.heatmap(selected_data.corr(), annot=True)
plt.show()
```



After creating the plot above, what can you conclude? Are there any features you can combine to form a new one? If so, try it out and see how it affects your final results in the end.

TYPE YOUR ANSWER HERE

The heapmap helps us identify variables that are highly correlated with each other. A high correlation is indicated with values closer to 1 or -1. The variables that have moderate to high positive/negative correlations like symboling, wheelbase, carlenght, carwidth, carheight, curbweight, enginesize, etc. to name a few help us decide on what variables to consider to keep when building and testing our model. We can also combine some hight correlated variables like citympg and highwaympg which both have a correlation of 0.97.

4.5 For example, citympg and highwaympg can be combined into a single feature. Create a new column called 'fuel_economy' that's a combination of the 2.

In [23]:
```
### WRITE YOUR CODE HERE ###
cars_df['fuel_economy'] = (cars_df['citympg'] + cars_df['highwaympg']) / 2
print(cars_df[['citympg', 'highwaympg', 'fuel_economy']].head())
```

```
   citympg  highwaympg  fuel_economy
0       21          27          24.0
1       21          27          24.0
2       19          26          22.5
3       24          30          27.0
4       18          22          20.0
```

After visual analysis, which variables do you believe to be significant when predicting price, and why?

[3 Points]

TYPE YOUR ANSWER HERE

After analyzing the bar graphs, scatter plot, box plot, and heatmap we can identify variables that have significant influence in predicting price. Based on the distribution graphs and box plot we can begin by saying that engine type, cylinder number, fuel system, drivewheel, engine location, fuel type gas, number of doors, aspiration in the categorical section showed high correlation with price. The scatter plots of car length, width, heigh, weight, and engine size, bore ratio, stroke, horsepower, wheelbase, city and highway mpg all showed either positive or negative linear correlation with price. These conclusions are futher supported by the correlation matrix (heatmap) created. The variable that showed moderate to high positive and negative correlation are variables that have significant influence in predicting the price. The plots before the heatmap visually showed how each variable correlated with price.

Part 5: Data Pre-Processing

[5 Points]

Perform the following

1. Convert your categorical variables into dummy variables
2. Scale the data using a scaler of your choice

3. Split your data into a training and testing set, with test size of 0.30

In [24]:
```
### WRITE YOUR CODE HERE ###
# from practice of week 4, week 5, and week 7
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

cars_df = pd.get_dummies(cars_df, drop_first=True)
```

In [28]:
```
# columns to only be considered based on the sample coefficient output on hw
expected_features = [
    'symboling', 'wheelbase', 'carlength', 'carwidth', 'carheight', 'curbweight',
    'enginesize', 'boreratio', 'stroke', 'compressionratio', 'horsepower', 'peakrpm',
    'fuel_economy', 'fueltype_gas', 'aspiration_turbo', 'doornumber_two',
    'carbody_hardtop', 'carbody_hatchback', 'carbody_sedan', 'carbody_wagon',
    'drivewheel_fwd', 'drivewheel_rwd', 'enginelocation_rear', 'enginetype_dohcv',
    'enginetype_l', 'enginetype_ohc', 'enginetype_ohcf', 'enginetype_ohcv',
    'enginetype_rotor', 'cylindernumber_five', 'cylindernumber_four',
    'cylindernumber_six', 'cylindernumber_three', 'cylindernumber_twelve',
    'cylindernumber_two', 'fuelsystem_2bbl', 'fuelsystem_4bbl', 'fuelsystem_idi',
    'fuelsystem_mfi', 'fuelsystem_mpfi', 'fuelsystem_spdi', 'fuelsystem_spfi'
]

X = cars_df.drop(columns=[col for col in cars_df.columns if col not in expected_featur

y = cars_df['price']
```

In [29]:
```
# Scale the data and reassign X to a new DataFrame
scaler = StandardScaler()
X_columns = X.columns
X_scaled = scaler.fit_transform(X)
X = pd.DataFrame(data = X_scaled, columns = X_columns)
# scale y
scaler_y = StandardScaler()
y = scaler_y.fit_transform(y.values.reshape(-1,1)).flatten()

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=

# print(cars_df.columns)
```

Part 6: Model Creation and Evaluation

[15 Points]

Perform the following using sklearn

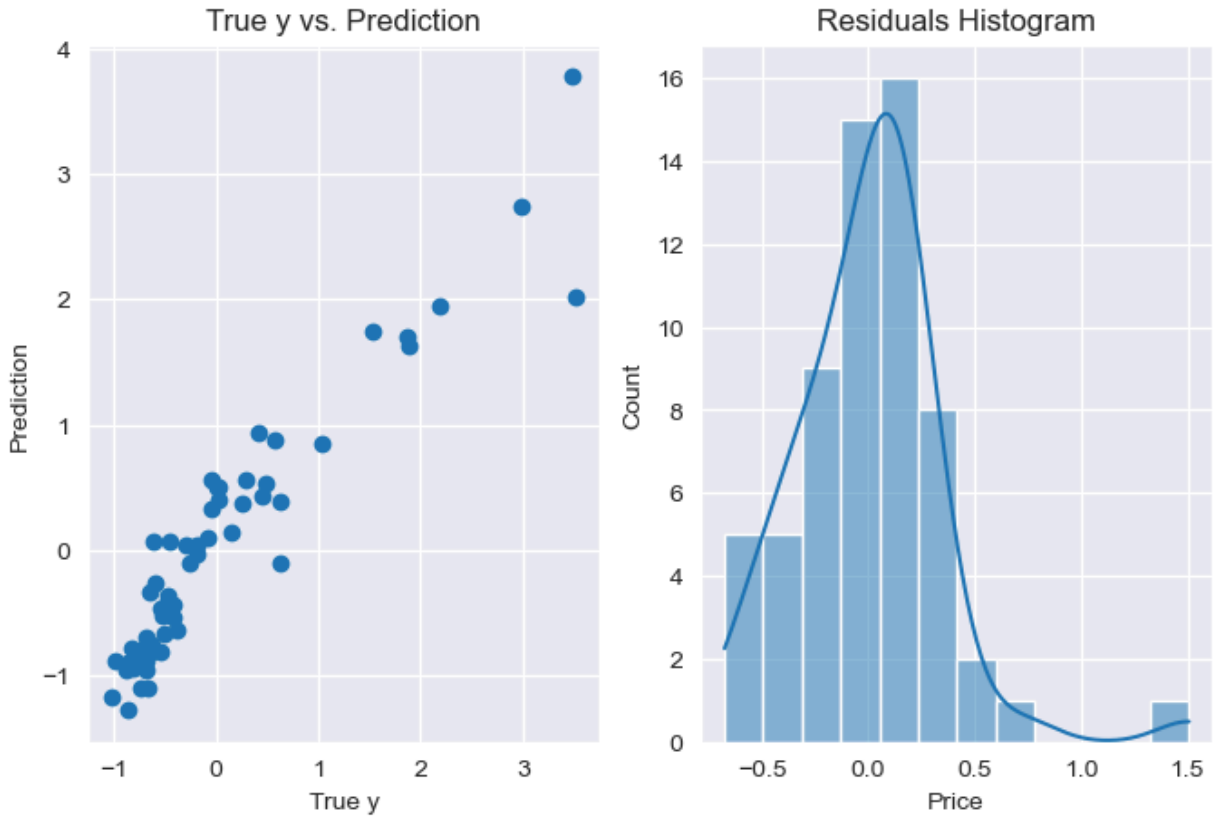
1. Create a linear regression model, and train (fit) it on the training data.
2. Run the test data through your model to obtain predictions. Save these predictions into a variable called 'predictions'.
4. Create a scatter plot of the true price labels vs the predicted price value of your model.
5. Create a histogram of the residuals

6. Print the R^2 of your model

Note: You don't need to obtain the same results as us. If you made any changes earlier and obtain better results, then even better. Just make sure you're not scoring significantly lower than our obtained values.

[8 Points]

In [30]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score

lm = LinearRegression()
lm.fit(X_train, y_train)

fig, axes = plt.subplots(nrows = 1, ncols=2)

predictions = lm.predict(X_test)
axes[0].scatter(y_test, predictions)
axes[0].set(title='True y vs. Prediction', xlabel ='True y', ylabel='Prediction')


residuals = y_test - predictions
sns.histplot(residuals, kde=True, ax=axes[1])
axes[1].set(title='Residuals Histogram', xlabel ='Price', ylabel='Count')

plt.show()

print('R2 Score: ', r2_score(y_test, predictions))
```



```
R2 Score:  0.89655265628398
```

Lastly, create a dataframe of your model's coefficents. For example, we obtained the coefficients below.

[7 Points]

In [31]:
```
### WRITE YOUR CODE HERE, WHICH SHOULD REPRODUCE THE BELOW IF DONE CORRECTLY ###
coeff_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coefficient'])
coeff_df
```

Out[31]:

	Coefficient
symboling	1.467697e-02
wheelbase	3.741429e-02
carlength	-9.176459e-02
carwidth	1.738382e-01
carheight	3.018805e-02
curbweight	2.465155e-01
enginesize	6.126339e-01
boreratio	-5.575562e-02
stroke	-1.602166e-01
compressionratio	-4.682939e-02
horsepower	2.384646e-02
peakrpm	1.106497e-01
fuel_economy	-2.649687e-02
fueltype_gas	-3.280601e-02
aspiration_turbo	9.244647e-02
doornumber_two	-1.135213e-02
carbody_hardtop	-4.118011e-02
carbody_hatchback	-1.544803e-01
carbody_sedan	-1.054529e-01
carbody_wagon	-1.503869e-01
drivewheel_fwd	-2.113964e-02
drivewheel_rwd	1.603523e-02
enginelocation_rear	1.336516e-01
enginetype_dohcv	-4.530519e-02
enginetype_l	-2.936111e-02
enginetype_ohc	1.308010e-01
enginetype_ohcf	-2.259193e-03
enginetype_ohcv	-1.481000e-01
enginetype_rotor	3.762625e-03
cylindernumber_five	-2.033360e-01
cylindernumber_four	-4.353723e-01
cylindernumber_six	-2.559219e-01
cylindernumber_three	5.551115e-17
cylindernumber_twelve	-1.027384e-01

	Coefficient
cylindernumber_two	3.762625e-03
fuelsystem_2bbl	3.043863e-03
fuelsystem_4bbl	5.276404e-04
fuelsystem_idi	3.280601e-02
fuelsystem_mfi	5.281549e-31
fuelsystem_mpfi	1.532362e-02
fuelsystem_spdi	-5.361942e-02
fuelsystem_spfi	-5.608494e-03

What do **your** coefficients mean? Write a conclusion answering the problem statement we listed earlier, and what you've learned from the model.

TYPE YOUR CONCLUSION HERE

The r-squared value of my model is 0.896, this tells us that 89.6% of the variation in car prices can be explained by the independent variables used in the model. This number tells us how well my model fits the data and an 0.896 indicates a pretty good fit and a good predictor. The coefficients tells us how each attribute or our independet variable influence our price (or our dependent variable) so a higher coefficient indicated a strong influence. Based on the output of my model, enginesize has the highest coefficient value of +0.61 indicating that a increase in engine is directly proportional to increase in price. cylindernumber_four has a coefficient value of -0.44 indicating a negative correlation meaning that an increase of cylindernumber_four means a decrease in car's price. Based on my model's coefficient output, engine size, curb weight, car width, enginelocation_rear, drivetype_rwd, and aspiration_turbo all have positve correlation with highest to lowest in the order written indicating positive impact on car prices.

The variables used to build my model are listed in the output showing the coefficients. All those variable have significant influence in predicting the prices as showen by the graphs and the r-squared value of my model. The coefficients show how well these variables describe the price of a car.

In [ ]: