Data Introduction

- The data here we have taken is about automobiles and the data available from the kaggle website.The data set consists of three types of entities:

1. The specialization of an auto in terms of various characteristics
2. It’s assigned insurance risk rating
3. It’s normalized losses in use as compared to other cars

Importing the data

- **Importing the libraries**

In [1]:
```
import pandas as pd
import numpy as np
```

load data and store in dataframe df:

In [2]:
```
path='https://s3-api.us-geo.objectstorage.softlayer.net/cf-courses-data/CognitiveClass/DA0101EN/automobileEDA.csv'
df = pd.read_csv(path)
df.head()
```

Out[2]:

	symboling	normalized-losses	make	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	...	compression-ratio	h
0	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...	9.0	1
1	3	122	alfa-romero	std	two	convertible	rwd	front	88.6	0.811148	...	9.0	1
2	1	122	alfa-romero	std	two	hatchback	rwd	front	94.5	0.822681	...	9.0	1
3	2	164	audi	std	four	sedan	fwd	front	99.8	0.848630	...	10.0	1
4	2	164	audi	std	four	sedan	4wd	front	99.4	0.848630	...	8.0	1

5 rows × 29 columns

Exploring the data

- **Analyse the data with feature selection and visualization**
- Matplotlib and seaborn are the libraries for visualization of data

```
In [4]: import matplotlib.pyplot as plt
        import seaborn as sns
        %matplotlib inline
```

When visualising individual variables, it is important to first understand what type of variable you are dealing with. This will help us find the right visualisation method for that variable.

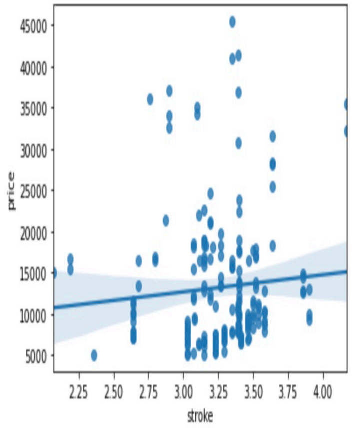
```
In [5]: # list the data types for each column
        print(df.dtypes)

symboling              int64
normalized-losses      int64
make                  object
aspiration            object
num-of-doors          object
body-style            object
drive-wheels          object
engine-location       object
wheel-base           float64
length               float64
width                float64
height               float64
curb-weight            int64
engine-type           object
num-of-cylinders      object
engine-size            int64
fuel-system           object
bore                 float64
stroke               float64
compression-ratio    float64
horsepower           float64
peak-rpm             float64
city-mpg               int64
highway-mpg            int64
price                float64
city-L/100km         float64
horsepower-binned     object
diesel                 int64
gas                    int64
dtype: object
```

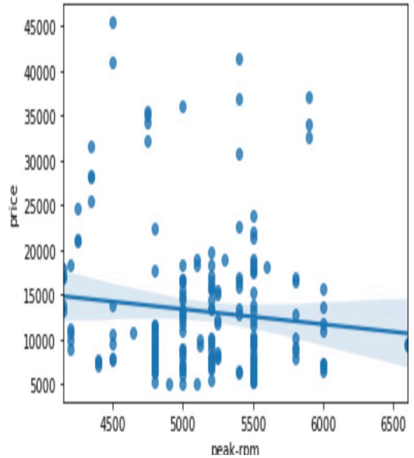
A great way to visualize these variables is by using scatterplots with fitted lines. In order to start understanding the (linear) relationship between an individual variable and the price. We can do this by using "regplot", which plots the scatterplot plus the fitted regression line for the data.

In [15]:
```python
sns.regplot(x="stroke", y="price", data=df)
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x7fed72ae6d10>

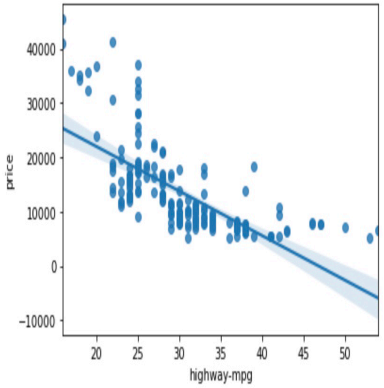


Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x7fed72c03150>



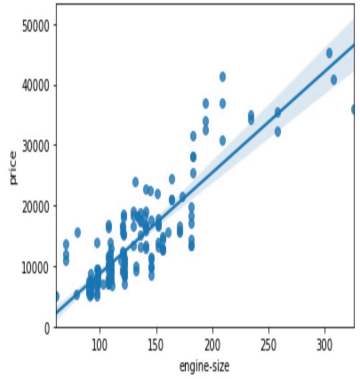
In [10]:
```python
sns.regplot(x="highway-mpg", y="price", data=df)
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x7fed72bee310>



In [8]:
```python
sns.regplot(x="engine-size", y="price", data=df)
plt.ylim(0,)
```

Out[8]: (0.0, 53284.69121042168)



Statistical analysis

- Let's first take a look at the variables by utilizing a description method.
- The **describe** function automatically computes basic statistics for all continuous variables. Any NaN values are automatically skipped in these statistics.

In [19]:
```
df.describe()
```

Out[19]:

	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore
count	201.000000	201.00000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000
mean	0.840796	122.00000	98.797015	0.837102	0.915126	53.766667	2555.666667	126.875622	3.330692
std	1.254802	31.99625	6.066366	0.059213	0.029187	2.447822	517.296727	41.546834	0.268072
min	-2.000000	65.00000	86.600000	0.678039	0.837500	47.800000	1488.000000	61.000000	2.540000
25%	0.000000	101.00000	94.500000	0.801538	0.890278	52.000000	2169.000000	98.000000	3.150000
50%	1.000000	122.00000	97.000000	0.832292	0.909722	54.100000	2414.000000	120.000000	3.310000
75%	2.000000	137.00000	102.400000	0.881788	0.925000	55.500000	2926.000000	141.000000	3.580000
max	3.000000	256.00000	120.900000	1.000000	1.000000	59.800000	4066.000000	326.000000	3.940000

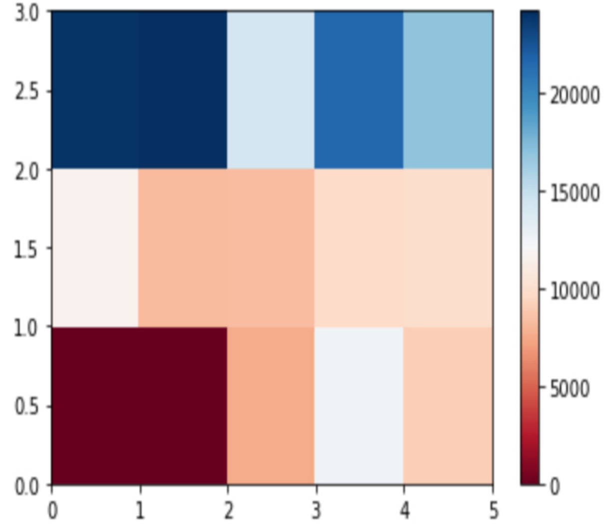
The default setting of "describe" skips variables of type object. We can apply the method "describe" on the variables of type 'object' as follows:

In [20]:
```
df.describe(include=['object'])
```

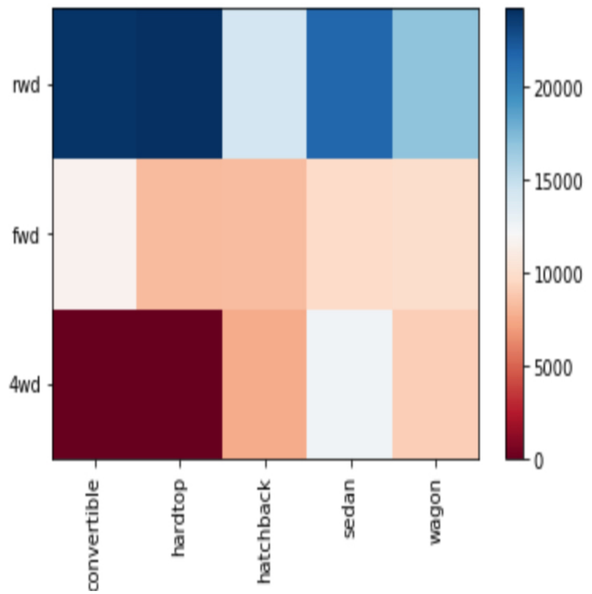
Out[20]:

	make	aspiration	num-of-doors	body-style	drive-wheels	engine-location	engine-type	num-of-cylinders	fuel-system	horsepower-binned
count	201	201	201	201	201	201	201	201	201	200
unique	22	2	2	5	3	2	6	7	8	3
top	toyota	std	four	sedan	fwd	front	ohc	four	mpfi	Low
freq	32	165	115	94	118	198	145	157	92	115

Let's use a heat map to visualise the relationship between Variables.



The heatmap plots the target variable (price) proportional to colour with respect to the variables in the vertical and horizontal axis respectively. This allows us to visualise how the variables are related .



- The main question we want to answer in this module, is "What are the main characteristics which have the most impact on the car price?".
- To get a better measure of the important characteristics, we look at the correlation of these variables with the car price, in other words: how is the car price dependent on this variable?

In [36]:
```
df.corr()
```

Out[36]:

	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	strok
symboling	1.000000	0.466264	-0.535987	-0.365404	-0.242423	-0.550160	-0.233118	-0.110581	-0.140019	-0.00
normalized-losses	0.466264	1.000000	-0.056661	0.019424	0.086802	-0.373737	0.099404	0.112360	-0.029862	0.05
wheel-base	-0.535987	-0.056661	1.000000	0.876024	0.814507	0.590742	0.782097	0.572027	0.493244	0.15
length	-0.365404	0.019424	0.876024	1.000000	0.857170	0.492063	0.880665	0.685025	0.608971	0.12
width	-0.242423	0.086802	0.814507	0.857170	1.000000	0.306002	0.866201	0.729436	0.544885	0.18
height	-0.550160	-0.373737	0.590742	0.492063	0.306002	1.000000	0.307581	0.074694	0.180449	-0.06
curb-weight	-0.233118	0.099404	0.782097	0.880665	0.866201	0.307581	1.000000	0.849072	0.644060	0.16
engine-size	-0.110581	0.112360	0.572027	0.685025	0.729436	0.074694	0.849072	1.000000	0.572609	0.20
bore	-0.140019	-0.029862	0.493244	0.608971	0.544885	0.180449	0.644060	0.572609	1.000000	-0.05
stroke	-0.008245	0.055563	0.158502	0.124139	0.188829	-0.062704	0.167562	0.209523	-0.055390	1.00
compression-ratio	-0.182196	-0.114713	0.250313	0.159733	0.189867	0.259737	0.156433	0.028889	0.001263	0.18
horsepower	0.075819	0.217299	0.371147	0.579821	0.615077	-0.087027	0.757976	0.822676	0.566936	0.09
peak-rpm	0.279740	0.239543	-0.360305	-0.285970	-0.245800	-0.309974	-0.279361	-0.256733	-0.267392	-0.06
city-mpg	-0.035527	-0.225016	-0.470606	-0.665192	-0.633531	-0.049800	-0.749543	-0.650546	-0.582027	-0.03
highway-mpg	0.036233	-0.181877	-0.543304	-0.698142	-0.680635	-0.104812	-0.794889	-0.679571	-0.591309	-0.03
price	-0.082391	0.133999	0.584642	0.690628	0.751265	0.135486	0.834415	0.872335	0.543155	0.08
city-L/100km	0.066171	0.238567	0.476153	0.657373	0.673363	0.003811	0.785353	0.745059	0.554610	0.03
diesel	-0.196735	-0.101546	0.307237	0.211187	0.244356	0.281578	0.221046	0.070779	0.054458	0.24
gas	0.196735	0.101546	-0.307237	-0.211187	-0.244356	-0.281578	-0.221046	-0.070779	-0.054458	-0.24

- **Hypothesis testing**
- Pearson coefficient
- The P-value is the probability value that the correlation between these two variables is statistically significant. Normally, we choose a significance level of 0.05, which means that we are 95% confident that the correlation between the variables is significant.
- By convention, when the
- p-value is 0.001: we say there is strong evidence that the correlation is significant.
- the p-value is 0.05: there is moderate evidence that the correlation is significant.
- the p-value is 0.1: there is weak evidence that the correlation is significant.
- the p-value is 0.1: there is no evidence that the correlation is significant.

ANOVA: Analysis of variance

- ANOVA is to test whether there are significant differences between the means of two or more groups. It returns two parameters:
- **F-test score**: ANOVA assumes the means of all groups are the same, calculates how much the actual means deviate from the assumption, and reports it as the F-test score. A larger score means there is a larger difference between the means.
- **P-value**: P-value tells how statistically significant is our calculated score value.
- If our price variable is strongly correlated with the variable we are analyzing, expect ANOVA to return a sizeable F-test score and a small p-value.

Conclusion: Important Variables

- We now have a better idea of what our data looks like and which variables are important to take into account when predicting the car price. We have narrowed it down to the following variables:
- Continuous numerical variables:
- Length
- Width
- Curb-weight
- Engine-size
- Horsepower
- City-mpg
- Highway-mpg
- Wheel-base
- Bore
- Categorical variables:
- Drive-wheels
- As we now move into building machine learning models to automate our analysis, feeding the model with variables that meaningfully affect our target variable will improve our model's prediction performance.