Importing the Dataset

```
pip install ucimlrepo
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.6-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.6
```

```
from ucimlrepo import fetch_ucirepo

# fetch dataset
automobile = fetch_ucirepo(id=10)

# data (as pandas dataframes)
X = automobile.data.features
y = automobile.data.targets
```

```
X
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	fuel-system	engine-size	...	length	wheel-base	engine-location	drive-wheels	
0	13495.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	130	...	168.8	88.6	front	rwd	conv
1	16500.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	130	...	168.8	88.6	front	rwd	conv
2	16500.0	26	19	5000.0	154.0	9.0	3.47	2.68	mpfi	152	...	171.2	94.5	front	rwd	hat
3	13950.0	30	24	5500.0	102.0	10.0	3.40	3.19	mpfi	109	...	176.6	99.8	front	fwd	
4	17450.0	22	18	5500.0	115.0	8.0	3.40	3.19	mpfi	136	...	176.6	99.4	front	4wd	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
200	16845.0	28	23	5400.0	114.0	9.5	3.15	3.78	mpfi	141	...	188.8	109.1	front	rwd	
201	19045.0	25	19	5300.0	160.0	8.7	3.15	3.78	mpfi	141	...	188.8	109.1	front	rwd	
202	21485.0	23	18	5500.0	134.0	8.8	2.87	3.58	mpfi	173	...	188.8	109.1	front	rwd	
203	22470.0	27	26	4800.0	106.0	23.0	3.40	3.01	idi	145	...	188.8	109.1	front	rwd	
204	22625.0	25	19	5400.0	114.0	9.5	3.15	3.78	mpfi	141	...	188.8	109.1	front	rwd	

205 rows × 25 columns

```
y
```

	symboling
0	3
1	3
2	1
3	2
4	2
...	...
200	-1
201	-1
202	-1
203	-1
204	-1

205 rows × 1 columns

Next steps: View recommended plots

Concatenating the Dataset

```
!pip install hvplot
```

```
Collecting hvplot
  Downloading hvplot-0.9.2-py2.py3-none-any.whl (1.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 8.7 MB/s eta 0:00:00
Requirement already satisfied: bokeh>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from hvplot) (3.3.4)
Requirement already satisfied: colorcet>=2 in /usr/local/lib/python3.10/dist-packages (from hvplot) (3.1.0)
Requirement already satisfied: holoviews>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from hvplot) (1.17.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from hvplot) (2.0.3)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.10/dist-packages (from hvplot) (1.25.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from hvplot) (24.0)
Requirement already satisfied: panel>=0.11.0 in /usr/local/lib/python3.10/dist-packages (from hvplot) (1.3.8)
Requirement already satisfied: param<3.0,>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from hvplot) (2.1.0)
Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (3.1.3)
Requirement already satisfied: contourpy>=1 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (1.2.1)
Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (9.4.0)
Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (6.0.1)
Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (6.3.3)
Requirement already satisfied: xyzservices>=2021.09.1 in /usr/local/lib/python3.10/dist-packages (from bokeh>=1.0.0->hvplot) (2024.4.0)
Requirement already satisfied: pyviz-comms>=0.7.4 in /usr/local/lib/python3.10/dist-packages (from holoviews>=1.11.0->hvplot) (3.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->hvplot) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->hvplot) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas->hvplot) (2024.1)
Requirement already satisfied: markdown in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (3.6)
Requirement already satisfied: markdown-it-py in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (3.0.0)
Requirement already satisfied: linkify-it-py in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (2.0.3)
Requirement already satisfied: mdit-py-plugins in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (0.4.0)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (2.31.0)
Requirement already satisfied: tqdm>=4.48.0 in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (4.66.2)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (6.1.0)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/dist-packages (from panel>=0.11.0->hvplot) (4.11.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from Jinja2>=2.9->bokeh>=1.0.0->hvplot) (2.1.
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->hvplot) (1.16.0
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->panel>=0.11.0->hvplot) (0.5.1)
Requirement already satisfied: uc-micro-py in /usr/local/lib/python3.10/dist-packages (from linkify-it-py->panel>=0.11.0->hvplot) (1.0.3
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.10/dist-packages (from markdown-it-py->panel>=0.11.0->hvplot) (0.1.2
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->panel>=0.11.0->hvplot
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->panel>=0.11.0->hvplot) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->panel>=0.11.0->hvplot) (2.0
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->panel>=0.11.0->hvplot) (202
Installing collected packages: hvplot
Successfully installed hvplot-0.9.2
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import hvplot.pandas

from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.linear_model import LinearRegression
%matplotlib inline
```

```
data = pd.concat([X, y], axis = 1)
data
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	fuel-system	engine-size	...	wheel-base	engine-location	drive-wheels	body-style	c
0	13495.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	130	...	88.6	front	rwd	convertible	
1	16500.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	130	...	88.6	front	rwd	convertible	
2	16500.0	26	19	5000.0	154.0	9.0	3.47	2.68	mpfi	152	...	94.5	front	rwd	hatchback	
3	13950.0	30	24	5500.0	102.0	10.0	3.40	3.19	mpfi	109	...	99.8	front	fwd	sedan	
4	17450.0	22	18	5500.0	115.0	8.0	3.40	3.19	mpfi	136	...	99.4	front	4wd	sedan	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
200	16845.0	28	23	5400.0	114.0	9.5	3.15	3.78	mpfi	141	...	109.1	front	rwd	sedan	
201	19045.0	25	19	5300.0	160.0	8.7	3.15	3.78	mpfi	141	...	109.1	front	rwd	sedan	
202	21485.0	23	18	5500.0	134.0	8.8	2.87	3.58	mpfi	173	...	109.1	front	rwd	sedan	
203	22470.0	27	26	4800.0	106.0	23.0	3.40	3.01	idi	145	...	109.1	front	rwd	sedan	
204	22625.0	25	19	5400.0	114.0	9.5	3.15	3.78	mpfi	141	...	109.1	front	rwd	sedan	

205 rows × 26 columns

Checking for NaN Values

```
missing_values = data.isnull().sum()
print(missing_values)
```

```
price                4
highway-mpg          0
city-mpg             0
peak-rpm             2
horsepower           2
compression-ratio    0
stroke               4
bore                 4
fuel-system          0
engine-size          0
num-of-cylinders     0
engine-type          0
curb-weight          0
height               0
width                0
length               0
wheel-base           0
engine-location      0
drive-wheels         0
body-style           0
num-of-doors         2
aspiration           0
fuel-type            0
make                 0
normalized-losses    41
symboling            0
dtype: int64
```

Dropping categorical columns

```
data.drop(columns = ['fuel-system', 'engine-location', 'drive-wheels', 'body-style', 'aspiration', 'fuel-type', 'make', 'engine-type'], inpl
data
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	engine-size	num-of-cylinders	curb-weight	height	width	length	wheel-base
0	13495.0	27	21	5000.0	111.0	9.0	2.68	3.47	130	4	2548	48.8	64.1	168.8	88.6
1	16500.0	27	21	5000.0	111.0	9.0	2.68	3.47	130	4	2548	48.8	64.1	168.8	88.6
2	16500.0	26	19	5000.0	154.0	9.0	3.47	2.68	152	6	2823	52.4	65.5	171.2	94.5
3	13950.0	30	24	5500.0	102.0	10.0	3.40	3.19	109	4	2337	54.3	66.2	176.6	99.8
4	17450.0	22	18	5500.0	115.0	8.0	3.40	3.19	136	5	2824	54.3	66.4	176.6	99.4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	16845.0	28	23	5400.0	114.0	9.5	3.15	3.78	141	4	2952	55.5	68.9	188.8	109.1
201	19045.0	25	19	5300.0	160.0	8.7	3.15	3.78	141	4	3049	55.5	68.8	188.8	109.1
202	21485.0	23	18	5500.0	134.0	8.8	2.87	3.58	173	6	3012	55.5	68.9	188.8	109.1
203	22470.0	27	26	4800.0	106.0	23.0	3.40	3.01	145	6	3217	55.5	68.9	188.8	109.1
204	22625.0	25	19	5400.0	114.0	9.5	3.15	3.78	141	4	3062	55.5	68.9	188.8	109.1

Next steps: View recommended plots

Getting all the columns with null values

```
null_columns = []

for x in data.columns:
    if data[x].isnull().any():
        null_columns.append(x)

null_columns
```

```
['price',
 'peak-rpm',
 'horsepower',
 'stroke',
 'bore',
 'num-of-doors',
 'normalized-losses']
```

Mean imputation

```
for x in null_columns:
  data[x] = data[x].fillna(data[x].mean())

data
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	engine-size	c
0	13495.0	27	21	5000.0	111.0	9.0	2.68	3.47	130	
1	16500.0	27	21	5000.0	111.0	9.0	2.68	3.47	130	
2	16500.0	26	19	5000.0	154.0	9.0	3.47	2.68	152	
3	13950.0	30	24	5500.0	102.0	10.0	3.40	3.19	109	
4	17450.0	22	18	5500.0	115.0	8.0	3.40	3.19	136	
...	...	...	...	...	...	...	...	...	...	
200	16845.0	28	23	5400.0	114.0	9.5	3.15	3.78	141	
201	19045.0	25	19	5300.0	160.0	8.7	3.15	3.78	141	
202	21485.0	23	18	5500.0	134.0	8.8	2.87	3.58	173	
203	22470.0	27	26	4800.0	106.0	23.0	3.40	3.01	145	
204	22625.0	25	19	5400.0	114.0	9.5	3.15	3.78	141	

Next steps: View recommended plots

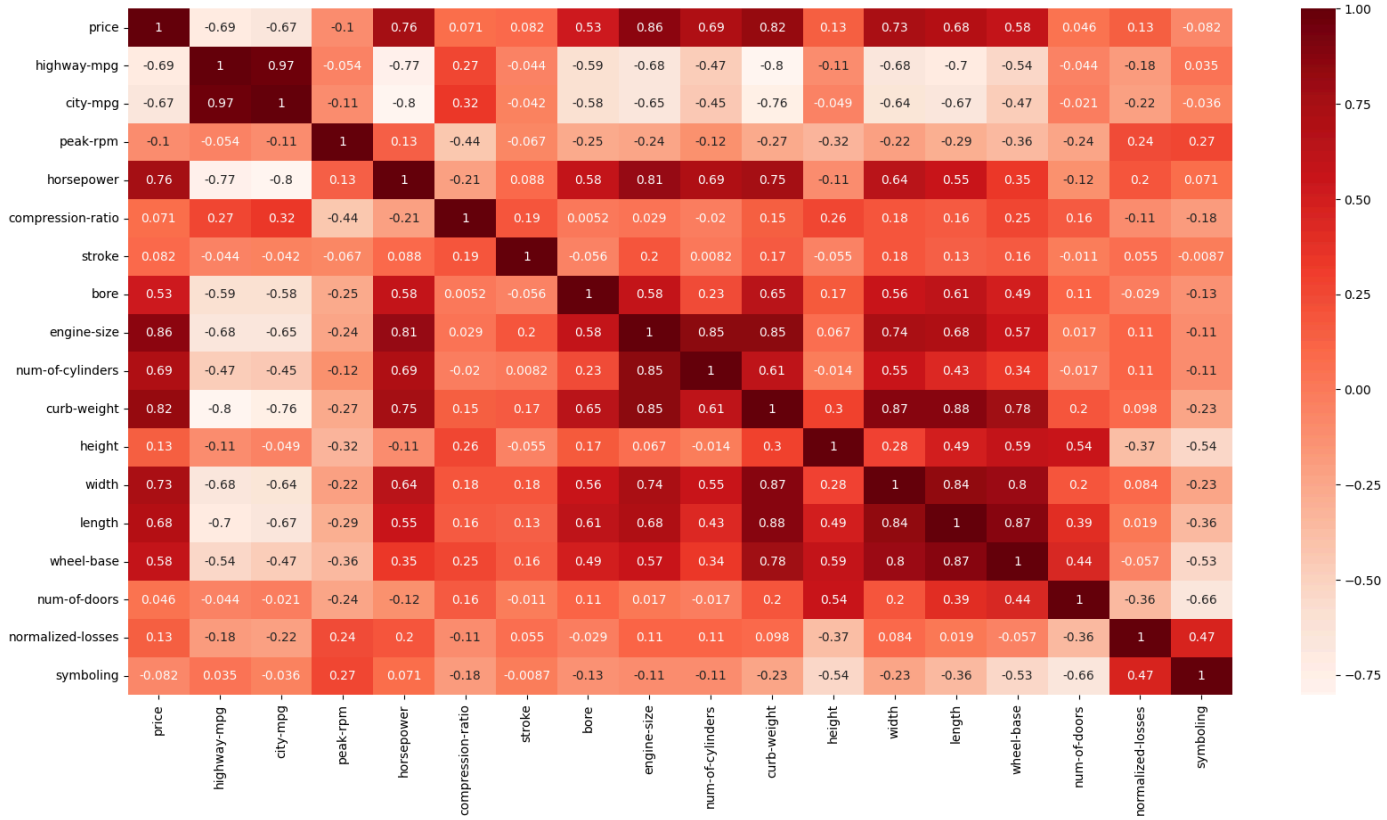
Checking again if there is null

```
missing_values = data.isnull().sum()
print(missing_values)
```

```
price               0
highway-mpg         0
city-mpg            0
peak-rpm            0
horsepower          0
compression-ratio   0
stroke              0
bore                0
engine-size         0
num-of-cylinders    0
curb-weight         0
height              0
width               0
length              0
wheel-base          0
num-of-doors        0
normalized-losses   0
symboling           0
dtype: int64
```

Correlation of the Columns using heatmap

```
plt.figure(figsize = (20,10))
ax = sns.heatmap(data.corr(), annot = True, cmap = 'Reds')
```



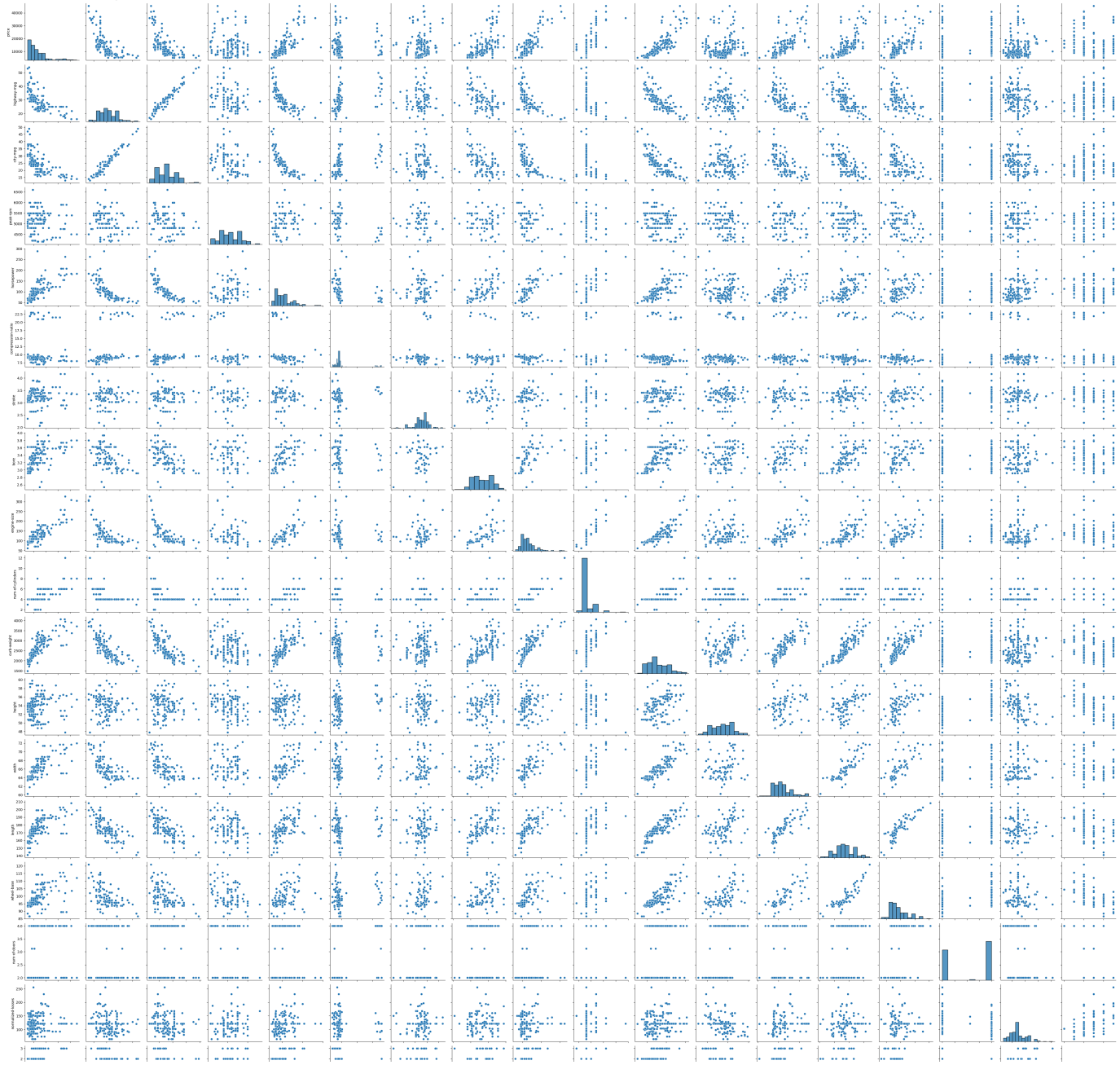
Observing our heatmap, we can see good correlations or good relevance of 2 columns, we can use those columns for our predictions

---

Pairplotting the Dataframe

```
sns.pairplot(data)
```

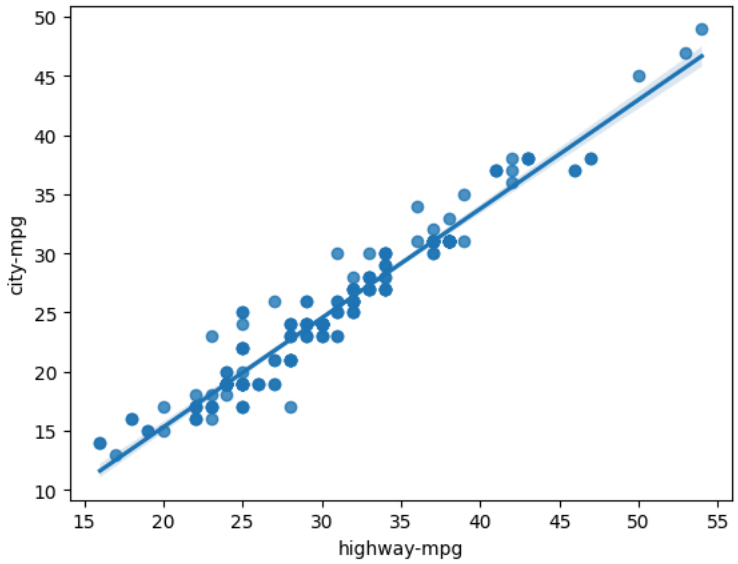
```
<seaborn.axisgrid.PairGrid at 0x79c7b442ec20>
```



Based on our correlation on heatmap, lets choose highway-mpg with city mpg and prize with engine size since they have a good correlation

```
sns.regplot(x = data['highway-mpg'], y = data['city-mpg'])
```

```
<Axes: xlabel='highway-mpg', ylabel='city-mpg'>
```

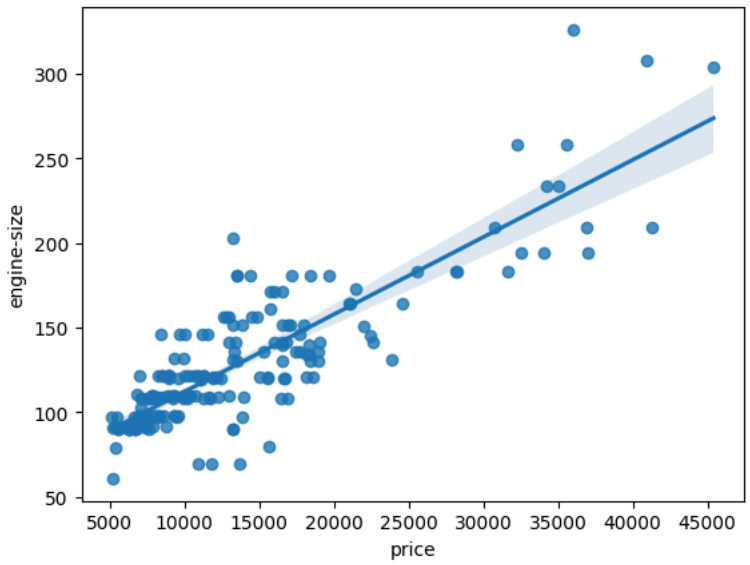



```
data['highway-mpg'].corr(data['city-mpg'])
```

```
0.9713370423425061
```

```
sns.regplot(x = data['price'], y = data['engine-size'])
```

```
<Axes: xlabel='price', ylabel='engine-size'>
```




```
data['price'].corr(data['engine-size'])
```

```
0.8617522436859719
```

The correlation coefficient between city-mpg and highwat-mpg is 0.97

The correlation coefficient between price and engine-size is 0.86

Scale of correlation coefficient	Value
$0 < r \leq 0.19$	Very Low Correlation
$0.2 \leq r \leq 0.39$	Low Correlation
$0.4 \leq r \leq 0.59$	Moderate Correlation
$0.6 \leq r \leq 0.79$	High Correlation

Both of them are above 0.8 which mean that if one of them goes up, the other will also go up in otherwords, they are directly proportional to each other.

We can also try with negative correlation

We can plot high-way mpg and curb-weight to see their correlation