```
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
```

```
!pip install hvplot
```

```
Collecting hvplot
  Downloading hvplot-0.9.2-py2.py3-none-any.whl (1.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 10.0 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) (2.1.5)
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) (3.3.2)
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) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->panel>=0.11.0->hvplot) (2024.2.2)
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
```

LINEAR REGERESSION ANALYSIS (AUTOMOBILE)

import

[ ] ↳ *1 cell hidden*

DATA PRE-PROCESSING + EDA

```
Xlia
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	fuel-system	engine-size	...	length	wheel-base	engine-location	drive-wheels	body-style	num-of-doors	as
0	13495.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	130	...	168.8	88.6	front	rwd	convertible	2.0	
1	16500.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	130	...	168.8	88.6	front	rwd	convertible	2.0	
2	16500.0	26	19	5000.0	154.0	9.0	3.47	2.68	mpfi	152	...	171.2	94.5	front	rwd	hatchback	2.0	
3	13950.0	30	24	5500.0	102.0	10.0	3.40	3.19	mpfi	109	...	176.6	99.8	front	fwd	sedan	4.0	
4	17450.0	22	18	5500.0	115.0	8.0	3.40	3.19	mpfi	136	...	176.6	99.4	front	4wd	sedan	4.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
200	16845.0	28	23	5400.0	114.0	9.5	3.15	3.78	mpfi	141	...	188.8	109.1	front	rwd	sedan	4.0	
201	19045.0	25	19	5300.0	160.0	8.7	3.15	3.78	mpfi	141	...	188.8	109.1	front	rwd	sedan	4.0	
202	21485.0	23	18	5500.0	134.0	8.8	2.87	3.58	mpfi	173	...	188.8	109.1	front	rwd	sedan	4.0	
203	22470.0	27	26	4800.0	106.0	23.0	3.40	3.01	idi	145	...	188.8	109.1	front	rwd	sedan	4.0	
204	22625.0	25	19	5400.0	114.0	9.5	3.15	3.78	mpfi	141	...	188.8	109.1	front	rwd	sedan	4.0	

205 rows × 25 columns

```
ylia
```

	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

```
ae = pd.concat([Xlia,ylia], axis=1)
```

```
ae
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	fuel-system	engine-size	...	wheel-base	engine-location	drive-wheels	body-style	num-of-doors	aspiration
0	13495.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	130	...	88.6	front	rwd	convertible	2.0	std
1	16500.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	130	...	88.6	front	rwd	convertible	2.0	std
2	16500.0	26	19	5000.0	154.0	9.0	3.47	2.68	mpfi	152	...	94.5	front	rwd	hatchback	2.0	std
3	13950.0	30	24	5500.0	102.0	10.0	3.40	3.19	mpfi	109	...	99.8	front	fwd	sedan	4.0	std
4	17450.0	22	18	5500.0	115.0	8.0	3.40	3.19	mpfi	136	...	99.4	front	4wd	sedan	4.0	std
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	16845.0	28	23	5400.0	114.0	9.5	3.15	3.78	mpfi	141	...	109.1	front	rwd	sedan	4.0	std
201	19045.0	25	19	5300.0	160.0	8.7	3.15	3.78	mpfi	141	...	109.1	front	rwd	sedan	4.0	turbo
202	21485.0	23	18	5500.0	134.0	8.8	2.87	3.58	mpfi	173	...	109.1	front	rwd	sedan	4.0	std
203	22470.0	27	26	4800.0	106.0	23.0	3.40	3.01	idi	145	...	109.1	front	rwd	sedan	4.0	turbo
204	22625.0	25	19	5400.0	114.0	9.5	3.15	3.78	mpfi	141	...	109.1	front	rwd	sedan	4.0	turbo

205 rows × 26 columns

```
ae.info()
print('\n',ae.shape)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   price              201 non-null    float64
 1   highway-mpg        205 non-null    int64  
 2   city-mpg           205 non-null    int64  
 3   peak-rpm           203 non-null    float64
 4   horsepower         203 non-null    float64
 5   compression-ratio  205 non-null    float64
 6   stroke             201 non-null    float64
 7   bore               201 non-null    float64
 8   fuel-system        205 non-null    object 
 9   engine-size        205 non-null    int64  
 10  num-of-cylinders   205 non-null    int64  
 11  engine-type        205 non-null    object 
 12  curb-weight        205 non-null    int64  
 13  height             205 non-null    float64
 14  width              205 non-null    float64
 15  length             205 non-null    float64
 16  wheel-base         205 non-null    float64
 17  engine-location    205 non-null    object 
 18  drive-wheels       205 non-null    object 
 19  body-style         205 non-null    object 
 20  num-of-doors       203 non-null    float64
 21  aspiration         205 non-null    object 
 22  fuel-type          205 non-null    object 
 23  make               205 non-null    object 
 24  normalized-losses  164 non-null    float64
 25  symboling          205 non-null    int64  
dtypes: float64(12), int64(6), object(8)
memory usage: 41.8+ KB

 (205, 26)
```

```
ae.set_index('make')
```

make	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	fuel-system	engine-size	...	length	wheel-base	engine-location	drive-wheels	body-style	num-of-doors
alfa-romero	13495.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	130	...	168.8	88.6	front	rwd	convertible	2.0
alfa-romero	16500.0	27	21	5000.0	111.0	9.0	2.68	3.47	mpfi	130	...	168.8	88.6	front	rwd	convertible	2.0
alfa-romero	16500.0	26	19	5000.0	154.0	9.0	3.47	2.68	mpfi	152	...	171.2	94.5	front	rwd	hatchback	2.0
audi	13950.0	30	24	5500.0	102.0	10.0	3.40	3.19	mpfi	109	...	176.6	99.8	front	fwd	sedan	4.0
audi	17450.0	22	18	5500.0	115.0	8.0	3.40	3.19	mpfi	136	...	176.6	99.4	front	4wd	sedan	4.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
volvo	16845.0	28	23	5400.0	114.0	9.5	3.15	3.78	mpfi	141	...	188.8	109.1	front	rwd	sedan	4.0
volvo	19045.0	25	19	5300.0	160.0	8.7	3.15	3.78	mpfi	141	...	188.8	109.1	front	rwd	sedan	4.0
volvo	21485.0	23	18	5500.0	134.0	8.8	2.87	3.58	mpfi	173	...	188.8	109.1	front	rwd	sedan	4.0
volvo	22470.0	27	26	4800.0	106.0	23.0	3.40	3.01	idi	145	...	188.8	109.1	front	rwd	sedan	4.0
volvo	22625.0	25	19	5400.0	114.0	9.5	3.15	3.78	mpfi	141	...	188.8	109.1	front	rwd	sedan	4.0

205 rows × 25 columns

```
ae.describe()
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	engine-size	num-of-cylinders	curb-weight	height
count	201.000000	205.000000	205.000000	203.000000	203.000000	205.000000	201.000000	201.000000	205.000000	205.000000	205.000000	205.000000
mean	13207.129353	30.751220	25.219512	5125.369458	104.256158	10.142537	3.255423	3.329751	126.907317	4.380488	2555.565854	53.724878
std	7947.066342	6.886443	6.542142	479.334560	39.714369	3.972040	0.316717	0.273539	41.642693	1.080854	520.680204	2.443522
min	5118.000000	16.000000	13.000000	4150.000000	48.000000	7.000000	2.070000	2.540000	61.000000	2.000000	1488.000000	47.800000
25%	7775.000000	25.000000	19.000000	4800.000000	70.000000	8.600000	3.110000	3.150000	97.000000	4.000000	2145.000000	52.000000
50%	10295.000000	30.000000	24.000000	5200.000000	95.000000	9.000000	3.290000	3.310000	120.000000	4.000000	2414.000000	54.100000
75%	16500.000000	34.000000	30.000000	5500.000000	116.000000	9.400000	3.410000	3.590000	141.000000	4.000000	2935.000000	55.500000
max	45400.000000	54.000000	49.000000	6600.000000	288.000000	23.000000	4.170000	3.940000	326.000000	12.000000	4066.000000	59.800000

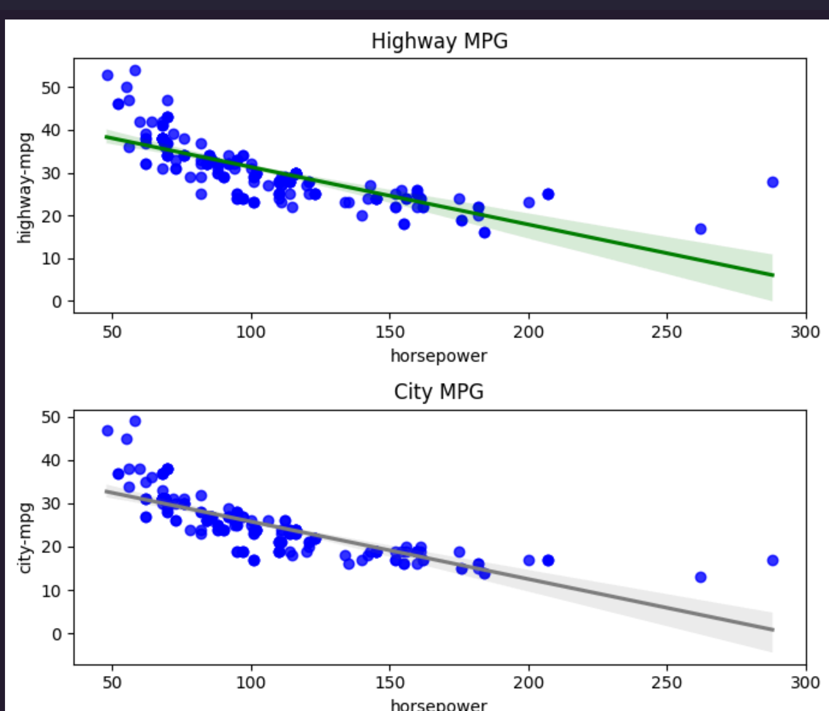
I will be focusing on the possible factors that affect the MPG of a vehicle on roads.

HORSEPOWER

This plot suggests that there is be a ***negative correlation between mpg and horsepower***.

The higher the horsepower is of a car, the less fuel efficient it is thus having lower miles per gallon.

```
fig, axes = plt.subplots(2, 1, figsize=(7, 6))

# highway MPG
sns.regplot(ax = axes[0], x=ae['horsepower'], y=ae['highway-mpg'],scatter_kws={"color": "blue"}, line_kws={"color": "green"})
axes[0].set_title('Highway MPG')

# city MPG
sns.regplot(ax = axes[1], x=ae['horsepower'], y=ae['city-mpg'],scatter_kws={"color": "blue"}, line_kws={"color": "gray"})
axes[1].set_title('City MPG')

plt.tight_layout()

plt.show()
```

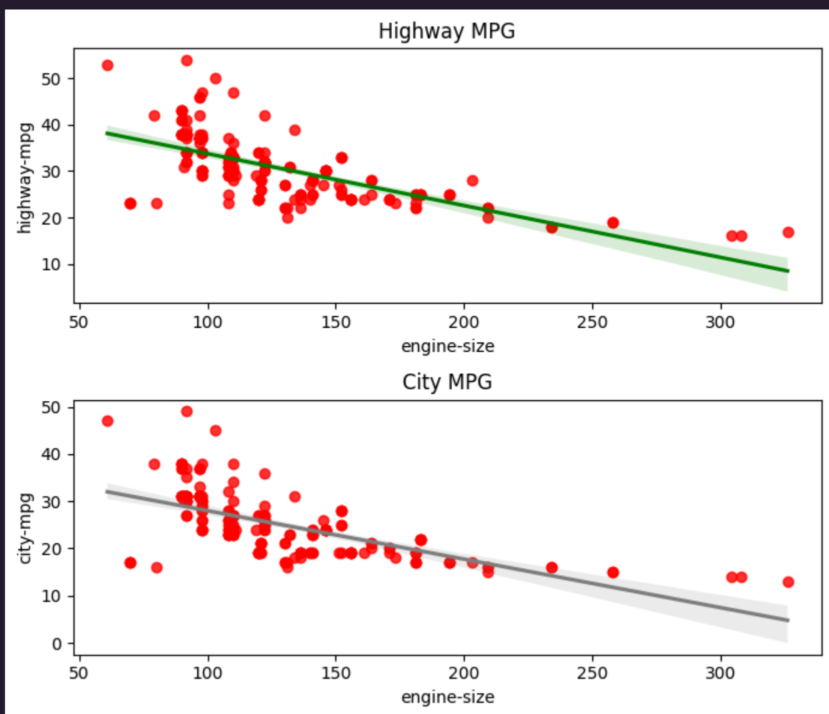


ENGINE SIZE

This plot suggests that there is a ***negative correlation between a car's engine size and mpg***.

Cars that have bigger engine displacement consumes bigger amount of fuel therefore having lesser mpg than small engines.

```
fig, axes = plt.subplots(2, 1, figsize=(7, 6))

# highway MPG
sns.regplot(ax=axes[0], data=ae, x='engine-size', y='highway-mpg',
            scatter_kws={"color": "red"}, line_kws={"color": "green"})
axes[0].set_title('Highway MPG')

# city MPG
sns.regplot(ax=axes[1], data=ae, x='engine-size', y='city-mpg',
            scatter_kws={"color": "red"}, line_kws={"color": "gray"})
axes[1].set_title('City MPG')

plt.tight_layout()

plt.show()
```

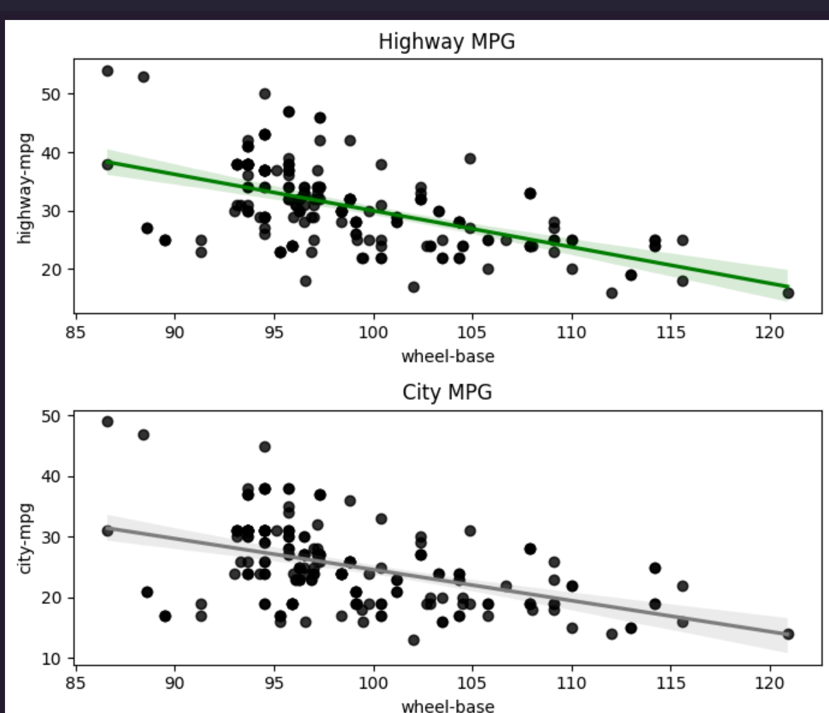


WHEELBASE

This plot suggests that there might be a slight ***negative correlation between a car's wheelbase and mpg***.

The bigger wheels a car has, the less fuel efficient it is thus having lower miles per gallon.

```
fig, axes = plt.subplots(2, 1, figsize=(7, 6))

# highway MPG
sns.regplot(ax=axes[0], data=ae, x='wheel-base', y='highway-mpg',
            scatter_kws={"color": "black"}, line_kws={"color": "green"})
axes[0].set_title('Highway MPG')

# city MPG
sns.regplot(ax=axes[1], data=ae, x='wheel-base', y='city-mpg',
            scatter_kws={"color": "black"}, line_kws={"color": "gray"})
axes[1].set_title('City MPG')

plt.tight_layout()

plt.show()
```



ALL PREVIOUS REGRESSIONS SEEM TO BE NEGATIVE LET'S HAVE A POSITIVE REGRESSION :D

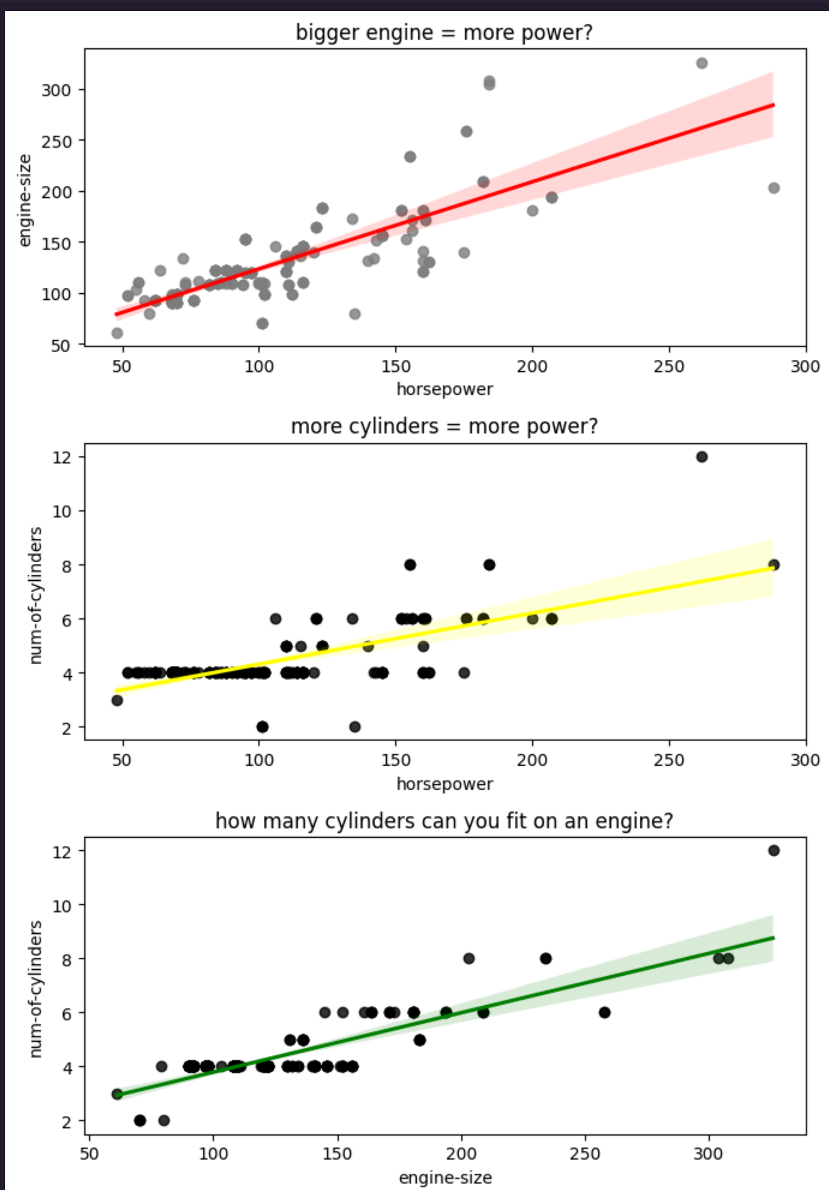
engine size to power

the bigger engine a car has the more horsepower it can get.

number of cylinders to power

the more the cylinders of an engine the greater power it can give to the car.

```
fig, axes = plt.subplots(3, 1, figsize=(7, 10))

sns.regplot(ax=axes[0], data=ae, x='horsepower', y='engine-size',
            scatter_kws={"color": "gray"}, line_kws={"color": "red"})
axes[0].set_title('bigger engine = more power?')

sns.regplot(ax=axes[1], data=ae, x='horsepower', y='num-of-cylinders',
            scatter_kws={"color": "black"}, line_kws={"color": "yellow"})
axes[1].set_title('more cylinders = more power?')

sns.regplot(ax=axes[2], data=ae, x='engine-size', y='num-of-cylinders',
            scatter_kws={"color": "black"}, line_kws={"color": "green"})
axes[2].set_title('how many cylinders can you fit on an engine?')

plt.tight_layout()

plt.show()
```



how many cylinders can fit in a engine

assuming the engine size is in cubic inches, the maximum size is 326, which is around 5.3 liters, and we can see in the plot that there one car that has 12 cylinders in a 5.3 liter engine.

in the dataset, the maximum possible cylinders inside a 5.3 liter engine is 12.

```
aefc = ae.drop(['fuel-system', 'engine-type', 'engine-location', 'drive-wheels', 'body-style', 'aspiration', 'fuel-type', 'make'],axis=1)
#for correlation
```

```
aefc
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	engine-size	num-of-cylinders	curb-weight	height	width	length	wheel-base	num-of-doors	normalized-losses
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	2.0	Na
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	2.0	Na
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	2.0	Na
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	4.0	164
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	4.0	164
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
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	4.0	95
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	4.0	95
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	4.0	95
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	4.0	95
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	4.0	95

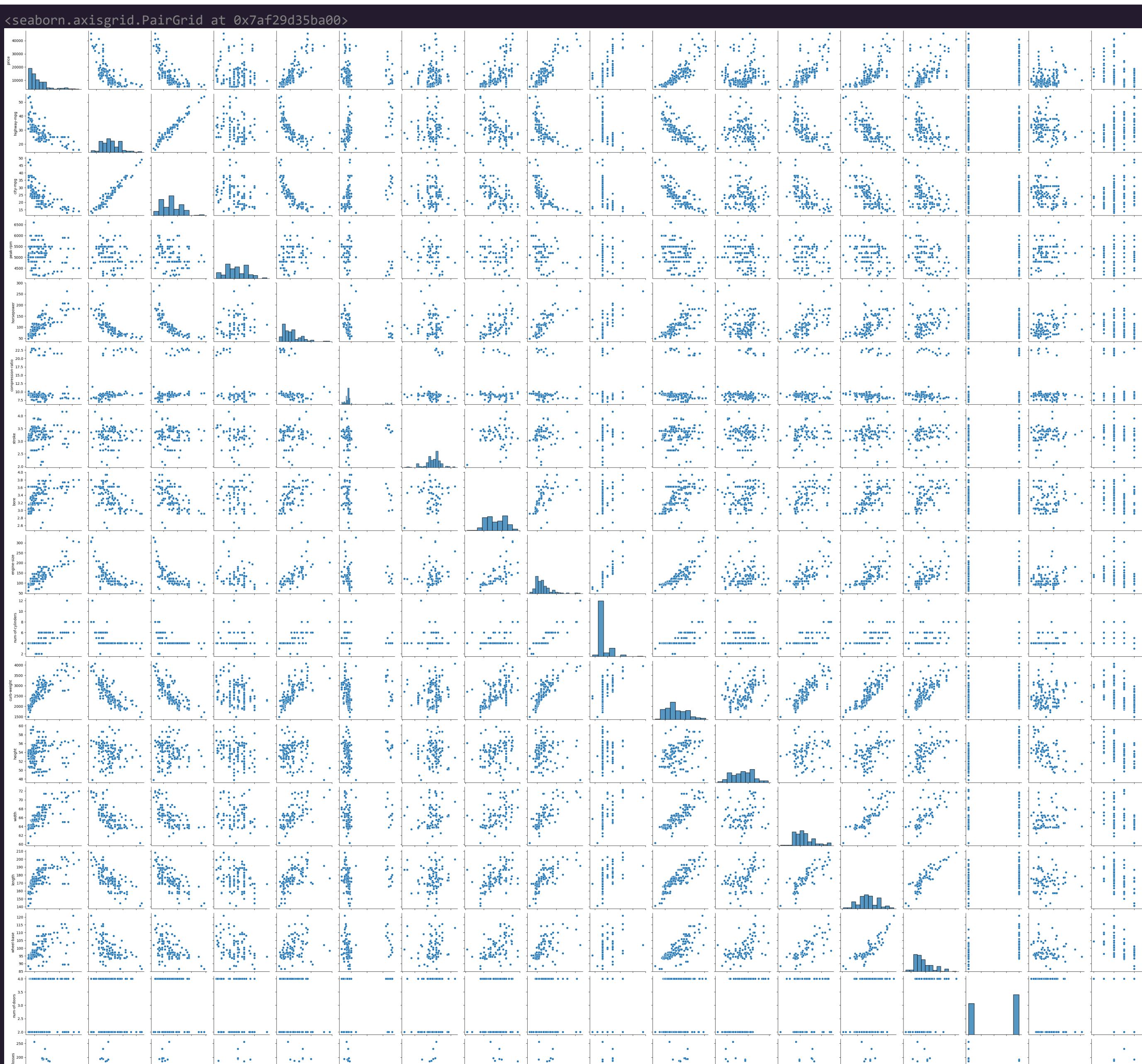
205 rows × 18 columns

```
1 aefc.corr()
```

	price	highway-mpg	city-mpg	peak-rpm	horsepower	compression-ratio	stroke	bore	engine-size	num-of-cylinders	curb-weight	height	width
price	1.000000	-0.704692	-0.686571	-0.101649	0.810533	0.071107	0.082310	0.543436	0.872335	0.708645	0.834415	0.135486	0.751265
highway-mpg	-0.704692	1.000000	0.971337	-0.054257	-0.770908	0.265201	-0.044528	-0.594572	-0.677470	-0.466666	-0.797465	-0.107358	-0.677218
city-mpg	-0.686571	0.971337	1.000000	-0.113788	-0.803620	0.324701	-0.042906	-0.594584	-0.653658	-0.445837	-0.757414	-0.048640	-0.642704
peak-rpm	-0.101649	-0.054257	-0.113788	1.000000	0.130971	-0.436221	-0.071493	-0.264269	-0.244618	-0.124434	-0.266306	-0.322272	-0.219957
horsepower	0.810533	-0.770908	-0.803620	0.130971	1.000000	-0.205874	0.090254	0.577273	0.810773	0.691633	0.751034	-0.110711	0.642482
compression-ratio	0.071107	0.265201	0.324701	-0.436221	-0.205874	1.000000	0.186170	0.005203	0.028971	-0.020002	0.151362	0.261214	0.181129
stroke	0.082310	-0.044528	-0.042906	-0.071493	0.090254	0.186170	1.000000	-0.055909	0.206675	0.008578	0.168929	-0.056999	0.182956
bore	0.543436	-0.594572	-0.594584	-0.264269	0.577273	0.005203	-0.055909	1.000000	0.594090	0.243553	0.649045	0.176195	0.559204
engine-size	0.872335	-0.677470	-0.653658	-0.244618	0.810773	0.028971	0.206675	0.594090	1.000000	0.846031	0.850594	0.067149	0.735433
num-of-cylinders	0.708645	-0.466666	-0.445837	-0.124434	0.691633	-0.020002	0.008578	0.243553	0.846031	1.000000	0.609727	-0.013995	0.545007
curb-weight	0.834415	-0.797465	-0.757414	-0.266306	0.751034	0.151362	0.168929	0.649045	0.850594	0.609727	1.000000	0.295572	0.867032
height	0.135486	-0.107358	-0.048640	-0.322272	-0.110711	0.261214	-0.056999	0.176195	0.067149	-0.013995	0.295572	1.000000	0.279210
width	0.751265	-0.677218	-0.642704	-0.219957	0.642482	0.181129	0.182956	0.559204	0.735433	0.545007	0.867032	0.279210	1.000000
length	0.690628	-0.704662	-0.670909	-0.287325	0.555003	0.158414	0.129739	0.607480	0.683360	0.430672	0.877728	0.491029	0.841118
wheel-base	0.584642	-0.544082	-0.470414	-0.361052	0.352297	0.249786	0.161477	0.490378	0.569329	0.339507	0.776386	0.589435	0.795144
num-of-doors	0.046532	-0.044507	-0.020812	-0.242485	-0.124963	0.165799	-0.010697	0.114501	0.017519	-0.016530	0.195683	0.547651	0.202072
normalized-losses	0.203254	-0.210768	-0.258502	0.264597	0.295772	-0.132654	0.065627	-0.036167	0.167365	0.175380	0.119893	-0.432335	0.105073
symboling	-0.082391	0.034606	-0.035823	0.274573	0.071622	-0.178515	-0.008965	-0.134205	-0.105790	-0.113129	-0.227691	-0.541038	-0.232919

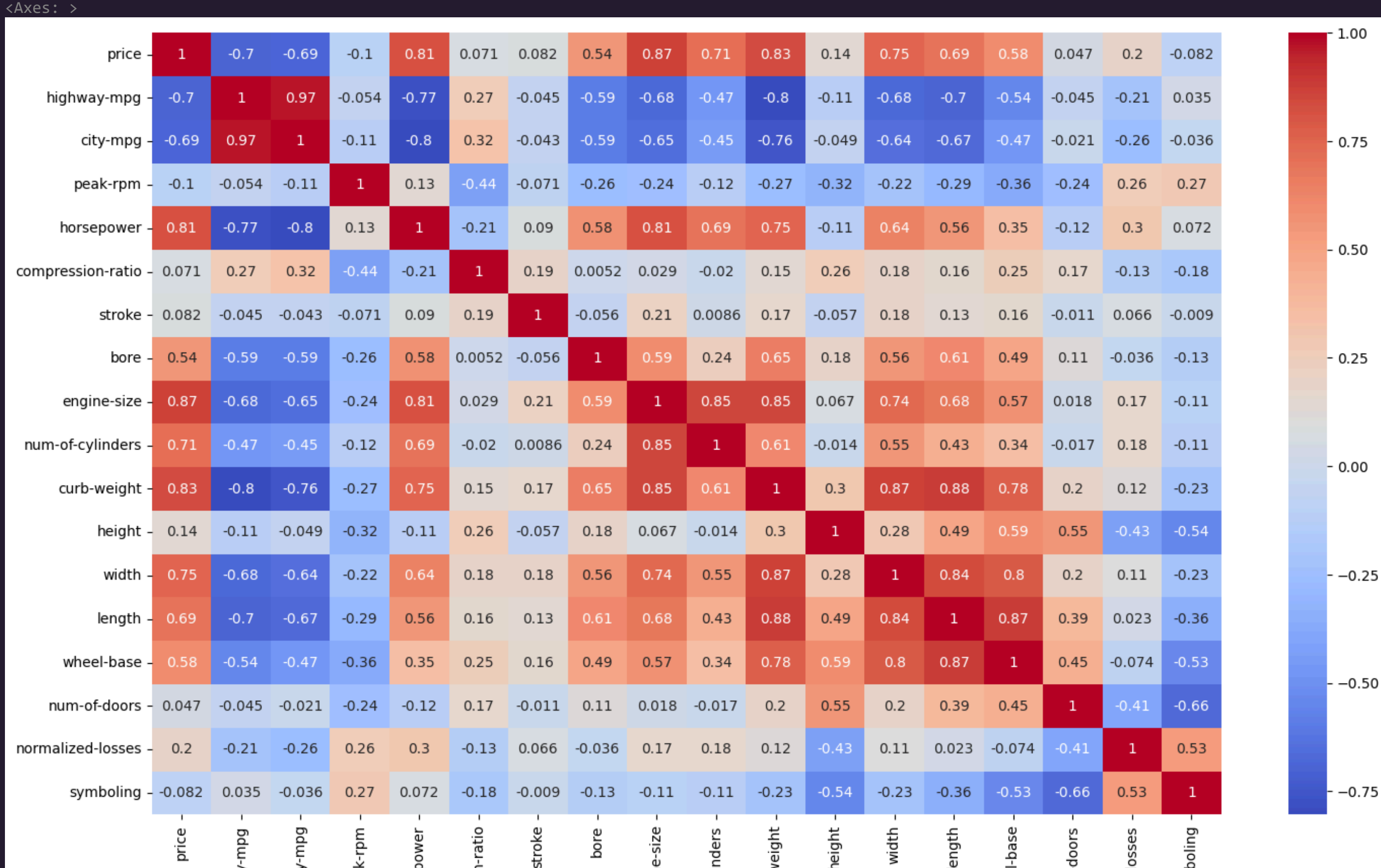
```
1 sns.pairplot(ae)
```

<seaborn.axisgrid.PairGrid at 0x7af29d35ba00>



```
1 plt.figure(figsize=(17, 10))
2 aehm = sns.heatmap(aefc.corr(), annot=True, cmap='coolwarm')
3 aehm
```

<Axes: >



LOGISTIC REGERESSION ANALYSIS (WINE)

import

[ ] ↳ 1 cell hidden

DATA PRE-PROCESSING + EDA

```
1 Xlow
```

	Alcohol	Malicacid	Ash	Alcalinity_of_ash	Magnesium	Total_phenols	Flavanoids	Nonflavanoid_phenols	Proanthocyanins	Color_intensity	Hue	0D28
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	
...	...	...	...	...	...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.06	7.70	0.64	
174	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	1.41	7.30	0.70	
175	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	1.35	10.20	0.59	
176	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	1.46	9.30	0.60	
177	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	1.35	9.20	0.61	

178 rows × 13 columns

```
1 ylow
```

	class
0	1
1	1
2	1
3	1
4	1
...	...
173	3
174	3
175	3
176	3
177	3

178 rows × 1 columns

```
1 we = pd.concat([Xlow,ylow], axis=1)
```

```
1 we
```

	Alcohol	Malicacid	Ash	Alcalinity_of_ash	Magnesium	Total_phenols	Flavanoids	Nonflavanoid_phenols	Proanthocyanins	Color_intensity	Hue	0D28
0	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	
1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	
2	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	
3	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	
4	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	
...	...	...	...	...	...	...	...	...	...	...	...	
173	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.06	7.70	0.64	
174	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	1.41	7.30	0.70	
175	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	1.35	10.20	0.59	
176	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	1.46	9.30	0.60	
177	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	1.35	9.20	0.61	

178 rows × 14 columns

```
1 we.info()
2 print('\n',we.shape)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 178 entries, 0 to 177
Data columns (total 14 columns):
 #   Column                        Non-Null Count  Dtype  
---  ------                        --------------  -----  
 0   Alcohol                       178 non-null    float64
 1   Malicacid                     178 non-null    float64
 2   Ash                           178 non-null    float64
 3   Alcalinity_of_ash             178 non-null    float64
 4   Magnesium                     178 non-null    int64  
 5   Total_phenols                 178 non-null    float64
 6   Flavanoids                    178 non-null    float64
 7   Nonflavanoid_phenols          178 non-null    float64
 8   Proanthocyanins               178 non-null    float64
 9   Color_intensity               178 non-null    float64
 10  Hue                           178 non-null    float64
 11  0D280_0D315_of_diluted_wines  178 non-null    float64
 12  Proline                       178 non-null    int64  
 13  class                         178 non-null    int64  
dtypes: float64(11), int64(3)
```

```
memory usage: 19.6 KB

 (178, 14)
```

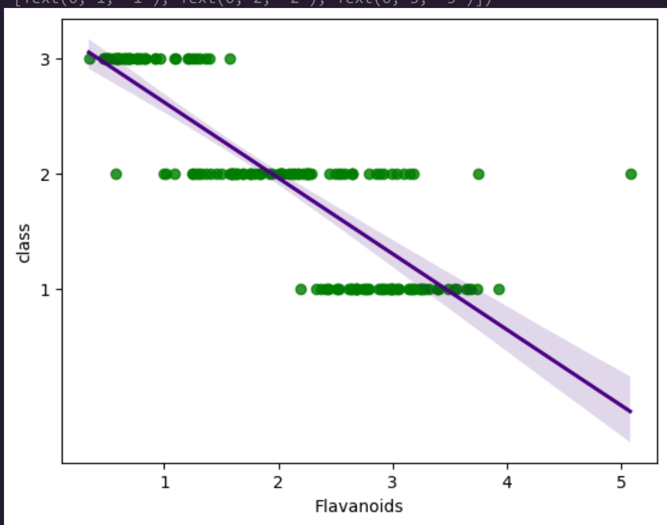
Class and Flavanoid

Wines that are higher class tend to have higher flavanoid content, possibly from the grapes used to make the wine grow to conditions that leads to more flavanoids.

Wines that are lower class have lower flavanoid content, probably due to the processing techniques to produce the wine.

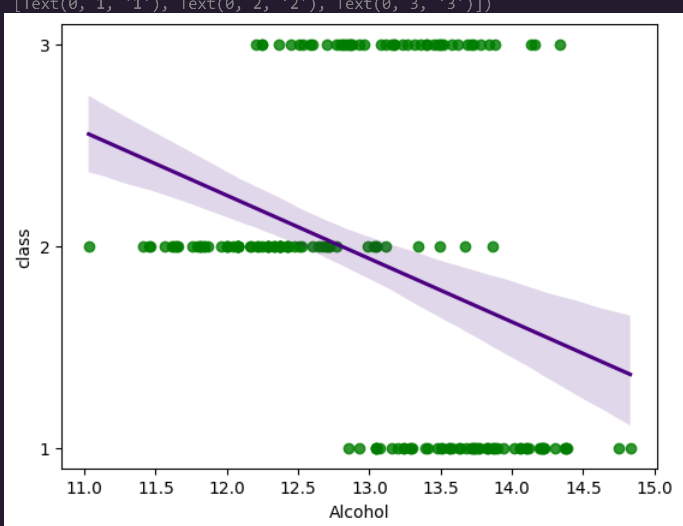
```
sns.regplot(x=we['Flavanoids'], y=we['class'],scatter_kws={"color": "green"}, line_kws={"color": "indigo"})
plt.yticks(we['class'].unique())
```

```
([<matplotlib.axis.YTick at 0x7af2837f9d50>,
  <matplotlib.axis.YTick at 0x7af27f9d5210>,
  <matplotlib.axis.YTick at 0x7af27f0e2980>],
 [Text(0, 1, '1'), Text(0, 2, '2'), Text(0, 3, '3')])
```




```
sns.regplot(x=we['Alcohol'], y=we['class'],scatter_kws={"color": "green"}, line_kws={"color": "indigo"})
plt.yticks(we['class'].unique())
```

```
([<matplotlib.axis.YTick at 0x7af283272680>,
  <matplotlib.axis.YTick at 0x7af2837e28c0>,
  <matplotlib.axis.YTick at 0x7af283579c30>],
 [Text(0, 1, '1'), Text(0, 2, '2'), Text(0, 3, '3')])
```




```
we.describe()
```

	Alcohol	Malicacid	Ash	Alcalinity_of_ash	Magnesium	Total_phenols	Flavanoids	Nonflavanoid_phenols	Proanthocyanins	Color_intensit
count	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.000000	178.00000
mean	13.000618	2.336348	2.366517	19.494944	99.741573	2.295112	2.029270	0.361854	1.590899	5.05809
std	0.811827	1.117146	0.274344	3.339564	14.282484	0.625851	0.998859	0.124453	0.572359	2.31828
min	11.030000	0.740000	1.360000	10.600000	70.000000	0.980000	0.340000	0.130000	0.410000	1.28000
25%	12.362500	1.602500	2.210000	17.200000	88.000000	1.742500	1.205000	0.270000	1.250000	3.22000
50%	13.050000	1.865000	2.360000	19.500000	98.000000	2.355000	2.135000	0.340000	1.555000	4.69000
75%	13.677500	3.082500	2.557500	21.500000	107.000000	2.800000	2.875000	0.437500	1.950000	6.20000
max	14.830000	5.800000	3.230000	30.000000	162.000000	3.880000	5.080000	0.660000	3.580000	13.00000

```
wefhm = we.corr()
wefhm
```

	Alcohol	Malicacid	Ash	Alcalinity_of_ash	Magnesium	Total_phenols	Flavanoids	Nonflavanoid_phenols	Proanthocyani
Alcohol	1.000000	0.094397	0.211545	-0.310235	0.270798	0.289101	0.236815	-0.155929	0.1366
Malicacid	0.094397	1.000000	0.164045	0.288500	-0.054575	-0.335167	-0.411007	0.292977	-0.2207
Ash	0.211545	0.164045	1.000000	0.443367	0.286587	0.128980	0.115077	0.186230	0.0096
Alcalinity_of_ash	-0.310235	0.288500	0.443367	1.000000	-0.083333	-0.321113	-0.351370	0.361922	-0.1973
Magnesium	0.270798	-0.054575	0.286587	-0.083333	1.000000	0.214401	0.195784	-0.256294	0.2364
Total_phenols	0.289101	-0.335167	0.128980	-0.321113	0.214401	1.000000	0.864564	-0.449935	0.6124

```
1 sns.pairplot(we)
```

<seaborn.axisgrid.PairGrid at 0x7af286d198a0>