Perform predictive modeling on mtcars.csv dataset from which select few columns as an input and output as mpg(mileage per gallon)

So on the basis of inputs selected we have to predict its mileage

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

In [2]:
```
df = pd.read_csv('mtcars.csv')
df.head()
```

Out[2]:

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	21.0	6.0	160.0	110.0	3.90	2.620	16.46	0.0	1.0	4.0	4.0
1	21.0	6.0	160.0	110.0	3.90	2.875	17.02	0.0	1.0	4.0	4.0
2	22.8	4.0	108.0	93.0	3.85	2.320	18.61	1.0	1.0	4.0	1.0
3	21.4	6.0	258.0	110.0	3.08	3.215	19.44	1.0	0.0	3.0	1.0
4	18.7	8.0	360.0	175.0	3.15	3.440	17.02	0.0	0.0	3.0	2.0

In [3]:
```
df.info()
```

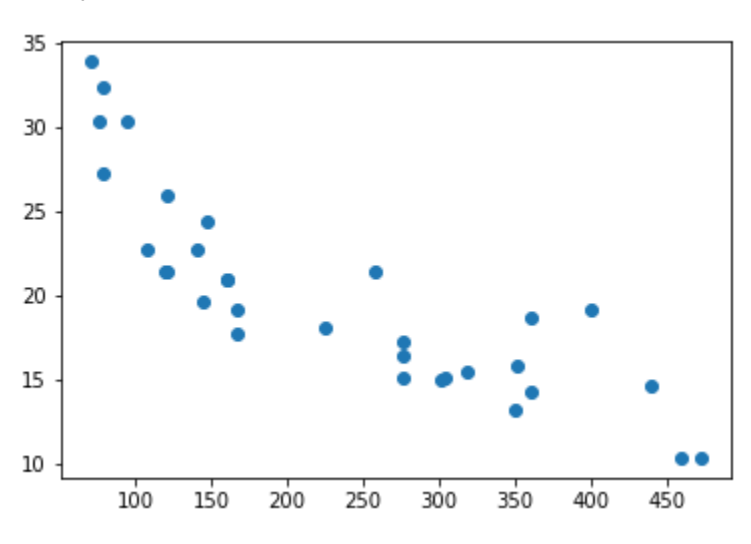
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32 entries, 0 to 31
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   mpg     32 non-null     float64
 1   cyl     32 non-null     float64
 2   disp    32 non-null     float64
 3   hp      32 non-null     float64
 4   drat    32 non-null     float64
 5   wt      32 non-null     float64
 6   qsec    32 non-null     float64
 7   vs      32 non-null     float64
 8   am      32 non-null     float64
 9   gear    32 non-null     float64
 10  carb    32 non-null     float64
dtypes: float64(11)
memory usage: 2.9 KB
```

In [4]:
```
df.isna().sum()
```

Out[4]:
```
mpg     0
cyl     0
disp    0
hp      0
drat    0
wt      0
qsec    0
vs      0
am      0
gear    0
carb    0
dtype: int64
```

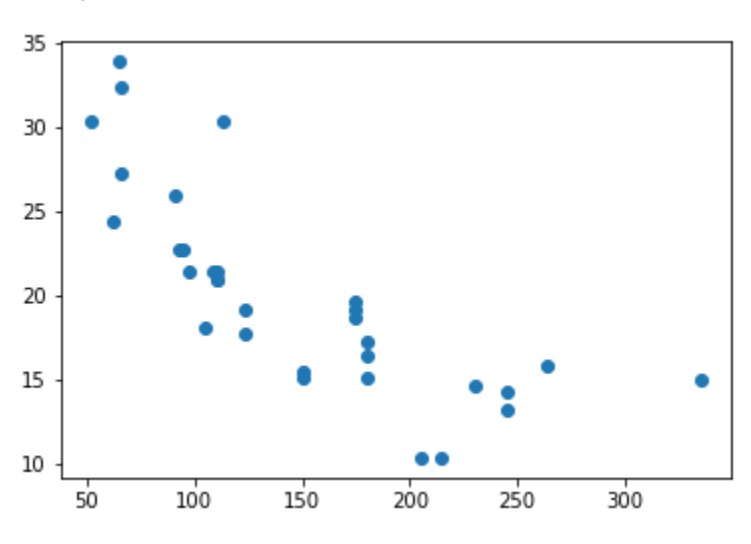
In [5]:
```
plt.scatter(df.disp,df.mpg)
```

Out[5]: <matplotlib.collections.PathCollection at 0x15f8c05e9d0>



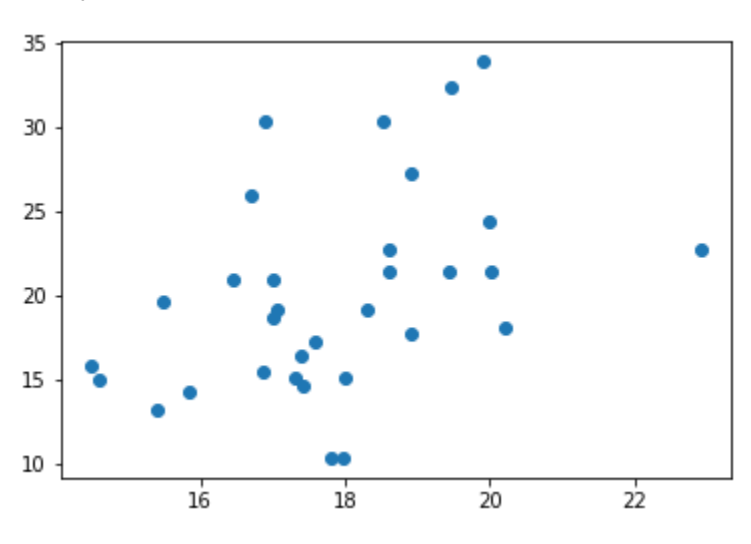
In [6]:
```
plt.scatter(df.hp,df.mpg)
```

Out[6]: <matplotlib.collections.PathCollection at 0x15f8d129310>



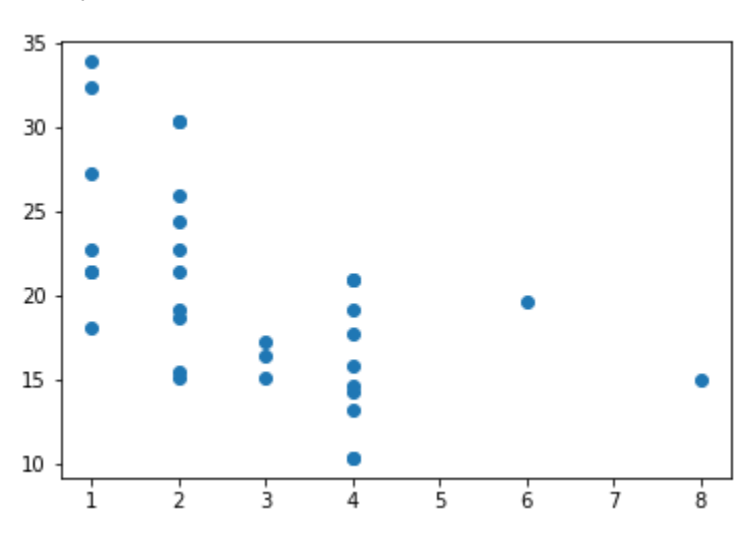
In [7]:
```
plt.scatter(df.qsec,df.mpg)
```

Out[7]: <matplotlib.collections.PathCollection at 0x15f8d189250>



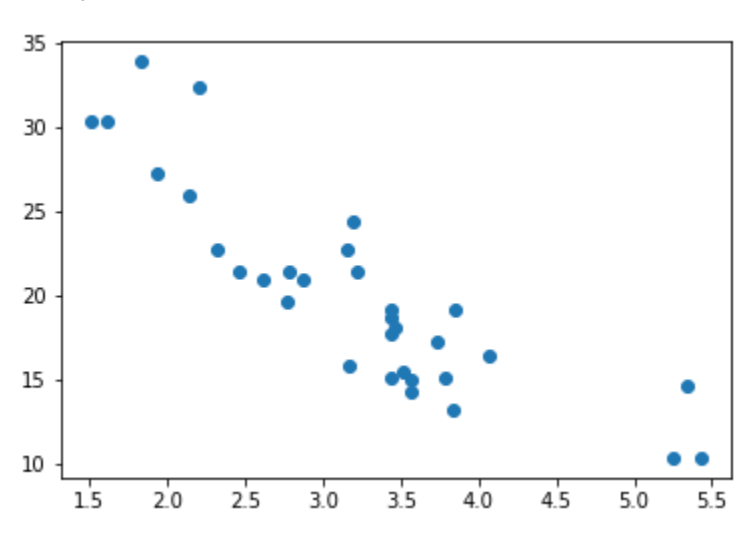
In [8]:
```
plt.scatter(df.carb,df.mpg)
```

Out[8]: <matplotlib.collections.PathCollection at 0x15f8d1b7fa0>



In [9]:
```
plt.scatter(df.wt,df.mpg)
```

Out[9]: <matplotlib.collections.PathCollection at 0x15f8d22eb20>



In [10]:
```
x = df[['disp','hp','wt']]
x[:3]
```

Out[10]:

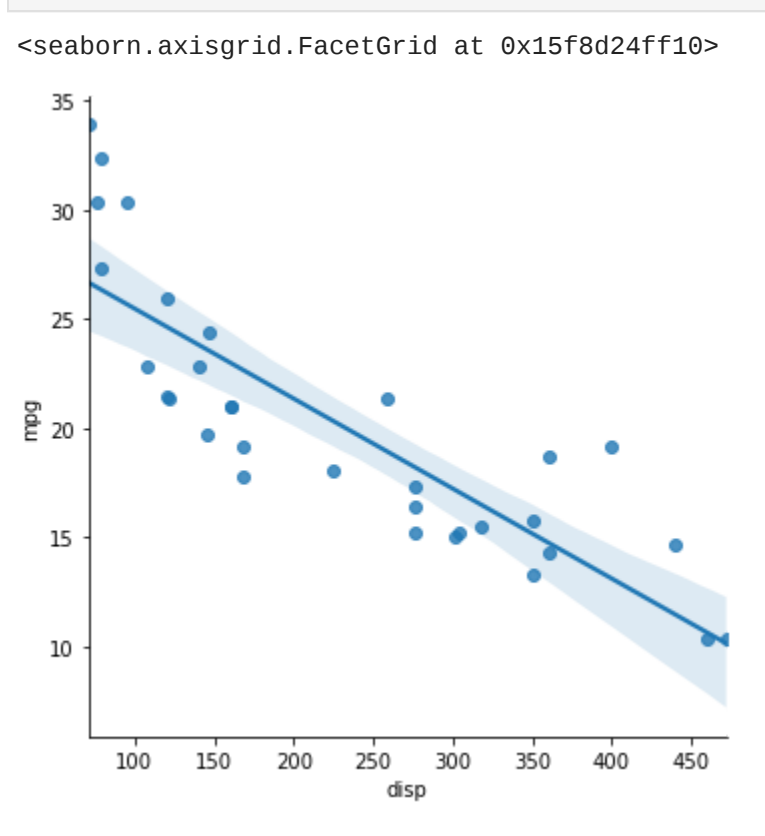
	disp	hp	wt
0	160.0	110.0	2.620
1	160.0	110.0	2.875
2	108.0	93.0	2.320

In [11]:
```
pip install seaborn
```

```
Note: you may need to restart the kernel to use updated packages.
'C:\Users\Nilesh' is not recognized as an internal or external command,
operable program or batch file.
```

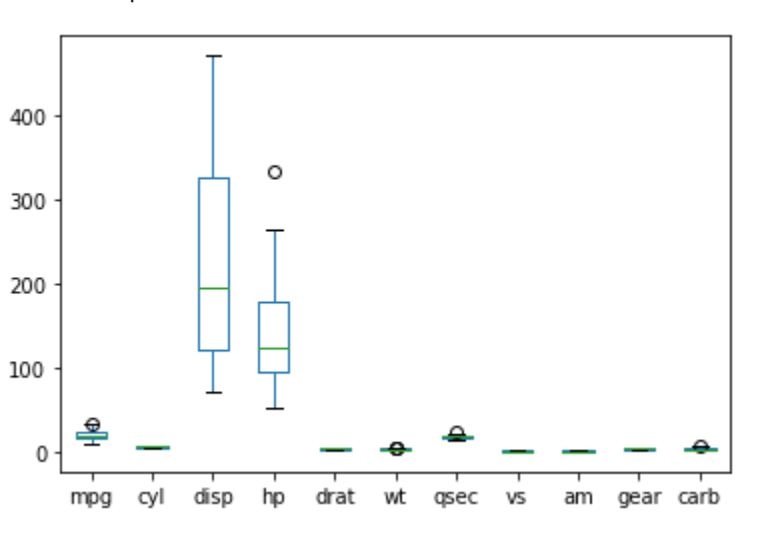
In [14]:
```
sns.lmplot(x = 'disp',y = 'mpg',data=df)
```

Out[14]: <seaborn.axisgrid.FacetGrid at 0x15f8d24ff10>



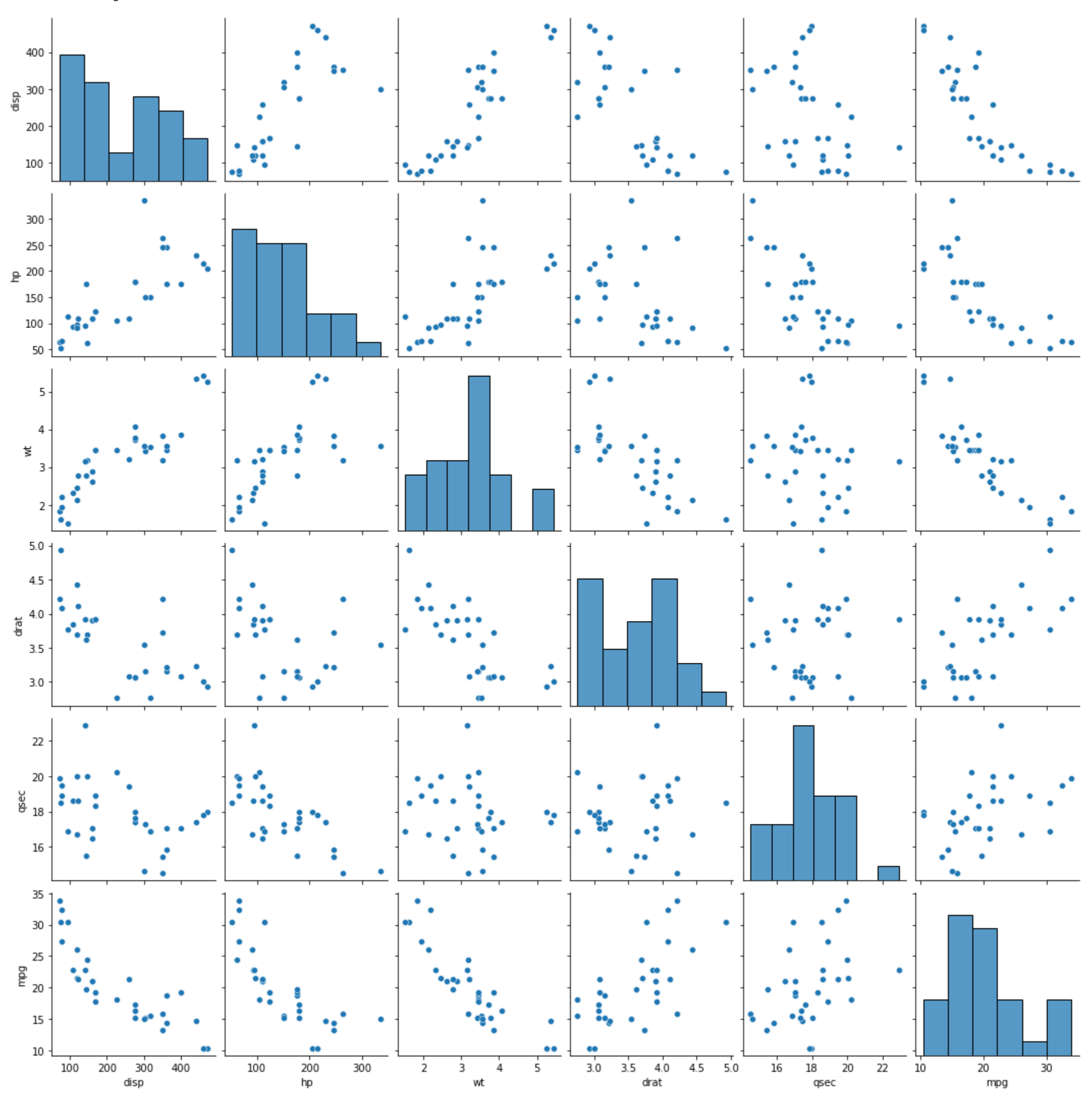
In [15]:
```
df.plot(kind = 'box')
```

Out[15]: <AxesSubplot:>



In [16]:
```
sns.pairplot(df[['disp','hp','wt','drat','qsec','mpg']])
```

Out[16]: <seaborn.axisgrid.PairGrid at 0x15f8e53c640>



In [18]:
```
y = df.mpg
y[:3]
```

Out[18]:
```
0    21.0
1    21.0
2    22.8
Name: mpg, dtype: float64
```

In [19]:
```
x[:3]
```

Out[19]:

	disp	hp	wt
0	160.0	110.0	2.620
1	160.0	110.0	2.875
2	108.0	93.0	2.320

In [20]:
```
# create linear class model
from sklearn.linear_model import LinearRegression
```

In [21]:
```
model = LinearRegression()
model
```

Out[21]: LinearRegression()

In [22]:
```
model.fit(x,y)
```

Out[22]: LinearRegression()

In [23]:
```
# slope
model.coef_
```

Out[23]: array([-9.37009081e-04, -3.11565508e-02, -3.80089058e+00])

In [24]:
```
x.columns
```

Out[24]: Index(['disp', 'hp', 'wt'], dtype='object')

In [25]:
```
#intercept
model.intercept_
```

Out[25]: 37.10550526903182

In [26]:
```
# prediction
x[:3]
```

Out[26]:

	disp	hp	wt
0	160.0	110.0	2.620
1	160.0	110.0	2.875
2	108.0	93.0	2.320

In [27]:
```
y[:3]
```

Out[27]:
```
0    21.0
1    21.0
2    22.8
Name: mpg, dtype: float64
```

In [29]:
```
test = x[:3]
test
```

Out[29]:

	disp	hp	wt
0	160.0	110.0	2.620
1	160.0	110.0	2.875
2	108.0	93.0	2.320

In [30]:
```
model.predict(test)
```

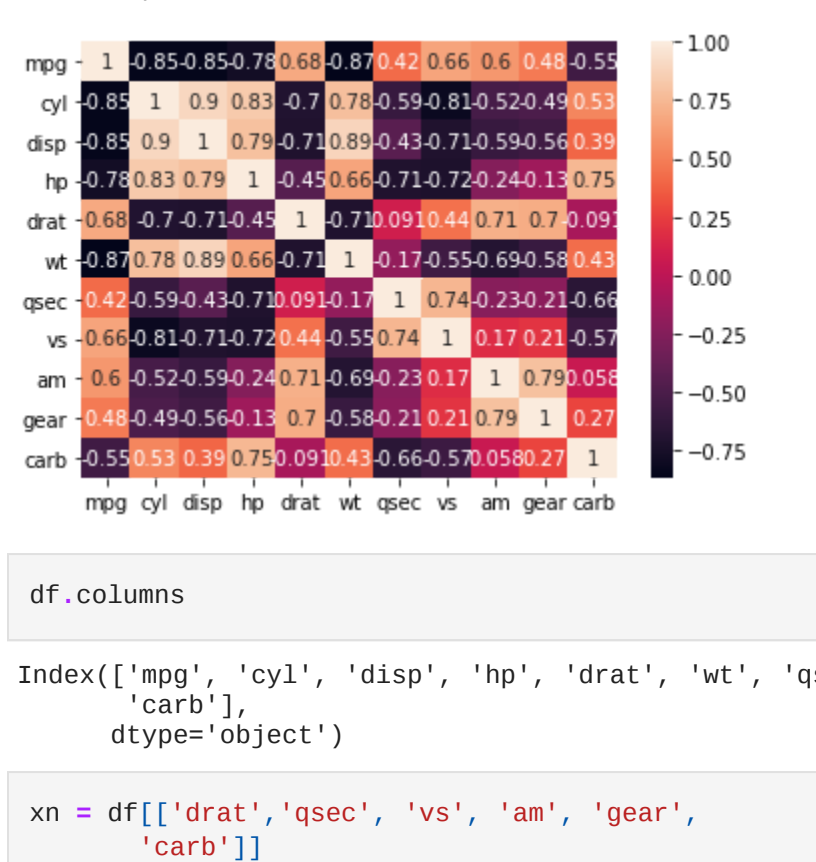
Out[30]: array([23.5700299 , 22.6008028 , 25.28868291])

In [31]:
```
# check score
model.score(x,y)*100
```

Out[31]: 82.68361424946447

In [33]:
```
sns.heatmap(df.corr(),annot=True)
```

Out[33]: <AxesSubplot:>



In [34]:
```
df.columns
```

Out[34]:
```
Index(['mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am', 'gear',
       'carb'],
      dtype='object')
```

In [36]:
```
xn = df[['drat','qsec', 'vs', 'am', 'gear',
        'carb']]
xn[:3]
```

Out[36]:

	drat	qsec	vs	am	gear	carb
0	3.90	16.46	0.0	1.0	4.0	4.0
1	3.90	17.02	0.0	1.0	4.0	4.0
2	3.85	18.61	1.0	1.0	4.0	1.0

In [37]:
```
model.fit(xn,y)
```

Out[37]: LinearRegression()

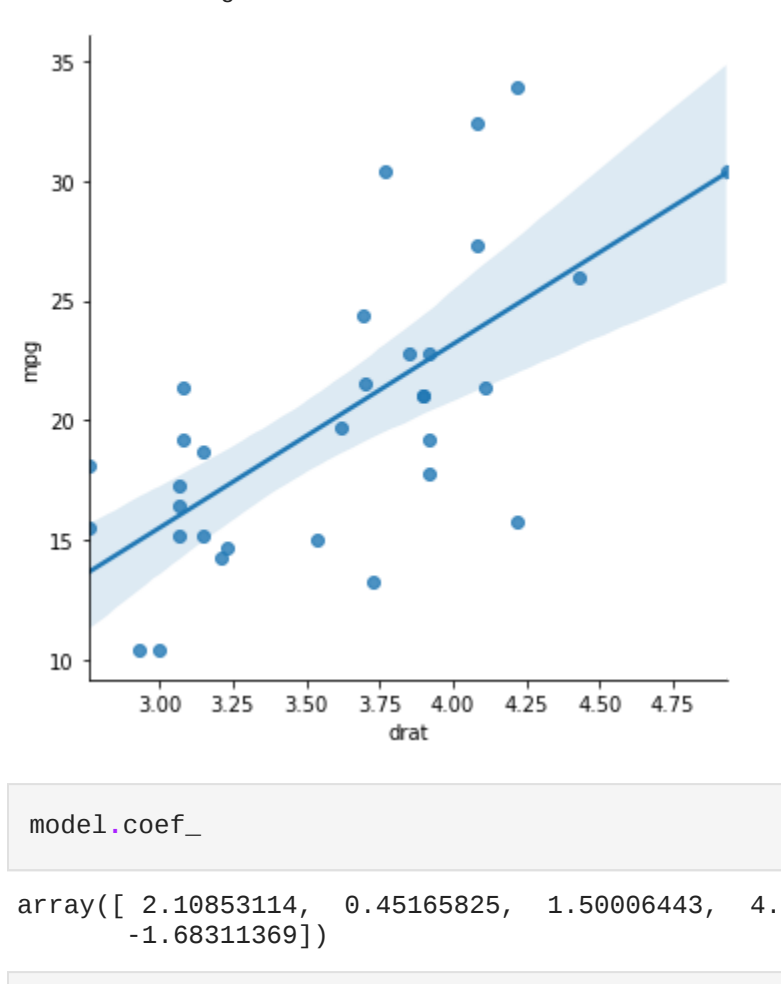
In [38]:
```
model.score(xn,y)*100
```

Out[38]: 81.73146698607448

In [39]:
```
sns.lmplot('drat','mpg',data=df)
```

```
C:\Users\Nilesh koli\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid pos
itional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[39]: <seaborn.axisgrid.FacetGrid at 0x15f92ff45b0>



In [40]:
```
model.coef_
```

Out[40]:
```
array([ 2.10853114,  0.45165825,  1.50006443,  4.06713222,  1.70150963,
       -1.68311369])
```

In [41]:
```
model.predict(xn[:3])
```

Out[41]: array([20.39479785, 20.64772646, 27.80984202])

In [ ]: