In [1]:
```
#dodda.prudhvi sai
#assignment-04
#simple linear regression
```

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

In [2]:
```
dataset =pd.read_csv('student_scores.csv')
```

In [3]:
```
dataset
```

Out[3]:

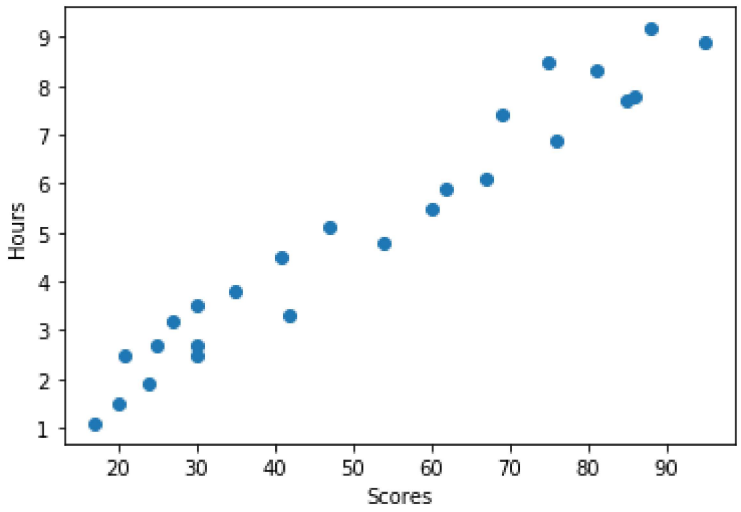
	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [4]:
```
dataset.isnull().any()
```

Out[4]:
```
Hours     False
Scores    False
dtype: bool
```

In [7]:
```
plt.scatter(dataset['Scores'],dataset['Hours'])
plt.xlabel('Scores')
plt.ylabel('Hours')
```

Out[7]:
```
Text(0, 0.5, 'Hours')
```



In [8]:
```
#independent
x=dataset.iloc[:,0:1]
y=dataset.iloc[:,1:]
```

In [10]:
```

x=dataset.iloc[:,0:1].values
y=dataset.iloc[:,1:].values
```

In [11]:
```
x.shape
```

Out[11]:
```
(25, 1)
```

In [12]:
```
y.shape
```

Out[12]:
```
(25, 1)
```

In [13]:
```
from sklearn.model_selection import train_test_split
```

In [14]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

In [15]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
```

In [16]:
```
lr.fit(x_train,y_train)
```

Out[16]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [17]:
```
y_pred=lr.predict(x_test)
```

In [18]:
```
y_pred
```

Out[18]:
```
array([[16.88414476],
       [33.73226078],
       [75.357018  ],
       [26.79480124],
       [60.49103328]])
```

In [19]:
```
from sklearn.metrics import r2_score
accuracy=r2_score(y_test,y_pred)
```

In [20]:
```
accuracy
```

Out[20]:
```
0.9454906892105356
```

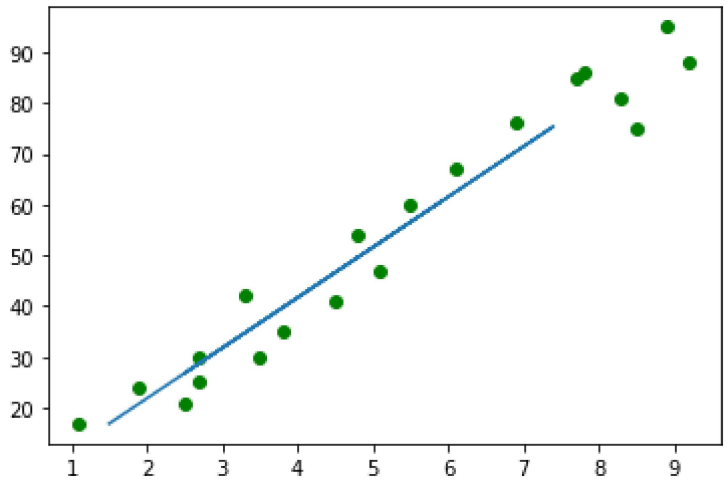
In [21]:
```
y_pred_Scores=lr.predict([[15]])
```

In [22]:
```
y_pred_Scores
```

Out[22]:
```
array([[150.67800725]])
```

In [23]:
```
plt.scatter(x_train,y_train,color='green')
plt.plot(x_test,lr.predict(x_test))
```

Out[23]:
```
[<matplotlib.lines.Line2D at 0x1a263659f48>]
```



In [24]:
```
lr.intercept_#intercept
```

Out[24]:
```
array([2.01816004])
```

In [25]:
```
lr.coef_#slope
```

Out[25]:
```
array([[9.91065648]])
```

In [2]:
```
#multi linear regression
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [3]:
```
dataset=pd.read_csv('petrol_consumption.csv')
```

In [4]:
```
dataset
```

20	7.00	4593	7834	0.663
21	8.00	4983	602	0.602
22	9.00	4897	2449	0.511
23	9.00	4258	4686	0.517
24	8.50	4574	2619	0.551
25	9.00	3721	4746	0.544
26	8.00	3448	5399	0.548
27	7.50	3846	9061	0.579
28	8.00	4188	5975	0.563
29	9.00	3601	4650	0.493
30	7.00	3640	6905	0.518

In [5]:
```
type(dataset)
```

Out[5]:
```
pandas.core.frame.DataFrame
```

In [7]:
```
dataset.isnull().any()
```

Out[7]:
```
Petrol_tax                      False
Average_income                  False
Paved_Highways                  False
Population_Driver_licence(%)    False
Petrol_Consumption              False
dtype: bool
```

In [8]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48 entries, 0 to 47
Data columns (total 5 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Petrol_tax                      48 non-null     float64
 1   Average_income                  48 non-null     int64  
 2   Paved_Highways                  48 non-null     int64  
 3   Population_Driver_licence(%)    48 non-null     float64
 4   Petrol_Consumption              48 non-null     int64  
dtypes: float64(2), int64(3)
memory usage: 2.0 KB
```

In [10]:
```
dataset.head()
```

Out[10]:

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licence(%)	Petrol_Consumption
0	9.0	3571	1976	0.525	541
1	9.0	4092	1250	0.572	524
2	9.0	3865	1586	0.580	561
3	7.5	4870	2351	0.529	414
4	8.0	4399	431	0.544	410

In [12]:
```
x=dataset.drop(['Petrol_Consumption'],axis=1).values
y=dataset['Petrol_Consumption'].values
```

In [13]:
```
y
```

Out[13]:
```
array([541, 524, 561, 414, 410, 457, 344, 467, 464, 498, 580, 471, 525,
       508, 566, 635, 603, 714, 865, 640, 649, 540, 464, 547, 460, 566,
       577, 631, 574, 534, 571, 554, 577, 628, 487, 644, 640, 704, 648,
       968, 587, 699, 632, 591, 782, 510, 610, 524], dtype=int64)
```

In [14]:
```
x
```

Out[14]:
```
array([[9.0000e+00, 3.5710e+03, 1.9760e+03, 5.2500e-01],
       [9.0000e+00, 4.0920e+03, 1.2500e+03, 5.7200e-01],
       [9.0000e+00, 3.8650e+03, 1.5860e+03, 5.8000e-01],
       [7.5000e+00, 4.8700e+03, 2.3510e+03, 5.2900e-01],
       [8.0000e+00, 4.3990e+03, 4.3100e+02, 5.4400e-01],
       [1.0000e+01, 5.3420e+03, 1.3330e+03, 5.7100e-01],
       [8.0000e+00, 5.3190e+03, 1.1868e+04, 4.5100e-01],
       [8.0000e+00, 5.1260e+03, 2.1380e+03, 5.5300e-01],
       [8.0000e+00, 4.4470e+03, 8.5770e+03, 5.2900e-01],
       [7.0000e+00, 4.5120e+03, 8.5070e+03, 5.5200e-01],
       [8.0000e+00, 4.3910e+03, 5.9390e+03, 5.3000e-01],
       [7.5000e+00, 5.1260e+03, 1.4186e+04, 5.2500e-01],
       [7.0000e+00, 4.8170e+03, 6.9300e+03, 5.7400e-01],
       [7.0000e+00, 4.2070e+03, 6.5800e+03, 5.4500e-01],
       [7.0000e+00, 4.3320e+03, 8.1590e+03, 6.0800e-01],
       [7.0000e+00, 4.3180e+03, 1.0340e+04, 5.8600e-01],
       [7.0000e+00, 4.2060e+03, 8.5080e+03, 5.7200e-01],
       [7.0000e+00, 3.7180e+03, 4.7250e+03, 5.4000e-01],
       [7.0000e+00, 4.7160e+03, 5.9150e+03, 7.2400e-01],
       [8.5000e+00, 4.3410e+03, 6.0100e+03, 6.7700e-01],
       [7.0000e+00, 4.5930e+03, 7.8340e+03, 6.6300e-01],
       [8.0000e+00, 4.9830e+03, 6.0200e+02, 6.0200e-01],
       [9.0000e+00, 4.8970e+03, 2.4490e+03, 5.1100e-01],
       [9.0000e+00, 4.2580e+03, 4.6860e+03, 5.1700e-01],
       [8.5000e+00, 4.5740e+03, 2.6190e+03, 5.5100e-01],
       [9.0000e+00, 3.7210e+03, 4.7460e+03, 5.4400e-01],
       [8.0000e+00, 3.4480e+03, 5.3990e+03, 5.4800e-01],
       [7.5000e+00, 3.8460e+03, 9.0610e+03, 5.7900e-01],
       [8.0000e+00, 4.1880e+03, 5.9750e+03, 5.6300e-01],
       [9.0000e+00, 3.6010e+03, 4.6500e+03, 4.9300e-01],
       [7.0000e+00, 3.6400e+03, 6.9050e+03, 5.1800e-01],
       [7.0000e+00, 3.3330e+03, 6.5940e+03, 5.1300e-01],
       [8.0000e+00, 3.0630e+03, 6.5240e+03, 5.7800e-01],
       [7.5000e+00, 3.3570e+03, 4.1210e+03, 5.4700e-01],
       [8.0000e+00, 3.5280e+03, 3.4950e+03, 4.8700e-01],
       [6.5800e+00, 3.8020e+03, 7.8340e+03, 6.2900e-01],
       [5.0000e+00, 4.0450e+03, 1.7782e+04, 5.6600e-01],
       [7.0000e+00, 3.8970e+03, 6.3850e+03, 5.8600e-01],
       [8.5000e+00, 3.6350e+03, 3.2740e+03, 6.6300e-01],
       [7.0000e+00, 4.3450e+03, 3.9050e+03, 6.7200e-01],
       [7.0000e+00, 4.4490e+03, 4.6390e+03, 6.2600e-01],
       [7.0000e+00, 3.6560e+03, 3.9850e+03, 5.6300e-01],
       [7.0000e+00, 4.3000e+03, 3.6350e+03, 6.0300e-01],
       [7.0000e+00, 3.7450e+03, 2.6110e+03, 5.0800e-01],
       [6.0000e+00, 5.2150e+03, 2.3020e+03, 6.7200e-01],
       [9.0000e+00, 4.4760e+03, 3.9420e+03, 5.7100e-01],
       [7.0000e+00, 4.2960e+03, 4.0830e+03, 6.2300e-01],
       [7.0000e+00, 5.0020e+03, 9.7940e+03, 5.9300e-01]])
```

In [15]:
```
x.shape
```

Out[15]:
```
(48, 4)
```

In [17]:
```
y.shape
```

Out[17]:
```
(48,)
```

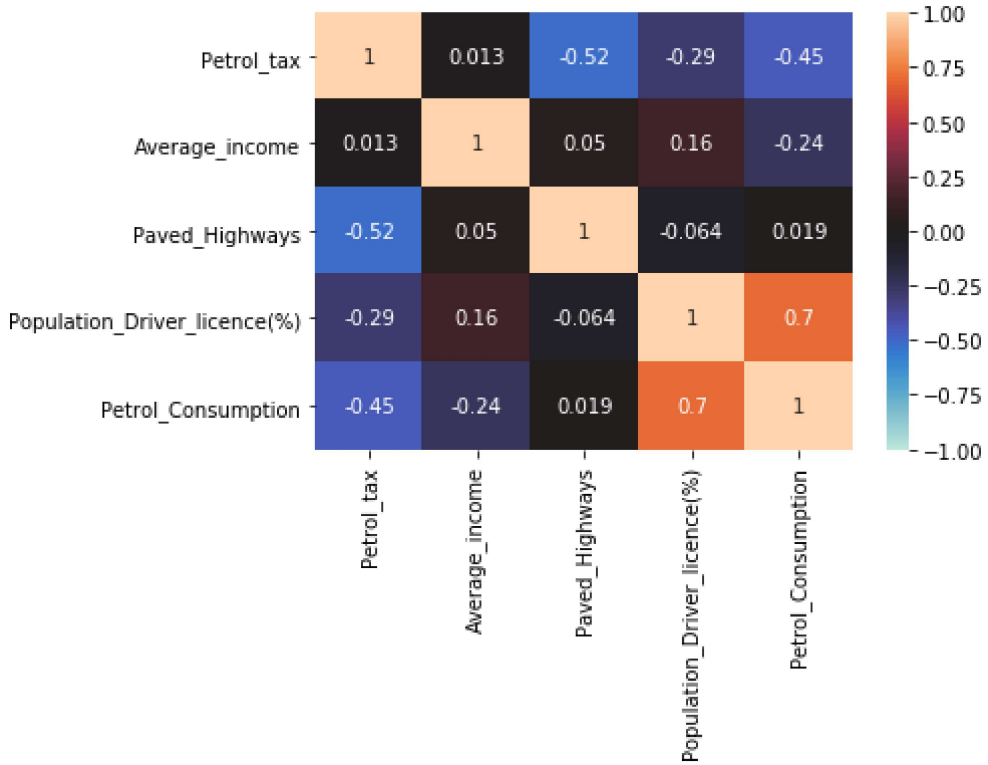
In [18]:
```
dataset.corr()
```

Out[18]:

	Petrol_tax	Average_income	Paved_Highways	Population_Driver_licenc
Petrol_tax	1.000000	0.012665	-0.522130	-0.28
Average_income	0.012665	1.000000	0.050163	0.15
Paved_Highways	-0.522130	0.050163	1.000000	-0.06
Population_Driver_licence(%)	-0.288037	0.157070	-0.064129	1.00
Petrol_Consumption	-0.451280	-0.244862	0.019042	0.69

In [19]:
```
import seaborn as sns
sns.heatmap(dataset.corr(),annot=True,vmin=-1,vmax=1,center=0)
```

Out[19]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x15657e4b108>
```



In [35]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3, random_state=1)
```

In [36]:
```
from sklearn.linear_model import LinearRegression
mlr=LinearRegression(normalize=True)
mlr.fit(x_train,y_train)
```

Out[36]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=True)
```

In [37]:
```
y_pred=mlr.predict(x_test)
y_pred
```

Out[37]:
```
array([620.58093461, 467.51687024, 739.21337689, 645.93715567,
       591.85595105, 486.34203341, 583.80501833, 598.93055594,
       708.77595273, 622.27676584, 565.00406531, 570.30820315,
       571.26024921, 508.77150871, 702.78484107])
```

In [38]:
```
 mlr.predict([[9.0,3571,1976,0.525]])
```

Out[38]:
```
array([534.03190111])
```

In [39]:
```
from sklearn.metrics import r2_score
accuracy=r2_score(y_test,y_pred)
```

In [40]:
```
accuracy
```

Out[40]:
```
0.6483616335976214
```

In [41]:
```
mlr.intercept_
```

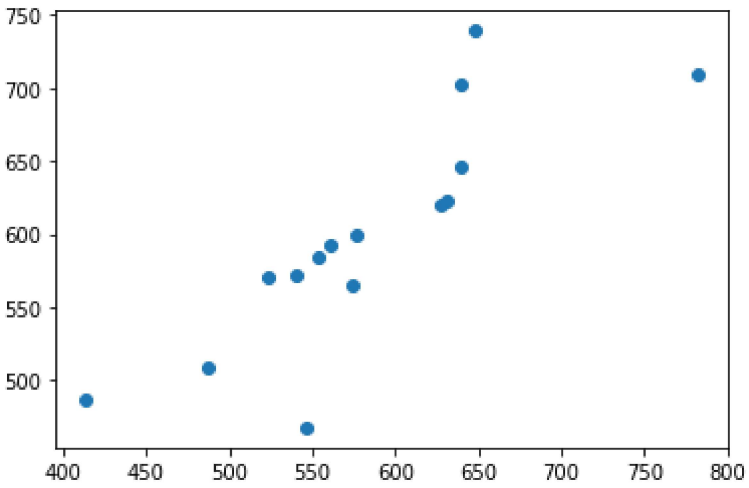
Out[41]:
```
304.6754038181706
```

In [42]:
```
mlr.coef_
```

Out[42]:
```
array([-2.85437070e+01, -7.36087171e-02, -1.65354087e-03,  1.43309331e+03])
```

In [45]:
```
plt.scatter(y_test,y_pred)
```

Out[45]:
```
<matplotlib.collections.PathCollection at 0x15658a58c08>
```



In [ ]: