Roll No. 412039

AIM: To Implement Multiple Linear Regression using Python.

Code & Output:

In [2]:

```
import pandas as pd
path_to_file = 'Downloads/petrol_consumption.csv'
df = pd.read_csv(path_to_file)
```

In [3]:

```
df.head()
```

Out[3]:

	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 [4]:

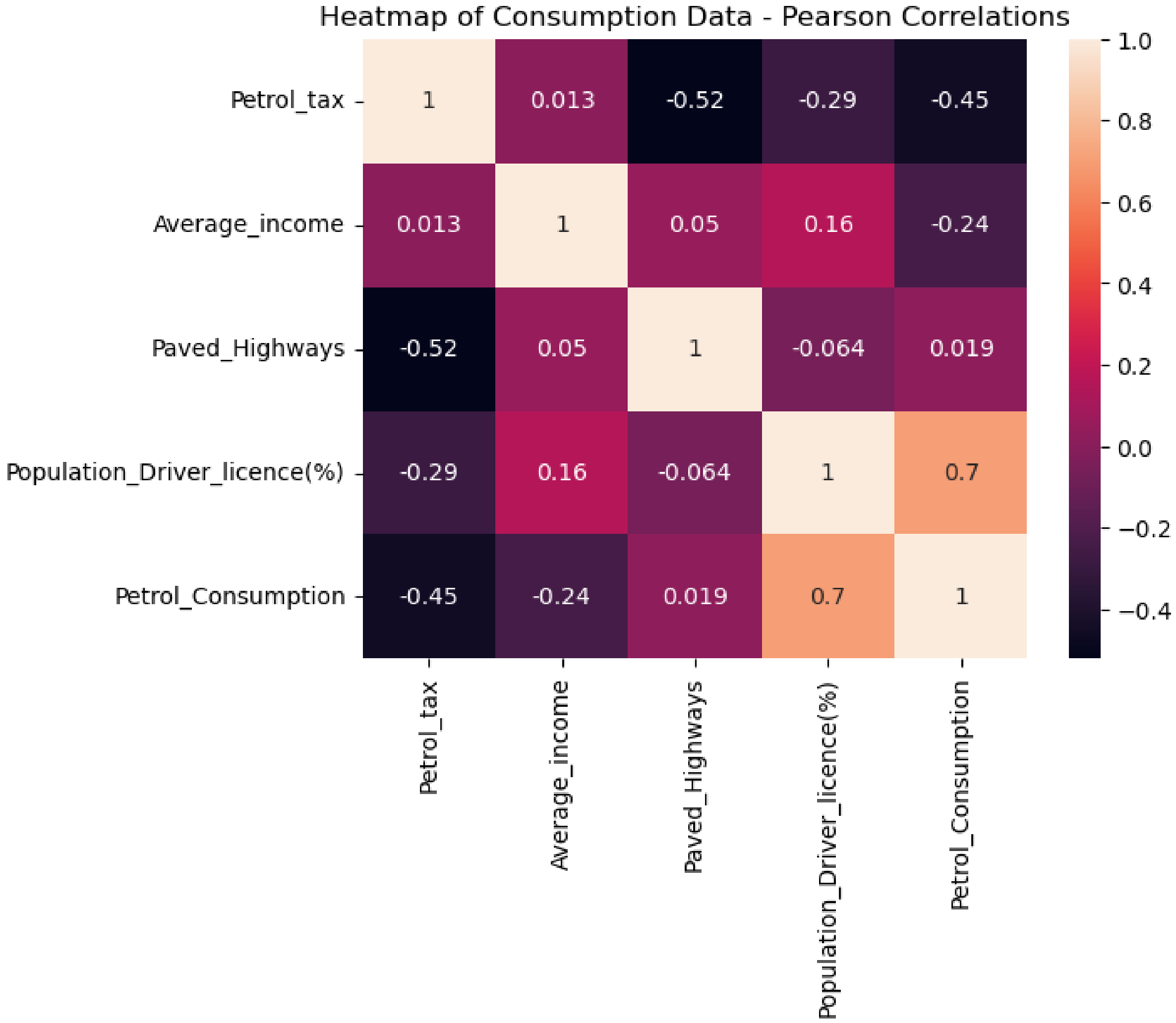
```
print(df.describe().round(2).T)
```

```
                              count     mean      std      min      25%  \
Petrol_tax                     48.0     7.67     0.95     5.00     7.00   
Average_income                 48.0  4241.83   573.62  3063.00  3739.00   
Paved_Highways                 48.0  5565.42  3491.51   431.00  3110.25   
Population_Driver_licence(%)   48.0     0.57     0.06     0.45     0.53   
Petrol_Consumption             48.0   576.77   111.89   344.00   509.50   

                                  50%      75%       max  
Petrol_tax                       7.50     8.12     10.00  
Average_income                4298.00  4578.75   5342.00  
Paved_Highways                4735.50  7156.00  17782.00  
Population_Driver_licence(%)     0.56     0.60      0.72  
Petrol_Consumption             568.50   632.75    968.00  
```

In [7]:

```
correlations = df.corr()
# annot=True displays the correlation values
sns.heatmap(correlations, annot=True).set(title='Heatmap of Consumption Data - Pearson Correlations')
```



In [8]:

```
y = df['Petrol_Consumption']
X = df[['Average_income', 'Paved_Highways',
       'Population_Driver_licence(%)', 'Petrol_tax']]
```

In [9]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size=0.2, random_state=42)
```

In [11]:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[11]:

```
LinearRegression()
```

In [12]:

```
regressor.intercept_
```

Out[12]:

```
361.4508790666836
```

In [13]:

```
regressor.coef_
```

Out[13]:

```
array([-5.65355145e-02, -4.38217137e-03,  1.34686930e+03, -3.69937459e+01])
```

In [15]:

```
feature_names = X.columns
print(feature_names)
```

```
Index(['Average_income', 'Paved_Highways', 'Population_Driver_licence(%)',
       'Petrol_tax'],
      dtype='object')
```

In [16]:

```
feature_names = X.columns
model_coefficients = regressor.coef_

coefficients_df = pd.DataFrame(data = model_coefficients,
                              index = feature_names,
                              columns = ['Coefficient value'])
print(coefficients_df)
```

```
                              Coefficient value
Average_income                        -0.056536
Paved_Highways                        -0.004382
Population_Driver_licence(%)        1346.869298
Petrol_tax                           -36.993746
```

In [17]:

```
y_pred = regressor.predict(X_test)
```

In [18]:

```
results = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
print(results)
```

```
    Actual   Predicted
27     631  606.692665
40     587  673.779442
26     577  584.991490
43     591  563.536910
24     460  519.058672
37     704  643.461003
12     525  572.897614
19     640  687.077036
4      410  547.609366
25     566  530.037630
```

In [21]:

```
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print(f'Mean absolute error: {mae:.2f}')
print(f'Mean squared error: {mse:.2f}')
print(f'Root mean squared error: {rmse:.2f}')
```

```
Mean absolute error: 53.47
Mean squared error: 4083.26
Root mean squared error: 63.90
```

In [22]:

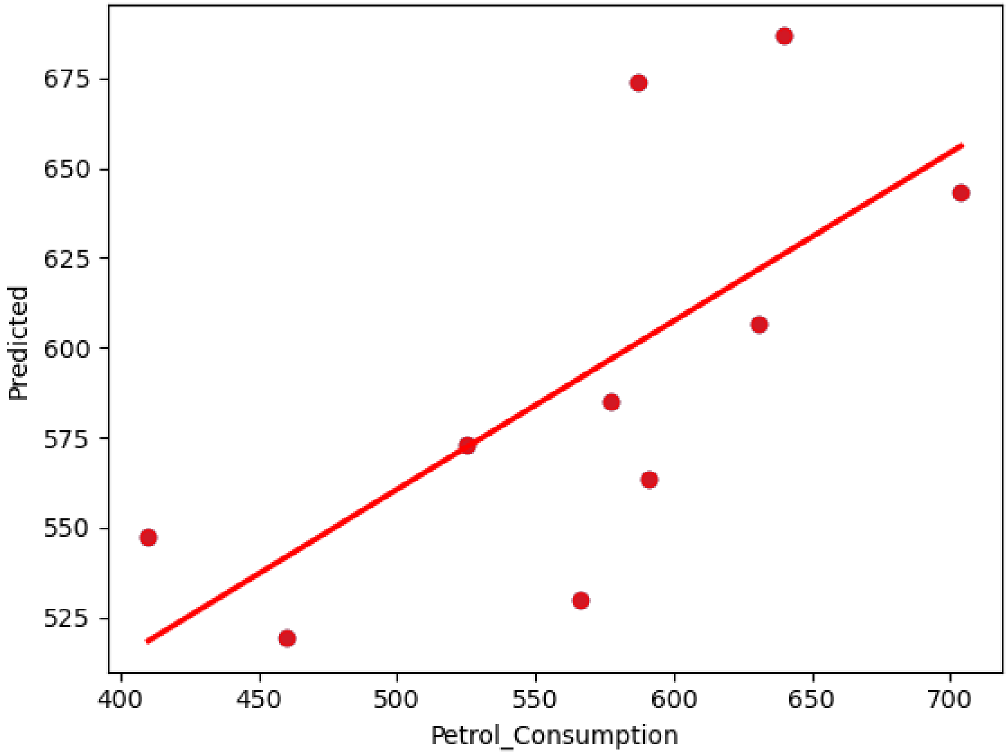
```
regressor.score(X_train, y_train)
```

Out[22]:

```
0.7068781342155135
```

In [23]:

```
plt.scatter(y_test,y_pred);
plt.xlabel('Actual');
plt.ylabel('Predicted');
sns.regplot(x=y_test,y=y_pred,ci=None,color ='red');
```



Conclusion: Thus, we implemented multiple Linear Regression using sklearn library of python.