Simple Linear Regression

Using scikit-learn to implement simple Linear Regression. Check how Co2Emission is related to Co2Emissions. Create a model using train set,test using test set .

```python
In [1]:  import matplotlib.pyplot as plt
         import pandas as pd
         import pylab as pl
         import numpy as np
         %matplotlib inline
```

```python
In [2]:  df = pd.read_csv("FuelConsumptionCo2.csv")
         df.head()
```

Out[2]:

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	TRANSMISSION	FUE
0	2014	ACURA	ILX	COMPACT	2.0	4	AS5	
1	2014	ACURA	ILX	COMPACT	2.4	4	M6	
2	2014	ACURA	ILX HYBRID	COMPACT	1.5	4	AV7	
3	2014	ACURA	MDX 4WD	SUV - SMALL	3.5	6	AS6	
4	2014	ACURA	RDX AWD	SUV - SMALL	3.5	6	AS6	

Creating train and test dataset:

Train/Test Split dataset to mutually exclusive. We can use 80% of the entire data for training, and the 20% for testing. We create a mask using np.random.rand().

```python
In [4]:  cdf = df[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB','CO2EMISSIONS']]
         msk = np.random.rand(len(df)) < 0.8
         train = cdf[msk]
         test = cdf[~msk]
```

Linear Regression

Training model using Training Set

In [5]:
```python
from sklearn import linear_model
regr = linear_model.LinearRegression()
train_x = np.asanyarray(train[['ENGINESIZE']])
train_y = np.asanyarray(train[['CO2EMISSIONS']])
regr.fit (train_x, train_y)

# The coefficients and intercept
print ('Coefficients: ', regr.coef_)
print ('Intercept: ',regr.intercept_)
```
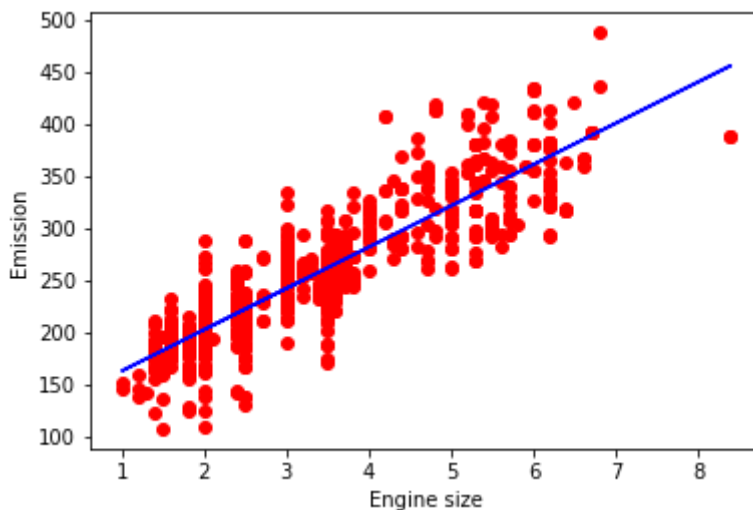
```
Coefficients:  [[39.57984166]]
Intercept:  [123.83703388]
```

Plotting to find how regression line fits:

In [6]:
```python
plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS,  color='red')
plt.plot(train_x, regr.coef_[0][0]*train_x + regr.intercept_[0], '-b')
plt.xlabel("Engine size")
plt.ylabel("Emission")
```

Out[6]: Text(0, 0.5, 'Emission')

Evaluate using R2 score

In [7]:
```python
from sklearn.metrics import r2_score
test_x = np.asanyarray(test[['ENGINESIZE']])
test_y = np.asanyarray(test[['CO2EMISSIONS']])
test_y_hat = regr.predict(test_x)
```

In [8]:
```python
print("R2-score: %.2f" % r2_score(test_y_hat , test_y) )
```

```
R2-score: 0.66
```

In [ ]: