The information in this dataset pertains to the fluctuation of a vehicle's CO2 emissions based on various characteristics. It was obtained from the Canadian Government's open data website and is a consolidated version spanning seven years.

In [1]:
```
# import the required libraries

import matplotlib.pyplot as plt
import pandas as pd
import pylab as pl
import numpy as np
import urllib
%matplotlib inline
```

In [2]:
```
# Downloading Data

url = 'https://open.canada.ca/data/en/dataset/98f1a129-f628-4ce4-b24d-6f16bf24dd64?utm_me
```

In [3]:
```
#Reading the data to make sense of it

df = pd.read_csv("FuelConsumption.csv")

# take a look at the dataset
df.head()
```

Out[3]:

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

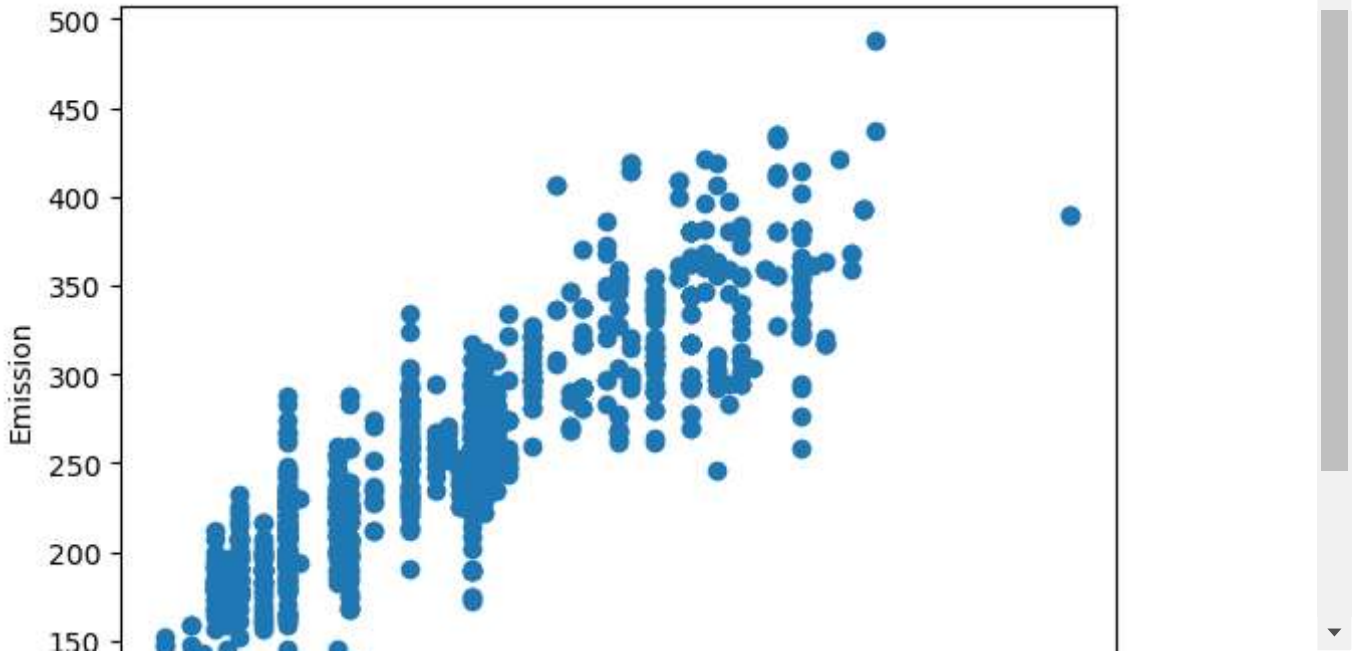
Let's select some features that we want to use for regression.

In [4]:
```
#selecting features that we want to use for regression

cdf = df[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_CITY','FUELCONSUMPTION_HWY','FUELCONS
cdf.head(9)
```

Out[4]:

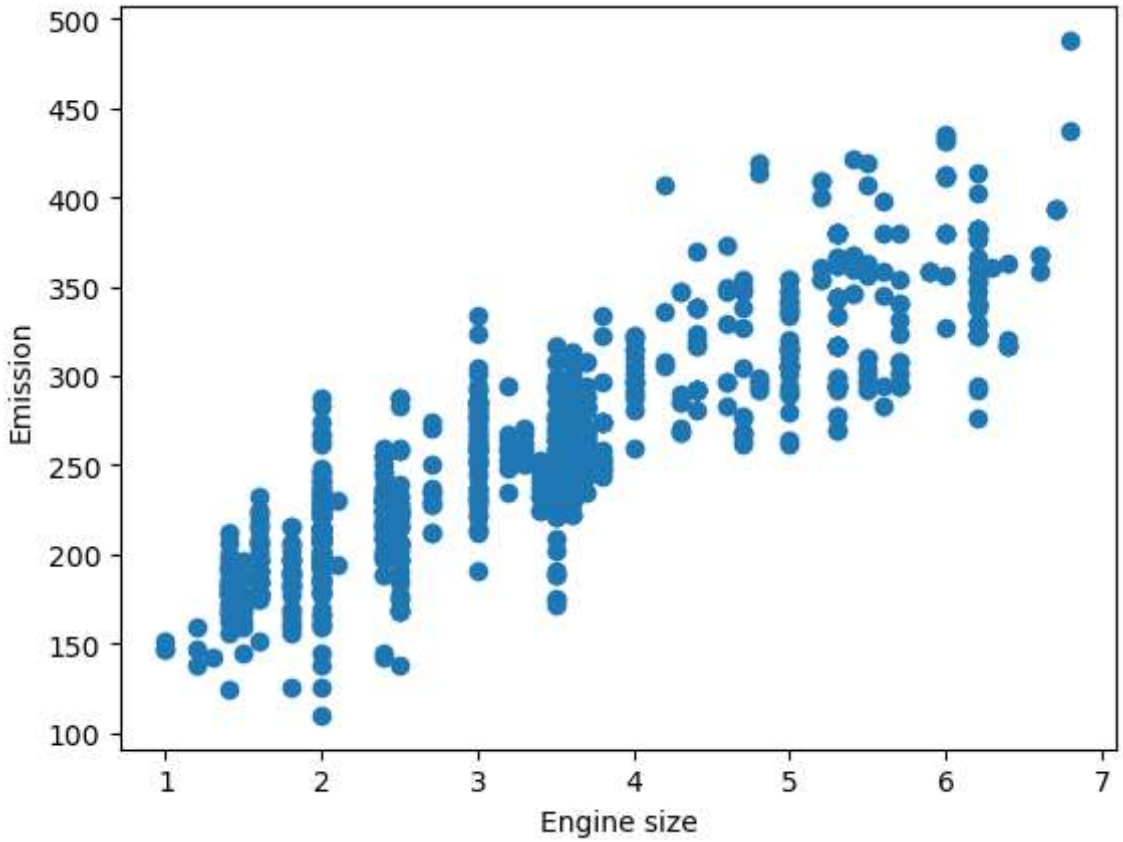
	ENGINESIZE	CYLINDERS	FUELCONSUMPTION_CITY	FUELCONSUMPTION_HWY	FUELCONSUMPTION_COM
0	2.0	4	9.9	6.7	8
1	2.4	4	11.2	7.7	9
2	1.5	4	6.0	5.8	5
3	3.5	6	12.7	9.1	11
4	3.5	6	12.1	8.7	10
5	3.5	6	11.9	7.7	10
6	3.5	6	11.8	8.1	10
7	3.7	6	12.8	9.0	11
8	3.7	6	13.4	9.5	11

In [5]:
```
# Plotting emission values with respect to Engine size

plt.scatter(cdf.ENGINESIZE, cdf.CO2EMISSIONS)
plt.xlabel("Engine size")
plt.ylabel("Emission")
plt.show()
```



In [6]:
```
# Creating train and test dataset

msk = np.random.rand(len(df)) < 0.8
train = cdf[msk]
test = cdf[~msk]
```

In [7]:
```
# Train data distribution

plt.scatter(train.ENGINESIZE, train.CO2EMISSIONS)
plt.xlabel("Engine size")
plt.ylabel("Emission")
plt.show()
```



In [8]:
```
#Multiple Regression Model
#Coefficient and Intercept are the parameters of the fitted line

from sklearn import linear_model
regr = linear_model.LinearRegression()
x = np.asanyarray(train[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB']])
y = np.asanyarray(train[['CO2EMISSIONS']])
regr.fit (x, y)
# The coefficients
print ('Coefficients: ', regr.coef_)
```

```
Coefficients:  [[11.61272261  6.82806543  9.45328819]]
```

In [9]:
```
#Prediction

y_hat= regr.predict(test[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB']])
x = np.asanyarray(test[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_COMB']])
y = np.asanyarray(test[['CO2EMISSIONS']])
print("Residual sum of squares: %.2f"
      % np.mean((y_hat - y) ** 2))

# Explained variance score: 1 is perfect prediction
print('Variance score: %.2f' % regr.score(x, y))
```

```
Residual sum of squares: 548.71
Variance score: 0.87

C:\Users\reube\anaconda3\lib\site-packages\sklearn\base.py:413: UserWarning: X has feat
ure names, but LinearRegression was fitted without feature names
  warnings.warn(
```

In [10]:

```
# UsingnFUELCONSUMPTION_CITY and FUELCONSUMPTION_HWY instead of FUELCONSUMPTION_COMB

regr = linear_model.LinearRegression()
x = np.asanyarray(train[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_CITY','FUELCONSUMPTIO
y = np.asanyarray(train[['CO2EMISSIONS']])
regr.fit (x, y)
print ('Coefficients: ', regr.coef_)
y_= regr.predict(test[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_CITY','FUELCONSUMPTION_H
x = np.asanyarray(test[['ENGINESIZE','CYLINDERS','FUELCONSUMPTION_CITY','FUELCONSUMPTION_
y = np.asanyarray(test[['CO2EMISSIONS']])
print("Residual sum of squares: %.2f"% np.mean((y_ - y) ** 2))
print('Variance score: %.2f' % regr.score(x, y))
```

```
Coefficients:  [[11.6002967   6.73650792  5.49621353  3.86735376]]
Residual sum of squares: 547.73
Variance score: 0.87

C:\Users\reube\anaconda3\lib\site-packages\sklearn\base.py:413: UserWarning: X has feat
ure names, but LinearRegression was fitted without feature names
  warnings.warn(
```