Multi Linear Regression

In [1]:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

In [2]:

```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\jesus\anaconda3\lib\site-p
ackages (1.0.5)
Requirement already satisfied: python-dateutil>=2.6.1 in c:\users\jesus\anac
onda3\lib\site-packages (from pandas) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in c:\users\jesus\anaconda3\lib
\site-packages (from pandas) (2020.1)
Requirement already satisfied: numpy>=1.13.3 in c:\users\jesus\appdata\roami
ng\python\python38\site-packages (from pandas) (1.19.1)
Requirement already satisfied: six>=1.5 in c:\users\jesus\anaconda3\lib\site
-packages (from python-dateutil>=2.6.1->pandas) (1.15.0)
Note: you may need to restart the kernel to use updated packages.
```

Business Use Case

In [3]:

```
df = pd.read_csv("https://raw.githubusercontent.com/AP-State-Skill-Development-Corporation/
df1 = pd.read_csv('C:\\Users\\Jesus\\Desktop\\Online Training\\Machine-Learning-Using-Pytho
df2 = pd.read_csv('FuelConsumptionCo2.csv')
df.head()
```

Out[3]:

	MODELYEAR	MAKE	MODEL	VEHICLECLASS	ENGINESIZE	CYLINDERS	TRANSMISSION
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

In [4]:

```
df.shape
```

Out[4]:

```
(1067, 13)
```

In [5]:

```
df.columns
```

Out[5]:

```
Index(['MODELYEAR', 'MAKE', 'MODEL', 'VEHICLECLASS', 'ENGINESIZE', 'CYLINDER
S',
       'TRANSMISSION', 'FUELTYPE', 'FUELCONSUMPTION_CITY',
       'FUELCONSUMPTION_HWY', 'FUELCONSUMPTION_COMB',
       'FUELCONSUMPTION_COMB_MPG', 'CO2EMISSIONS'],
      dtype='object')
```

Understanding the Data

`FuelConsumption.csv` :

We have downloaded a fuel consumption dataset, **`FuelConsumption.csv`** , which contains model-specific fuel consumption ratings and estimated carbon dioxide emissions for new light-duty vehicles for retail sale in Canada. [Dataset source (http://open.canada.ca/data/en/dataset/98f1a129-f628-4ce4-b24d-6f16bf24dd64)](http://open.canada.ca/data/en/dataset/98f1a129-f628-4ce4-b24d-6f16bf24dd64)

- **MODELYEAR** e.g. 2014
- **MAKE** e.g. Acura
- **MODEL** e.g. ILX
- **VEHICLE CLASS** e.g. SUV
- **ENGINE SIZE** e.g. 4.7
- **CYLINDERS** e.g 6
- **TRANSMISSION** e.g. A6
- **FUELTYPE** e.g. z
- **FUEL CONSUMPTION in CITY(L/100 km)** e.g. 9.9
- **FUEL CONSUMPTION in HWY (L/100 km)** e.g. 8.9
- **FUEL CONSUMPTION COMB (L/100 km)** e.g. 9.2
- **CO2 EMISSIONS (g/km)** e.g. 182 --> low --> 0

In [6]:

```
df['CO2EMISSIONS'].head(10)
```

Out[6]:

```
0    196
1    221
2    136
3    255
4    244
5    230
6    232
7    255
8    267
9    212
Name: CO2EMISSIONS, dtype: int64
```

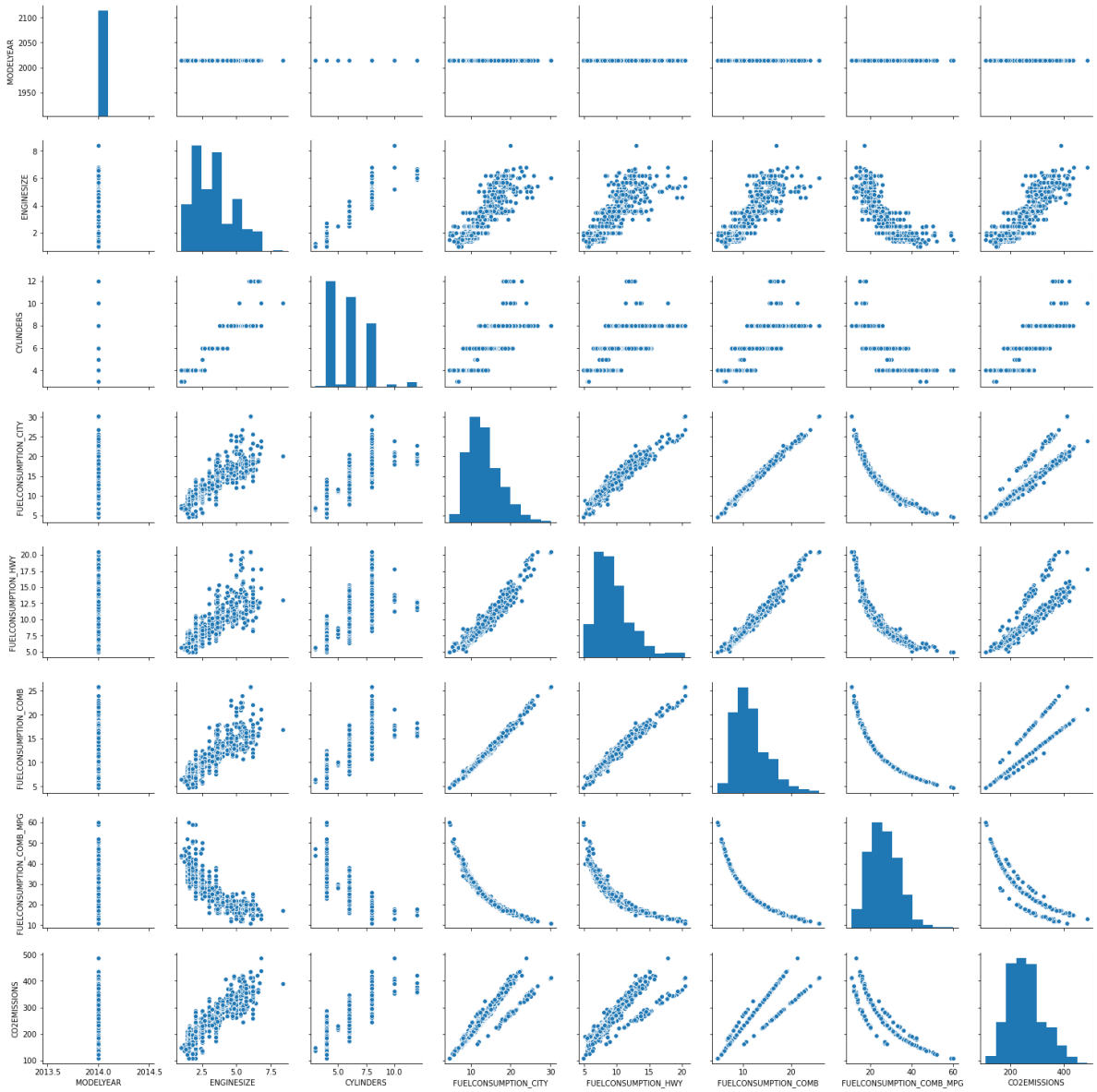
In [7]:

```
import seaborn as sns

sns.pairplot(df)
```

Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x1ed68702550>
```



Multi Linear Regression/ Linear Regression With multiple varibles

- Engine Size
- Fuel Consumption_city
- Fuel Consumption_hwy
- Fuel Consumption Comb

$$Y = M * x + C$$

$$Y = m_1 * x_1 + m_2 * x_2 + m_3 * x_3 + \ldots\ldots\ldots\ldots + m_n * x_n + C + E$$

$$Y = m_1 * ES + m_2 * fuelc + m_3 * fuelconhwy + .. + C$$

In [8]:

```
X = df[['ENGINESIZE','FUELCONSUMPTION_CITY', 'FUELCONSUMPTION_HWY', 'FUELCONSUMPTION_COMB']
y = df[['CO2EMISSIONS']]
```

In [9]:

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(X, y, train_size = 0.75, random_state =
```

In [10]:

```
x_train.shape, x_test.shape
```

Out[10]:

```
((800, 4), (267, 4))
```

In [11]:

```
x_train.head()
```

Out[11]:

	ENGINESIZE	FUELCONSUMPTION_CITY	FUELCONSUMPTION_HWY	FUELCONSUMPTION_C
232	2.4	10.0	7.0	
420	5.8	15.9	9.9	
602	1.8	9.3	6.4	
158	2.5	10.5	7.2	
256	1.4	8.7	7.0	

In [12]:

```
x_train.head()
```

Out[12]:

	ENGINESIZE	FUELCONSUMPTION_CITY	FUELCONSUMPTION_HWY	FUELCONSUMPTION_C
232	2.4	10.0	7.0	
420	5.8	15.9	9.9	
602	1.8	9.3	6.4	
158	2.5	10.5	7.2	
256	1.4	8.7	7.0	

In [13]:

```
from sklearn.linear_model import LinearRegression
```

In [14]:

```
model = LinearRegression()

model.fit(x_train, y_train)
```

Out[14]:

```
LinearRegression()
```

In [15]:

```
x_test.head(1).values
```

Out[15]:

```
array([[ 5. , 20.6, 13.6, 17.5]])
```

In [16]:

```
y_test.head(1).values
```

Out[16]:

```
array([[280]], dtype=int64)
```

In [17]:

```
model.predict([[ 5. , 20.6, 13.6, 17.5]])
```

Out[17]:

```
array([[348.03212316]])
```

In [18]:

```
x_test.head(2).values
```

Out[18]:

```
array([[ 5. , 20.6, 13.6, 17.5],
       [ 1.6,  8.6,  6.8,  7.8]])
```

In [19]:

```
y_test.head(2).values
```

Out[19]:

```
array([[280],
       [179]], dtype=int64)
```

In [20]:

```
model.predict([[ 1.6,  8.6,  6.8,  7.8]])
```

Out[20]:

```
array([[184.19313927]])
```

In [21]:

```
from sklearn.metrics import mean_squared_error

mean_squared_error(y_test, model.predict(x_test)) ** 0.5
```

Out[21]:

```
20.886832384194694
```

In [22]:

```
model.score(x_test, model.predict(x_test))
```

Out[22]:

```
1.0
```

Non - Linear Regression

Non-linear regressions are a relationship between independent variables x and a dependent variable y which result in a non-linear function modeled data. Essentially any relationship that is not linear can be termed as non-linear, and is usually represented by the polynomial of k degrees (maximum power of x).

$$y = ax^3 + bx^2 + cx + d$$

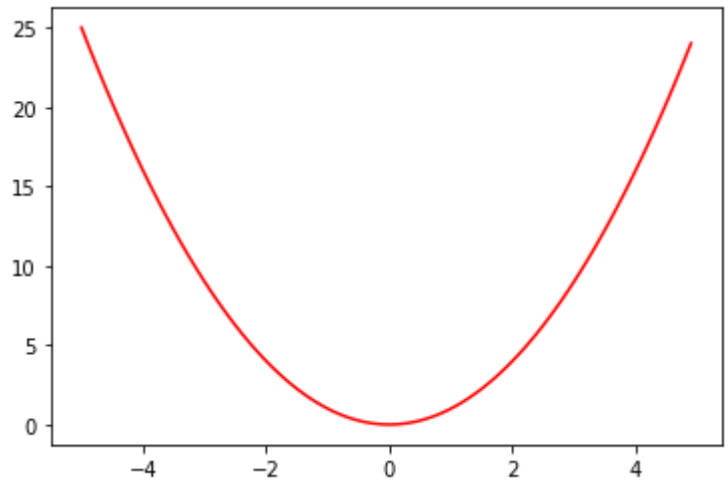
In [23]:

```
x = np.arange(-5, 5, 0.1)

plt.plot(x,x ** 2, c = 'r')
```

Out[23]:

```
[<matplotlib.lines.Line2D at 0x1ed77fbe430>]
```

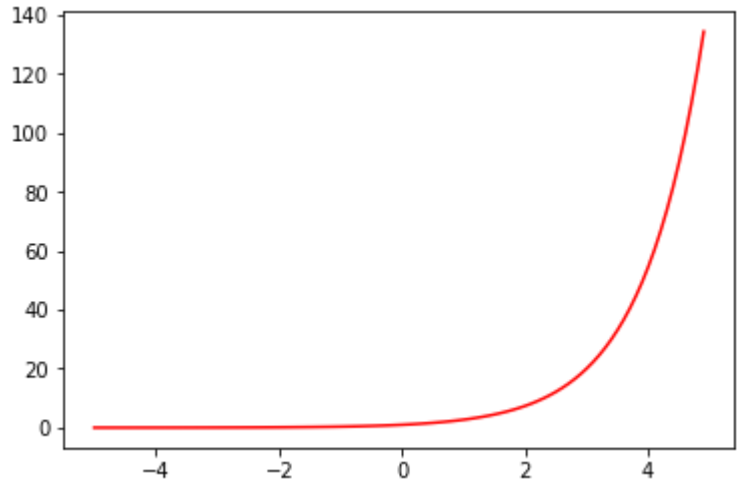


In [24]:

```
plt.plot(x,np.exp(x), c = 'r')
```

Out[24]:

```
[<matplotlib.lines.Line2D at 0x1ed7800e100>]
```



$$y = m_1 * x^0 + m_2 * x^2 + - - - - - - - - + m_n * x^n$$

$$a1 = x^0$$
$$a2 = x^2$$
$$.$$

$$y = m_1 * a_1 + m_2 * a_2 + m_3 * a_3 + \ldots . m_n * a_n$$

1. Polynomial Features
2. Multi Linear Regress

In [25]:

```
X = df['FUELCONSUMPTION_COMB_MPG'].values.reshape(-1, 1)
y = df['CO2EMISSIONS']
```

In [26]:

```
from sklearn.preprocessing import PolynomialFeatures
```

In [27]:

```
poly = PolynomialFeatures(degree = 2)
```

In [28]:

```
x_poly = poly.fit_transform(X)

x_poly
```

Out[28]:

```
array([[1.000e+00, 3.300e+01, 1.089e+03],
       [1.000e+00, 2.900e+01, 8.410e+02],
       [1.000e+00, 4.800e+01, 2.304e+03],
       ...,
       [1.000e+00, 2.400e+01, 5.760e+02],
       [1.000e+00, 2.500e+01, 6.250e+02],
       [1.000e+00, 2.200e+01, 4.840e+02]])
```

In [29]:

```
X
```

Out[29]:

```
array([[33],
       [29],
       [48],
       ...,
       [24],
       [25],
       [22]], dtype=int64)
```

In [30]:

```
model2 = LinearRegression()

model2.fit(x_poly, y)
```

Out[30]:

```
LinearRegression()
```

In [31]:

```
mean_squared_error(y, model2.predict(x_poly)) ** 0.5
```

Out[31]:

```
24.355929375947483
```

In [40]:

```
Y_pred = model2.predict(x_poly)
plt.scatter(X,y, label = 'Original Data')
plt.plot(X, Y_pred, 'ro', label = 'Predicted line')
plt.xlabel('FUELCONSUMPTION_COMB_MPG')
plt.ylabel('CO2EMISSIONS')
plt.legend()
```

Out[40]:

```
<matplotlib.legend.Legend at 0x1ed7963adf0>
```