1. Linear Regression with Single Variable

In [2]:

```
import pandas as pd
```

Step1: Define Business Use Case

Our Use case is to predict the salary of a person based on Years of Experience

In [3]:

```
df = pd.read_csv("https://raw.githubusercontent.com/AP-State-Skill-Development-Corporation/
```

Step2: Data Exploration

In [4]:

```
df.head()
```

Out[4]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

In [5]:

```
df.shape
```

Out[5]:

```
(30, 2)
```

In [6]:

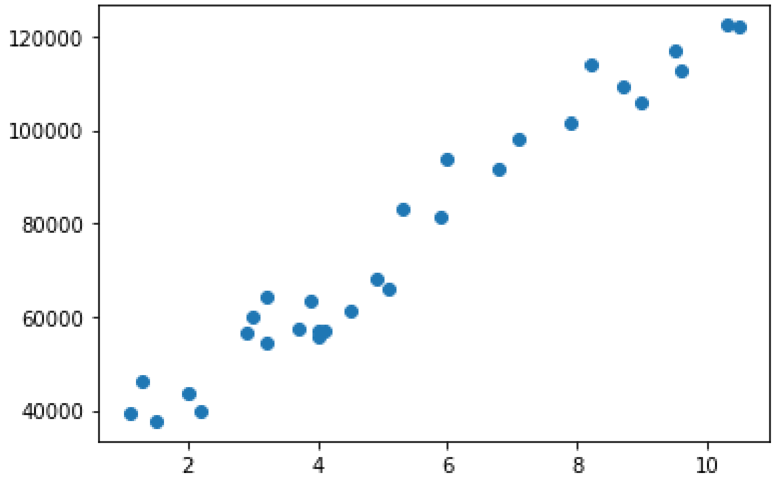
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     float64
dtypes: float64(2)
memory usage: 608.0 bytes
```

In [7]:

```
import matplotlib.pyplot as plt

plt.scatter(df['YearsExperience'], df['Salary'])
plt.show()
```



In [8]:

```
df
```

Out[8]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [9]:

```
df.corr()
```

Out[9]:

	YearsExperience	Salary
YearsExperience	1.000000	0.978242
Salary	0.978242	1.000000

Step3: Select Algorithm

Based on the data exploration we have found that YearsExperience is Positively Linearly Coreleated with the salary so we have selected the linear regression

$$Salary = M * YearExperience + C$$

Predict the output values based on the input values

In [27]:

```
df['YearsExperience'].values
```

Out[27]:

```
array([ 1.1,  1.3,  1.5,  2. ,  2.2,  2.9,  3. ,  3.2,  3.2,  3.7,  3.9,
        4. ,  4. ,  4.1,  4.5,  4.9,  5.1,  5.3,  5.9,  6. ,  6.8,  7.1,
        7.9,  8.2,  8.7,  9. ,  9.5,  9.6, 10.3, 10.5])
```

In [16]:

```
x = df['YearsExperience'].values.reshape(-1, 1)
y = df['Salary']
```

In [28]:

```
x.shape
```

Out[28]:

```
(30, 1)
```

In [12]:

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

Step4: Build the model

In [13]:

```
model = LinearRegression()
```

In [17]:

```
model.fit(x, y)
```

Out[17]:

```
LinearRegression()
```

In [18]:

```
model.coef_
```

Out[18]:

```
array([9449.96232146])
```

In [19]:

```
model.intercept_
```

Out[19]:

```
25792.20019866871
```

$$Y = M * X + C$$

so by building the model we have calculated the coefficient/slope and intercept as in the above cells

$$salary = 9449.962 * X + 25792.200$$

In [20]:

```
model.predict([[11]])
```

Out[20]:

```
array([129741.78573467])
```

In [21]:

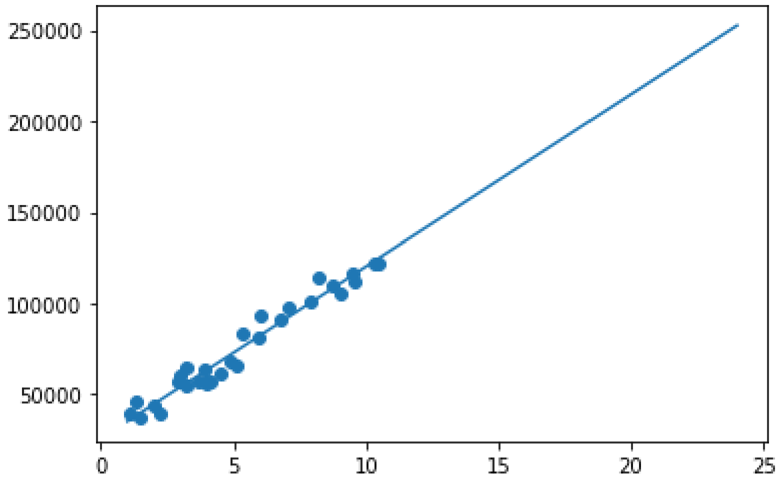
```
model.predict([[12]])
```

Out[21]:

```
array([139191.74805613])
```

In [31]:

```
import numpy as np
new = np.arange(1, 25).reshape(-1, 1)
plt.scatter(df['YearsExperience'], df['Salary'])
plt.plot(new, model.predict(new))
plt.show()
```



Step6: Evaluate

In [32]:

```
model.score(x, y)
```

Out[32]:

```
0.9569566641435086
```

Linear Regression with Multiple Variables

Step1: Define Business Use Case

Our Use case is to predict the CO2Emissions of a person based on few features

In [33]:

```
co2 = pd.read_csv('https://raw.githubusercontent.com/AP-State-Skill-Development-Corporation
```

Step2: Data Exploration

In [34]:

```
co2.head()
```

Out[34]:

	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 [35]:

```
co2.columns
```

Out[35]:

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

In [36]:

```
co2.shape
```

Out[36]:

```
(1067, 13)
```

In [38]:

```
co2['MAKE'].value_counts().shape
```

Out[38]:

```
(39,)
```

In [39]:

```
co2['MAKE'].value_counts()
```

Out[39]:

```
FORD             90
CHEVROLET        86
BMW              64
MERCEDES-BENZ    59
TOYOTA           49
AUDI             49
GMC              49
PORSCHE          44
VOLKSWAGEN       42
DODGE            39
MINI             36
NISSAN           33
KIA              33
CADILLAC         32
JEEP             31
MAZDA            27
HYUNDAI          24
SUBARU           23
LEXUS            22
JAGUAR           22
HONDA            21
INFINITI         21
CHRYSLER         19
LAND ROVER       19
MITSUBISHI       16
BUICK            16
RAM              13
ACURA            12
VOLVO            11
LINCOLN          11
FIAT             10
SCION             9
BENTLEY           8
ASTON MARTIN      7
ROLLS-ROYCE       7
MASERATI          6
LAMBORGHINI       3
SMART             2
SRT               2
Name: MAKE, dtype: int64
```

In [40]:

```
co2['VEHICLECLASS'].value_counts()
```

Out[40]:

```
MID-SIZE                    178
COMPACT                     172
SUV - SMALL                 154
SUV - STANDARD              110
FULL-SIZE                    86
TWO-SEATER                   71
SUBCOMPACT                   65
PICKUP TRUCK - STANDARD      62
MINICOMPACT                  47
STATION WAGON - SMALL        36
VAN - PASSENGER              25
VAN - CARGO                  22
MINIVAN                      14
PICKUP TRUCK - SMALL         12
SPECIAL PURPOSE VEHICLE       7
STATION WAGON - MID-SIZE      6
Name: VEHICLECLASS, dtype: int64
```

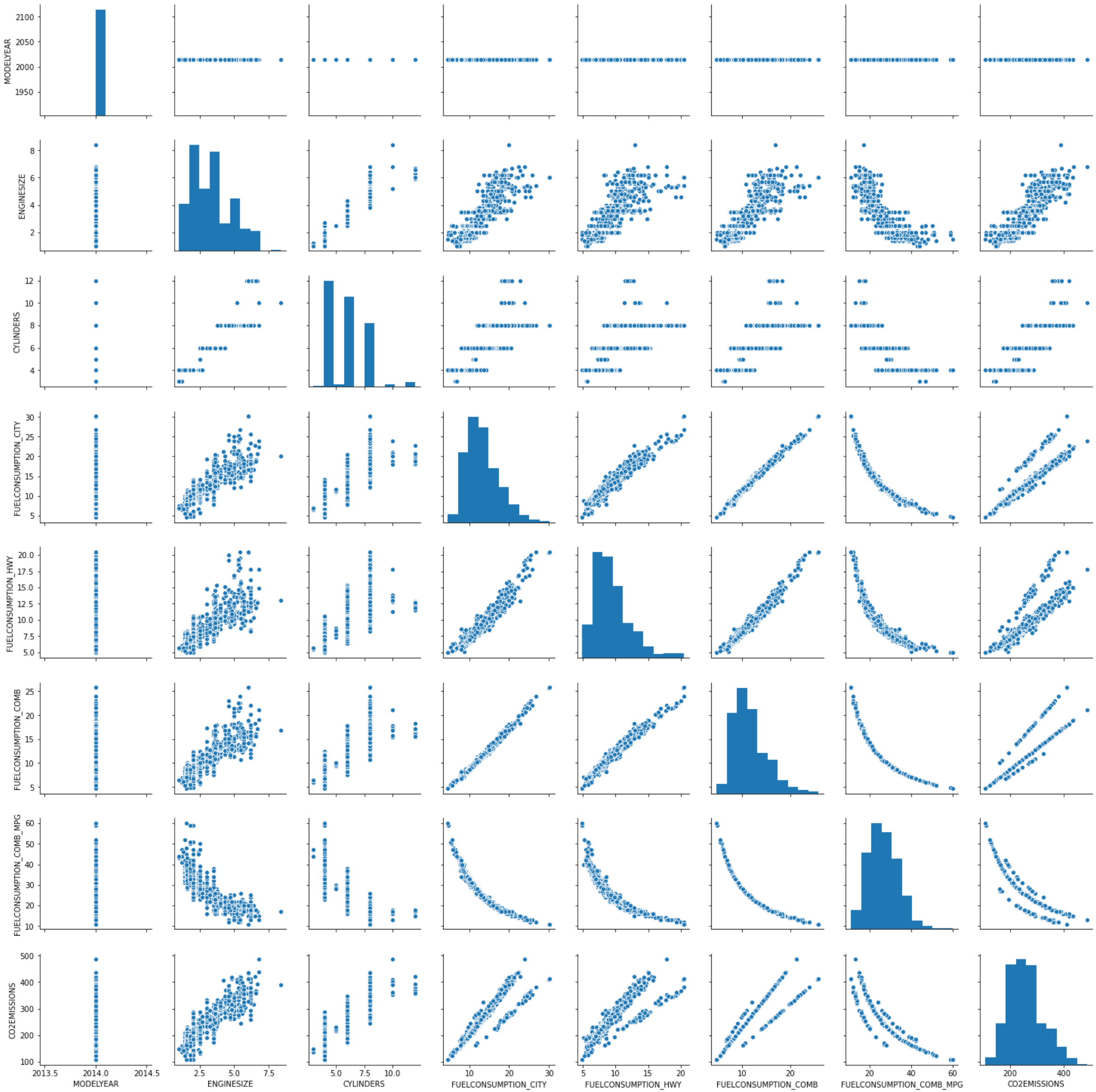
In [7]:

```
import seaborn as sns

sns.pairplot(df)
```

Out[7]:

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



Step3: Select Algorithm

Based on the data exploration we have found that `CO2Emissions` is Positively Linearly Coreleated with the `FUELCONSUMPTION_CITY` , `FUELCONSUMPTION_HWY` , `FUELCONSUMPTION_COMB` so we have selected the linear regression with multiple variables

In [43]:

```
co2.columns
```

Out[43]:

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

In [45]:

```
x = co2[['FUELCONSUMPTION_CITY','FUELCONSUMPTION_HWY', 'FUELCONSUMPTION_COMB']]
y = co2['CO2EMISSIONS']
```

Equation for Linear Regression with Multiple Variables

$$y = m_1x_1 + m_2x_2 + m_3x_3 + \ldots\ldots m_nx_n + c$$

splitting the entire data in to two parts

1. Training Part
2. Testing Part

In [46]:

```
from sklearn.model_selection import train_test_split

x_tr, x_tt, y_tr, y_tt = train_test_split(x, y, test_size = 0.3, random_state = 42)
```

In [61]:

```
train_test_split(x, y, test_size = 0.3, random_state = 42)
```

Out[61]:

```
[      FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY  FUELCONSUMPTION_COMB
 820                   14.0                 10.3                  12.3
 902                   13.1                  8.7                  11.1
 350                   20.6                 15.5                  18.3
 5                     11.9                  7.7                  10.0
 310                   18.3                 12.6                  15.7
 ...                    ...                  ...                   ...
 330                   14.2                  9.4                  12.0
 466                   11.5                  8.2                  10.0
 121                   16.2                 10.9                  13.8
 1044                  10.0                  6.9                   8.6
 860                   19.7                 14.3                  17.3

 [746 rows x 3 columns],
      FUELCONSUMPTION_CITY  FUELCONSUMPTION_HWY  FUELCONSUMPTION_COMB
 732                  15.4                 10.4                  13.2
 657                  11.3                  7.6                   9.6
 168                  15.1                  9.9                  12.8
 86                   11.4                  7.3                   9.6
 411                  10.5                  7.1                   9.0
 ..                    ...                  ...                   ...
 82                   10.4                  6.7                   8.7
 436                  23.5                 17.7                  20.9
 457                  16.3                 11.4                  14.1
 497                   8.3                  6.9                   7.7
 853                   9.1                  8.5                   8.8

 [321 rows x 3 columns],
 820     283
 902     255
 350     421
 5       230
 310     251
        ...
 330     276
 466     230
 121     317
 1044    198
 860     398
 Name: CO2EMISSIONS, Length: 746, dtype: int64,
 732    304
 657    221
 168    294
 86     221
 411    207
       ...
 82     200
 436    334
 457    324
 497    177
 853    202
 Name: CO2EMISSIONS, Length: 321, dtype: int64]
```

In [48]:

```
x_tr.shape, x_tt.shape
```

Out[48]:

```
((746, 3), (321, 3))
```

Step4: Build the model

In [50]:

```
from sklearn.linear_model import LinearRegression

model.fit(x_tr, y_tr)
```

Out[50]:

```
LinearRegression()
```

In [53]:

```
y_pred = model.predict(x_tt)
```

In [54]:

```
x_tt.head(1)
```

Out[54]:

	FUELCONSUMPTION_CITY	FUELCONSUMPTION_HWY	FUELCONSUMPTION_COMB
732	15.4	10.4	13.2

In [55]:

```
y_tt.head(1)
```

Out[55]:

```
732    304
Name: CO2EMISSIONS, dtype: int64
```

In [56]:

```
y_pred[0]
```

Out[56]:

```
286.82307123945753
```

Step6: Evaluate

In [57]:

```
model.score(x_tt, y_tt)
```

Out[57]:

```
0.8113937336428083
```

In [58]:

```
model.intercept_
```

Out[58]:

```
73.5306782666596
```

In [60]:

```
model.coef_
```

Out[60]:

```
array([10.01652918, -6.39753184,  9.51304353])
```