What is Simple Linear Regression?

Simple Linear Regression is finding the best relationship between the input variable x (independent variable) and the expected variable y (dependent variable). The linear relationship between these two variables can be represented by a straight line called regression line.

Formula :- $y = b0 + b1x$

What do terms represent?

- y is the response or the target variable
- x is the feature
- $b1$ is the coefficient of x
- $b0$ is the intercept

Estimating ("Learning") Model Coefficients

The coefficients are estimated using the **least-squares criterion**, i.e., the best fit line has to be calculated that minimizes the **sum of squared residuals** (or "sum of squared errors").

Diving into the code

Dividing the code into steps for better understanding:

1. Load the dataset.
2. Visualize the data.
3. Training Simple Linear Regression Model.

Step1 : Load Dataset

In [1]:

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

In [2]:

```
dataset = pd.read_csv('data/Salary_Data.csv')
```

In [3]:

```
dataset.head()
```

Out[3]:

	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 [17]:

```
dataset.shape
```

Out[17]:

```
(30, 2)
```

Here, 'YearsExperience' is the independent variable and 'Salary' is the dependent variable which will be predicted based on the value of 'YearsExperience'.

In [4]:

```
X = dataset.drop(['Salary'],axis=True)
y = dataset['Salary']
```

In [5]:

```
#Now Split The Data
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 [6]:

```
X_train.shape,X_test.shape,y_train.shape,y_test.shape
```

Out[6]:

```
((24, 1), (6, 1), (24,), (6,))
```

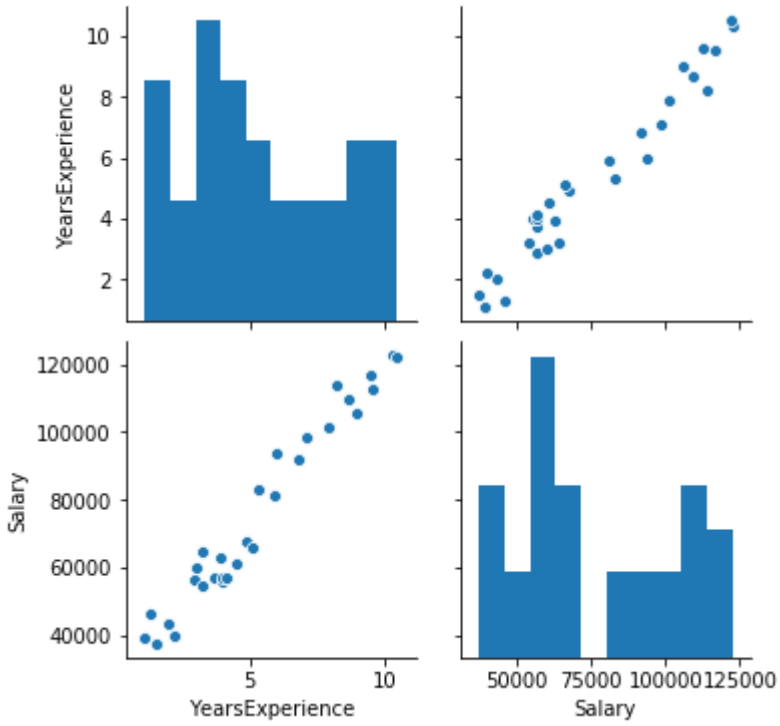
Step 2: Visualize the data.

In [7]:

```
# Visualize Whole Data set
sns.pairplot(dataset)
```

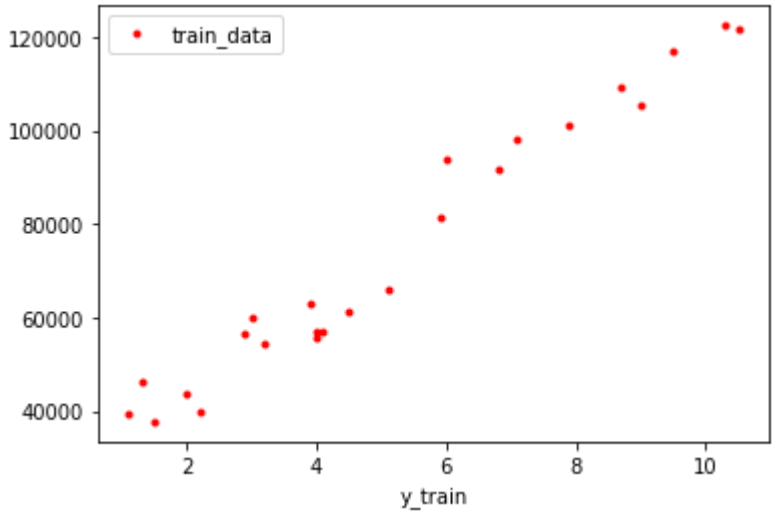
Out[7]:

```
<seaborn.axisgrid.PairGrid at 0x1459f6c6488>
```



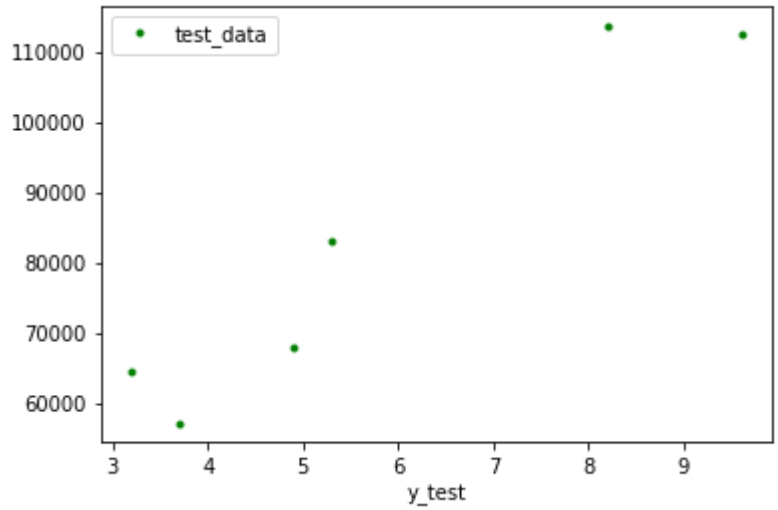
In [8]:

```
# Visualize Only Train data
plt.plot(X_train,y_train,'r.',label='train_data')
plt.xlabel('X_train')
plt.xlabel('y_train')
plt.legend()
plt.show()
```



In [9]:

```
# Visualize Only Test data
plt.plot(X_test,y_test,'g.',label='test_data')
plt.xlabel('X_test')
plt.xlabel('y_test')
plt.legend()
plt.show()
```



In The Above three graph clealr shows that the data is linearly deparable so here we use Linear Regression

Ok, now we hope the dataset is pretty clear by now. We will move to the next step now!

Step 3 :Training Simple Linear Regression Model.

In [10]:

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

Out[10]:

```
LinearRegression()
```

Predicting the Test set results

In [11]:

```
y_pred = regressor.predict(X_test).round(1)
```

In [12]:

```
calculation = pd.DataFrame(np.c_[y_test,y_pred], columns = ["Original Salary","Predict
 Salary"])
calculation
```

Out[12]:

	Original Salary	Predict Salary
0	112635.0	115790.2
1	67938.0	71498.3
2	113812.0	102596.9
3	83088.0	75267.8
4	64445.0	55477.8
5	57189.0	60189.7

Visualising the Training set results

In [16]:

```
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Train set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

Visualising the Training set results

In [14]:

```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

This is All About Simple Linear Regresion .