In [1]:

```
import numpy as np
import pandas as pd
```

In [2]:

```
data = pd.read_csv(r"C:\Users\ABHISHEK\Desktop\Salary_Data.csv")
```

In [3]:

```
data
```

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
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 [4]:

```
data.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]:

```
data.describe()
```

Out[5]:

	YearsExperience	Salary
count	30.000000	30.000000
mean	5.313333	76003.000000
std	2.837888	27414.429785
min	1.100000	37731.000000
25%	3.200000	56720.750000
50%	4.700000	65237.000000
75%	7.700000	100544.750000
max	10.500000	122391.000000

In [6]:

```
# to find missing values
data.isnull().any()
```

Out[6]:

```
YearsExperience    False
Salary             False
dtype: bool
```

In [7]:

```
data.dtypes
```

Out[7]:

```
YearsExperience    float64
Salary             float64
dtype: object
```

In [8]:

```
# Converting dataframe into numpy array
x = data.iloc[:, 0:1].values
y = data.iloc[:, 1:2].values
```

In [9]:

```
x
```

Out[9]:

```
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 [10]:

```
y
```

Out[10]:

```
array([[ 39343.],
       [ 46205.],
       [ 37731.],
       [ 43525.],
       [ 39891.],
       [ 56642.],
       [ 60150.],
       [ 54445.],
       [ 64445.],
       [ 57189.],
       [ 63218.],
       [ 55794.],
       [ 56957.],
       [ 57081.],
       [ 61111.],
       [ 67938.],
       [ 66029.],
       [ 83088.],
       [ 81363.],
       [ 93940.],
       [ 91738.],
       [ 98273.],
       [101302.],
       [113812.],
       [109431.],
       [105582.],
       [116969.],
       [112635.],
       [122391.],
       [121872.]])
```

In [11]:

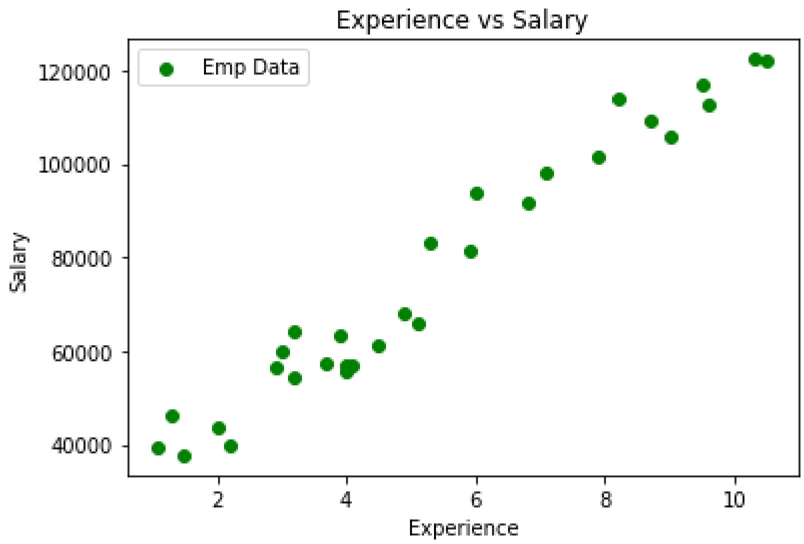
```
x[0:5], y[0:5]
```

Out[11]:

```
(array([[1.1],
        [1.3],
        [1.5],
        [2. ],
        [2.2]]), array([[39343.],
        [46205.],
        [37731.],
        [43525.],
        [39891.]]))
```

In [14]:

```
# Lets plot and check whether the x and y data is linear or not
import matplotlib.pyplot as plt
plt.scatter(x, y, label = 'Emp Data', color='g')
plt.title("Experience vs Salary")
plt.xlabel("Experience")
plt.ylabel("Salary")
plt.legend()
plt.show()
```



In [15]:

```
# now split the data into train and test sets
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=0)
```

In [16]:

```
x.shape, y.shape
```

Out[16]:

```
((30, 1), (30, 1))
```

In [17]:

```
#Automatically x_train,y_train rows will be 24 ie 80%
x_train.shape, y_train.shape
```

Out[17]:

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

In [18]:

```
#Lets x_test and y_test has 6 rows ie 20% of 30
x_test.shape, y_test.shape
```

Out[18]:

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

In [19]:

```
# Now train your model with train_data
from sklearn.linear_model import LinearRegression

regressor = LinearRegression()

regressor.fit(x_train, y_train)
```

Out[19]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [20]:

```
#as model got trained now we need to test the model
y_pred = regressor.predict(x_test)
```

In [21]:

```
y_pred
```

Out[21]:

```
array([[ 40748.96184072],
       [122699.62295594],
       [ 64961.65717022],
       [ 63099.14214487],
       [115249.56285456],
       [107799.50275317]])
```

In [22]:

```
y_test
```

Out[22]:

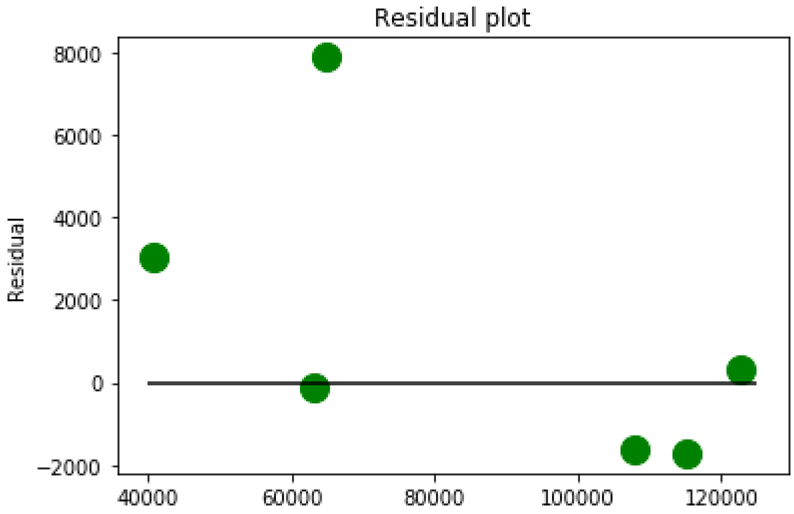
```
array([[ 37731.],
       [122391.],
       [ 57081.],
       [ 63218.],
       [116969.],
       [109431.]])
```

In [23]:

```
plt.scatter(y_pred, y_pred-y_test, c='g', s=200)
plt.hlines(y=0, xmin=40000, xmax=125000)
plt.title('Residual plot')
plt.ylabel('Residual')
```

Out[23]:

```
Text(0, 0.5, 'Residual')
```



In [24]:

```
y_pred-y_test
```

Out[24]:

```
array([[ 3017.96184072],
       [  308.62295594],
       [ 7880.65717022],
       [ -118.85785513],
       [-1719.43714544],
       [-1631.49724683]])
```

In [25]:

```
# Now check the accuracy
from sklearn.metrics import r2_score
accuracy = r2_score(y_test, y_pred)
```

In [26]:

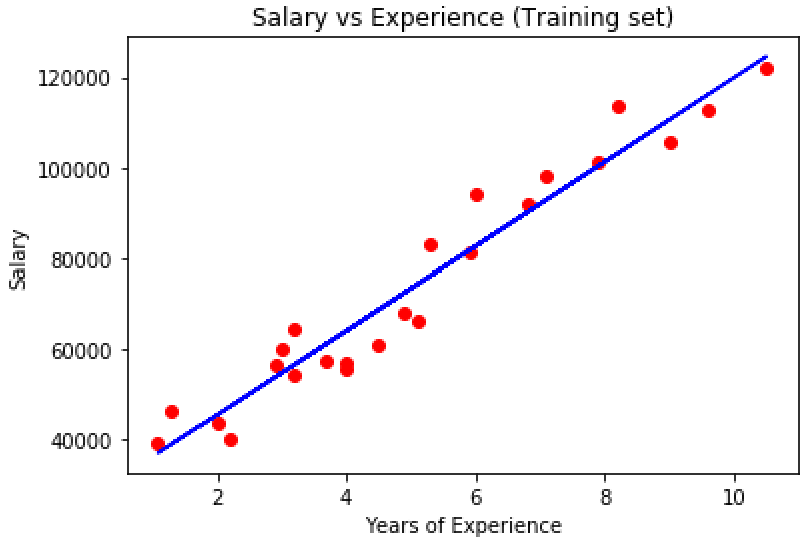
```
accuracy
```

Out[26]:

```
0.988169515729126
```

In [27]:

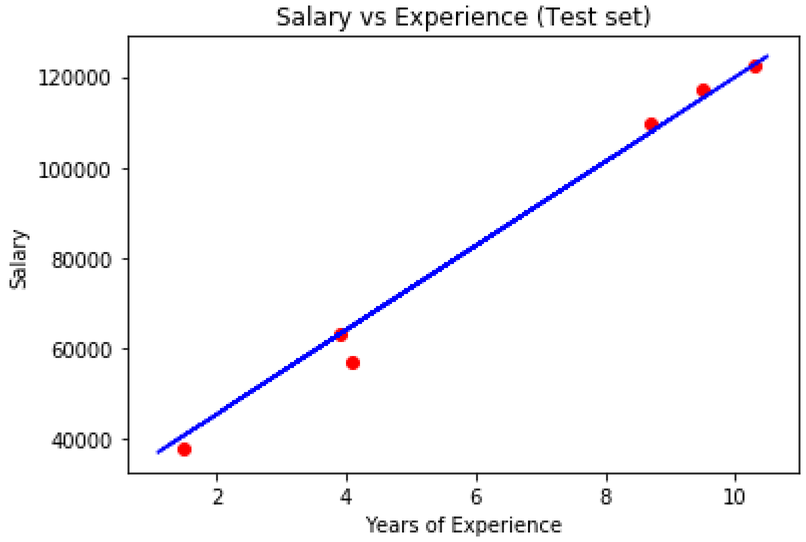
```
# Let’s start with the code for plotting the training set:
# We’ll plot the actual values (from the dataset) in red, and our model’s predictions in blue.
# This way, we’ll be able to easily differentiate the two.
plt.scatter(x_train, y_train, color = 'red')
plt.plot(x_train, regressor.predict(x_train), color = 'blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```



In [28]:

```

# Now let’s look at the plot for the test set:
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()
```

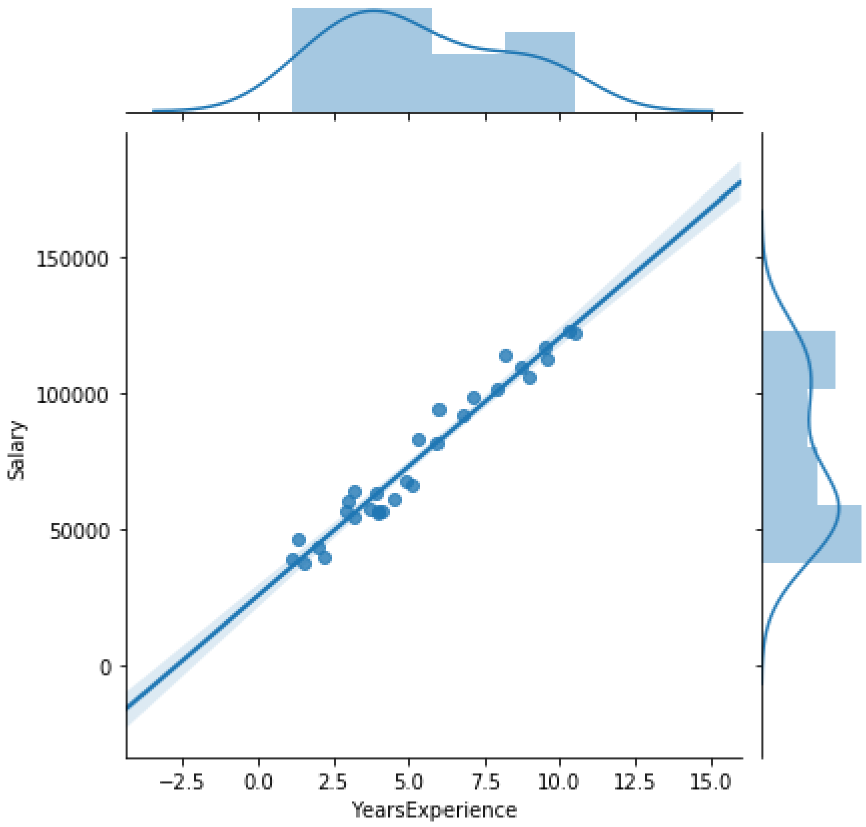


In [29]:

```
# To visualize distribution of data
# Regression line is drawn over the points
import seaborn as sns
sns.jointplot(x=data['YearsExperience'], y=data['Salary'], data=data, kind='reg')
```

Out[29]:

```
<seaborn.axisgrid.JointGrid at 0x1e19d88c7f0>
```



In [ ]:

```
```