In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import statsmodels.formula.api as smf
```

In [2]:

```
# import dataset
dataset=pd.read_csv('Salary_Data.csv')
dataset
```

Out[2]:

	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 [3]:

```
dataset.info()
```

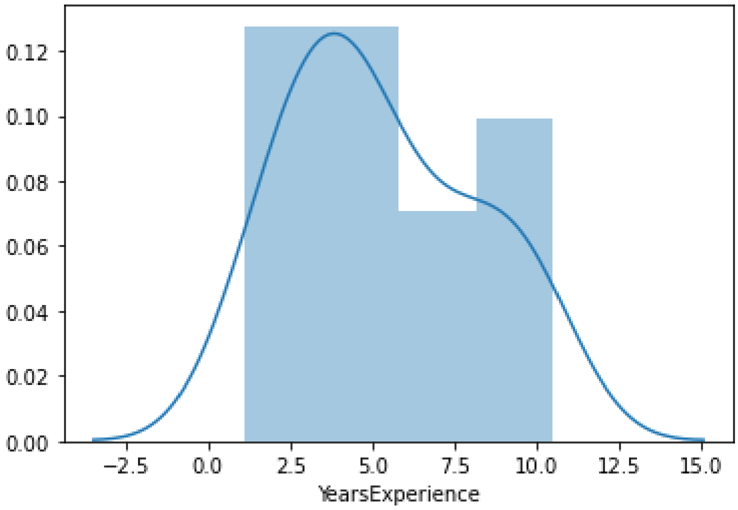
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     float64
dtypes: float64(2)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

In [4]:

```
sns.distplot(dataset['YearsExperience'])
```

Out[4]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20e9d2287c0>
```

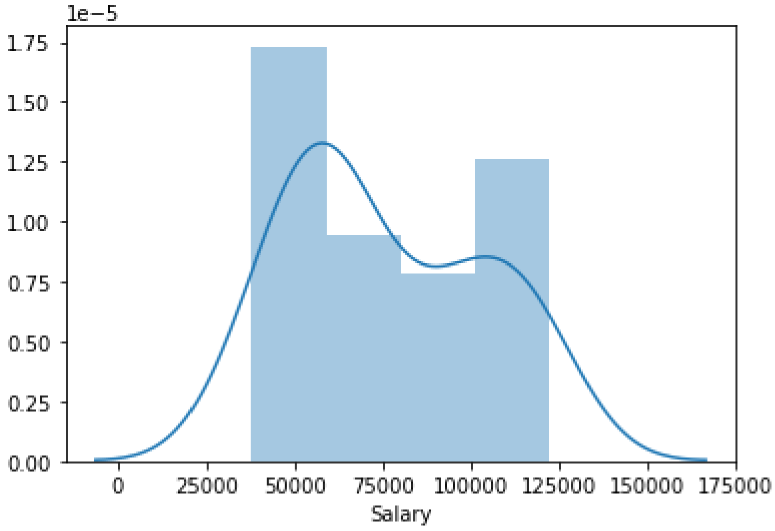


In [5]:

```
sns.distplot(dataset['Salary'])
```

Out[5]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20ea2a61520>
```



In [6]:

```
# correlation analysis
dataset.corr()
```

Out[6]:

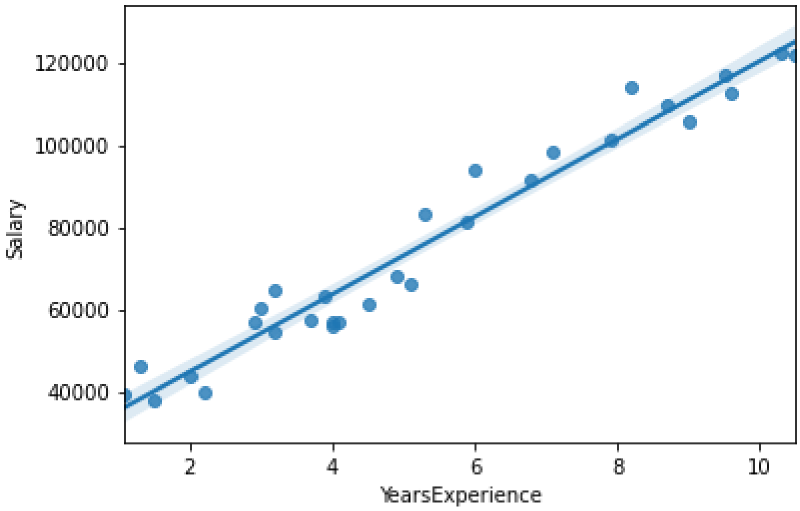
	YearsExperience	Salary
YearsExperience	1.000000	0.978242
Salary	0.978242	1.000000

In [7]:

```
sns.regplot(x=dataset['YearsExperience'],y=dataset['Salary'])
```

Out[7]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x20ea2ad6190>
```



In [8]:

```
# model building
model=smf.ols("Salary~YearsExperience",data=dataset).fit()
```

In [9]:

```
# model testing
# Finding Pvalues and tvalues
model.tvalues, model.pvalues
```

Out[9]:

```
(Intercept           11.346940
 YearsExperience     24.950094
 dtype: float64,
 Intercept           5.511950e-12
 YearsExperience     1.143068e-20
 dtype: float64)
```

In [10]:

```
# Finding Rsquared values
model.rsquared , model.rsquared_adj
```

Out[10]:

```
(0.9569566641435086, 0.9554194021486339)
```

In [11]:

```
# model prediction
# Manual prediction for say 3 Years Experience
Salary = (25792.200199) + (9449.962321)*(3)
Salary
```

Out[11]:

```
54142.087162
```

In [12]:

```
# Automatic Prediction for say 3 & 5 Years Experience
new_data=pd.Series([3,5])
new_data
```

Out[12]:

```
0    3
1    5
dtype: int64
```

In [13]:

```
data_pred=pd.DataFrame(new_data,columns=['YearsExperience'])
data_pred
```

Out[13]:

	YearsExperience
0	3
1	5

In [14]:

```
model.predict(data_pred)
```

Out[14]:

```
0    54142.087163
1    73042.011806
dtype: float64
```

In [ ]: