In [1]:
```
import numpy as np
import pandas as pd
import os
os.getcwd()
```

Out[1]:
```
'C:\\Users\\Gaurav\\Untitled Folder'
```

In [2]:
```
os.listdir()
```

Out[2]:
```
['.ipynb_checkpoints',
 '1 .LIST_TUPLE_SETS_DICT.ipynb',
 '10 .Visualization Practice.ipynb',
 '11.LINEAR REGRESSION.ipynb',
 '2  .STRING.ipynb',
 '3 .Untitled.ipynb',
 '4 .ERROR_MAPS.ipynb',
 '5 .NUMPY.ipynb',
 '6 .PANDAS.ipynb',
 '7 .PANDAS.ipynb',
 '8 .MATPLOTLIB.ipynb',
 '9  .SEABORN.ipynb',
 'a .LINEAR_LOGISTIC   IRIS REGRESSION.ipynb',
 'Churn-Data.csv',
 'iris.csv',
 'king.png',
 'pp-2018.csv',
 'salaryData.csv',
 'TITANIC',
 'winemag-data-130k-v2.csv',
 'yearsofexperince.csv']
```

Preprocessing

In [4]:
```
df=pd.read_csv('salaryData.csv')
df.head()
```

Out[4]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

Data Exploration

In [5]:
```
df.shape
```

Out[5]:
```
(31, 2)
```

In [6]:
```
df.info()
```

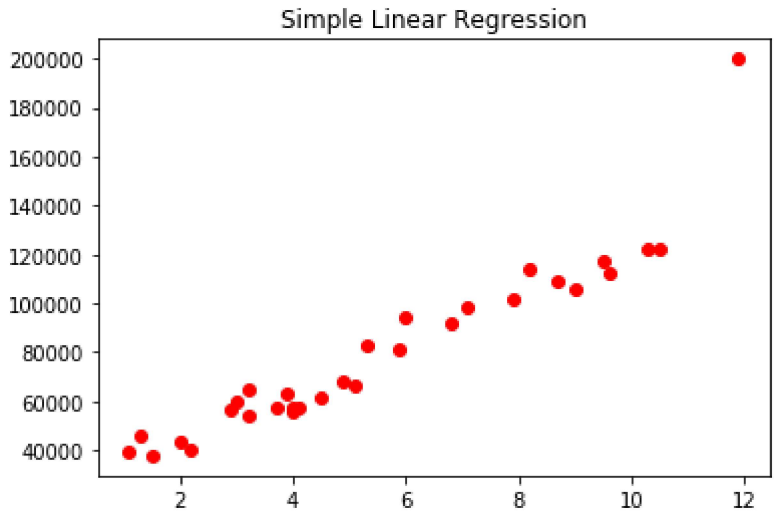
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31 entries, 0 to 30
Data columns (total 2 columns):
YearsExperience    31 non-null float64
Salary             31 non-null int64
dtypes: float64(1), int64(1)
memory usage: 576.0 bytes
```

In [7]:
```
df.describe()
```

Out[7]:

	YearsExperience	Salary
count	31.000000	31.000000
mean	5.525806	80002.903226
std	3.030618	34963.913711
min	1.100000	37731.000000
25%	3.200000	56799.500000
50%	4.900000	66029.000000
75%	8.050000	103442.000000
max	11.900000	200000.000000

In [12]:
```
import matplotlib.pyplot as plt
plt.scatter(df["YearsExperience"],df["Salary"],color='red')
plt.title("Simple Linear Regression")
plt.show()
```



Data Manipulation

In [9]:
```
emp=df.iloc[:,0].values

salary=df.iloc[:,1].values
```

Data Split

In [19]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(emp,salary,test_size=0.2,random_state
```

In [20]:
```
xtrain=pd.DataFrame(xtrain)
xtest=pd.DataFrame(xtest)
ytrain=pd.DataFrame(ytrain)
ytest=pd.DataFrame(ytest)
```

Model Making

In [13]:
```
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
```

In [25]:
```
regressor.fit(xtrain,ytrain)
```

Out[25]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [26]:
```
print(regressor.coef_)
beta1=regressor.coef_
```

```
[[11777.8286543]]
```

In [28]:
```
print(regressor.intercept_)
beta0=regressor.intercept_
```

```
[17116.49933095]
```

```
y=beta0+beta1*salary
```

In [31]:
```
pred=regressor.predict(xtrain)
```

Evaluate Model

In [32]:
```
from sklearn.metrics import mean_squared_error
print("     Root Mean Squared Error")
print("---------------------------------")
np.sqrt(mean_squared_error(pred,ytrain))
```

```
     Root Mean Squared Error
---------------------------------
```

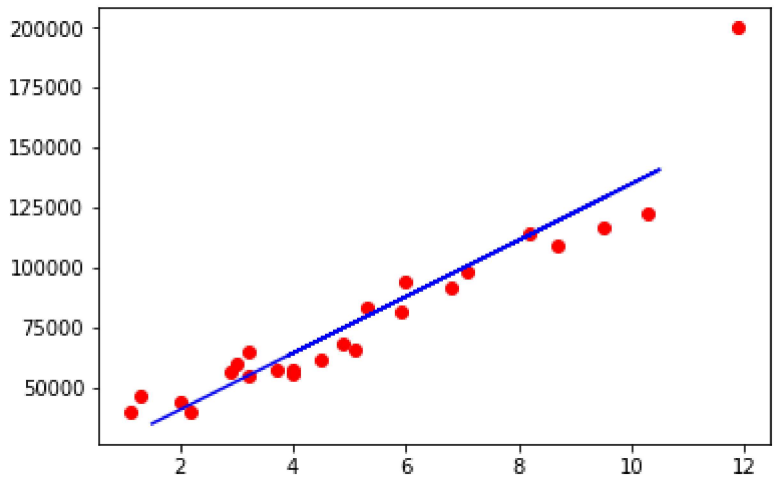
Out[32]: 11741.920913494481

In [33]:
```
from sklearn.metrics import mean_squared_error
print("     Root Mean Squared Error")
print("---------------------------------")
pred=regressor.predict(xtest)
np.sqrt(mean_squared_error(pred,ytest))
```

```
     Root Mean Squared Error
---------------------------------
```

Out[33]: 12703.131268414472

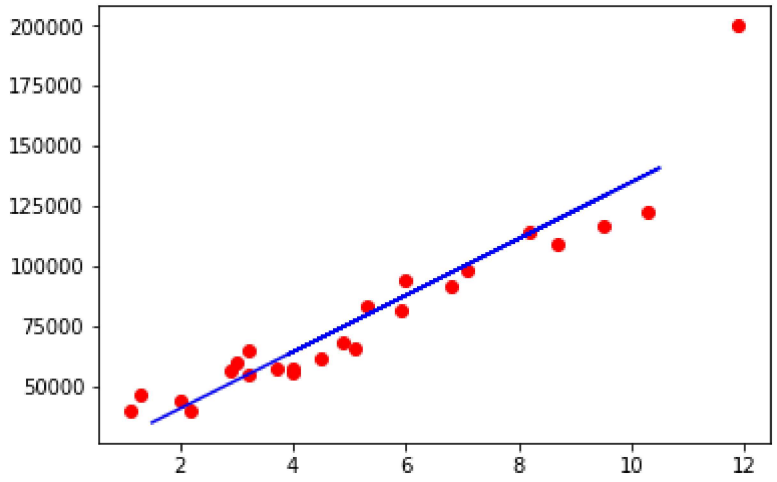
In [40]:
```
plt.scatter(xtrain,ytrain,color="red")
plt.plot(xtest,pred,color="blue")
plt.show()
```



Model Creation

In [41]:
```
def linear(xtrain,ytrain,xtest,ytest):
    global regressor
    regressor=LinearRegression()
    regressor.fit(xtrain,ytrain)
    pred=regressor.predict(xtest)
    plt.scatter(xtrain,ytrain,color="red")
    plt.plot(xtest,pred,color="blue")
    plt.show()
    return np.sqrt(mean_squared_error(pred,ytest))
```

In [42]:
```
linear(xtrain,ytrain,xtest,ytest)
```



Out[42]: 12703.131268414472

Model Improvement

In [49]:
```
df=df.iloc[:30,:]
```

In [50]:
```
df.head()
```

Out[50]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

In [52]:
```
df['YearsExperience']=np.round(df['YearsExperience'])
df.head()
```

Out[52]:

	YearsExperience	Salary
0	1.0	39343
1	1.0	46205
2	2.0	37731
3	2.0	43525
4	2.0	39891

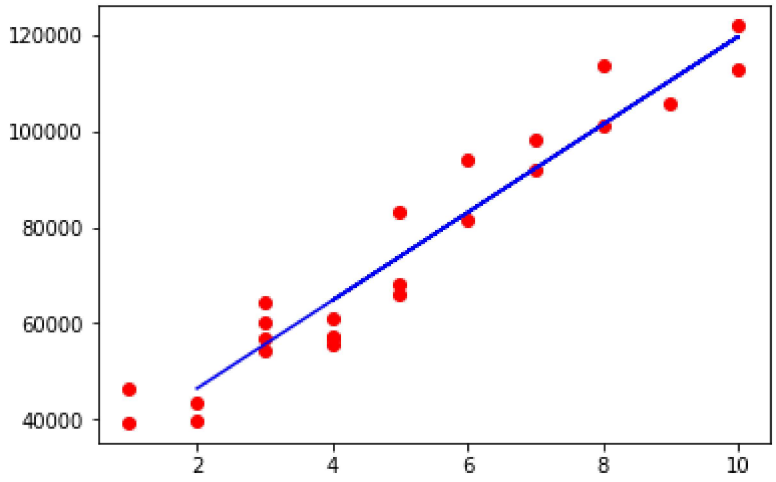
In [53]:
```
emp=df.iloc[:,0].values

salary=df.iloc[:,1].values
```

In [56]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(emp,salary,test_size=0.2,random_state

xtrain=pd.DataFrame(xtrain)
xtest=pd.DataFrame(xtest)
ytrain=pd.DataFrame(ytrain)
ytest=pd.DataFrame(ytest)
```

In [57]:
```
linear(xtrain,ytrain,xtest,ytest)
```



Out[57]: 5041.08726482858

In [ ]:
```
#We see that our model has imporved and RMSE came down a lot
```

Mode Deployed

In [58]:
```
data=pd.read_csv('yearsofexperince.csv')
data.head()
```

Out[58]:

	experience
0	7.6
1	4.0

In [60]:
```
data['experience']=np.round(data['experience'])
regressor.predict(data)
```

Out[60]:
```
array([[101337.58028455],
       [ 64735.75101626]])
```