Implementation of Univariate Linear Regression

AIM:

To implement univariate Linear Regression to fit a straight line using least squares.

Equipments Required:

1. Hardware – PCs
2. Anaconda – Python 3.7 Installation / Jupyter notebook

Algorithm

1. Get the independent variable X and dependent variable Y.
2. Calculate the mean of the X -values and the mean of the Y -values.
3. Find the slope m of the line of best fit using the formula.

$$m = \frac{\sum_{i=1}^{n}(x_i-\overline{X})(y_i-\overline{Y})}{\sum_{i=1}^{n}(x_i-\overline{X})^2}$$

4. Compute the y -intercept of the line by using the formula:

$$b = \overline{Y} - m\overline{X}$$

5. Use the slope m and the y -intercept to form the equation of the line. 6. Obtain the straight line equation Y=mX+b and plot the scatterplot.

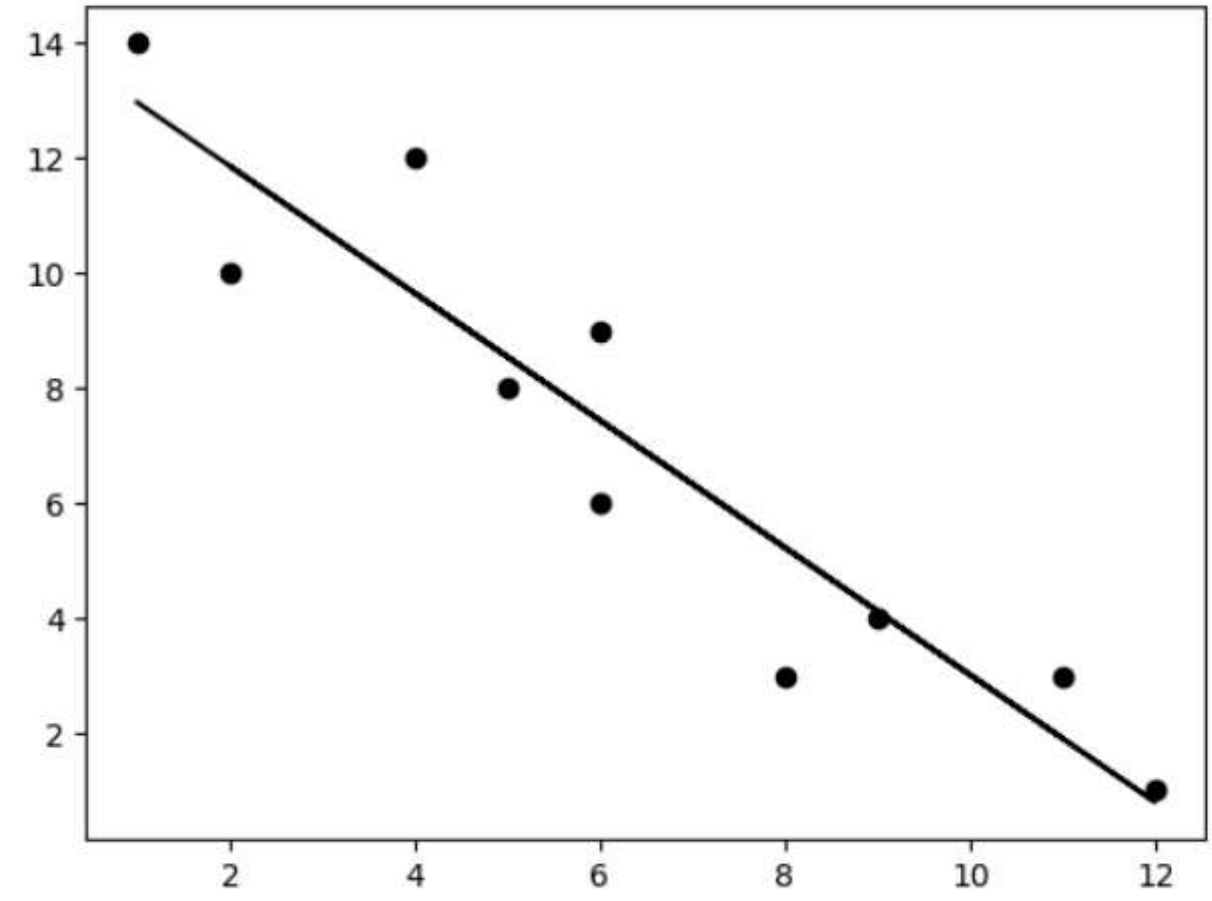
Program:

```
/*
Program to implement univariate Linear Regression to fit a straight line using least squ
Developed by: SHAHIN J
RegisterNumber:  212223040190

import numpy as np
import matplotlib.pyplot as plt

#input array - x and y

x = np.array(eval(input()))
y = np.array(eval(input()))

#Mean extraction

x_mean=np.mean(x)
y_mean=np.mean(y)

print(x_mean)
print(y_mean)

#formula Implementation

num, denom = 0, 0
```

```
for i in range(len(x)):
  num += ((x[i] - x_mean)*(y[i] - y_mean))
  denom += (x[i] - x_mean)**2

m = num/denom
b = y_mean - m * x_mean
print(m)
print(b)

y_predicted = m*x+b
y_predicted

plt.scatter(x,y,color='black')
plt.plot(x,y_predicted,color='black')
plt.show()

print(m*3+b)
*/
```

Output:

```
8,2,11,6,5,4,12,9,6,1
3,10,3,6,8,12,1,4,9,14
6.4
7.0
-1.1064189189189189
14.08108108108108
```




```
10.761824324324325
```

Result:

Thus the univariate Linear Regression was implemented to fit a straight line using least squares using python programming.