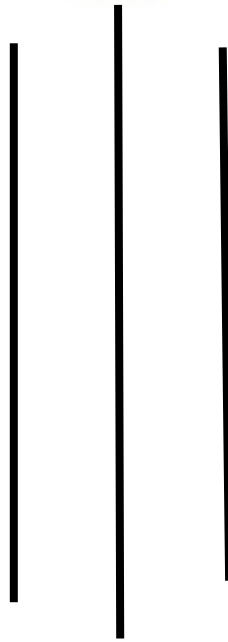


# NEPAL ENGINEERING COLLEGE

(AFFILIATED TO POKHARA UNIVERSITY)

Changunarayan, Bhaktapur



REPORT ON:  
Linear Regression

SUBMITTED BY:

NAME: Subash Khanal

CRN: 020-626

SUBMITTED TO:

Electrical and

Electronics

## **Experiment no:-7**

### **TITLE:-**

### **Linear Regression**

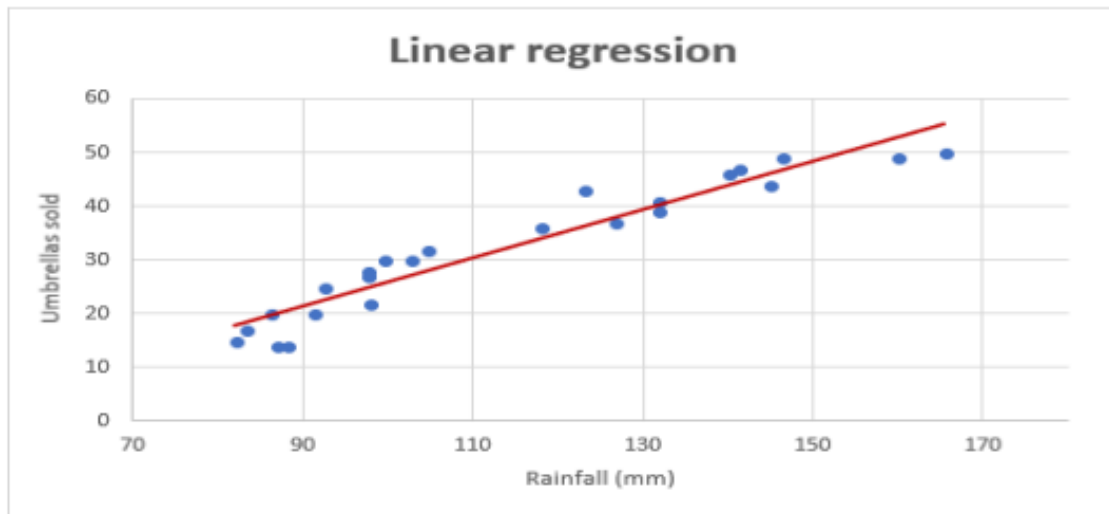
### **OBJECTIVE:-**

To fit the observations of two variables into a linear relationship between them and find slope and intercept using Matlab and C-programming

### **THEORY:-**

#### **Introduction:**

Linear regression is a technique used to model the relationships between observed variables. The idea behind simple linear regression is to "fit" the observations of two variables into a linear relationship between them. Graphically, the task is to draw the line that is "best-fitting" or "closest" to the points  $(x_i, y_i)$ , where  $x_i$  and  $y_i$  are observations of the two variables which are expected to depend linearly on each other. The linear relationship between  $x$  and  $y$  can be considered in the form of:  $y = a + bx$ . Our target is to find the value of  $a$  and  $b$  such that obtained line best fits the given data.



## Algorithm:

1. Start
2. Read number of data sets (n)
3. Read data x and y for i=1 to n
4. Compute  $\sum x$ ,  $\sum y$ ,  $\sum xy$ ,  $\sum x^2$ ,
5. Compute the value of a and b
6. Print out the equation
7. Stop

## Question:-

Implement above algorithm to fit a straight line to the following set of data:

x	1	2	3	4	5
y	3	4	5	6	8

## Using C-programming

Syntax:-

```
/* Linear Regression
prepared by:-
Subash khanal
crn: 020-626 */
#include <stdio.h>
#include<math.h>
#include<conio.h>
// function linearRegression
void linearRegression(float x[], float y[],
int n, float* slope, float* intercept) {
    float sum_x = 0, sum_y = 0, sum_xy = 0,
sum_xx = 0;
    for (int i = 0; i < n; i++) {
        sum_x += x[i];
        sum_y += y[i];
        sum_xy += x[i] * y[i];
        sum_xx += x[i] * x[i];
    }
    *slope = (n * sum_xy - sum_x * sum_y) / (n
* sum_xx - sum_x * sum_x);
    *intercept = (sum_y - *slope * sum_x) / n;
}

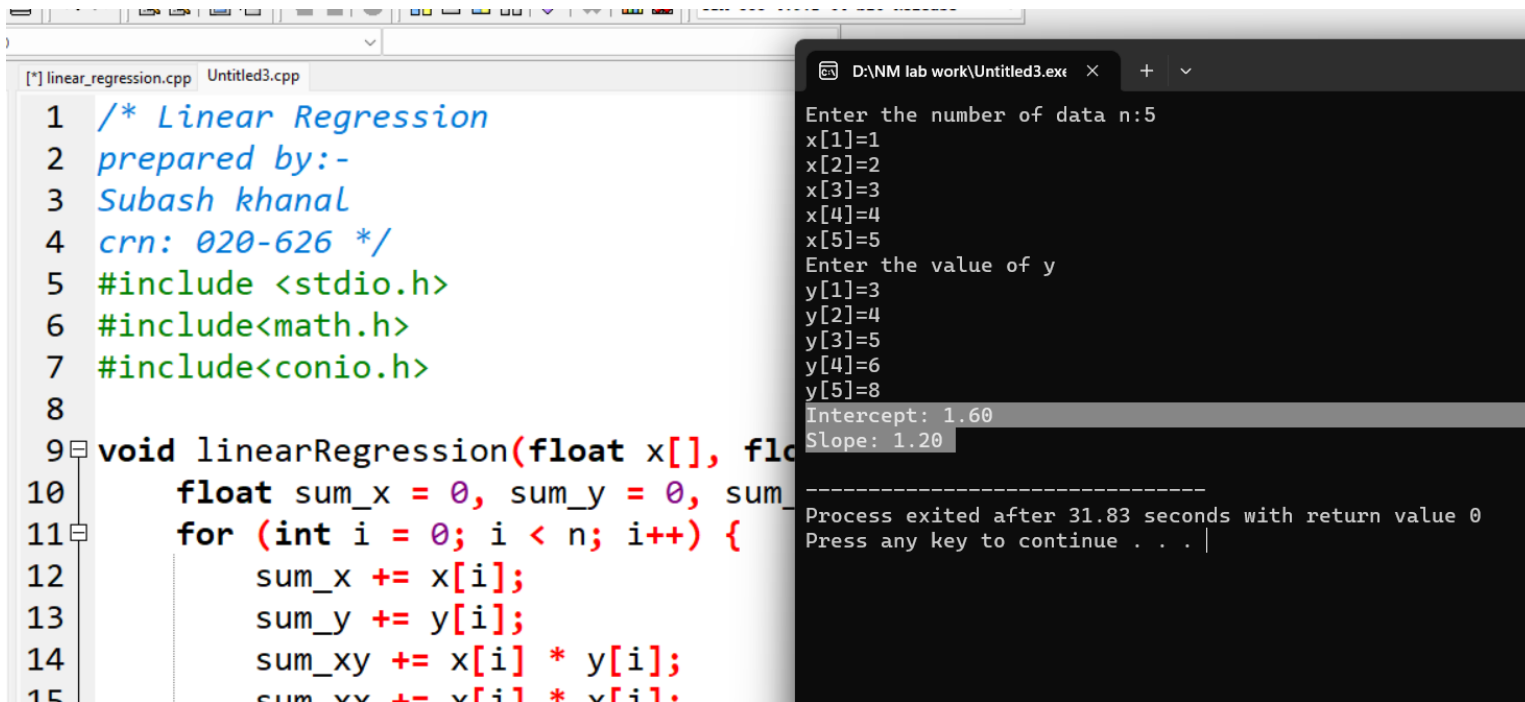
int main() {
    float x[100],y[100];
    int i,n;
```

```

    printf("Enter the number of data
n:"); //number of data in question
    scanf("%d",&n);
    for(i=0;i<n;i++) // for x
    {
        printf("x[%d]=",i+1);
        scanf("%f",&x[i]);
    }
    printf("Enter the value of y\n");
    for(i=0;i<n;i++)
    {
        printf("y[%d]=",i+1);
        scanf("%f",&y[i]);
    }
    float slope, intercept;
    linearRegression(x, y, n, &slope,
&intercept);
    //printing slope and intercept
    printf("Intercept: %.2f\n", intercept);
//intercept
    printf("Slope: %.2f\n", slope); //slope
    printf(" Linear Regression by:-\n Subash
khana1\n\n"); //Programmer details
    return 0;
    getch();
}

```

Output:-



The image shows a C++ IDE with two windows. The left window, titled 'linear\_regression.cpp', contains the following code:

```
1  /* Linear Regression
2  prepared by:-
3  Subash khalal
4  crn: 020-626 */
5  #include <stdio.h>
6  #include <math.h>
7  #include <conio.h>
8
9  void linearRegression(float x[], float y[], int n) {
10     float sum_x = 0, sum_y = 0, sum_xy = 0, sum_x2 = 0, sum_y2 = 0;
11     for (int i = 0; i < n; i++) {
12         sum_x += x[i];
13         sum_y += y[i];
14         sum_xy += x[i] * y[i];
15         sum_x2 += x[i] * x[i];
16         sum_y2 += y[i] * y[i];
17     }
```

The right window, titled 'D:\NM lab work\Untitled3.exe', shows the program's execution. It prompts the user to enter the number of data points (n=5) and the values of x and y. The output displays the calculated Intercept (1.60) and Slope (1.20), which are highlighted in the original image. The program then exits after 31.83 seconds.

```
Enter the number of data n:5
x[1]=1
x[2]=2
x[3]=3
x[4]=4
x[5]=5
Enter the value of y
y[1]=3
y[2]=4
y[3]=5
y[4]=6
y[5]=8
Intercept: 1.60
Slope: 1.20
-----
Process exited after 31.83 seconds with return value 0
Press any key to continue . . .
```

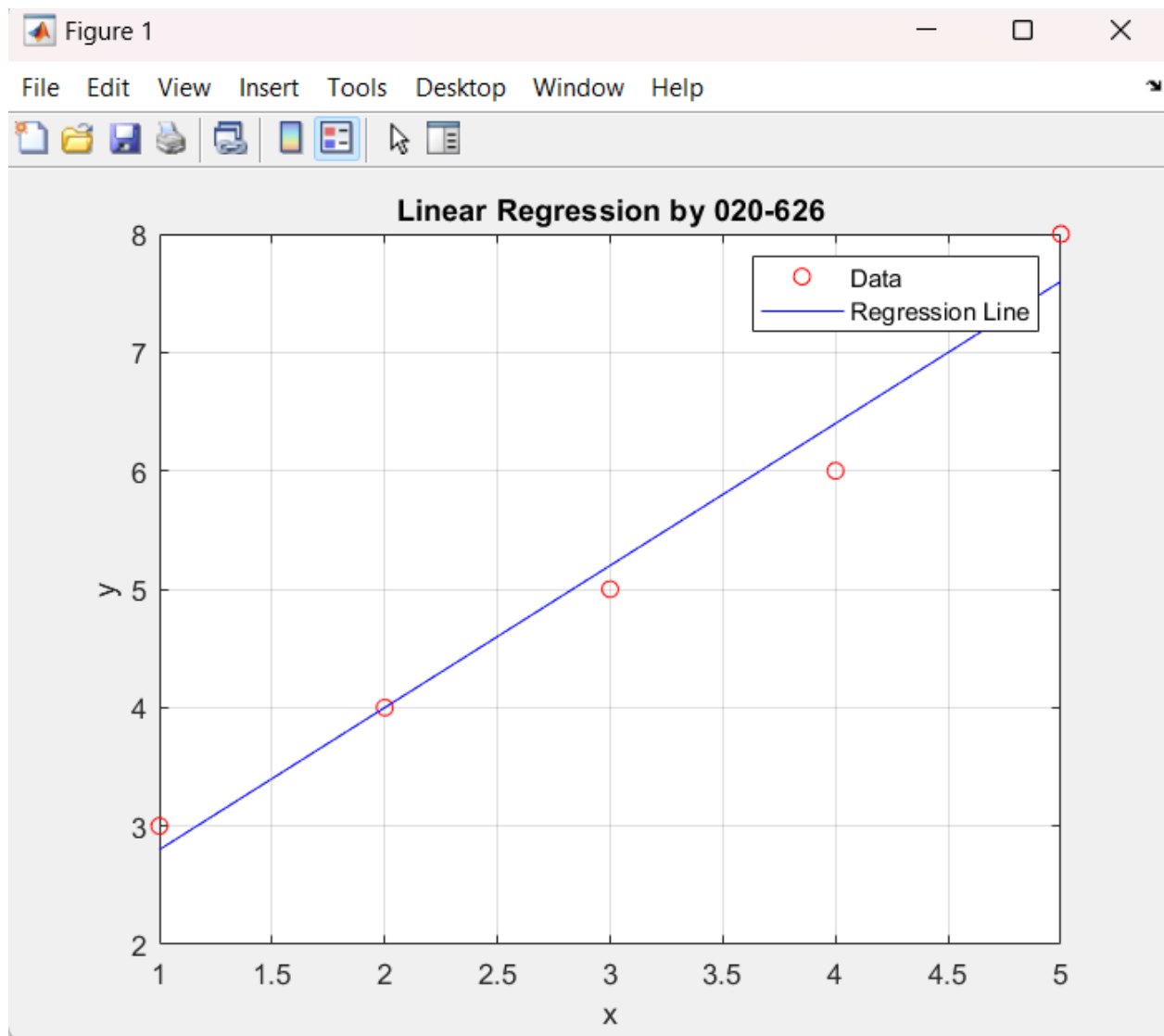
The value of Intercept and slope are found by using C-programming i.e. 1.60, 1.20 respectively which was highlighted on output screen.

## Using Matlab.

### Syntax:-

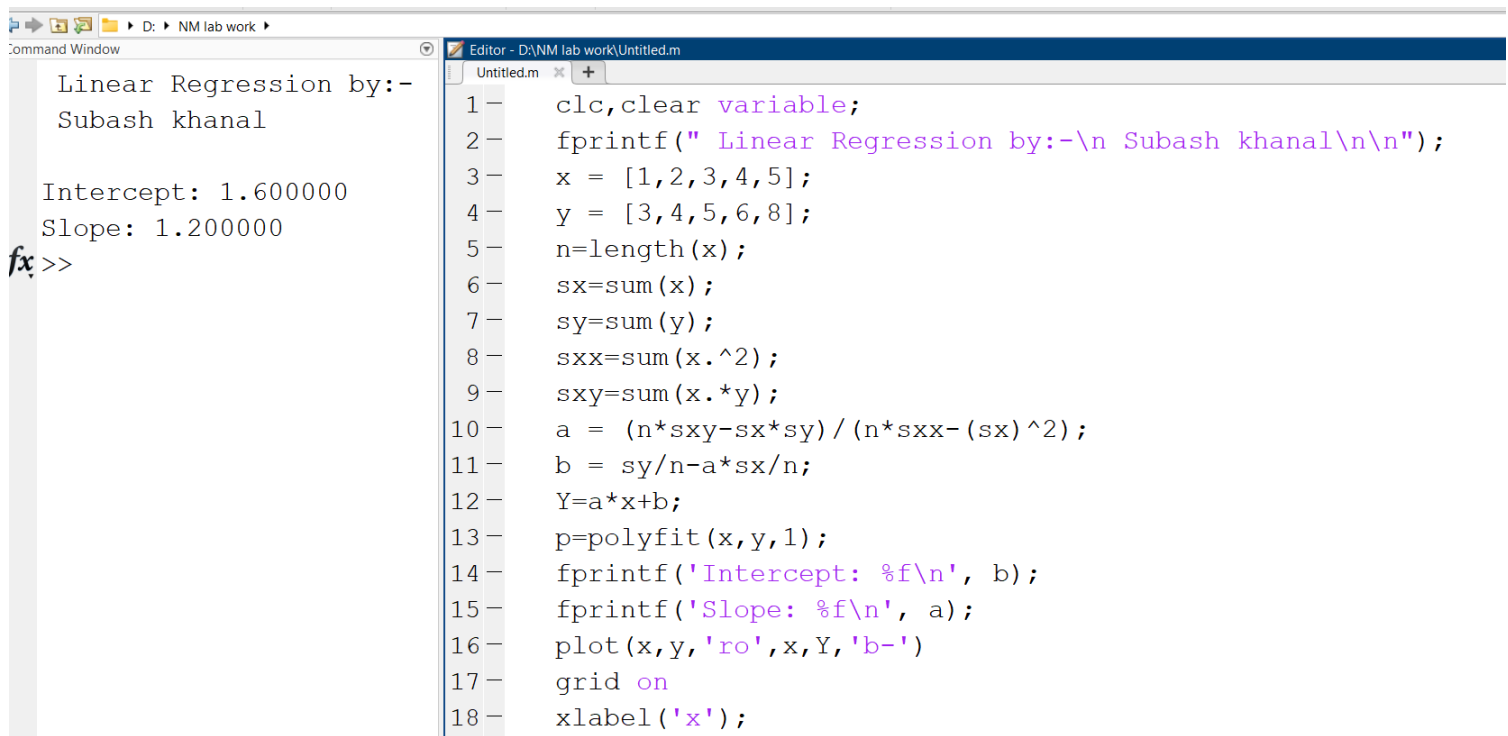
```
clc,clear variable;
%programmer details
fprintf(" Linear Regression by:-\n
Subash khanal\n\n");
x = [1,2,3,4,5];
y = [3,4,5,6,8];
n=length(x);
sx=sum(x);
sy=sum(y);
sxx=sum(x.^2);
sxy=sum(x.*y);
a = (n*sxy-sx*sy) / (n*sxx-(sx)^2);
b = sy/n-a*sx/n;
Y=a*x+b;
p=polyfit(x,y,1);
fprintf('Intercept: %f\n', b);
fprintf('Slope: %f\n', a);
plot(x,y,'ro',x,Y,'b-')
grid on
xlabel('x');
ylabel('y');
legend('Data', 'Regression Line');
title('Linear Regression by 020-
626');
```

## Graph from Matlab:-





## Output:-



The screenshot shows the MATLAB environment. The Command Window on the left displays the output of a linear regression calculation: "Linear Regression by:-", "Subash khanal", "Intercept: 1.600000", and "Slope: 1.200000". The Editor on the right shows the corresponding MATLAB script in a file named "Untitled.m". The script calculates the intercept and slope for a set of data points (x=[1,2,3,4,5], y=[3,4,5,6,8]) using the least squares method. The results are printed to the Command Window and plotted as a line with markers.

```
1- clc,clear variable;
2- fprintf(" Linear Regression by:-\n Subash khanal\n\n");
3- x = [1,2,3,4,5];
4- y = [3,4,5,6,8];
5- n=length(x);
6- sx=sum(x);
7- sy=sum(y);
8- sxx=sum(x.^2);
9- sxy=sum(x.*y);
10- a = (n*sxy-sx*sy)/(n*sxx-(sx)^2);
11- b = sy/n-a*sx/n;
12- Y=a*x+b;
13- p=polyfit(x,y,1);
14- fprintf('Intercept: %f\n', b);
15- fprintf('Slope: %f\n', a);
16- plot(x,y,'ro',x,Y,'b-')
17- grid on
18- xlabel('x');
```

Using Matlab the value of Intercept, Slope are found to 1.60, 1.20 respectively.

## Description:-

From above program of c-programming and Matlab it was clear that value of slope and intercept are same using linear Regression method.

## Conclusion:-

Hence, from above we can implement, fit and calculate the value of intercept and slope using the linear Regression method on Matlab and C-programming.