Experiment No. 2

Aim: Implementation of Linear Regression Algorithm.

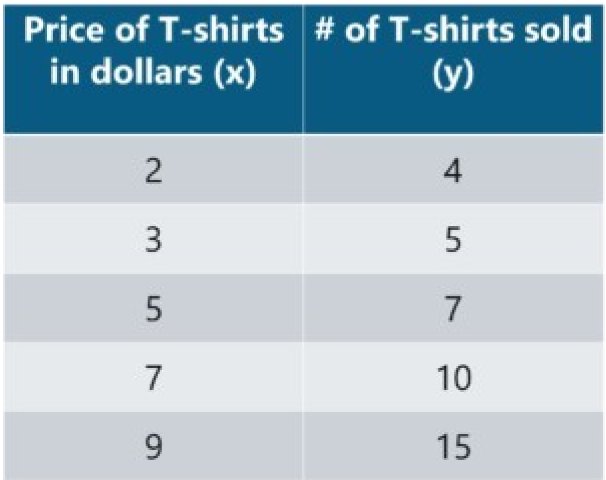
Objective: To implement Linear Regression in order to build a model that studies the relationship between an independent and dependent variable. The model will be evaluated by using least square regression method where RMSE and R-squared will be the model evaluation parameters..

Theory:

The least-squares method is a crucial statistical method that is practiced to find a regression line or a best-fit line for the given pattern. This method is described by an equation with specific parameters. The method of least squares is generously used in evaluation and regression. In regression analysis, this method is said to be a standard approach for the approximation of sets of equations having more equations than the number of unknowns. The method of least squares actually defines the solution for the minimization of the sum of squares of deviations or the errors in the result of each equation. Find the formula for sum of squares of errors, which help to find the variation in observed data. The least-squares method is often applied in data fitting.

Least Squares Regression Example

Tom who is the owner of a retail shop, found the price of different T-shirts vs the number of T-shirts sold at his shop over a period of one week.



Price of T-shirts in dollars (x)	# of T-shirts sold (y)
2	4
3	5
5	7
7	10
9	15

Let us use the concept of least squares regression to find the line of best fit for the above data.

Step 1: Calculate the slope 'm' by using the following formula:

$$m = \frac{n\sum xy - (\Sigma x)(\Sigma y)}{n\Sigma x^2 - (\Sigma x)^2}$$

After you substitute the respective values, m = 1.518 approximately.

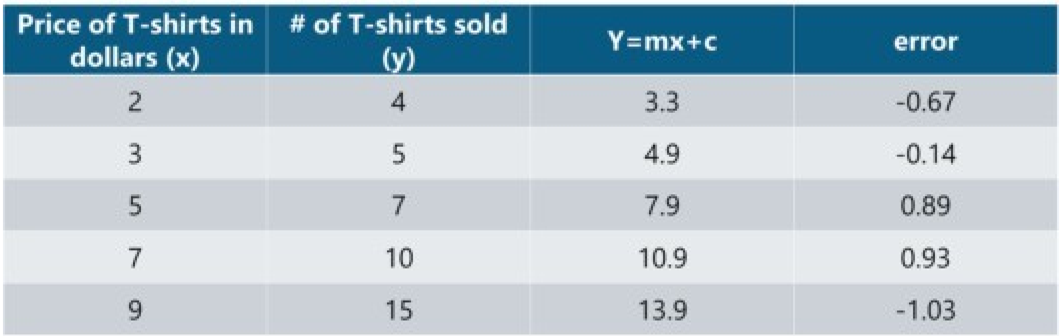
Step 2: Compute the y-intercept value

$$c = y - mx$$

After you substitute the respective values, c = 0.305 approximately.

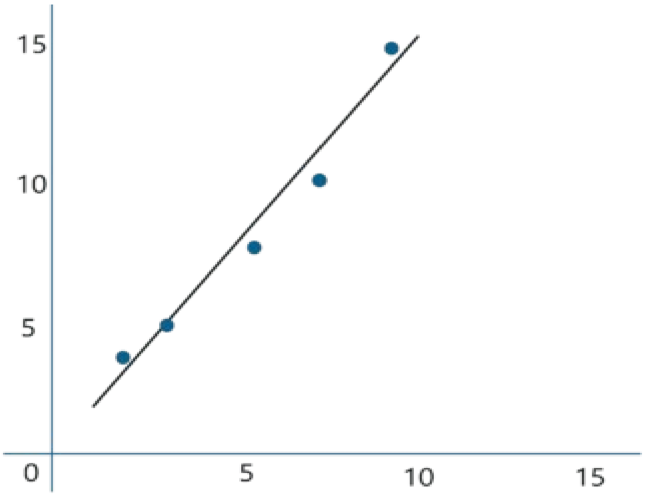
Step 3: Substitute the values in the final equation

$$y = mx + c$$



Price of T-shirts in dollars (x)	# of T-shirts sold (y)	Y=mx+c	error
2	4	3.3	-0.67
3	5	4.9	-0.14
5	7	7.9	0.89
7	10	10.9	0.93
9	15	13.9	-1.03

Let's construct a graph that represents the **y=mx + c** line of best fit:



Now Tom can use the above equation to estimate how many T-shirts of price $8 can he sell at the retail shop.

y = 1.518 x 8 + 0.305 = 12.45 T-shirts

This comes down to 13 T-shirts!

Dataset:

The data set contains the following variables:

- **Gender:** Male or female represented as binary variables
- **Age:** Age of an individual
- **Head size in cm^3:** An individuals head size in cm^3
- **Brain weight in grams:** The weight of an individual's brain measured in grams

These variables need to be analyzed in order to build a model that studies the relationship between the head size and brain weight of an individual.

Step 1: Import the required libraries

Step 2: Import the data set

Step 3: Assigning 'X' as independent variable and 'Y' as dependent variable

Step 4: Calculate the values of the slope and y-intercept

Step 5: Plotting the line of best fit

Step 6: Model Evaluation

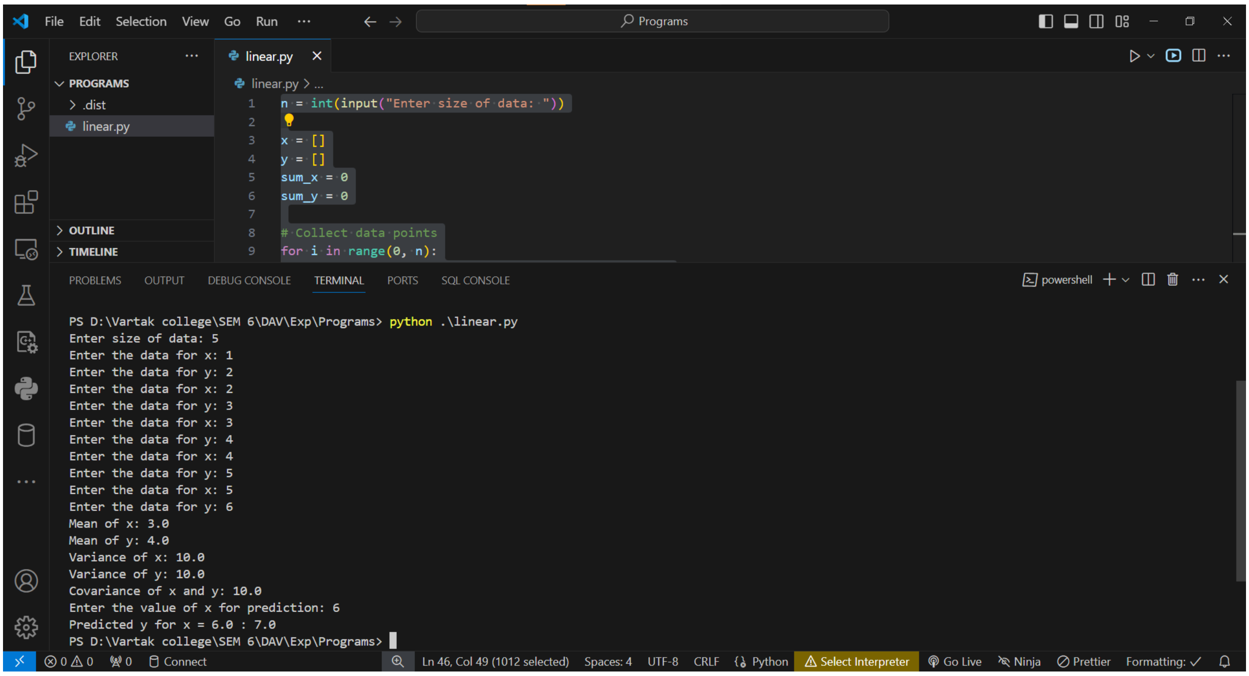
Implementation:

Linear Regression:

```
n = int(input("Enter size of data: "))
x = []
y = []
sum_x = 0
sum_y = 0
# Collect data points
for i in range(0, n):
    x.append(float(input("Enter the data for x: ")))
    y.append(float(input("Enter the data for y: ")))
    sum_x += x[i]
    sum_y += y[i]
# Calculate mean
x_bar = sum_x / n
y_bar = sum_y / n
print("Mean of x:", x_bar)
print("Mean of y:", y_bar)
sxx = 0
sxy = 0
```

```
syy = 0
# Calculate variances and covariance
for i in range(0, n):
    sxx += (x[i] - x_bar) ** 2
    syy += (y[i] - y_bar) ** 2
    sxy += (y[i] - y_bar) * (x[i] - x_bar)
print("Variance of x:", sxx)
print("Variance of y:", syy)
print("Covariance of x and y:", sxy)
# Calculate slope (b1) and intercept (b0)
b1 = sxy / sxx
b0 = y_bar - (b1 * x_bar)
# Specific value of x for prediction
x_new = float(input("Enter the value of x for prediction: "))
# Calculate predicted y for the specific value of x
y_pred = b0 + b1 * x_new
print("Predicted y for x =", x_new, ":", y_pred)
```

Output:



```
n = int(input("Enter size of data: "))

x = []
y = []
sum_x = 0
sum_y = 0

# Collect data points
for i in range(0, n):
```

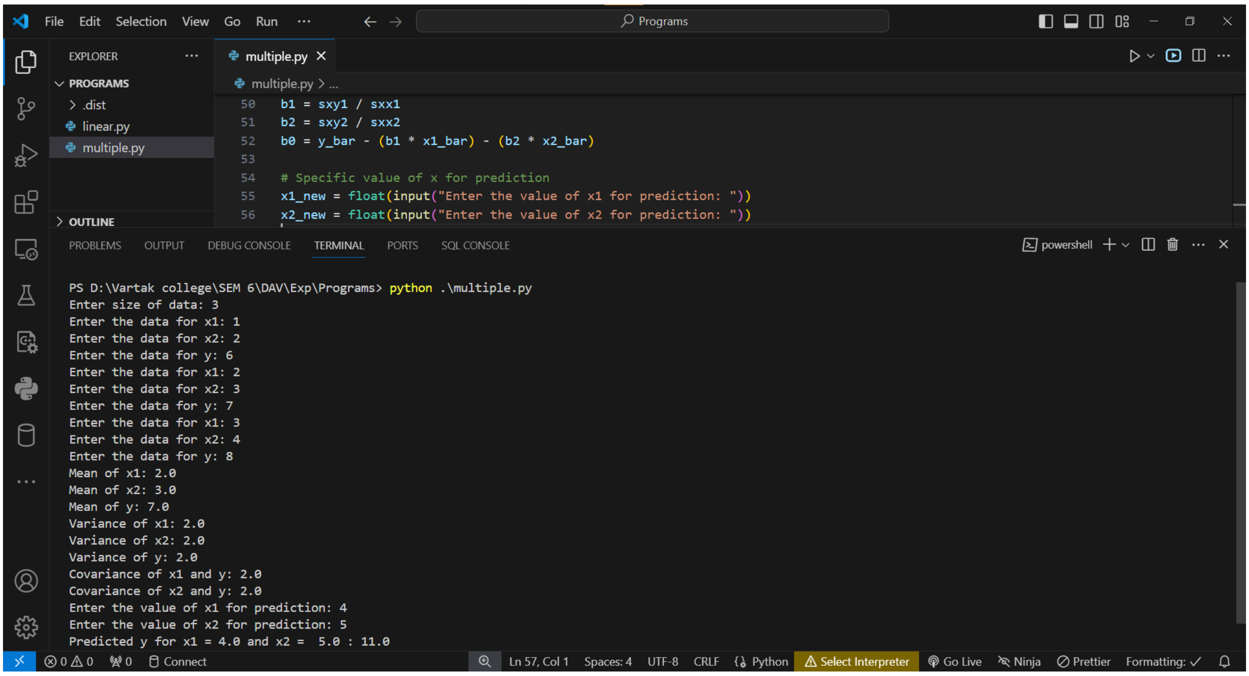
```
PS D:\Vartak college\SEM 6\DAV\Exp\Programs> python .\linear.py
Enter size of data: 5
Enter the data for x: 1
Enter the data for y: 2
Enter the data for x: 2
Enter the data for y: 3
Enter the data for x: 3
Enter the data for y: 4
Enter the data for x: 4
Enter the data for y: 5
Enter the data for x: 5
Enter the data for y: 6
Mean of x: 3.0
Mean of y: 4.0
Variance of x: 10.0
Variance of y: 10.0
Covariance of x and y: 10.0
Enter the value of x for prediction: 6
Predicted y for x = 6.0 : 7.0
PS D:\Vartak college\SEM 6\DAV\Exp\Programs>
```

Multiple Regression:

```
n = int(input("Enter size of data: "))
x1 = []
x2 = []
y = []
```

```
sum_x1 = 0
sum_x2 = 0
sum_y = 0
# Collect data points
for i in range(0, n):
    x1.append(float(input("Enter the data for x1: ")))
    x2.append(float(input("Enter the data for x2: ")))
    y.append(float(input("Enter the data for y: ")))
    sum_x1 += x1[i]
    sum_x2 += x2[i]
    sum_y += y[i]
# Calculate mean
x1_bar = sum_x1 / n
x2_bar = sum_x2 / n
y_bar = sum_y / n
print("Mean of x1:", x1_bar)
print("Mean of x2:", x2_bar)
print("Mean of y:", y_bar)
sxx1 = 0
sxx2 = 0
sxy1 = 0
sxy2 = 0
syy = 0
# Calculate variances and covariance
for i in range(0, n):
    sxx1 += (x1[i] - x1_bar) ** 2
    sxx2 += (x2[i] - x2_bar) ** 2
    syy += (y[i] - y_bar) ** 2
    sxy1 += (y[i] - y_bar) * (x1[i] - x1_bar)
    sxy2 += (y[i] - y_bar) * (x2[i] - x2_bar)
print("Variance of x1:", sxx1)
print("Variance of x2:", sxx2)
print("Variance of y:", syy)
print("Covariance of x1 and y:", sxy1)
print("Covariance of x2 and y:", sxy2)
# Calculate slope (b1) and intercept (b0)
b1 = sxy1 / sxx1
b2 = sxy2 / sxx2
b0 = y_bar - (b1 * x1_bar) - (b2 * x2_bar)
# Specific value of x for prediction
x1_new = float(input("Enter the value of x1 for prediction: "))
x2_new = float(input("Enter the value of x2 for prediction: "))
# Calculate predicted y for the specific value of x
y_pred = b0 + (b1 * x1_new) + (b2 * x2_new)
print("Predicted y for x1 =", x1_new, "and x2 = ", x2_new, ":", y_pred)
```

Output:


File Edit Selection View Go Run ···
Programs
EXPLORER
PROGRAMS
.dist
linear.py
multiple.py
OUTLINE
multiple.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL CONSOLE
powershell
Ln 57, Col 1 Spaces: 4 UTF-8 CRLF Python Select Interpreter Go Live Ninja Prettier Formatting: ✓
Connect

```
b1 = sxy1 / sxx1
b2 = sxy2 / sxx2
b0 = y_bar - (b1 * x1_bar) - (b2 * x2_bar)

# Specific value of x for prediction
x1_new = float(input("Enter the value of x1 for prediction: "))
x2_new = float(input("Enter the value of x2 for prediction: "))
```

```
PS D:\Vartak college\SEM 6\DAV\Exp\Programs> python .\multiple.py
Enter size of data: 3
Enter the data for x1: 1
Enter the data for x2: 2
Enter the data for y: 6
Enter the data for x1: 2
Enter the data for x2: 3
Enter the data for y: 7
Enter the data for x1: 3
Enter the data for x2: 4
Enter the data for y: 8
Mean of x1: 2.0
Mean of x2: 3.0
Mean of y: 7.0
Variance of x1: 2.0
Variance of x2: 2.0
Variance of y: 2.0
Covariance of x1 and y: 2.0
Covariance of x2 and y: 2.0
Enter the value of x1 for prediction: 4
Enter the value of x2 for prediction: 5
Predicted y for x1 = 4.0 and x2 =  5.0 : 11.0
```

Conclusion:

The implementation of Linear Regression and Multiple Regression in Python provides a practical demonstration of the least squares method for regression analysis. This method is widely used to find the best-fit line or plane that minimizes the sum of the squares of the residuals between observed and predicted values. By calculating the slope and intercept parameters, the regression models can make predictions based on the relationship between independent and dependent variables.

Comment on the Least Square Method used for regression:

The least squares method is a fundamental statistical technique for fitting a regression line to a set of data points. It aims to minimize the sum of the squared differences between the observed and predicted values, effectively capturing the overall trend in the data. In the context of linear regression, the method calculates the coefficients of the regression equation by minimizing the sum of the squared residuals.