MP-2 LAB ASSIGNMENT-2

Name: Aditya Pachar Date:19-03-2025

Roll no:245674001

Course:BSc Physics Hons

Question 1

Que 1. **Linear Fitting**

Fit a straight line to the x and y values given in the following table:

X	1	2	3	4	5	6	7
Y	0.5	2.5	2.0	4.0	3.5	6.0	5.5

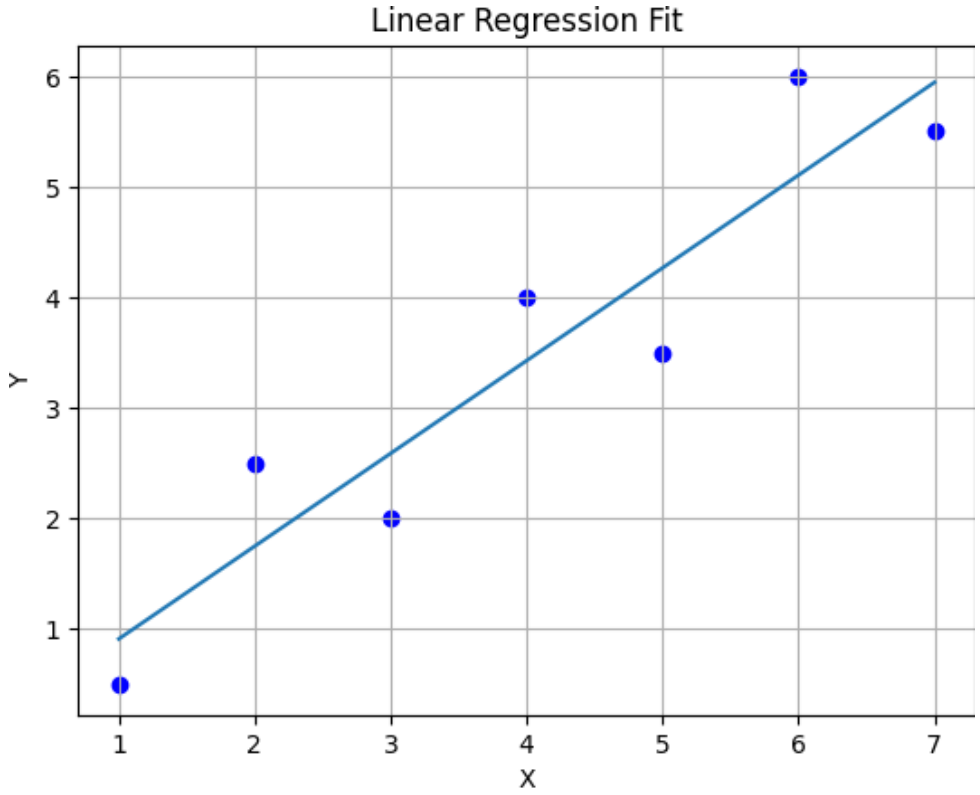
a) Display the equation of the regression line (Fitted line).
b) Plot both the fitted line and the given data points.
c) Estimate slope, intercept and correlation coefficient.
d) Show all values of x_i, y_i, x_i^2 and xy

```
Input: import numpy as np
Import pandas as pd
Import matplotlib.pyplot as plt

Elements = int(input("How many elements do you want to enter? "))
User_list_x=[]
For I in range (0,elements):
    X=float(input("Enter the x"+str(i+1)+ " coordinates:"))
    User_list_x.append(x)
Print(user_list_x)

User_list_y=[]
For I in range (0,elements):
    Y=float(input("Enter the y"+str(i+1)+ " coordinates:"))
    User_list_y.append(y)
Print(user_list_y)
User_list_x=np.array(user_list_x)
User_list_y=np.array(user_list_y)
X_sum=np.sum(user_list_x)
X_sq=np.multiply(user_list_x,user_list_x)
```

```
23    X_sqsum=np.sum(x_sq)
24    X_mean=np.mean(user_list_x)
25
26    Y_sum=np.sum(user_list_y)
27    Y_mean=np.mean(user_list_y)
28
29    Xi_X_yi=np.multiply(user_list_x,user_list_y)
30    Xi_X_yi_sum=np.sum(xi_X_yi)
31
32    A1=(elements*(xi_X_yi_sum)-(x_sum*y_sum))/((elements*x_sqsum)-
(x_sum)**2)
33    Ao=y_mean-a1*x_mean
34
35    Sy=np.sum((user_list_y-y_mean)**2)
36    S=np.sum((user_list_y-ao-a1*user_list_x))
37
38    Cc=((Sy-S)/Sy)**0.5
39    A=pd.DataFrame({'x':user_list_x,'y':user_list_y,'x**2':x_sq,'xi
X yi':xi_X_yi})
40    Print(a)
41    Print("Slope of the regression line",a1)
42    Print("Intercept of the regression line",ao)
43    Print("Correlation coefficient of the regression line",cc)
44    X=user_list_x
45    Y=a1*X+ao
46    Plt.plot(X, Y)
47    Plt.xlabel("X")
48    Plt.ylabel("Y")
49    Plt.scatter(X, user_list_y, color='blue', label="Data Points")
50    Plt.title("Linear Regression Fit")
51    Plt.grid()
52    Plt.show()
53    Output:
54    How many elements do you want to enter?  7
55    Enter the x1 coordinates: 1
56    Enter the x2 coordinates: 2
57    Enter the x3 coordinates: 3
58    Enter the x4 coordinates: 4
59    Enter the x5 coordinates: 5
60    Enter the x6 coordinates: 6
61    Enter the x7 coordinates: 7
62    [1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0]
63    Enter the y1 coordinates: .5
64    Enter the y2 coordinates: 2.5
65    Enter the y3 coordinates: 2
66    Enter the y4 coordinates: 4
67    Enter the y5 coordinates: 3.5
68    Enter the y6 coordinates: 6
```

```
Enter the y7 coordinates: 5.5
[0.5, 2.5, 2.0, 4.0, 3.5, 6.0, 5.5]
     X    y  x**2  xi X yi
0  1.0  0.5   1.0      0.5
1  2.0  2.5   4.0      5.0
2  3.0  2.0   9.0      6.0
3  4.0  4.0  16.0     16.0
4  5.0  3.5  25.0     17.5
5  6.0  6.0  36.0     36.0
6  7.0  5.5  49.0     38.5
Slope of the regression line 0.8392857142857143
Intercept of the regression line 0.07142857142857117
Correlation coefficient of the regression line 1.0
Trinket_plot.png
```

trinket plot.png



Question 2

Que 2. **Exponential**

Using the method of the least square fitting, find constant a and b such that the function $y=ae^{bx}$ fits the following data

X	1	3	5	7	9
Y	2.473	6.722	18.274	49.673	135.026

a) Display the equation of the regression line (Fitted line).
b) Plot both the fitted line and the given data points.
c) Show all values of x_i, y_i, x_i^2 and xy etc

```
#QUESTION2 import numpy as np
import matplotlib.pyplot as pl
print("Equation given: y=ae^bx\n lny=lna+bx \n Y=lny,A=lna\n Y=A+bx") X=np.array([1,3,5,7,9])
y=np.array([2.473,6.722,18.274,49.673,135.026])
#log values of y
Y=np.array([0.905,1.905,2.905,3.905,4.905])
x_sum=sum(X)
y_sum=sum(Y) xy=X*Y
xy_sum=sum(xy)
x_square=X**2
x2_sum=sum(x_square)
#n is the number of items entered n=5
#finding the value of b
b=((n*xy_sum)-(x_sum*y_sum))/((n*x2_sum)-(x_sum)**2)
#finding the value of A
A=((y_sum)-(b*x_sum))/n
print("Exponential regression fit:y=1.5e^0.5x")

# Given data points
X  = np.array([1, 3, 5, 7, 9])
Y  = np.array([2.473, 6.722, 18.274, 49.673, 135.026])

# Equation from manual calculation: y = 1.5 * e^(0.5x) a = 1.5
b = 0.5

# Generate fitted curve
X_line = np.linspace(min(X), max(X), 100)
Y_line = a * np.exp(b * X_line)

# Plot given data points
pl.scatter(X, Y, color='red', label='Given Data')
```

```
# Plot fitted exponential curve
pl.plot(X_line, Y_line, color='blue', label=r'Fitted Curve: $y = 1.5e^{0.5x}$')

# Labels and title pl.xlabel("X
values") pl.ylabel("Y values")
pl.title("Exponential Regression Fit")
pl.legend() pl.grid()
pl.show()
```

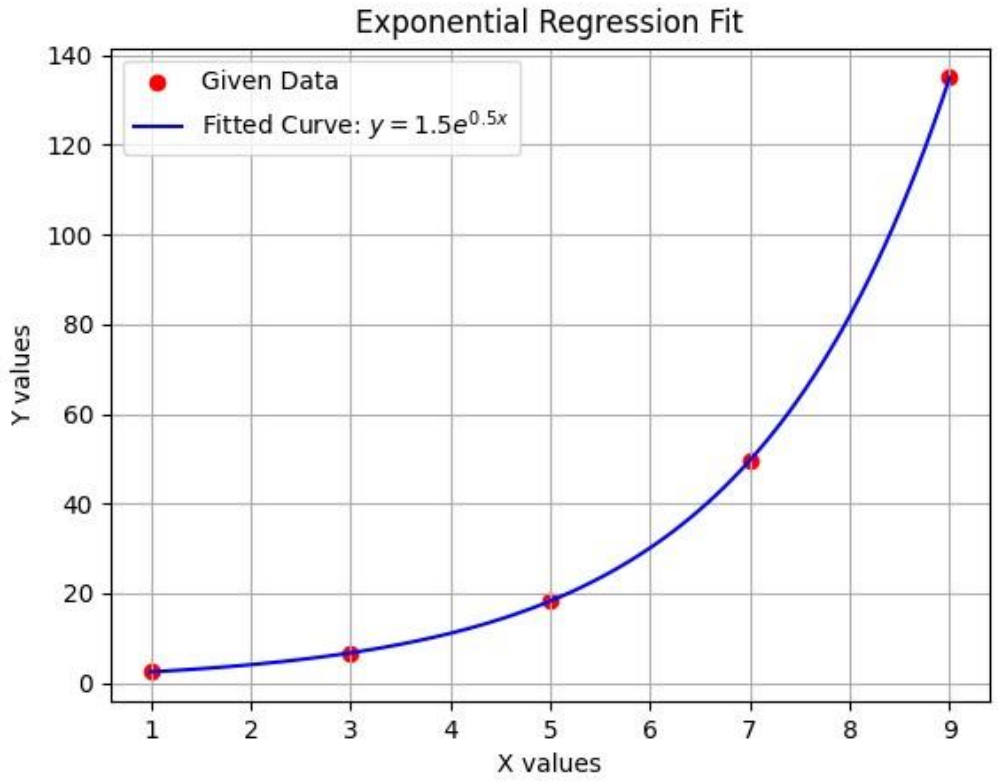
Output:

Equation given: y=ae^bx

lny=lna+bx Y=lny,A=lna

Y=A+bx

Exponential regression fit:y=1.5e^0.5x



Que 3. **Power Law Fitting**

(a) Fit a power equation $y = ax^b$ to the x and y values given in the following table.

x	1	2	3	4	5
y	0.5	1.7	3.4	5.7	8.4

(b) Display the computed values of the coefficients a and b.

(c) Plot both the fitted curve and the given data points.

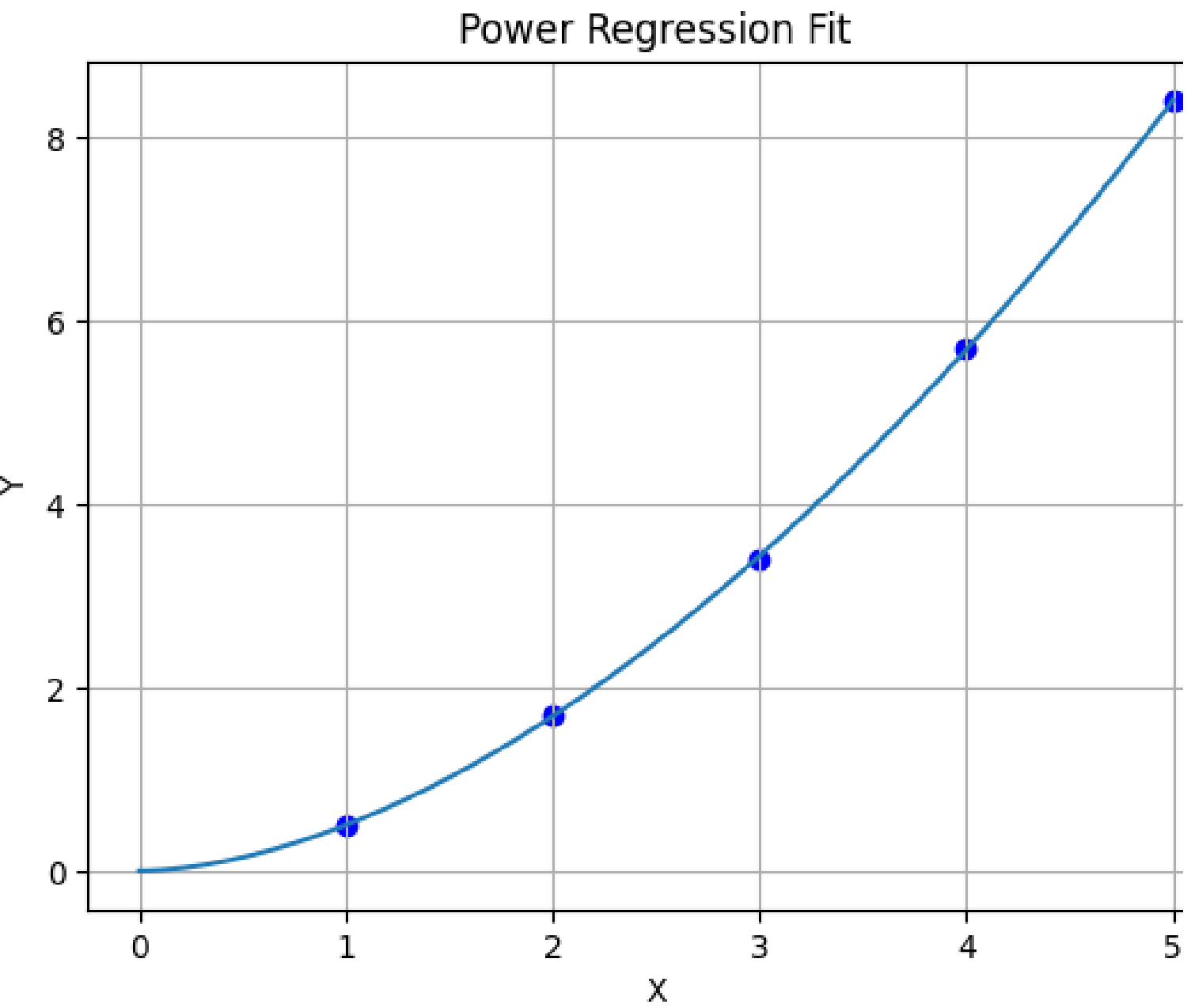
```
#Question3 import numpy as np
Import pandas as pd
Import matplotlib.pyplot as plt

Elements = int(input("How many elements do you want to enter? "))
User_list_x=[]
For I in range (0,elements):
   X=float(input("Enter the x"+str(i+1)+ " coordinates:"))
   User_list_x.append(x)
Print(user_list_x)
```

```
Print()
User_list_y=[]
For I in range (0,elements):
   Y=float(input("Enter the y"+str(i+1)+ " coordinates:"))
   User_list_y.append(y)
Print(user_list_y)
Print()
User_list_x=np.array(user_list_x)
User_list_y=np.array(user_list_y)

Y_prime=np.log(user_list_y)
X_prime=np.log(user_list_x)

X_sum=np.sum(x_prime)
X_sq=np.multiply(x_prime,x_prime)
X_sqsum=np.sum(x_sq)
X_mean=np.mean(x_prime)

Y_sum=np.sum(y_prime)
Y_mean=np.mean(y_prime)

Xi_X_yi=np.multiply(x_prime,y_prime)
Xi_X_yi_sum=np.sum(xi_X_yi)

A1=(elements*(xi_X_yi_sum)-(x_sum*y_sum))/((elements*x_sqsum)-(x_sum)**2)
Ao=y_mean-A1*x_mean

A=np.exp(Ao)
B=A1

DF=pd.DataFrame({'X':user_list_x,'Y':user_list_y,'ln X':x_prime,'ln Y':y_prime,'x**2':x_sq,'ln Xln
Y':xi_X_yi})
Print(DF)
Print()
Print("Equation of the regression line y="+str(f"{a:.3f}")+"x**"+str(f"{b:.3f}"))
Print("Here a is:",a)
Print("Here b is:",b)

X=np.linspace(0,max(user_list_x),100)
Y=a*X**b
Plt.plot(X, Y)
Plt.xlabel("X")
Plt.ylabel("Y")
Plt.scatter(user_list_x, user_list_y, color='blue', label="Data Points")
```

```
Plt.title("Power Regression Fit")
Plt.grid()
Plt.show()
```

Output:

```
How many elements do you want to enter?  5
Enter the x1 coordinates: 1
Enter the x2 coordinates: 2
Enter the x3 coordinates: 3
Enter the x4 coordinates: 4
Enter the x5 coordinates: 5
[1.0, 2.0, 3.0, 4.0, 5.0]

Enter the y1 coordinates: .5
Enter the y2 coordinates: 1.7
Enter the y3 coordinates: 3.4
Enter the y4 coordinates: 5.7
Enter the y5 coordinates: 8.4
[0.5, 1.7, 3.4, 5.7, 8.4]

   X    Y      ln X      ln Y      x**2  ln Xxln Y
    0    1.0  0.5  0.000000 -0.693147  0.000000  -0.000000
    1    1  2.0  1.7  0.693147  0.530628  0.480453   0.367803
    2    3.0  3.4  1.098612  1.223775  1.206949   1.344455
    3    3  4.0  5.7  1.386294  1.740466  1.921812   2.412798
    4    5.0  8.4  1.609438  2.128232  2.590290   3.425257
    5
Equation of the regression line y=0.501x**1.752
Here a is: 0.500933649097749
Here b is: 1.75172364807736
Trinket_plot.png
Trinket_plot.png
```


Power Regression Fit
Y
X
0
2
4
6
8
0
1
2
3
4
5

Question 4

Que 4. **Weighted Fitting**

(a) Fit the data given in the following table to a straight line.

x	0	2	4	6
y	10	15	18	25

(b) Display the equation of the regression line (fitted line).

(c) Plot both the fitted line and the given data points.

(d) Redo linear fitting to the given data assuming that the data points (2,15) and (4,18) are more significant and attach weights 5 and 10 respectively to two points.

(e) Display the equation of the regression line (Fitted line).

(f) Plot both the fitted line and the given data points.

```
import pandas as pd  import matplotlib.pyplot
as plt  data_x = np.array([0, 2, 4, 6])  data_y =
np.array([10, 15, 18, 25])  original_data_table
= pd.DataFrame({
'X Values': data_x,
'Y Values': data_y
})
print("Original Data Points:")
print(original_data_table)
print()
number_of_points = len(data_x)
total_x = sum(data_x)  total_y =
sum(data_y)  sum_xy = 0
sum_x_squared = 0  for i in
range(number_of_points):  sum_xy +=
data_x[i] * data_y[i]

   sum_x_squared += data_x[i] ** 2

total_x_times_total_y = total_x * total_y

calculation_summary = pd.DataFrame({
   'Sum X': [total_x],      'Sum Y':
[total_y],      'Sum XY':
[sum_xy],
   'Sum X²': [sum_x_squared],
```

```
    '(Sum X)(Sum Y)': [total_x_times_total_y]
})
print("Calculation Summary:")
print(calculation_summary)
print()

numerator = number_of_points * sum_xy - total_x * total_y  denominator =
number_of_points * sum_x_squared - total_x ** 2  slope = numerator / denominator

average_y = total_y / number_of_points  average_x = total_x /
number_of_points
intercept = average_y - slope * average_x

print(f"Simple Regression Equation: y = {slope:.3f}x + {intercept:.3f}")

weights = np.array([1, 5, 10, 1])
total_weight = sum(weights)

sum_weighted_x = 0  sum_weighted_y = 0
sum_weighted_xy = 0  sum_weighted_x2 = 0

for i in range(number_of_points):
    sum_weighted_x += weights[i] * data_x[i]      sum_weighted_y +=
weights[i] * data_y[i]      sum_weighted_xy += weights[i] * data_x[i] *
data_y[i]      sum_weighted_x2 += weights[i] * data_x[i] ** 2

weighted_numerator = sum_weighted_xy - (sum_weighted_x * sum_weighted_y) / total_weight
weighted_denominator = sum_weighted_x2 - (sum_weighted_x ** 2) / total_weight  weighted_slope =
weighted_numerator / weighted_denominator

weighted_intercept = (sum_weighted_y - weighted_slope * sum_weighted_x) / total_weight

print(f"Weighted Regression Equation: y = {weighted_slope:.3f}x + {weighted_intercept:.3f}")

plt.scatter(data_x, data_y, label='Original Data', color='green')

plt.plot(data_x, slope*data_x + intercept,
        color='red',
        label='Standard Fit')
plt.plot(data_x, weighted_slope*data_x + weighted_intercept,          color='blue',
        linestyle='--',
        label='Weighted Fit')
```

```
plt.xlabel("X Values")  plt.ylabel("Y
Values")
plt.title("Comparison of Regression Models")
plt.legend()  plt.grid(True)
plt.show()
```

Output:

X Values	Y Values
0	10
2	15
4	18
6	25

Sum X	Sum Y	Sum XY	Sum X^2	(Sum X)(Sum Y)
12	68	342	56	816

Simple Regression Equation:

$y = 2.100x + 10.400$

Weighted Regression Equation:

$y = 2.038x + 10.423$