Geometric Curve

Eqn: $y = ax^b$

may be written as :

$$\log y = (\log a) + b(\log x)$$

i.e. $Y = A + bX$

where Y = log y , A = log a , X = log x

The normal equations are :

$$i)\ \sum Y = An + b \sum X$$

$$ii)\ \sum XY = A \sum X + b \sum X^2$$

Example: - Determine the constants of the curve $y = ax^n$ which best fits the data given below:-

x:	4	5	6	7	8
y:	8	12.5	18	24.5	32

Solution:

Taking logarithms in both the sides in the equation $y = ax^n$ we have:

log y = (log a) + n(log x)

i.e. Y = A + nX

where: Y = log y , A = log a, X = log x.

The normal equations for determining the constants A and n in (i) are:

$$i)\ \sum Y = Ar + n\sum X$$

$$ii)\ \sum XY = A\ \sum X + n \sum X^2$$

where ‘r’ is the number of pair of observations, here ‘r’ = 5

x	y	X = log x	Rounded (X = log x)	Y = log y	Rounded (Y = log y)	X^2	XY
4	8	0.6021	0.60	0.9031	0.90	0.3600	0.5400
5	12.5	0.6990	0.70	1.0969	1.10	0.4900	0.7700
6	18	0.7782	0.78	1.2553	1.26	0.6084	0.9828
7	24.5	0.8451	0.85	1.3892	1.39	0.7225	1.1815
8	32	0.9031	0.90	1.5051	1.51	0.8100	1.3590
Total			3.83		6.16	2.9909	4.8333

Putting the values from Table we have,

6.16 = 5A + 3.83n-------------- (i)

4.8333 = 3.83A +2.9909n----- (ii)

Dividing (ii) by 5 and dividing (iii) by 3.83 we get:

1.232 = A + 0.766n --------- (iv)

1.262 = A + 0.781n ----------- (v)

Subtracting (v) from (iv) we get:

1.232 = A + 0.766n

(-)1.262 = (-)A + (-)0.781n

-0.030 = 0 - 0.015n

$$n = \frac{-0.030}{-0.015}$$

n = 2

Putting n = 2 in (iv) we get:

- ⇨ 1.232 = A + 0.766 x 2
- ⇨ 1.232 = A + 1.532
- ⇨ 1.232 – 1.532 = A
- ⇨ - 0.3 = A

we know :- A = log a

log a = -0.3

a = antilog(-0.3)

= $10^{(-0.3)}$

= 0.5

Therefore, y = ax^n

$= 0.5\,x^2$

= $0.5 \sum_{i=1}^{n} (x_i)$

Geometrical_Curve_fitting(Best Fit).py - C:\Users\Avinandan Bose\AppData\Local\Programs\Python\Python39\Machine Learning\Geometrical_Curve_fitting(Best Fit

File Edit Format Run Options Window Help

```python
import numpy as np

#Range
print("Enter a range")
n = int(input())

#Input x
x = [float(input(np.array([i])))for i in range(n)]
print("x:",x)

#Input y
y = [float(input(np.array([i])))for i in range(n)]
print("y:",y)

#X=log(x)
X = [np.log10(x[k])for k in range(n)]
print("X:",X)

#Rounded X=log(x) and its sum
round_X = np.round([X[k]for k in range(n)],2)
sum_round_X = np.round(sum(round_X),2)
print("Rounded X upto 2 decimal places:",round_X)
print("Sum of round X:",sum_round_X)

#Y=log(y)
Y = [np.log10(y[k])for k in range(n)]
print("Y:",Y)


#Rounded Y=log(y)
round_Y = np.round([Y[k]for k in range(n)],2)
sum_round_Y = np.round(sum(round_Y),2)
print("Rounded Y upto 2 decimal places:",round_Y)
print("Sum of round Y:",sum_round_Y)
```

```python
print("Putting these values we get equations:")
print("Eqn 1:",sum_round_Y,)

#Square of X (Rounded)
square_round_X =[round_X[k]**2 for k in range(n)]
sum_of_square_round_X  = sum(square_round_X)
print("Square of X:",square_round_X)
print("Sum of Square of X:",sum_of_square_round_X)


#X*Y
X_Y = [round_X[k]*round_Y[k] for k in range(n)]
sum_X_Y = sum(X_Y)
print("X * Y:",X_Y)
print("Sum of X * Y:",sum_X_Y)

print("Putting these values we get equations:")
print("Eqn 1:",sum_round_Y,"=",n,"A +",sum_round_X,"n")
print("Eqn 2:",sum_X_Y,"=",sum_round_X,"A +",sum_of_square_round_X,"n")

print("Dividing Eqn 1 with ",n, "we get:" )
div_1 = sum_round_Y/n
div_2 = n/n
div_3 = sum_round_X / n

print("Modified Eqn 1:",div_1,"=",div_2,"A +",div_3,"n")

print("Dividing Eqn 2 with ",sum_round_X, "we get:" )

div_a = sum_X_Y/sum_round_X
div_b = sum_round_X / sum_round_X
div_c = sum_of_square_round_X/ sum_round_X

print("Modified Eqn 2:",div_a,"=",div_b,"A +",div_c,"n")
```

```python
print("Subtracting Modified Eqn 2 from Modified Eqn 1 we get")

sub_1 = div_1 - div_a
sub_2 = div_2 - div_b
sub_3 = div_3 - div_c

print("Final Eqn :",sub_1,"=",sub_2,"A +",sub_3,"n")

N = np.round(sub_1/sub_3,1)
A = div_1-(div_3 *N)

print("N:",N)
print("A:",A)
print("We know A= log a, hence log a =",A )

#calculation of antilog

a = np.round(pow(10,A),1)

print("a=",a)

print(" Therefore y =",a,"x^",N)

y_final = [a*(pow(x[k],N))for k in range(n)]

print("Best fit acquired in geometrical curve , y =",y_final)

#plotting x and y
from matplotlib import pyplot as plt
plt.scatter(x,y,color='red')
plt.plot(x,y,color='violet')
plt.plot(x,y_final)
plt.title('Best Fit in a Exponential curve')
plt.show()
```

```
Enter a range
5
[0]4
[1]5
[2]6
[3]7
[4]8
x: [4.0, 5.0, 6.0, 7.0, 8.0]
[0]8
[1]12.5
[2]18
[3]24.5
[4]32
y: [8.0, 12.5, 18.0, 24.5, 32.0]
X: [0.6020599913279624, 0.6989700043360189, 0.7781512503836436, 0.8450980400142568, 0.9030899869919435]
Rounded X upto 2 decimal places: [0.6  0.7  0.78 0.85 0.9 ]
Sum of round X: 3.83
Y: [0.9030899869919435, 1.0969100130080565, 1.255272505103306, 1.3891660843645324, 1.505149978319906]
Rounded Y upto 2 decimal places: [0.9  1.1  1.26 1.39 1.51]
Sum of round Y: 6.16
Putting these values we get equations:
Eqn 1: 6.16
Square of X: [0.36, 0.48999999999999994, 0.6084, 0.7224999999999999, 0.81]
Sum of Square of X: 2.9909
X * Y: [0.54, 0.77, 0.9828, 1.1815, 1.359]
Sum of X * Y: 4.8333
Putting these values we get equations:
Eqn 1: 6.16 = 5 A + 3.83 n
Eqn 2: 4.8333 = 3.83 A + 2.9909 n
Dividing Eqn 1 with  5 we get:
Modified Eqn 1: 1.232 = 1.0 A + 0.766 n
Dividing Eqn 2 with  3.83 we get:
Modified Eqn 2: 1.261958224543081 = 1.0 A + 0.7809138381201044 n
Subtracting Modified Eqn 2 from Modified Eqn 1 we get
Final Eqn : -0.029958224543080947 = 0.0 A + -0.014913838120104383 n
N: 2.0
A: -0.30000000000000004
We know A= log a, hence log a = -0.30000000000000004
a= 0.5
 Therefore y = 0.5 x^ 2.0
Best fit acquired in geometrical curve , y = [8.0, 12.5, 18.0, 24.5, 32.0]
```

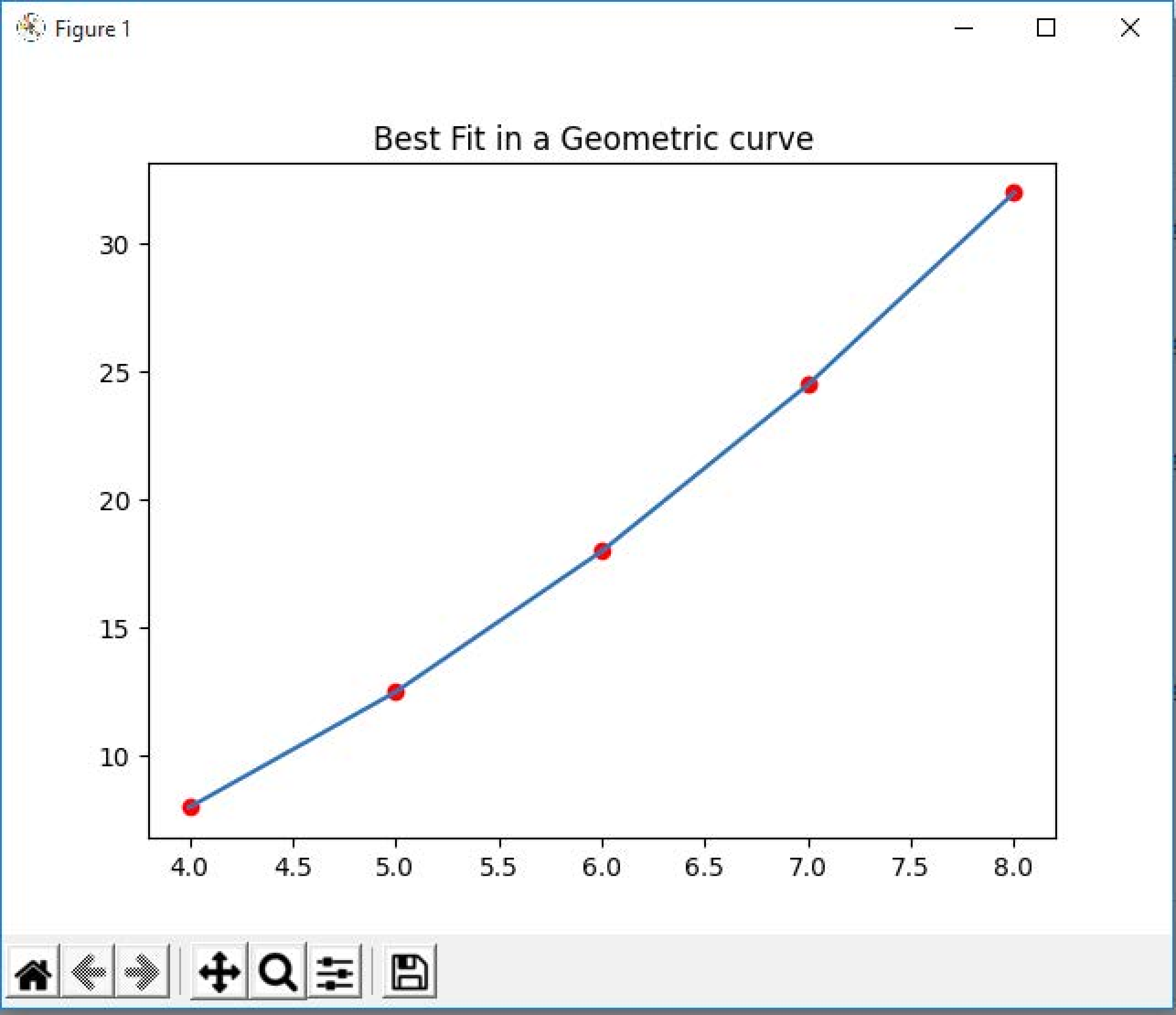
Figure 1
Best Fit in a Geometric curve
30
25
20
15
10
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0

```
Enter a range
5
[0]1.4
[1]1.5
[2]1.6
[3]1.7
[4]1.8
x: [1.4, 1.5, 1.6, 1.7, 1.8]
[0]2.5
[1]3.0
[2]4.5
[3]5.0
[4]5.5
y: [2.5, 3.0, 4.5, 5.0, 5.5]
X: [0.146128035678238, 0.17609125905568124, 0.2041199826559248, 0.2304489213782739, 0.25527250510330607]
Rounded X upto 2 decimal places: [0.15 0.18 0.2  0.23 0.26]
Sum of round X: 1.02
Y: [0.3979400086720376, 0.47712125471966244, 0.6532125137753437, 0.6989700043360189, 0.7403626894942439]
Rounded Y upto 2 decimal places: [0.4  0.48 0.65 0.7  0.74]
Sum of round Y: 2.97
Putting these values we get equations:
Eqn 1: 2.97
Square of X: [0.0225, 0.0324, 0.04000000000000001, 0.0529, 0.06760000000000001]
Sum of Square of X: 0.21540000000000004
X * Y: [0.06, 0.08639999999999999, 0.13, 0.161, 0.19240000000000002]
Sum of X * Y: 0.6298
Putting these values we get equations:
Eqn 1: 2.97 = 5 A + 1.02 n
Eqn 2: 0.6298 = 1.02 A + 0.21540000000000004 n
Dividing Eqn 1 with  5 we get:
Modified Eqn 1: 0.5940000000000001 = 1.0 A + 0.20400000000000001 n
Dividing Eqn 2 with  1.02 we get:
Modified Eqn 2: 0.6174509803921568 = 1.0 A + 0.21117647058823533 n
Subtracting Modified Eqn 2 from Modified Eqn 1 we get
Final Eqn : -0.023450980392156760 = 0.0 A + -0.007176470588235312 n
N: 3.3
A: -0.07919999999999994
We know A= log a, hence log a = -0.07919999999999994
a= 0.8
 Therefore y = 0.8 x^ 3.3
Best fit acquired in geometrical curve , y = [2.4283568362090064, 3.0492367257335093, 3.7729923944491413, 4.608625595960798,
5.565308633080833]
```

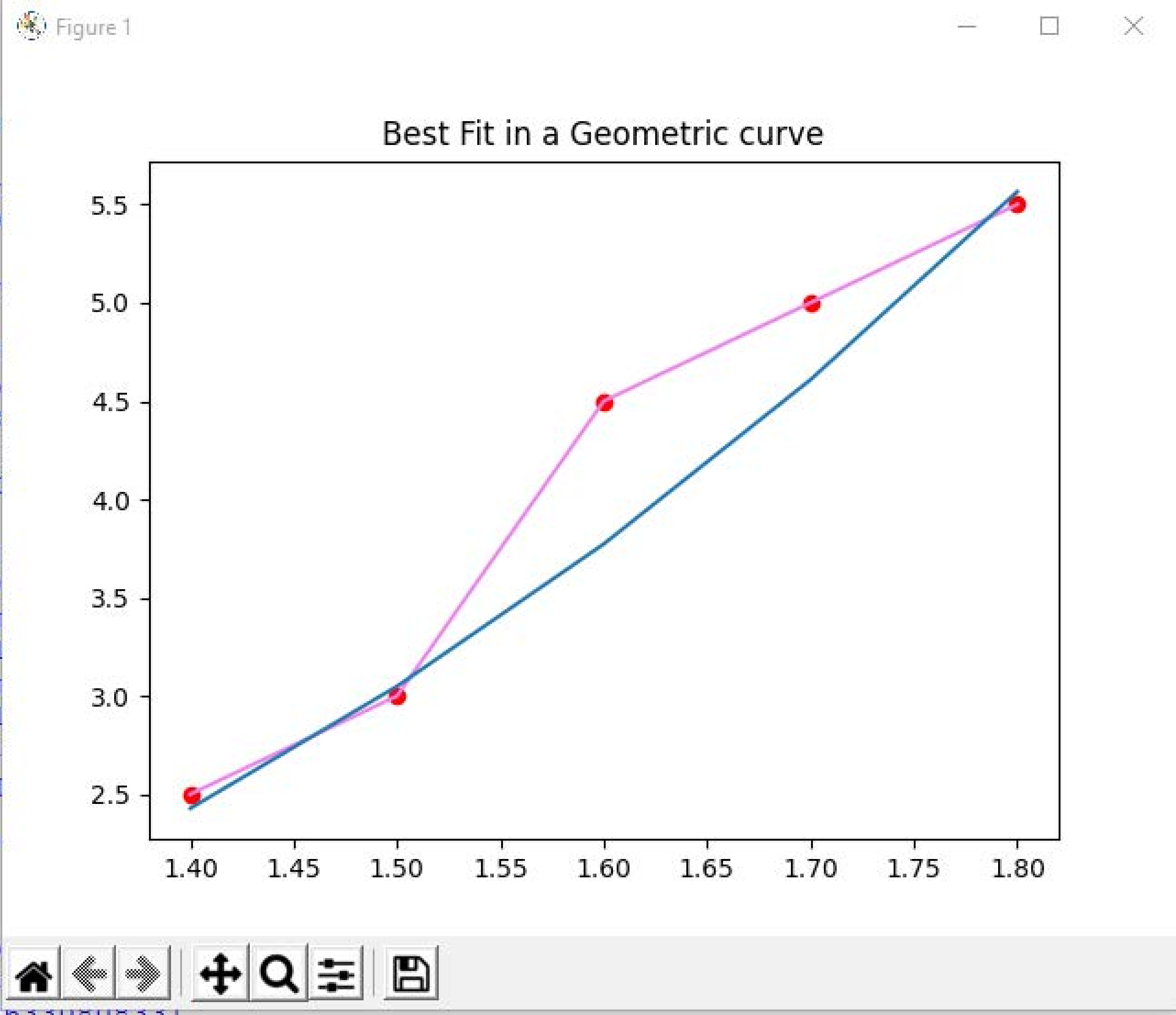
Figure 1
Best Fit in a Geometric curve
5.5
5.0
4.5
4.0
3.5
3.0
2.5
1.40
1.45
1.50
1.55
1.60
1.65
1.70
1.75
1.80