Assignment: 07

On Linear regression

PH1050 Computational Physics

Aarya Gosar
EP23B025
Engineering Physics
12th Oct 2023

Introduction

While analysing data, we often see two quantities being dependent on each other. In regression, we try to come up with a formula that predicts a quantity, given that we can measure the other quantity with certainity. In linear regression, we have a linear system of equations and we try to adjust the coefficient to match our data.

Aim

1) Generate a linear data and try to get the best fit line using a) formula, b)Mathematica functions
2) Find best fit curve of a multi parameter system using a) mathematica functions b) Linear Algebra
3) Generate a non linear data and use the function NonLinearModelFit

Code

Linear Data

Generating the data

I have used the data from my highschool spring mass oscillator experiment
X Axis : mass   Y Axis: T^2

In[246]:=

Observations :

PART - B To determine the mass of spring (m_s):

1. Mass of the hanger (m_0) = ..50.. g
2. Least count of the stop watch = ...1.... s

Sr No.	Total Mass M kg	Time t for 20 oscillation (s)			mean t (s)	Periodic time T = mean t/20 (s)	T^2 (S^2)
		1	2	3			
1	100	23	21	22	22	1·1	1·21
2	150	25	23	24	24	1·2	1·44
3	200	27	30	29	29	1·45	2·11
4	250	30	31	29	30	1·5	2·25
5	300	32	31	33	32	1·6	2·56

Calculations:

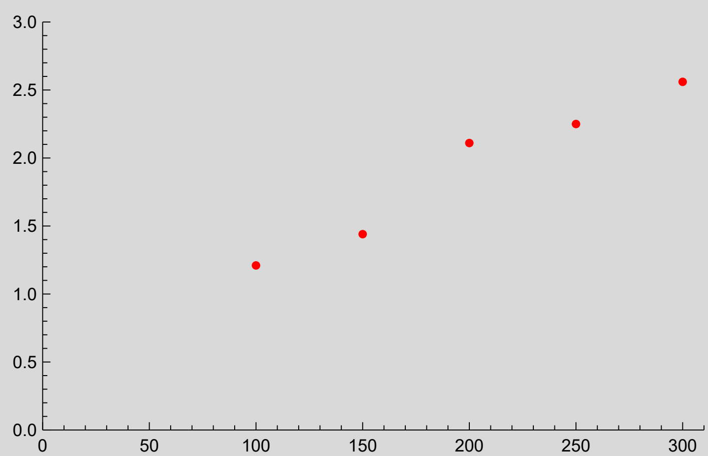
In[247]:=

```
X = {100,150,200,250,300};
y = {1.21,1.44,2.11,2.25,2.56};
n = Length[X];
combinedData = Table[{X[[i]],y[[i]]},{i,1,n,1}]
pltPoints = ListPlot[combinedData,PlotRange→{{0,310},{0,3}}, PlotStyle→{Red,Thick}]
```

Out[250]=

```
{{100, 1.21}, {150, 1.44}, {200, 2.11}, {250, 2.25}, {300, 2.56}}
```

Out[251]=



Best fit Line using Formula

In[252]:=

```
XiYi = Table[{X[[i]] * y[[i]]} , {i,1,n,1}];
Xi2 = Table[{X[[i]]^2} , {i,1,n,1}];
m = (n* Total[XiYi] - Total[X]*Total[y])/(n * (Total[Xi2])- Total[X]^2);
Print["M = "]
m = m[[1]]
c = (Total[y]*Total[Xi2] - Total[X]*Total[XiYi])/(n * (Total[Xi2])- Total[X]^2);
Print["C = "]
c = c[[1]]
datatest = Table[{x , m * x + c}, {x,-100,310}];
pltLine = ListLinePlot[datatest];
Show[pltLine,pltPoints]
```

M =

Out[256]=

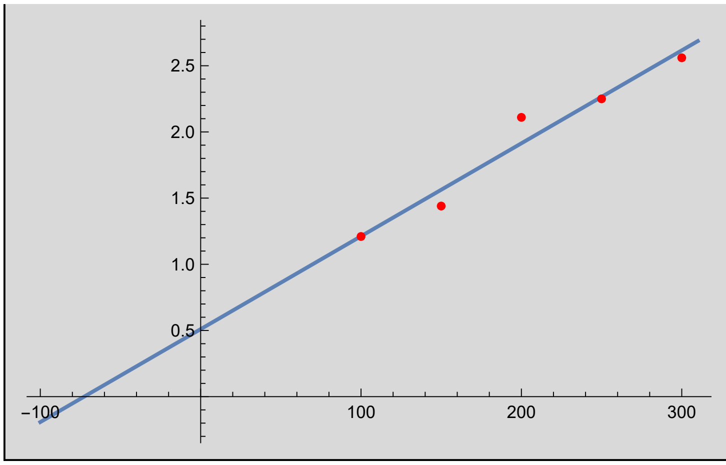
```
0.00702
```

C =

Out[259]=

```
0.51
```

Out[262]=



We can see that it cuts X - Axis at ~ -75 which is mass of hanger.
which was actually the case when I did the experiment

Using Linear Model fit

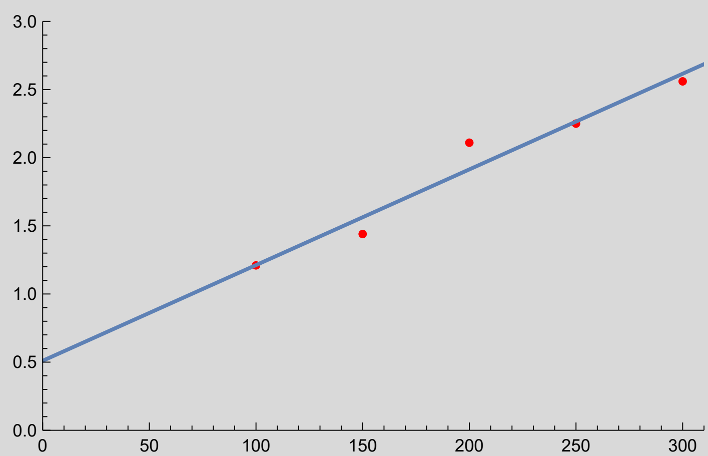
In[263]:=

```
LineFormula = LinearModelFit[combinedData,{x,1} , x]
pltFormula = ListLinePlot[Table[{x , LineFormula[x]}, {x,0,310,1}]];
Show[pltPoints,pltFormula]
```

Out[263]=

```
FittedModel[ 0.51 + 0.00702 x ]
```

Out[265]=



We Get the same result as before

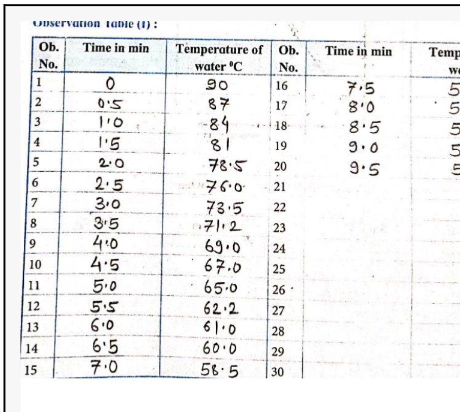
Multi Parameter

Generating data

This is experiment is from Newton’s law of cooling. Here we compare the temperature with time

X Axis: Time Y Axis: θ

In[266]:=



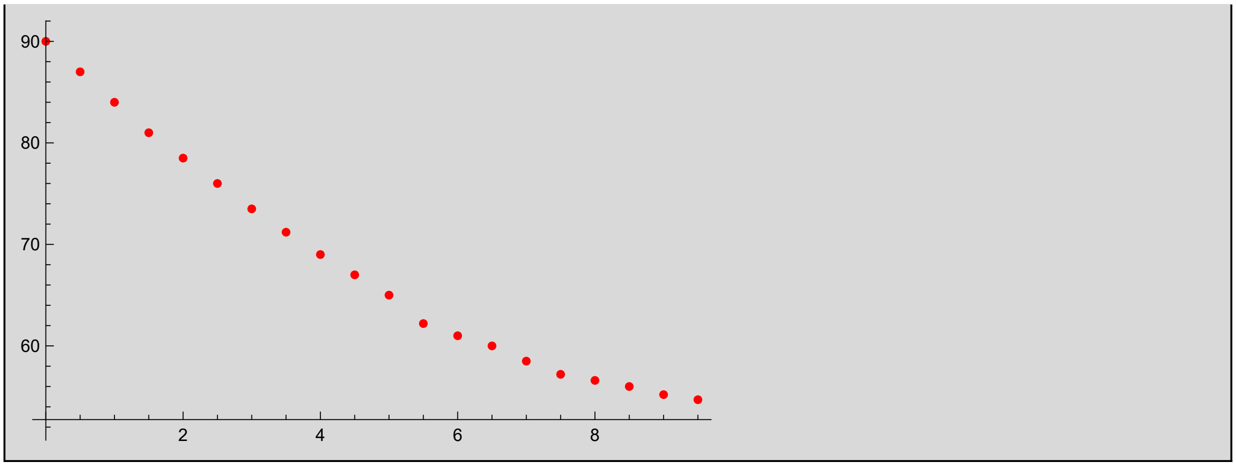
Observation Table (I) :

Ob. No.	Time in min	Temperature of water °C	Ob. No.	Time in min	Temp wa
1	0	90	16	7.5	5
2	0.5	87	17	8.0	5
3	1.0	84	18	8.5	5
4	1.5	81	19	9.0	5
5	2.0	78.5	20	9.5	5
6	2.5	76.0	21		
7	3.0	73.5	22		
8	3.5	71.2	23		
9	4.0	69.0	24		
10	4.5	67.0	25		
11	5.0	65.0	26		
12	5.5	62.2	27		
13	6.0	61.0	28		
14	6.5	60.0	29		
15	7.0	58.5	30		

In[267]:=

```
Clear["Global`*"]
θ = {90,87,84,81,78.5,76,73.5,71.2,69,67,65,62.2,61,60,58.5,57.2,56.6,56,55.2,54.7};
t = Table[time,{time,0,9.5,0.5}];
combinedData = Table[{ t[[i]], θ[[i]]} , {i,1,20,1}];
pltPoints = ListPlot[combinedData,PlotStyle→{Red,Thick}]
```

Out[271]=



Making the curve using FindFit function

I have chosen the function:

$$a3 + a2\, e^{-x} + \frac{a1}{1+x}$$

In[272]:=

```
params = FindFit[combinedData, a1 / (x + 1) + a2 Exp[-x] + a3 , {a1, a2, a3} , x]
FitCurve[x_] = a1 / (x + 1) + a2 Exp[-x] + a3 /. params
pltCurve = ListLinePlot[Table[{x, FitCurve[x]} , {x, 0, 10, 0.1}]];
Show[pltCurve, pltPoints]
```

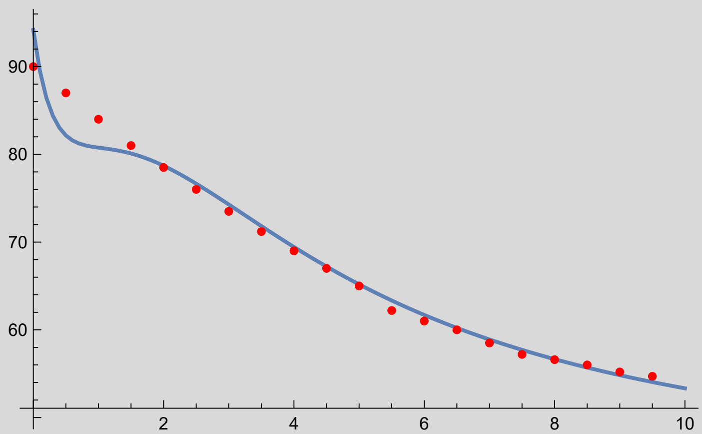
Out[272]=

{a1 → 166.113, a2 → -110.175, a3 → 38.2371}

Out[273]=

$$38.2371 - 110.175\, e^{-x} + \frac{166.113}{1 + x}$$

Out[275]=



Making the curve with Linear algebra

In[276]:=

```
paramMatrix = Table[{1 / (t[[i]] + 1), Exp[-t[[i]]] , 1} , {i, 1, 20, 1}];
{a1, a2, a3} =
 LinearSolve[Transpose[paramMatrix].paramMatrix , Transpose[paramMatrix].θ]
Print["The function is same as above"]
LinearFitCurve[x_] = a1 / (x + 1) + a2 Exp[-x] + a3
pltCurveLinear = ListLinePlot[Table[{x, LinearFitCurve[x]} , {x, 0, 10, 0.1}]];
Show[pltPoints, pltCurveLinear]
```

Out[277]=

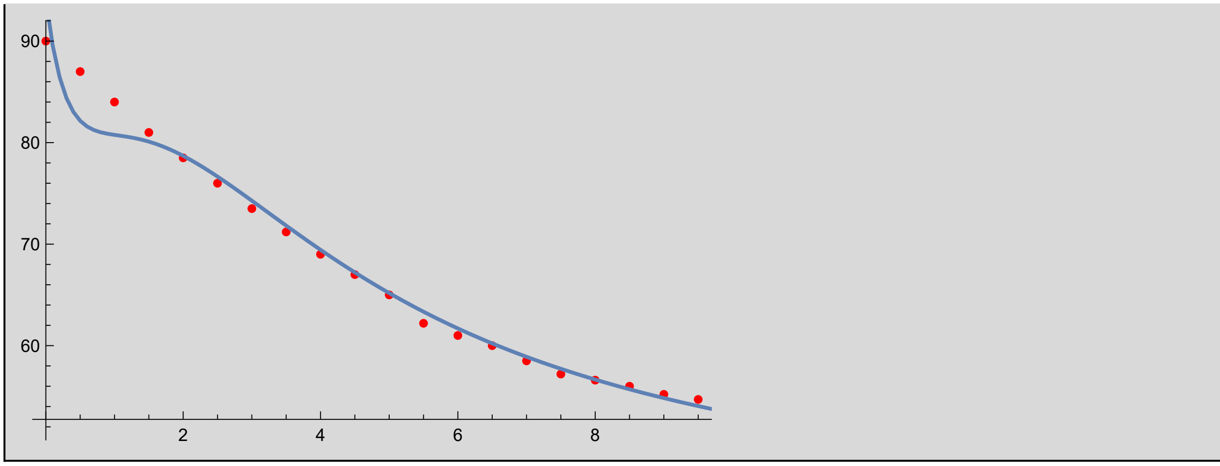
{166.113, -110.175, 38.2371}

The function is same as above

Out[279]=

$$38.2371 - 110.175\, e^{-x} + \frac{166.113}{1 + x}$$

Out[281]=



Non Linear fit

Generating data

Here I will take data from PH1030 experiment on transistor characteristics
The Ic vs Vce graph is non linear as we require

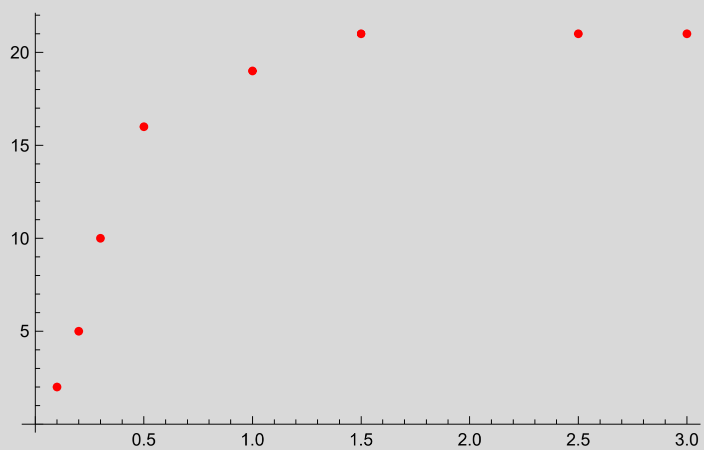
In[282]:=

$I_B = 75$ (μA)	
V_{CE} (v)	I_c (mA)
0.1	2
0.2	5
0.3	10
0.5	16
1	19
1.5	21
2.5	21

In[283]:=

```
Clear["Global`*"]
X = {0.1, 0.2, 0.3, 0.5, 1, 1.5, 2.5, 3};
y = {2, 5, 10, 16, 19, 21, 21, 21};
combined = Table[{X[[i]] , y[[i]]} , {i, 1, 8, 1}];
pltPoints = ListPlot[combined, PlotStyle → {Red, Thick, Bold}]
```

Out[287]=



NonLinear function

The function I have chose is
$b\, e^{-c x} + a \operatorname{Log}[x] + e \operatorname{Sin}[d x]$

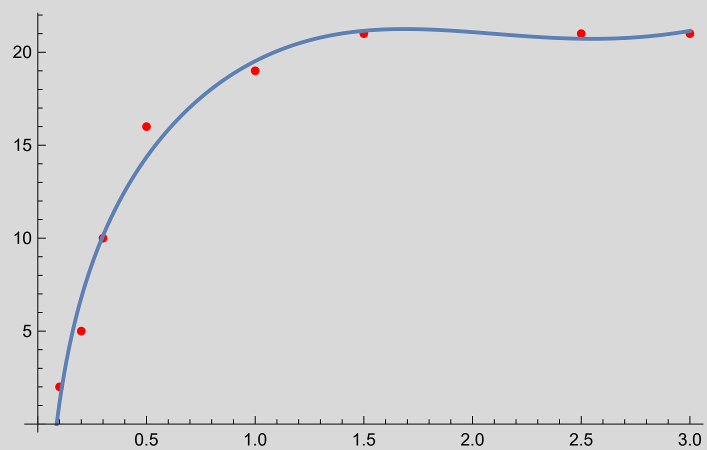
In[288]:=

```
NonLinearFunc =
 NonlinearModelFit[combined, a Log[x] + b Exp[- c x] + e Sin[d x], {a, b, c, d, e} , x]
pltNonLinear = ListLinePlot[Table[{x, NonLinearFunc[x]}, {x, 0, 3, 0.01}]];
Show[pltPoints, pltNonLinear]
```

Out[288]=

FittedModel[$19.2895\, e^{-0.0972941\, x} + 7.93789\, \mathrm{Log}[x] + 2.02666\, \mathrm{Sin}[1.64462\, x]$]

Out[290]=



Conclusion

As we can see, data from experiment is never precise. Regression helps us to find the correlation between two quantities subject to noise.
For simple systems we may choose to derive the equations ourselves or for complex systems we may use builtin functions of mathematica. We can see that the results are same irrespective of methods we choose.