ASSIGNMENT7: ON LINEAR AND NON LINEAR REGRESSION

SRIPATHI RAVI
EP23B021
DEPARTMENT OF PHYSICS

AIM

To get an experimental data and find the function that fits the data accurately using linear and non linear regression

PROBLEM STATEMENT

PART A:
1)Take a linear function and add noise to produce experimental data
2)Manually fit the data and find m and c
3)Solve for m and c using LinearModelFit and FindFit
PART B:
1) Take a multi parameter linear function and add gaussian noise
2)Using FindFit fit the data and extract fit parameters.
3)Convert the data to suitable matrix form and use LinearSolve to get the fit.
PART C:
1)Generate a non-linear function and add white noise.
2)Use NonLinearModelFit to fit the data and get the the fitting parameters

code organisation

Part A:
-defining linear function
-adding noise
-defining different summation for manually solving
-manually solving
-plotting graph
-using model fit to fit the data
-plotting graph
-using find fit to fit the data

-plotting graph
Part B:
-defining a multi parameter linear function
-fitting it using find fit
-plotting graph
-converting data into matrix
-solving matrix equation
-plotting graph
Part C:
-defining a non linear function and adding noise
-fitting it using NonLinearModelFit
-plotting graph

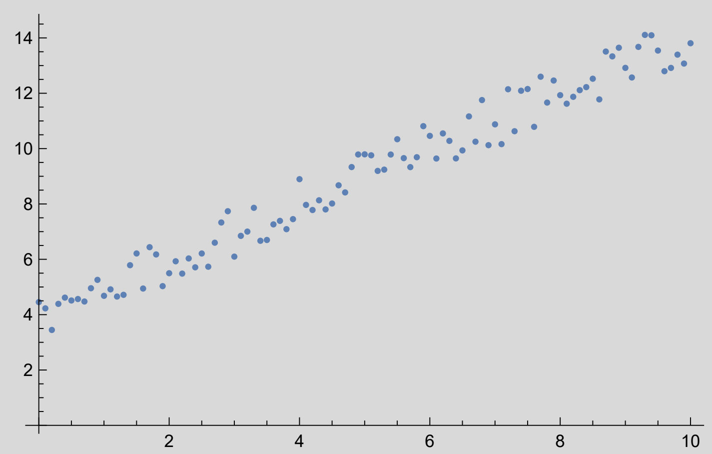
computational code

part a

In[210]:=

```
datan1 = Table[{x, 4 + x + RandomReal[{-1, 1}]}, {x, 0, 10, 0.1}];
a = ListPlot[datan1]
```

Out[211]=



In[213]:=

```
b1 = Sum[datan1[[i]][[1]], {i, 1, 100}]
```

Out[213]=

```
495.
```

In[214]:=
```
b2 = Sum[(datan1[[i]][[1]])^2, {i, 1, 100}]
```
Out[214]=
```
3283.5
```
In[215]:=
```
del = 100 b2 - b1^2
```
Out[215]=
```
83 325.
```
In[216]:=
```
ys = Sum[datan1[[i]][[2]], {i, 1, 100}]
```
Out[216]=
```
890.483
```
In[217]:=
```
xys = Sum[(datan1[[i]][[2]]) (datan1[[i]][[1]]), {i, 1, 100}]
```
Out[217]=
```
5252.9
```
In[218]:=
```
c = (b2 ys - b1 xys) / del
```
Out[218]=
```
3.88496
```
In[219]:=
```
m = (100 xys - b1 ys) / del
```
Out[219]=
```
1.01411
```

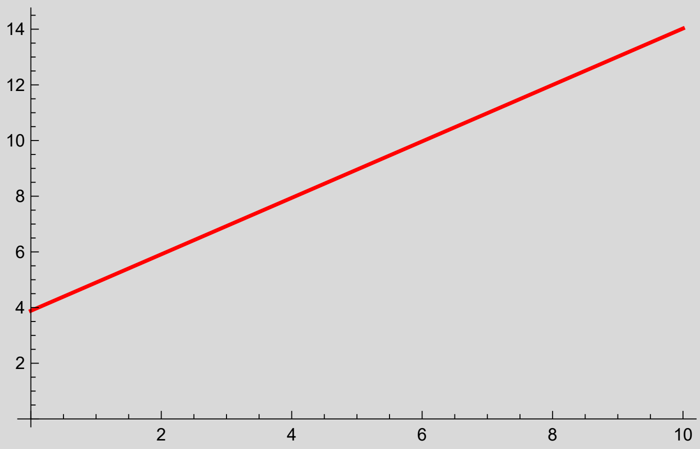
In[220]:=

```
f[x_] = m x + c
data1 = Table[{x, f[x]}, {x, 0, 10, 0.1}];
a1 = ListLinePlot[data1, PlotStyle → {Red}]
```

Out[220]=

```
3.88496 + 1.01411 x
```

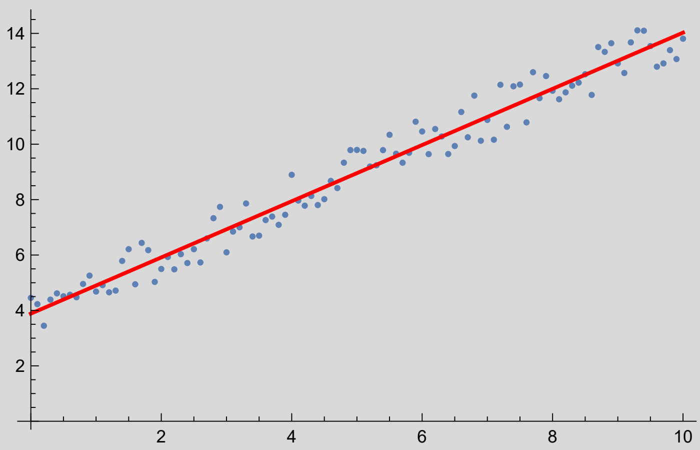
Out[222]=



In[223]:=

```
Show[a, a1]
```

Out[223]=



In[224]:=

```
lp = LinearModelFit[datan1, x, x]
```

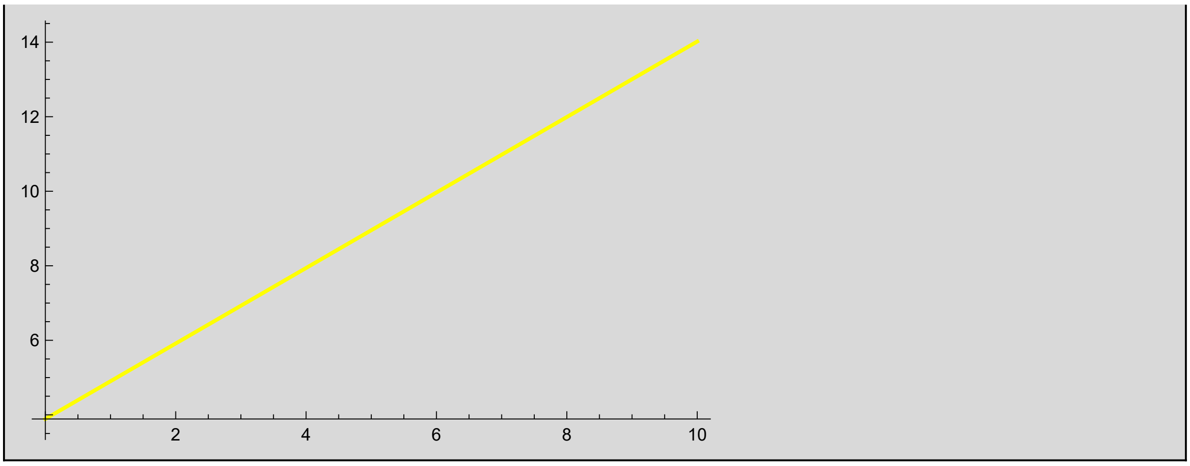
Out[224]=

```
FittedModel[ 3.88916 + 1.01284 x ]
```

In[225]:=

```
a3 = Plot[lp["BestFit"], {x, 0, 10}, PlotStyle → {Yellow}]
```

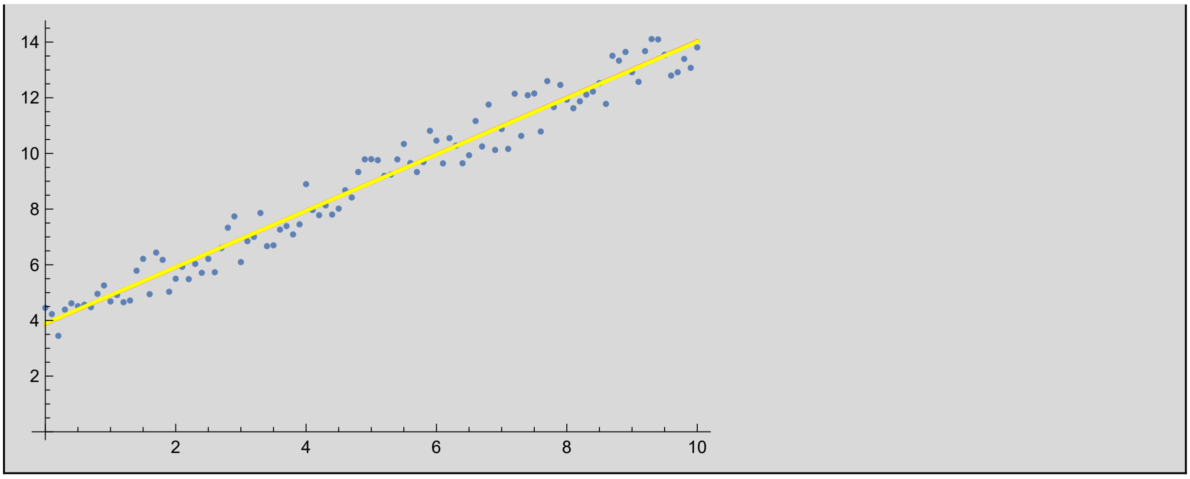
Out[225]=



In[226]:=

```
Show[a1, a, a3]
```

Out[226]=



In[230]:=

```
an = FindFit[datan1, m1 x + c1, {m1, c1}, x]
```

Out[230]=

```
{m1 → 1.01284, c1 → 3.88916}
```

In[232]:=

```
fn[x_] = m1 x + c1 /. an
```

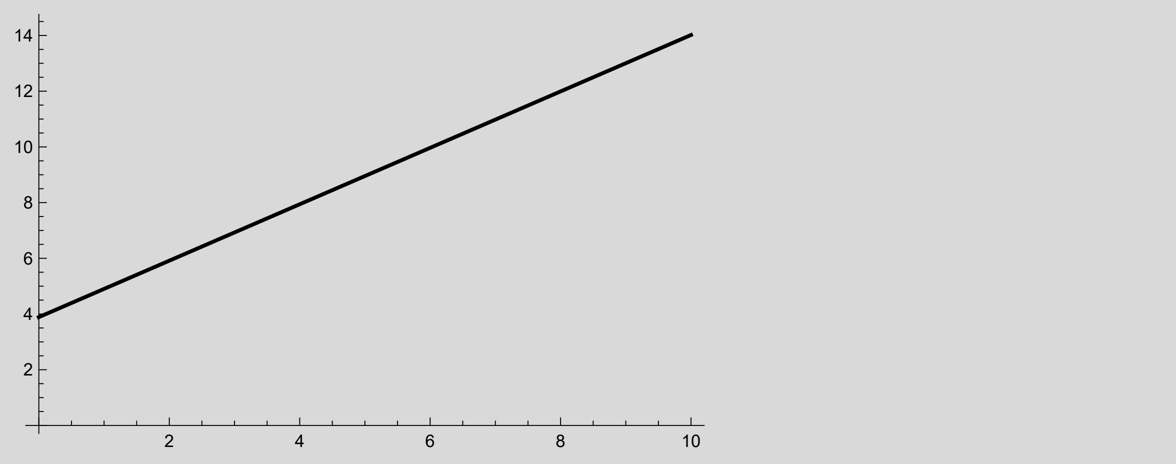
Out[232]=

```
3.88916 + 1.01284 x
```

In[243]:=

```
dataf = Table[{x, fn[x]}, {x, 0, 10, 0.1}];
af = ListLinePlot[dataf, PlotStyle → {Black}]
```

Out[244]=

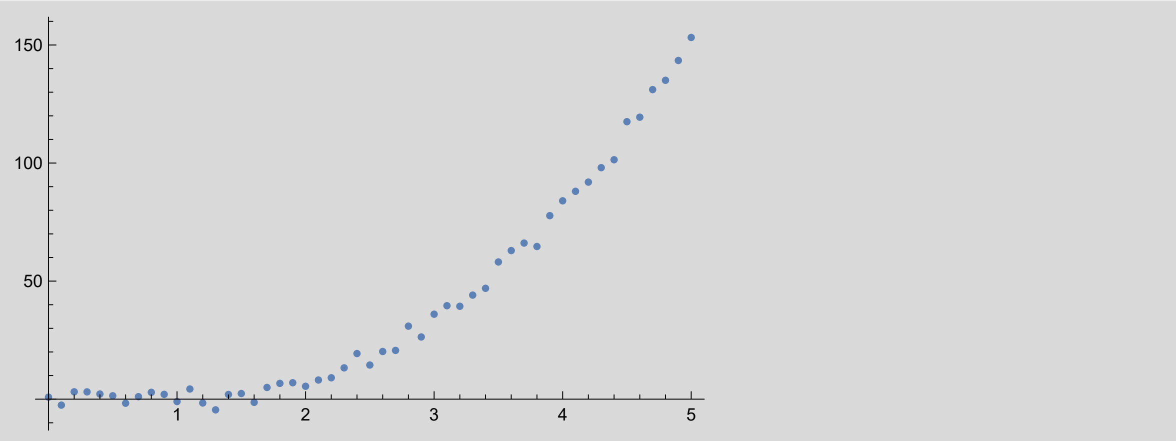


part b

In[164]:=

```
datax = Table[{x, 2 x^2 - 4 x + x^3 + RandomReal[{-5, 5}]}, {x, 0, 5, 0.1}];
a = ListPlot[datax]
```

Out[165]=



In[166]:=

```
f1x = FindFit[datax, d x + b5 x^2 + a5 x^3, {d, b5, a5}, x]
```

Out[166]=

{d → -3.94001, b5 → 2.19203, a5 → 0.951433}

In[167]:=

```
f2[x_] = d x + b5 x^2 + a5 x^3 /. f1x
```

Out[167]=

$-3.94001\, x + 2.19203\, x^2 + 0.951433\, x^3$

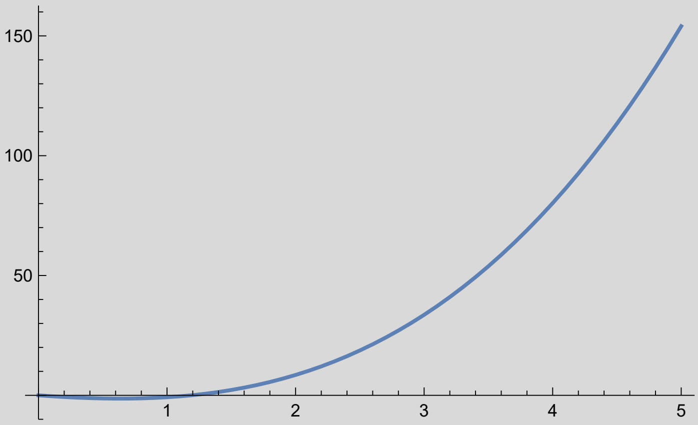
Out[12]=

```
0.
```

In[168]:=

```
data1 = Table[{x, f2[x]}, {x, 0, 5, 0.1}];
b = ListLinePlot[data1]
```

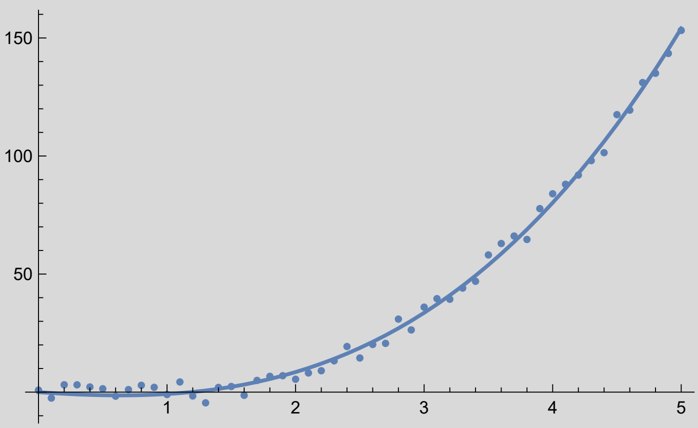
Out[169]=



In[170]:=

```
Show[a, b]
```

Out[170]=



In[117]:=

```
data2 = Table[{x, x^2, x^3}, {x, 0, 5, 0.1}];
mf = data2 // MatrixForm
```

Out[118]//MatrixForm=

0.	0.	0.
0.1	0.01	0.001
0.2	0.04	0.008
0.3	0.09	0.027
0.4	0.16	0.064
0.5	0.25	0.125
0.6	0.36	0.216
0.7	0.49	0.343
0.8	0.64	0.512
0.9	0.81	0.729
1.	1.	1.
1.1	1.21	1.331
1.2	1.44	1.728
1.3	1.69	2.197
1.4	1.96	2.744
1.5	2.25	3.375
1.6	2.56	4.096
1.7	2.89	4.913
1.8	3.24	5.832
1.9	3.61	6.859
2.	4.	8.
2.1	4.41	9.261
2.2	4.84	10.648
2.3	5.29	12.167
2.4	5.76	13.824
2.5	6.25	15.625
2.6	6.76	17.576
2.7	7.29	19.683
2.8	7.84	21.952
2.9	8.41	24.389
3.	9.	27.
3.1	9.61	29.791
3.2	10.24	32.768
3.3	10.89	35.937
3.4	11.56	39.304
3.5	12.25	42.875
3.6	12.96	46.656
3.7	13.69	50.653
3.8	14.44	54.872
3.9	15.21	59.319
4.	16.	64.
4.1	16.81	68.921
4.2	17.64	74.088
4.3	18.49	79.507
4.4	19.36	85.184
4.5	20.25	91.125
4.6	21.16	97.336
4.7	22.09	103.823
4.8	23.04	110.592
4.9	24.01	117.649
5.	25.	125.

In[126]:=

```
mft = MatrixForm[Transpose[data2]]
```

Out[126]//MatrixForm=

$$\begin{pmatrix} 0. & 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 & 0.7 & 0.8 & 0.9 & 1. & 1.1 & 1.2 & 1.3 \\ 0. & 0.01 & 0.04 & 0.09 & 0.16 & 0.25 & 0.36 & 0.49 & 0.64 & 0.81 & 1. & 1.21 & 1.44 & 1.69 \\ 0. & 0.001 & 0.008 & 0.027 & 0.064 & 0.125 & 0.216 & 0.343 & 0.512 & 0.729 & 1. & 1.331 & 1.728 & 2.197 & 2 \end{pmatrix}$$

Out[183]=

```
153.182
```

In[173]:=

```
data3 = Table[{datax⟦i⟧⟦2⟧}, {i, 1, 51, 1}]
```

Out[173]=

```
{{0.853366}, {-2.52398}, {3.12364}, {3.09493}, {2.18548}, {1.43664},
 {-1.71909}, {1.07017}, {2.89807}, {2.03648}, {-1.01269}, {4.31011},
 {-1.61928}, {-4.5277}, {1.99573}, {2.38742}, {-1.34879}, {4.96207},
 {6.6882}, {6.95496}, {5.45681}, {8.11575}, {9.06487}, {13.2603}, {19.3197},
 {14.4602}, {20.1895}, {20.6491}, {30.9271}, {26.3401}, {35.9858},
 {39.5818}, {39.3345}, {44.0728}, {46.9469}, {58.1065}, {62.9331}, {66.1105},
 {64.6597}, {77.7228}, {84.0143}, {88.0263}, {91.9329}, {98.0506}, {101.406},
 {117.524}, {119.419}, {131.108}, {135.045}, {143.438}, {153.182}}
```

In[154]:=

```
my = MatrixForm[data3]
```

Out[154]//MatrixForm=

$$\begin{pmatrix} 3.81802 \\ -3.50494 \\ 3.43641 \\ 2.78029 \\ 2.33592 \\ -1.49312 \\ -2.54149 \\ 2.85786 \\ -3.62776 \\ -5.31837 \\ -4.96221 \\ -0.293242 \\ -1.1462 \\ 1.48274 \\ -1.44004 \\ 0.469006 \\ 5.66938 \\ 3.75998 \\ 8.07239 \\ 9.19784 \\ 4.33674 \\ 8.97495 \\ 7.29868 \\ 13.5472 \\ 18.4973 \\ 22.3378 \\ 19.5778 \\ 18.4778 \\ 25.7134 \\ 29.6082 \\ 34.9027 \\ 39.0221 \\ 41.5957 \\ 44.6218 \\ 52.3697 \\ 48.4733 \\ 58.7321 \\ 63.3696 \\ 65.985 \\ 76.8731 \\ 77.7058 \\ 89.5875 \\ 96.9096 \\ 102.39 \\ 106.437 \\ 112.418 \\ 118.677 \\ 125.945 \\ 135.974 \\ 150.538 \\ 154.915 \end{pmatrix}$$

In[174]:=

```
Lsr = LinearSolve[Transpose[data2].data2, Transpose[data2].data3]
```

Out[174]=

```
{{-3.94001}, {2.19203}, {0.951433}}
```

In[184]:=

```
flsd = Table[{x, ({x, x^2, x^3}.Lsr)[[1]]}, {x, 0, 5, 0.1}]
```

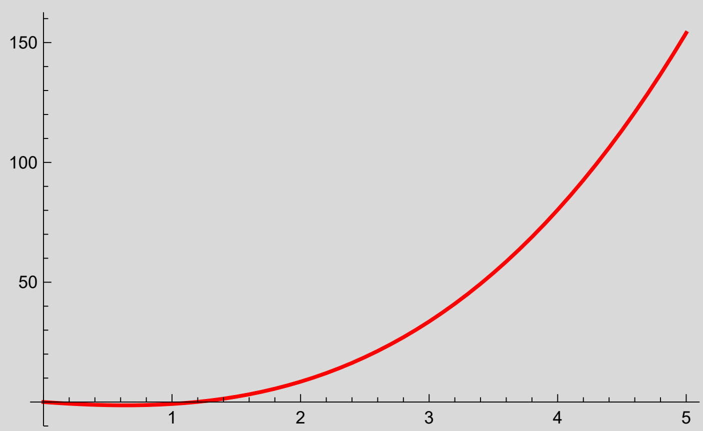
Out[184]=

```
{{0., 0.}, {0.1, -0.37113}, {0.2, -0.69271}, {0.3, -0.959033},
 {0.4, -1.16439}, {0.5, -1.30307}, {0.6, -1.36937}, {0.7, -1.35757},
 {0.8, -1.26198}, {0.9, -1.07687}, {1., -0.796548}, {1.1, -0.415298},
 {1.2, 0.0725871}, {1.3, 0.672816}, {1.4, 1.3911}, {1.5, 2.23314},
 {1.6, 3.20465}, {1.7, 4.31134}, {1.8, 5.55892}, {1.9, 6.95309}, {2., 8.49957},
 {2.1, 10.2041}, {2.2, 12.0723}, {2.3, 14.1099}, {2.4, 16.3227}, {2.5, 18.7163},
 {2.6, 21.2965}, {2.7, 24.0689}, {2.8, 27.0394}, {2.9, 30.2135}, {3., 33.5969},
 {3.1, 37.1955}, {3.2, 41.0149}, {3.3, 45.0608}, {3.4, 49.339}, {3.5, 53.855},
 {3.6, 58.6147}, {3.7, 63.6238}, {3.8, 68.8879}, {3.9, 74.4128}, {4., 80.2042},
 {4.1, 86.2677}, {4.2, 92.6092}, {4.3, 99.2342}, {4.4, 106.149}, {4.5, 113.358},
 {4.6, 120.868}, {4.7, 128.685}, {4.8, 136.813}, {4.9, 145.26}, {5., 154.03}}
```

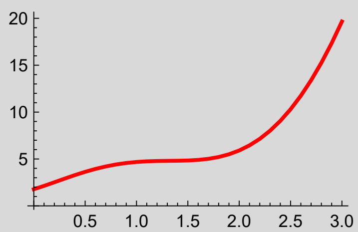
In[185]:=

```
lp = ListLinePlot[flsd, PlotStyle → {Red}]
```

Out[185]=



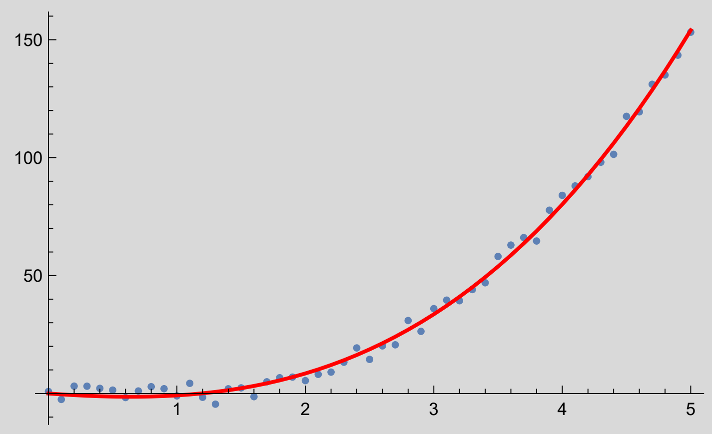
Out[179]=



In[186]:=

```
Show[a, lp]
```

Out[186]=

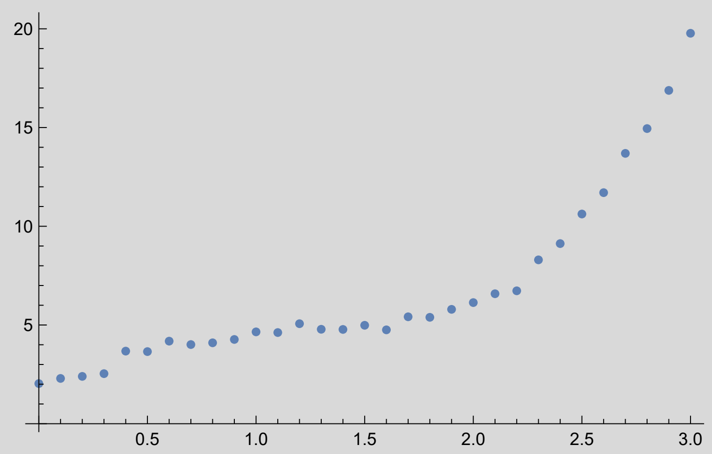


part c

In[198]:=

```
datanl = Table[
    {x, Exp[ x] + Exp[-2 x] + 2 Sin[2 x] + RandomReal[{-0.5, 0.5}]}, {x, 0, 3, 0.1}];
c = ListPlot[datanl]
```

Out[199]=



In[200]:=

```
nls = NonlinearModelFit[datanl, Exp[ h x] + a1 Exp[g x] + d Sin[e x], {h, g, a1, d, e}, x]
```

Out[200]=

FittedModel[$e^{-2.06498 x} + 0.966283 e^{1.01642 x} + 1.94633 \sin[1.94218 x]$]

FittedModel[$2.04375 e^{-1.32963 x} + e^{0.989534 x} + 1.80452 \sin[2.03841 x]$]

In[201]:=

```
nor = Normal[nls]
```

Out[201]=

$e^{-2.06498 x} + 0.966283 e^{1.01642 x} + 1.94633 \text{Sin}[1.94218 x]$

Out[88]=

$2.04375 e^{-1.32963 x} + e^{0.989534 x} + 1.80452 \text{Sin}[2.03841 x]$

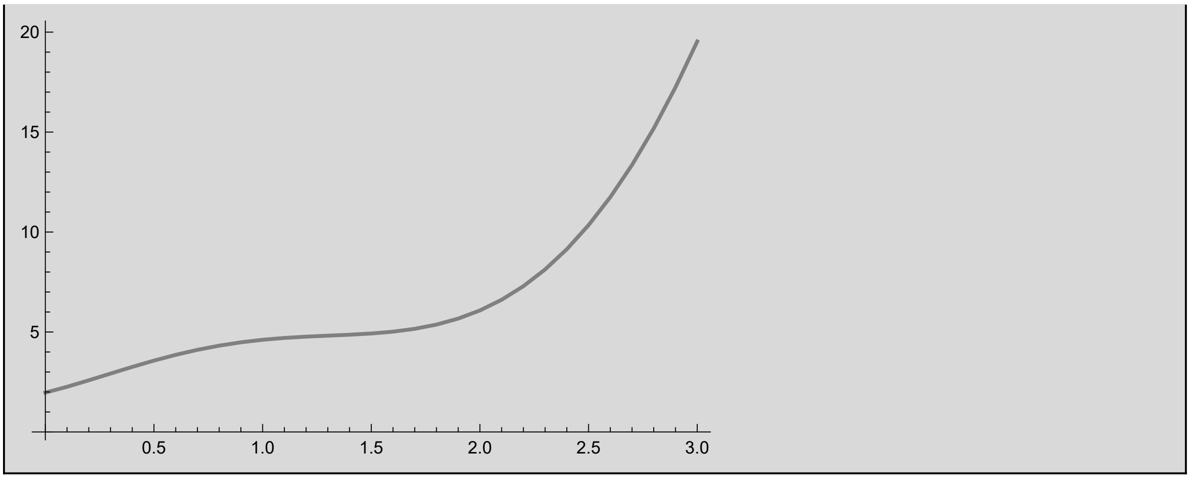
Out[89]=

4.8415

In[202]:=

```
datas = data = Table[{x, nls[x]}, {x, 0, 3, 0.1}];
b = ListLinePlot[datas, PlotStyle → {Gray}]
```

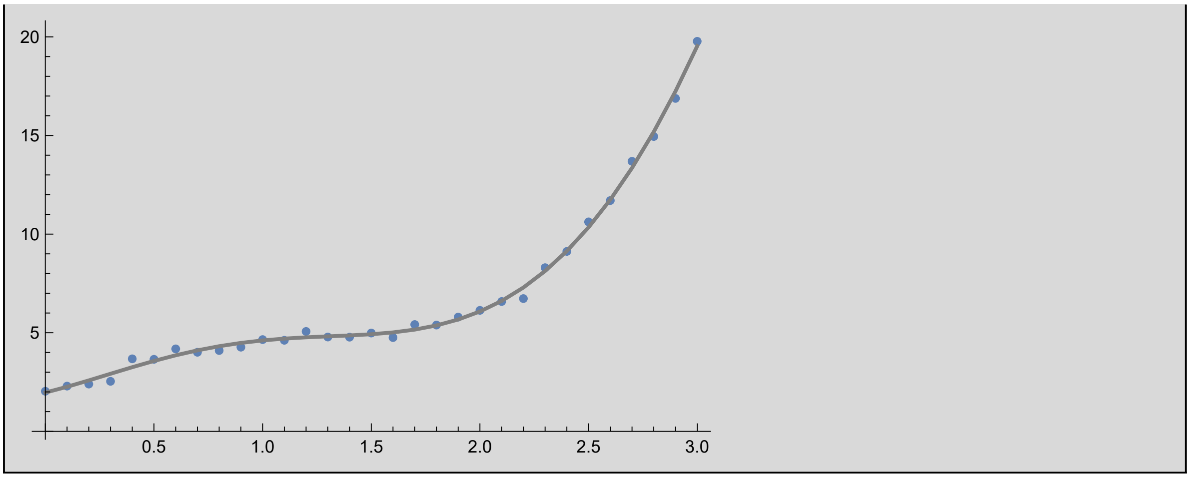
Out[203]=



In[204]:=

```
Show[c, b]
```

Out[204]=



result

Part a:
 experimental data:output 211
 manually fitted function: output222
 manually fitted function with experimental data: output223
 fitted using linear model fit: output 225
 fitted function using linear model fit with experimental data: output 226
 function fitted using find fit:output 244
Part b:
 experimental data: output 165
 data fitted using find fit: output 169

data fitted using find fit with experimental data: output 170
data fitted using matrix method: output 185
data fitted using matrix method along with experimental data: output 186

Part c:
experimental data:output 199
fitted data:203
fitted data along with experimental data:204

Linear and non linear regression is used to fit an experimental data (a function+noise)
into appropriate function. From the results above we can find out that the function fitted using regression is approximately same as the actual function.

discussion and comments

Linear and non linear regression is used to fit an experimental data to an appropriate function. This assignment was particularly useful in learning regression using mathematica
which can used in future for doing physics experiments.

references

mathematica stack exchange
mathematica documentation