Content

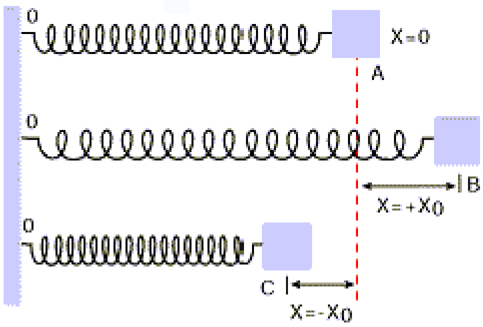
No.	Practical	Date	Remarks
1	Fitting a model to the dataset of mass and elongation of a spring		
2	Fitting a model to the dataset of radius and volume of a sphere		
3	Fitting a model to the dataset of time and population of a US in 1800s		
4	To obtain a natural cubic spline using Excel and MATLAB		
5	To simulate prey-predator model using MATLAB		
6	Program for multiple regression using R		
7	Program for statistical analysis of data using R		
8	To write a program using Monte-Carlo method.		
9	Simulate a model for epidemics using MATLAB		

Practical 1

AIM: In order to conduct an experiment to measure the stretch of a spring as a function of mass, a spring mass system is considered. The following data is obtained.

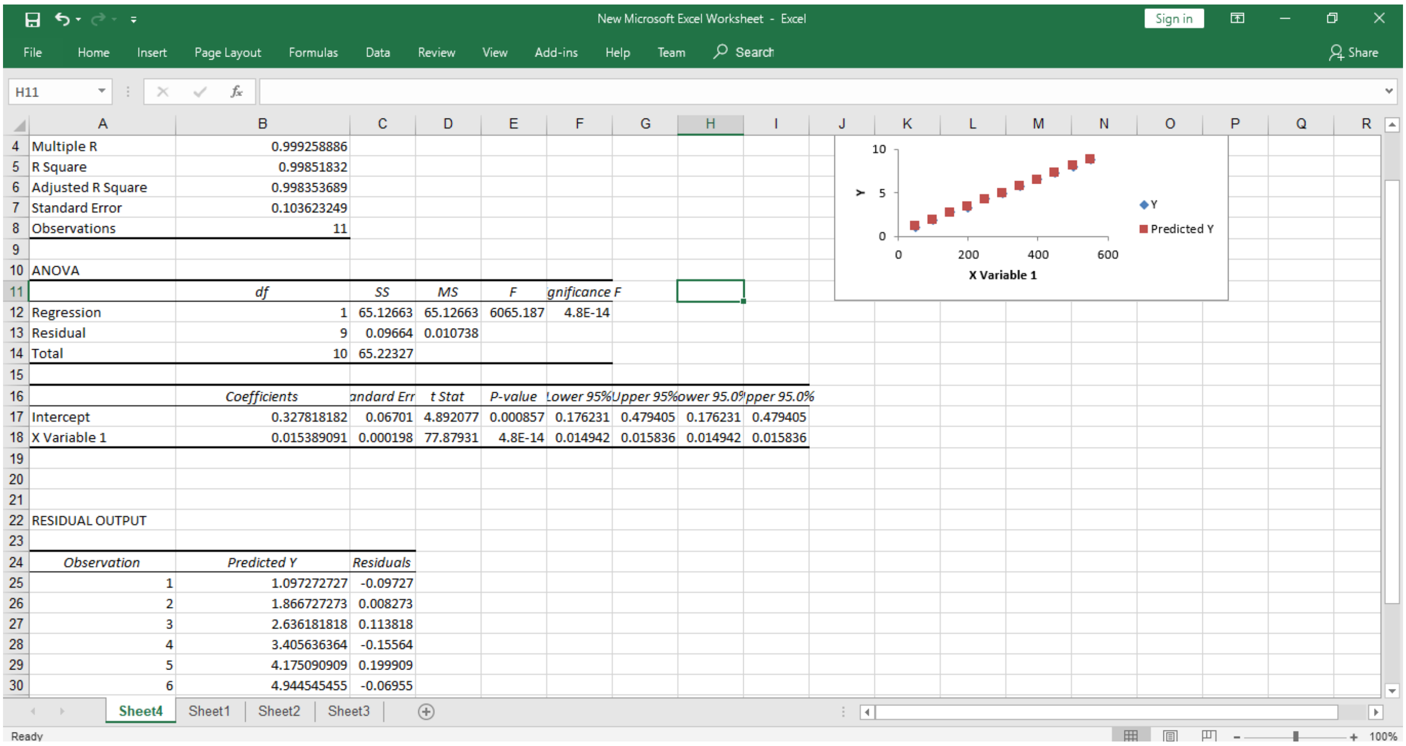
Mass	50	100	150	200	250	300	350	400	450	500	550
Elongation	1	1.875	2.75	3.25	4.375	4.875	5.765	6.5	7.25	8	8.75

Find a model that fits the above data well using A) Microsoft Excel B) MATLAB



USING EXCEL

1) Select Data and go to DATA tab and select Data analysis
2) Choose regression and select Y cells and X cells
3) Check confidence limit and select output cells.
4) Check mark line fit plot and press ok



Multiple R	0.999258886
R Square	0.99851832
Adjusted R Square	0.998353689
Standard Error	0.103623249
Observations	11

ANOVA

	df	SS	MS	F	gnificance F
Regression	1	65.12663	65.12663	6065.187	4.8E-14
Residual	9	0.09664	0.010738		
Total	10	65.22327			

	Coefficients	andard Err	t Stat	P-value	Lower 95%	Upper 95%	ower 95.0%	pper 95.0%
Intercept	0.327818182	0.06701	4.892077	0.000857	0.176231	0.479405	0.176231	0.479405
X Variable 1	0.015389091	0.000198	77.87931	4.8E-14	0.014942	0.015836	0.014942	0.015836

RESIDUAL OUTPUT

Observation	Predicted Y	Residuals
1	1.097272727	-0.09727
2	1.866727273	0.008273
3	2.636181818	0.113818
4	3.405636364	-0.15564
5	4.175090909	0.199909
6	4.944545455	-0.06955

USING MATLAB

```
x = [50:50:550];

y = [1,1.875,2.75,3.25,4.375,4.875,5.765,6.5,7.25,8,8.75];
```

```
disp('Linear Model Summary');
mdl = fitlm(x,y)
f = scatter(x,y);
hold on
ypred = mdl.feval(x);
h = plot(x,ypred);
legend('elongation vs mass','linear fit')
xlabel('Mass')
ylabel('Elongation')
hold off
```

OUTPUT

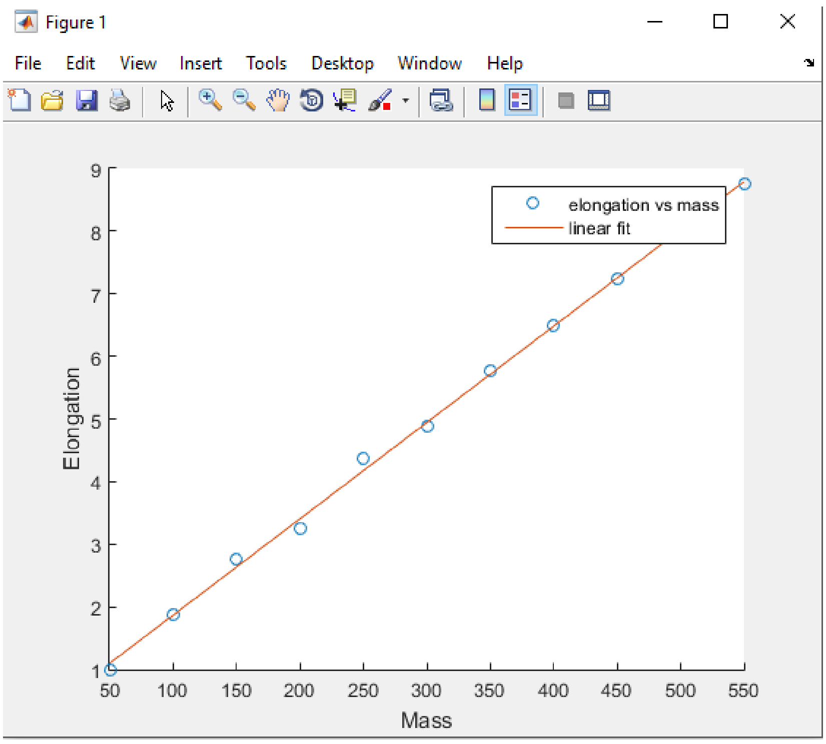
Linear Model Summary
mdl =

Linear regression model:
  y ~ 1 + x1

Estimated Coefficients:

	Estimate	SE	tStat	pValue
Intercept	0.32782	0.06701	4.8921	0.00085697
x1	0.015389	0.0001976	77.879	4.8022e-14

Number of observations: 11, Error degrees of freedom: 9
Root Mean Squared Error: 0.104
R-squared: 0.999, Adjusted R-Squared 0.998
F-statistic vs. constant model: 6.07e+03, p-value = 4.8e-14



Practical 2

AIM: The relation between the radius and the volume of a sphere is to be measured as table below-

Radius	1	2	3	4	5
Volume	4.19	33.51	113.10	268.08	523.60

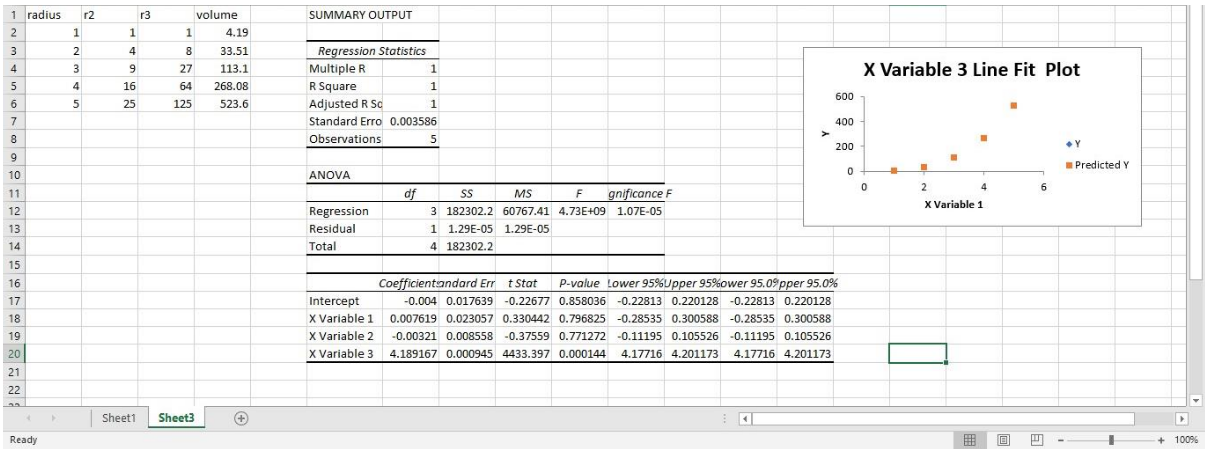
Use regressions to find the formula for the volume as a function of the radius. Fit a model for the given data using A) Microsoft Excel B) MATLAB

USING EXCEL

1) Select Data and go to DATA tab and select Data analysis
2) Choose regression and select Y cells and X cells
3) Check confidence limit and select output cells.
4) Check mark line fit plot and press ok

USING MATLAB

```
x = [1,2,3,4,5];
y = [4.19,33.51,113.10,268.08,523.60];
```



radius	r2	r3	volume
1	1	1	4.19
2	4	8	33.51
3	9	27	113.1
4	16	64	268.08
5	25	125	523.6

SUMMARY OUTPUT

Regression Statistics	
Multiple R	1
R Square	1
Adjusted R Sq	1
Standard Erro	0.003586
Observations	5

ANOVA

	df	*SS*	*MS*	*F*	*gnificance F*
Regression	3	182302.2	60767.41	4.73E+09	1.07E-05
Residual	1	1.29E-05	1.29E-05		
Total	4	182302.2			

	Coefficients	*andard Err*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	-0.004	0.017639	-0.22677	0.858036	-0.22813	0.220128	-0.22813	0.220128
X Variable 1	0.007619	0.023057	0.330442	0.796825	-0.28535	0.300588	-0.28535	0.300588
X Variable 2	-0.00321	0.008558	-0.37559	0.771272	-0.11195	0.105526	-0.11195	0.105526
X Variable 3	4.189167	0.000945	4433.397	0.000144	4.17716	4.201173	4.17716	4.201173


```
disp('Cubic Model Summary');
[mdl,gof] = fit(x',y','poly3')
f = scatter(x,y);
hold on
h = plot(mdl);
legend('Radius vs Volume','cubic fit')
xlabel('Radius')
```

```
ylabel('Volume')
hold off
```

OUTPUT

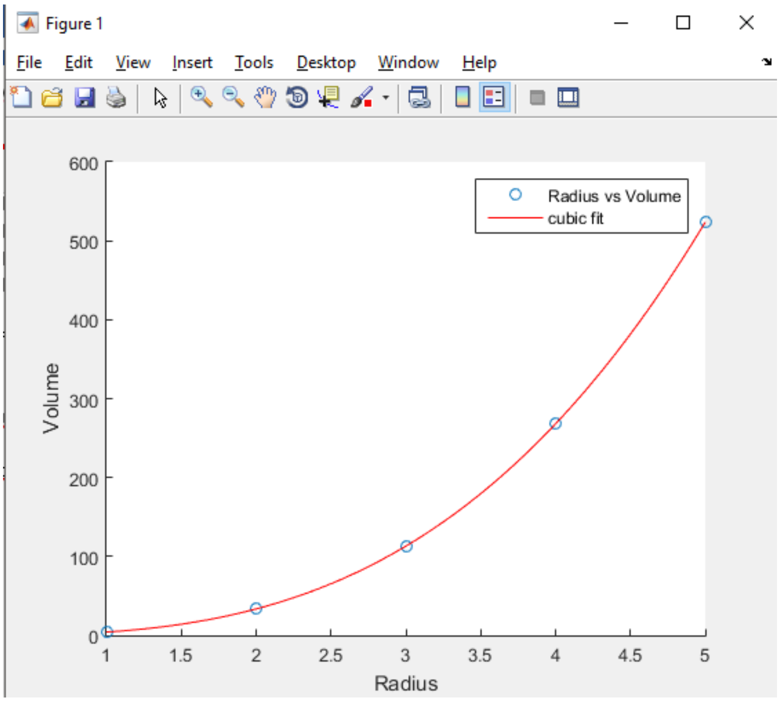
Cubic Model Summary

mdl =

Linear model Poly3:
mdl(x) = p1*x^3 + p2*x^2 + p3*x + p4
Coefficients (with 95% confidence bounds):
p1 = 4.189 (4.177, 4.201)
p2 = -0.003214 (-0.112, 0.1055)
p3 = 0.007619 (-0.2853, 0.3006)
p4 = -0.004 (-0.2281, 0.2201)

gof =

sse: 1.2857e-05
rsquare: 1.0000
dfe: 1
adjrsquare: 1.0000
rmse: 0.0036



Practical 3

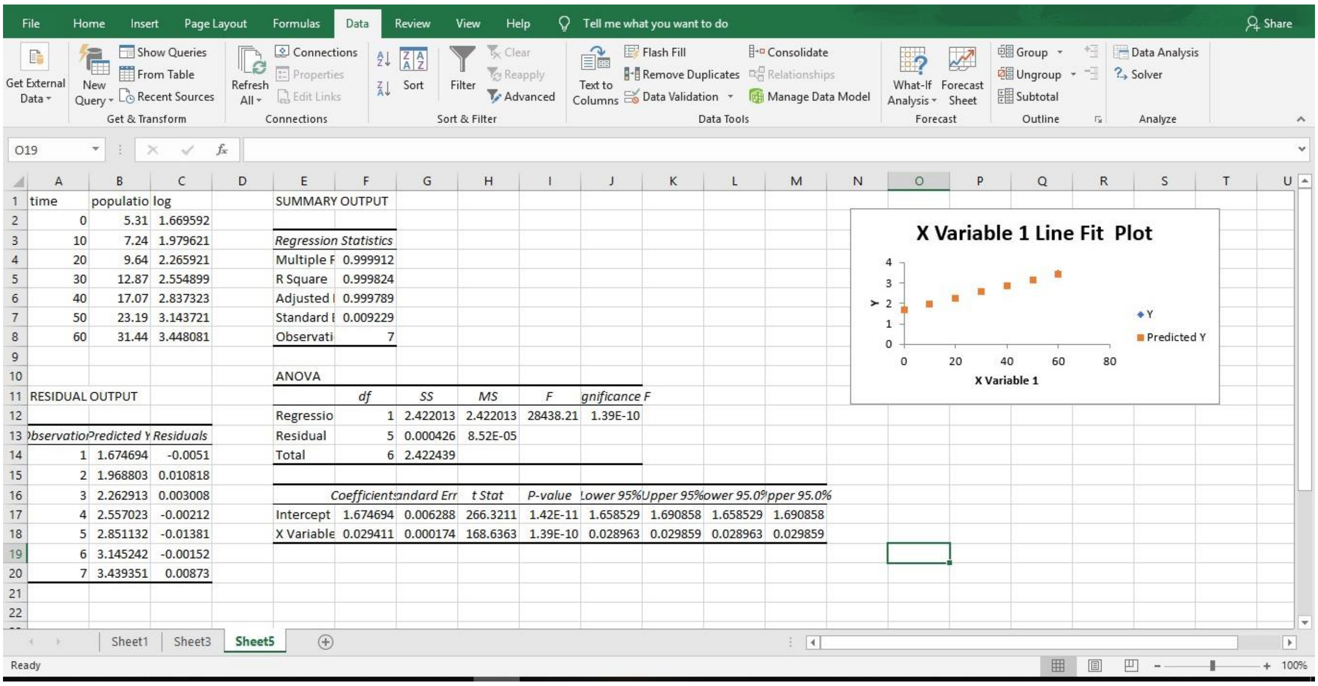
AIM: The size of population of US in 1800s has been measured and given in the table below, t=0 denotes year 1800.

Time	0	10	20	30	40	50	60
Population (millions)	5.31	7.24	9.64	12.87	17.07	23.19	31.44

Considering logarithm of P as a function of t, find a linear model for ln P and t. Fit exponential model for the given data using A) Excel B) MATLAB

USING EXCEL

1) Select Data and go to DATA tab and select Data analysis
2) Choose regression and select Y cells and X cells
3) Check confidence limit and select output cells.
4) Check mark line fit plot and press ok



USING MATLAB

```
x = [0:10:60];
y = [5.31,7.24,9.64,12.87,17.07,23.19,31.44];
p = log(y);
disp('Linear Model for ln(p) vs t');
lmdl = fitlm(x,p)
```

```
disp('Exponential Model for p vs t');
[emdl,gof] = fit(x',y','exp1')
f = scatter(x,y);
hold on
k = plot(emdl);
legend('Population vs Time','Exponential Fit')
xlabel('Time')
ylabel('Population')
```

OUTPUT

Linear Model for ln(p) vs t

lmdl =

Linear regression model:
y ~ 1 + x1

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	1.6747	0.0062882	266.32	1.4165e-11
x1	0.029411	0.0001744	168.64	1.3912e-10

Number of observations: 7, Error degrees of freedom:
5 Root Mean Squared Error: 0.00923
R-squared: 1, Adjusted R-Squared 1
F-statistic vs. constant model: 2.84e+04, p-value = 1.39e-

10 Exponential Model for p vs t

emdl =

General model Exp1:
emdl(x) = a*exp(b*x)
Coefficients (with 95% confidence bounds):
a = 5.277 (5.125, 5.428)
b = 0.02968 (0.02912, 0.03025)

gof =

sse: 0.1022
rsquare: 0.9998
dfe: 5
adjrsquare: 0.9998
rmse: 0.1430

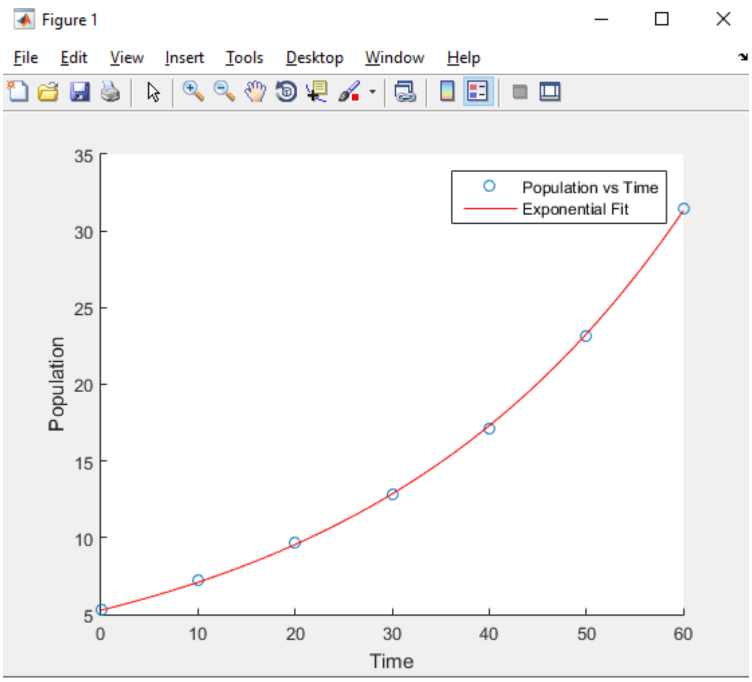


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Population vs Time
Exponential Fit
Population
Time
35
30
25
20
15
10
5
0
10
20
30
40
50
60

Practical 4

AIM: Given the following data points, obtain a natural cubic spline using Excel and MATLAB.

Given a set of data $(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_n, f(x_n))$

Assume $f''(x_0) = f''(x_n) = 0$,

other second derivatives can be evaluated by

$$(x_i - x_{i-1}) f''(x_{i-1}) + 2(x_{i+1} - x_{i-1}) f''(x_i) + (x_{i+1} - x_i) f''(x_{i+1}) = \frac{6}{(x_{i+1} - x_i)}[f(x_{i+1}) - f(x_i)] + \frac{6}{(x_i - x_{i-1})}[f(x_{i-1}) - f(x_i)]$$

$i = 1, 2, 3, \ldots, n-1$

The cubic function for each interval is

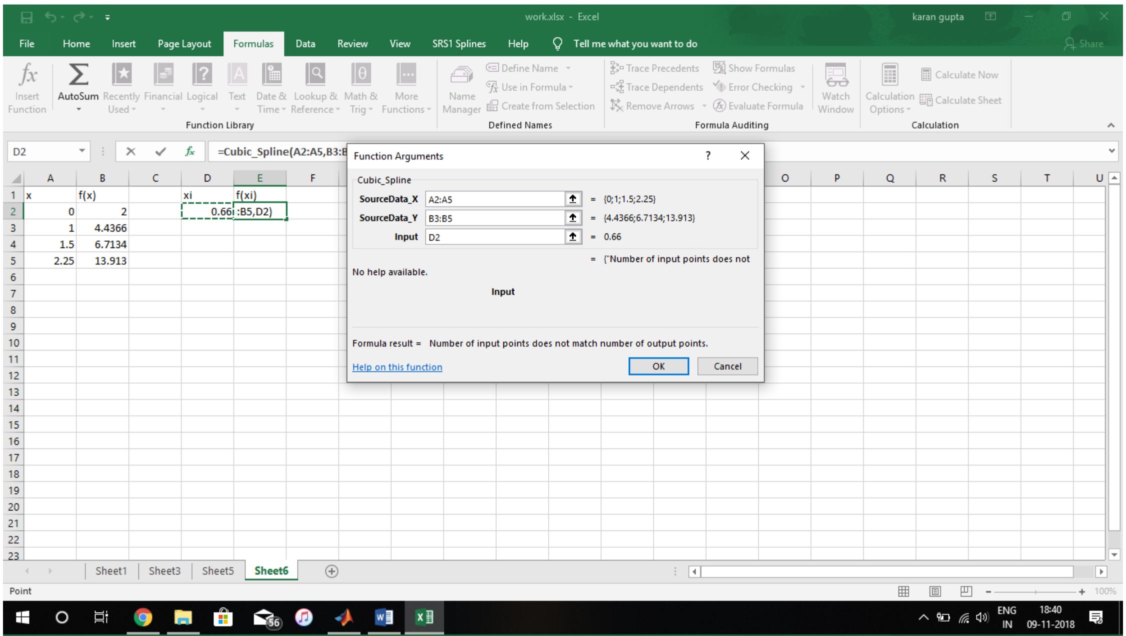
$$f_i(x) = \frac{f''(x_{i-1})}{6(x_i - x_{i-1})}(x_i - x)^3 + \frac{f''(x_i)}{6(x_i - x_{i-1})}(x - x_{i-1})^3 + \left[\frac{f(x_{i-1})}{x_i - x_{i-1}} - \frac{f''(x_{i-1})(x_i - x_{i-1})}{6}\right](x_i - x) + \left[\frac{f(x_i)}{x_i - x_{i-1}} - \frac{f''(x_i)(x_i - x_{i-1})}{6}\right](x - x_{i-1})$$

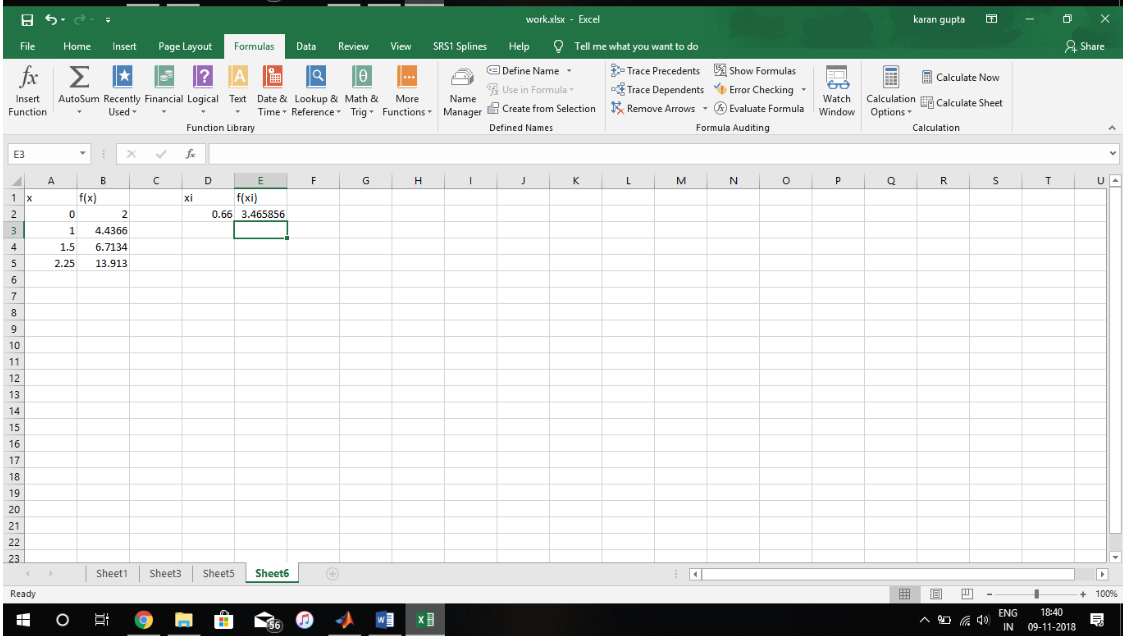
X	0	1	1.5	2.25
F(x)	2	4.4366	6.7134	13.9130

Also compute the spline value of f(0.66).

USING EXCEL

1) Install SRS1 Cubic spline extension into excel and enter the data
2) Select output cell and go to formulas and insert function
3) Select category as SRS1Splines.Functions25 and select x , y and input cells


work.xlsx - Excel
=Cubic_Spline(A2:A5,B3:B
Function Arguments
Cubic_Spline
SourceData_X A2:A5 = {0;1;1.5;2.25}
SourceData_Y B3:B5 = {4.4366;6.7134;13.913}
Input D2 = 0.66
= {"Number of input points does not
No help available.
Input
Formula result = Number of input points does not match number of output points.
Help on this function
OK
Cancel
x
f(x)
0
2
1
4.4366
1.5
6.7134
2.25
13.913
xi
f(xi)
0.66
:B5,D2)
Sheet1
Sheet3
Sheet5
Sheet6
Point


work.xlsx - Excel
x
f(x)
0
2
1
4.4366
1.5
6.7134
2.25
13.913
xi
f(xi)
0.66
3.465856
Sheet1
Sheet3
Sheet5
Sheet6
Ready

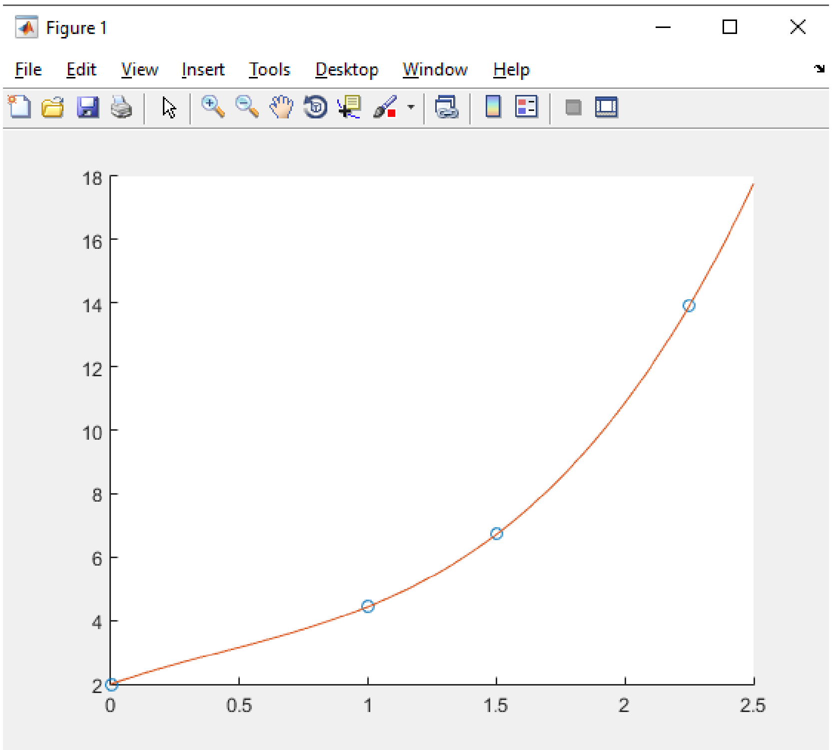
USING MATLAB

```
x = [0,1,1.5,2.25];
y = [2,4.4366,6.7134,13.9130];
xin = [0.66];
f = spline(x,y,xin)
xplot = [0:0.01:2.5];
yplot = spline(x,y,xplot);
h = scatter(x,y);
hold on
k = plot(xplot,yplot);
hold off
```

OUTPUT

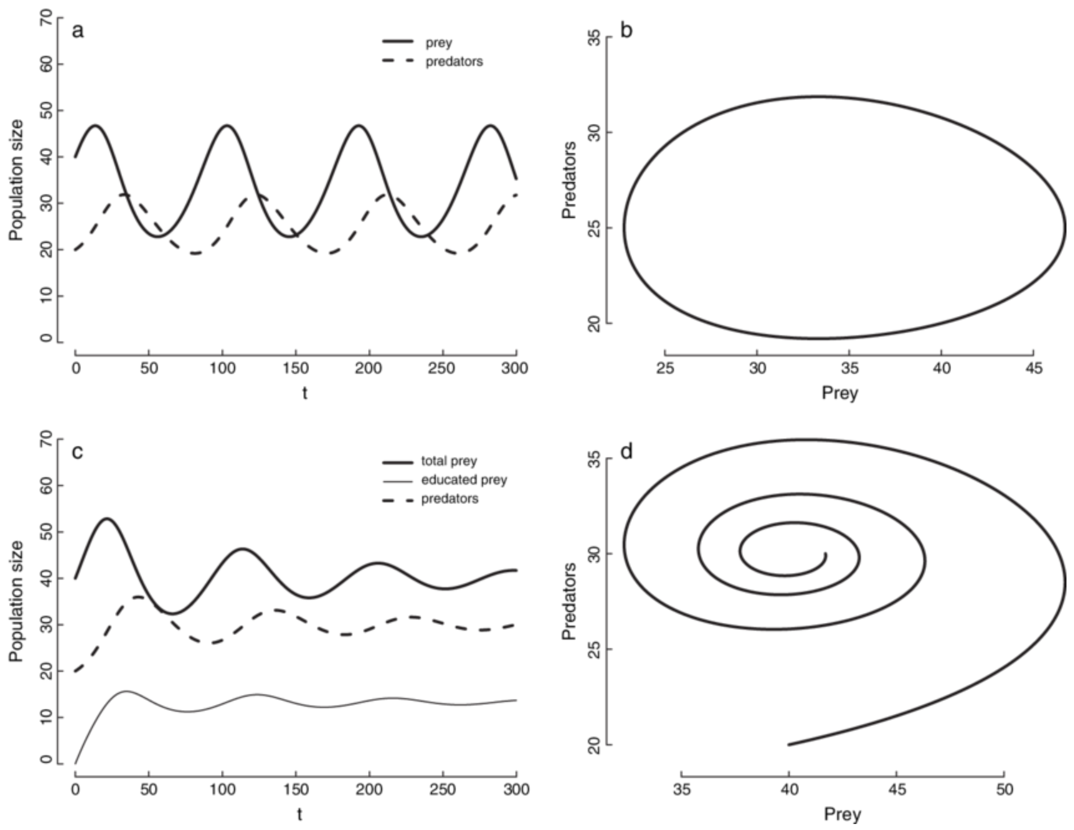
f =

  3.5114



Practical 5

AIM: Simulate prey-predator model using MATLAB.



CODE:

FUNCTION FILE FOR EQUATIONS:

```
function [dydt] = prey_predator(t,y,a,b)
    dydt = zeros(2,1);
    dydt(1) = y(1)*(a(1) - y(2)*b(1));
    dydt(2) = y(2)*(-a(2) + y(1)*b(2));
end
```

COMMAND WINDOW:

```
a = [1;1];
b = [0.1;0.1];
c0 = [0.1;0.1];
options = odeset('AbsTol',1e-20);
[t,c] = ode15s(@prey_predator,[0 100],c0,options,a,b);
plot (t,c(:,1)','-',t,c(:,2)','--');
legend ('prey','predator');
xlabel ('time');
```

OUTPUT:

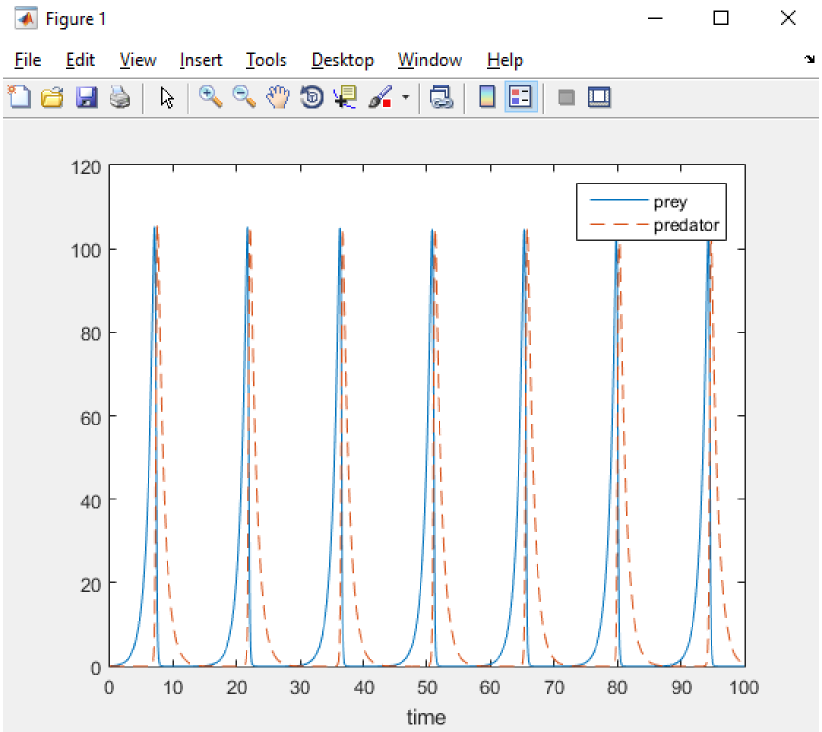
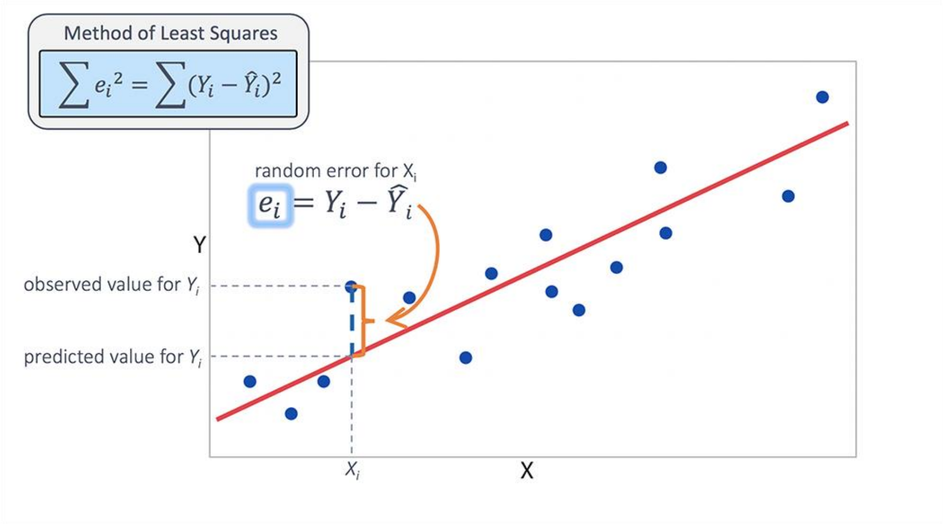


Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
120
100
80
60
40
20
0
0
10
20
30
40
50
60
70
80
90
100
time
prey
predator

Practical 6:

AIM: Program for multiple regression using R



Program:

```
input <- mtcars[,c("mpg","disp","hp","wt")]

# Create the relationship model.
model <- lm(mpg~disp+hp+wt, data = input)

# Show the model.
print(model)

# Get the Intercept and coefficients as vector elements.
cat("# # # # The Coefficient Values # # # ","\n")

a <- coef(model)[1]
print(a)

Xdisp <- coef(model)[2]
Xhp <- coef(model)[3]
Xwt <- coef(model)[4]

print(Xdisp)
print(Xhp)
print(Xwt)
```

Output

```
$Rscript main.r

Call:
lm(formula = mpg ~ disp + hp + wt, data = input)

Coefficients:
(Intercept)          disp             hp             wt
```

```
  37.105505    -0.000937    -0.031157    -3.800891

# # # # The Coefficient Values # # #
(Intercept)
   37.10551
         disp
-0.0009370091
         hp
-0.03115655
       wt
-3.800891
```

Practical 7

AIM: Program for Statistical Analysis of Data using R

Program:

```r
# load in packages we'll use
library(tidyverse) # utility functions
library(rpart) # for regression trees
library(randomForest) # for random forests

# read the data and store data in DataFrame titled melbourne_data
melbourne_data <- read_csv("../input/melb_data.csv")
```

```r
# print a summary of the data in Melbourne data
summary(melbourne_data)
```

Output:

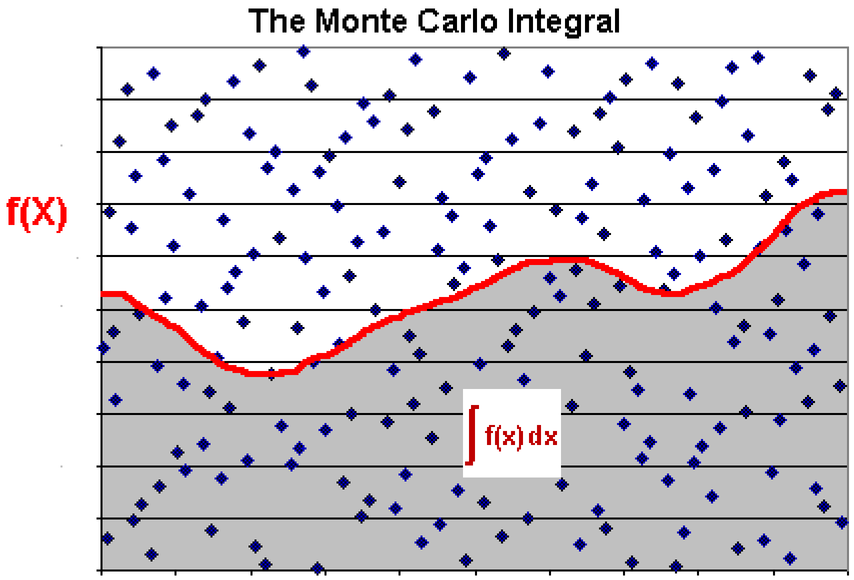
```
       X1            Suburb             Address              Rooms
 Min.   :    1   Length:18396       Length:18396       Min.   : 1.000
 1st Qu.: 5937   Class :character   Class :character   1st Qu.: 2.000
 Median :11820   Mode  :character   Mode  :character   Median : 3.000
 Mean   :11827                                         Mean   : 2.935
 3rd Qu.:17734                                         3rd Qu.: 3.000
 Max.   :23546                                         Max.   :12.000

     Type               Price             Method             SellerG
 Length:18396       Min.   :  85000   Length:18396       Length:18396
 Class :character   1st Qu.: 633000   Class :character   Class :character
 Mode  :character   Median : 880000   Mode  :character   Mode  :character
                    Mean   :1056697
                    3rd Qu.:1302000
                    Max.   :9000000

     Date              Distance        Postcode       Bedroom2
 Length:18396       Min.   : 0.00   Min.   :3000   Min.   : 0.000
 Class :character   1st Qu.: 6.30   1st Qu.:3046   1st Qu.: 2.000
 Mode  :character   Median : 9.70   Median :3085   Median : 3.000
                    Mean   :10.39   Mean   :3107   Mean   : 2.913
                    3rd Qu.:13.30   3rd Qu.:3149   3rd Qu.: 3.000
                    Max.   :48.10   Max.   :3978   Max.   :20.000
                    NA's   :1       NA's   :1      NA's   :3469
    Bathroom          Car            Landsize          BuildingArea
 Min.   :0.000   Min.   : 0.000   Min.   :     0.0   Min.   :    0.0
 1st Qu.:1.000   1st Qu.: 1.000   1st Qu.:   176.5   1st Qu.:   93.0
 Median :1.000   Median : 2.000   Median :   440.0   Median :  126.0
 Mean   :1.538   Mean   : 1.616   Mean   :   558.1   Mean   :  151.2
 3rd Qu.:2.000   3rd Qu.: 2.000   3rd Qu.:   651.0   3rd Qu.:  174.0
 Max.   :8.000   Max.   :10.000   Max.   :433014.0   Max.   :44515.0
 NA's   :3471    NA's   :3576     NA's   :4793       NA's   :10634
   YearBuilt      CouncilArea          Lattitude         Longtitude
```

Practical 8

AIM: Write a program using Monte-Carlo method to find approximate integration of f(x) = x, x^2, cos(πx)



[Ratio of dots below the Red line to all the points is the answer]

CODE:

FUNCTION:

```
function [value] = mmi(f,a,b)
    n = 0;
    for i=1:1000000
        x = (b-a).*rand + a;
        n = n + f(x);
    end
    value = ((b-a)/1000000)*n;
end
```

COMMAND:

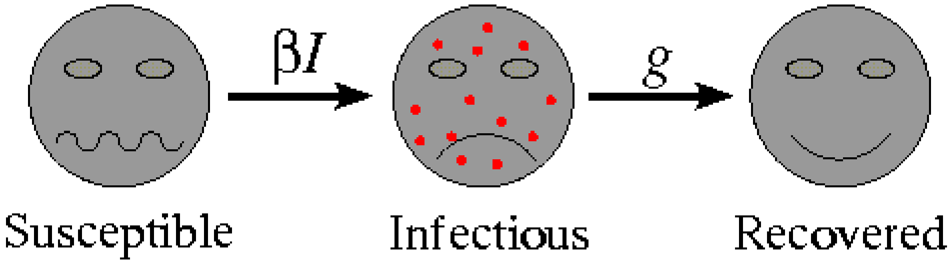
```
>> mmi(@(x) x,0,10)
ans =
  49.9396

>> mmi(@(x) x.^2,0,2)
ans =
   2.6703

>> mmi(@(x) cos(3.14*x),0,5)
ans =
   0.0088
```

Practical 9

AIM: Simulate a model for epidemics using MATLAB



CODE:

FUNCTION FILE FOR EQUATIONS:

```
function [dydt] = sir(t,y)
    a = 0.01;
    b = 0.1;
    dydt(1) =-a*y(1)*y(2);
    dydt(2) = a*y(1)*y(2)-b*y(2);
    dydt(3) = b*y(2);
    dydt = [dydt(1) dydt(2) dydt(3)]';
end
```

COMMAND WINDOW:

```
y0 = [99 1 0];
a = 0.01;
b = 0.1;
[t,y] = ode45('sir',[0 50],y0);
plot(t,y(:,1),t,y(:,2),t,y(:,3))
xlabel('time')
ylabel('values')
legend('susceptible','infected', 'recovered')
```

<u>OUTPUT</u>

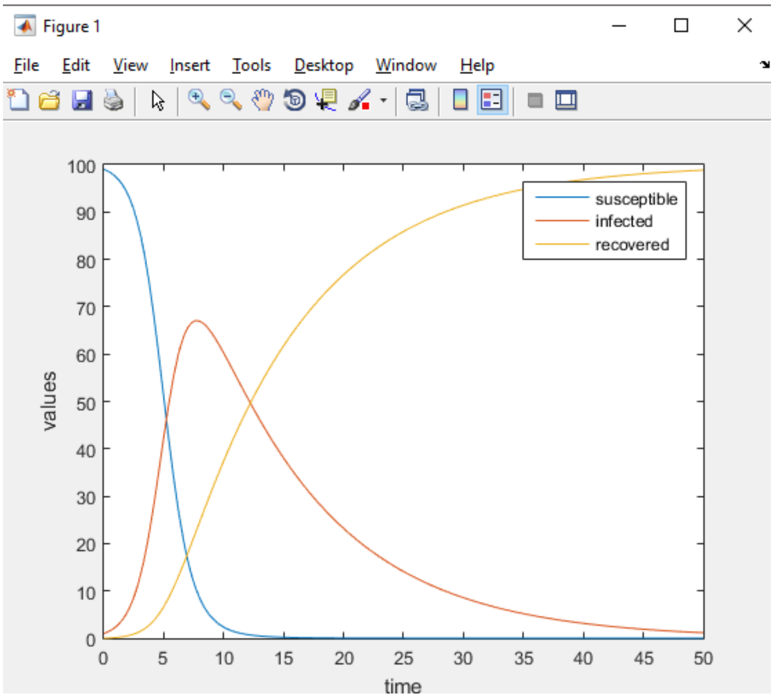


Figure 1
File Edit View Insert Tools Desktop Window Help
susceptible
infected
recovered
values
time
0
5
10
15
20
25
30
35
40
45
50
100
90
80
70
60
50
40
30
20
10