Content

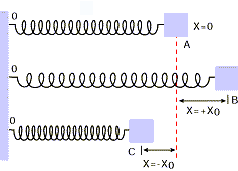
|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Practical** | **Date** | **Remarks** |
| **1** | Fitting a model to the dataset of mass and elongation of a spring |  |  |
| **2** | Fitting a model to the dataset of radius and volume of a sphere |  |  |
| **3** | Fitting a model to the dataset of 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.

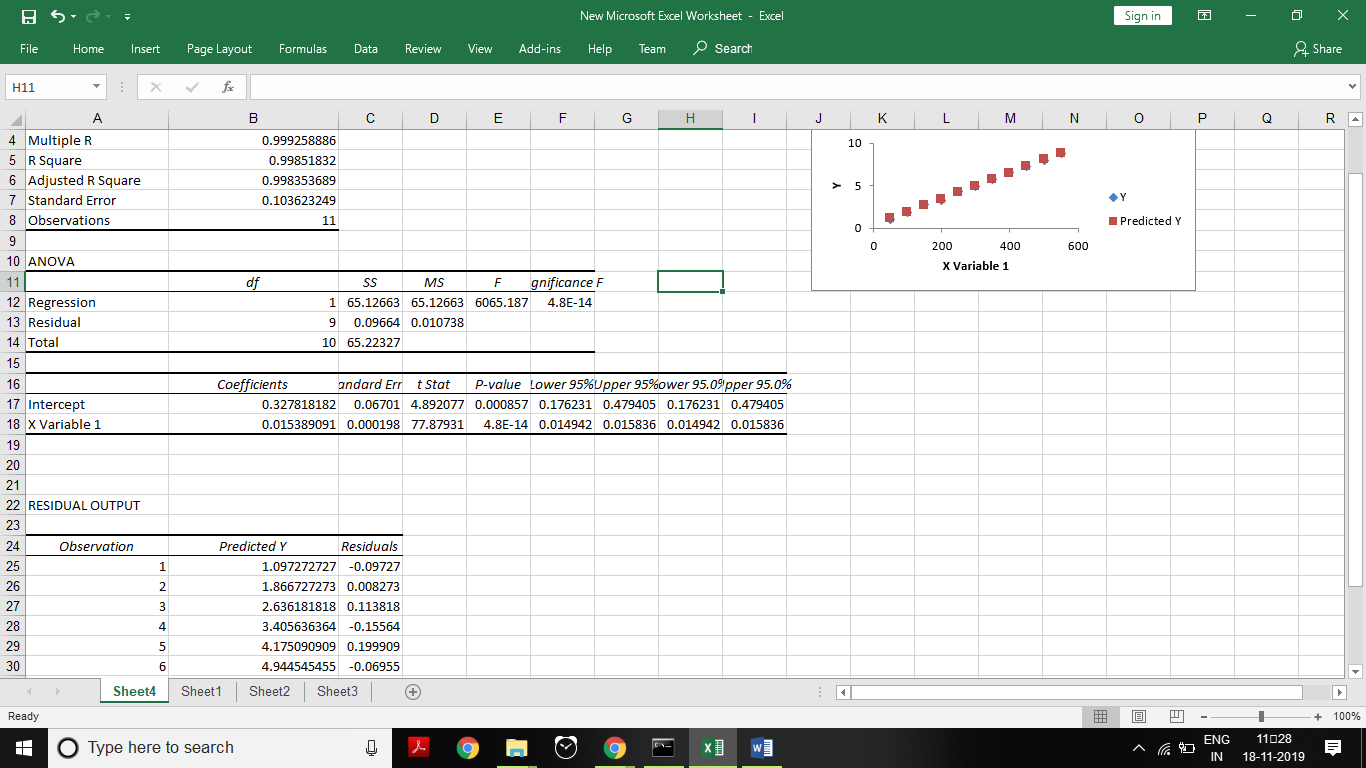
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mass | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 | 550 |
| Elongation | 1 | 1.875 | 2.75 | 3.25 | 4.375 | 4.875 | 5.765 | 6.5 | 7.25 | 8 | 8.75 |

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



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:

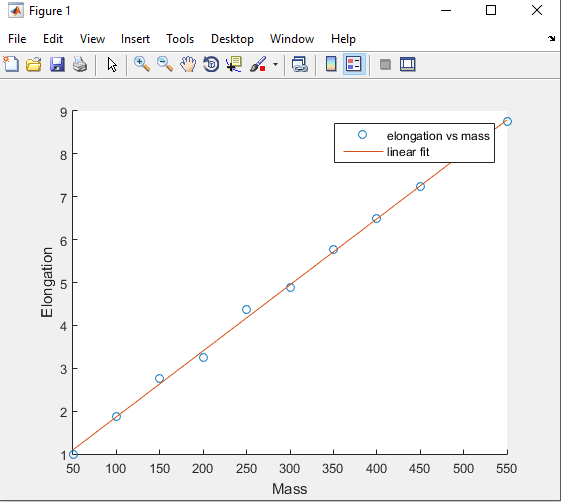
## Estimate SE tStat pValue

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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

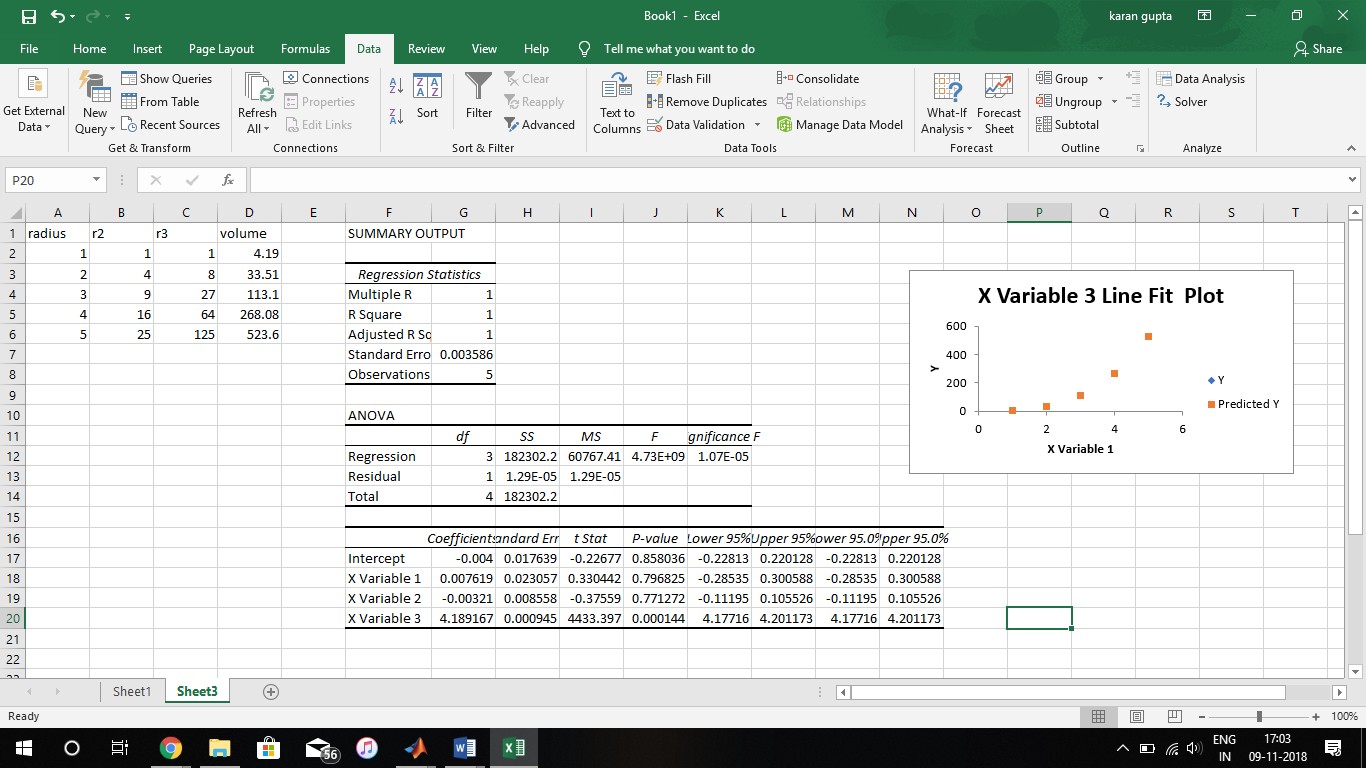
AIM**:** The relation between the radius and the volume of a sphere is to be measured as table below-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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];

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** =

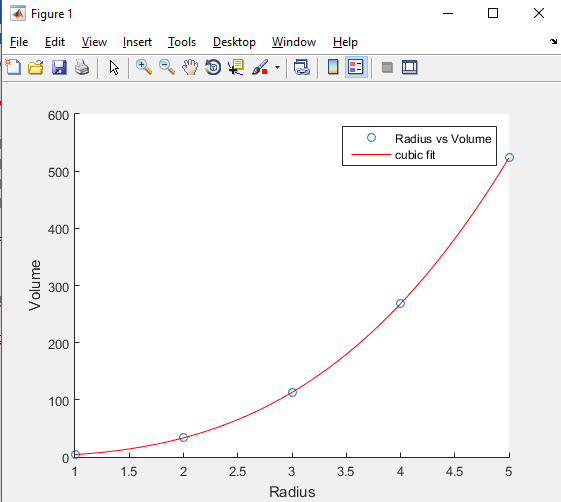
**sse**: 1.2857e-05

**rsquare**: 1.0000

**dfe**: 1

**adjrsquare**: 1.0000

**rmse**: 0.0036



Practical 3

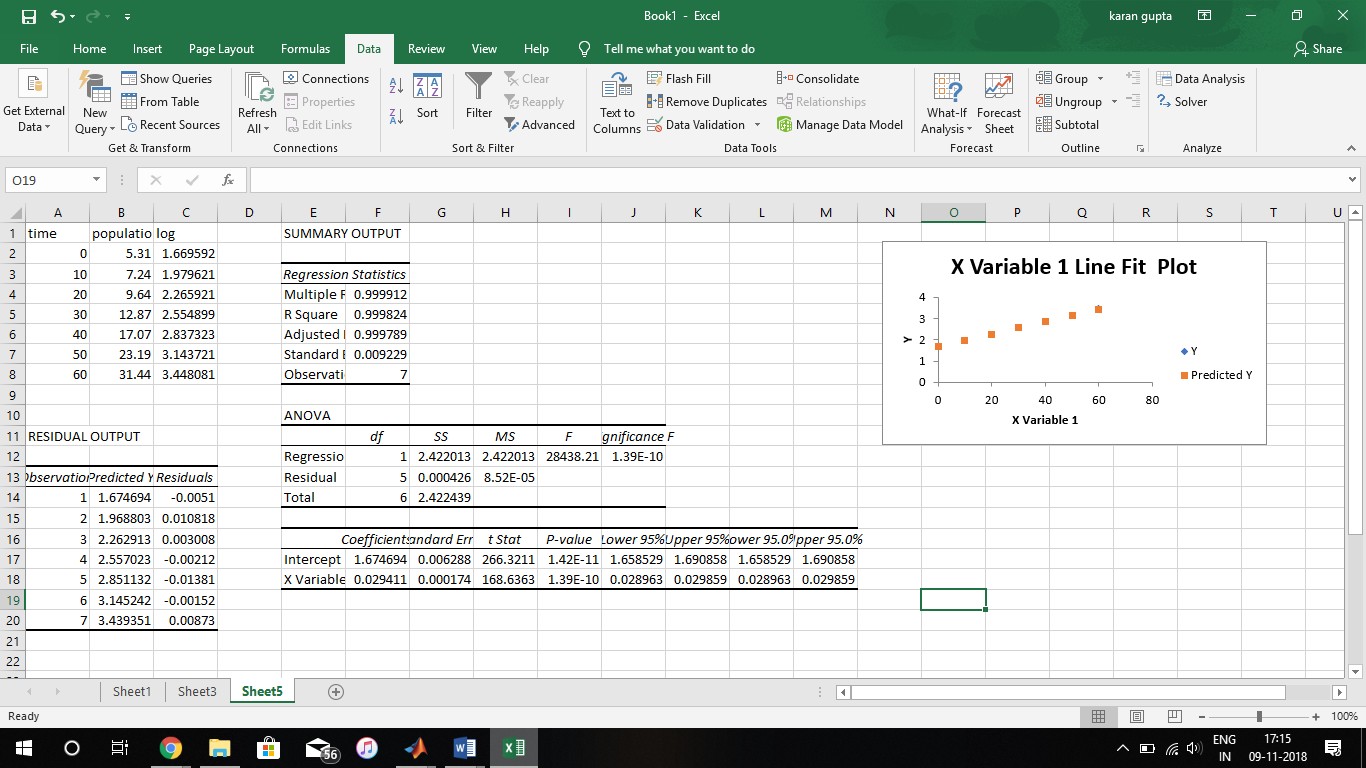
AIM**:** The size of population of US in 1800s has been measured and given in the table below, t=0 denotes year 1800.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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** =

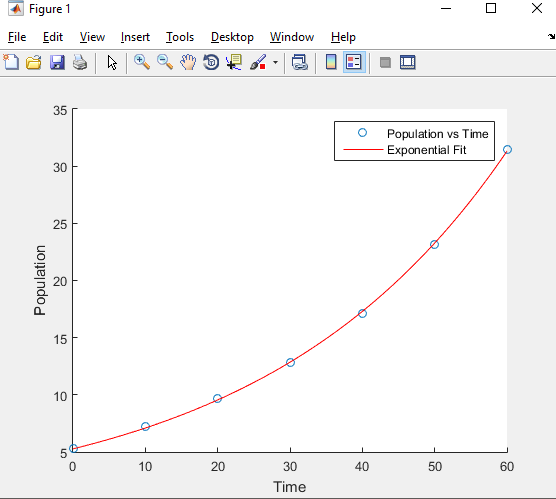
sse: 0.1022

rsquare: 0.9998

dfe: 5

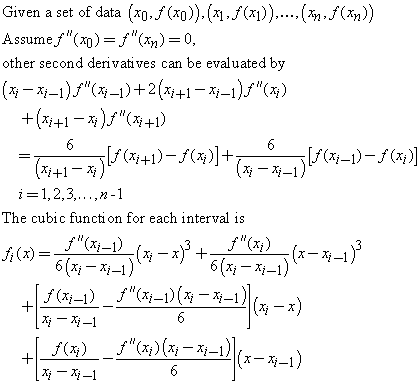
adjrsquare: 0.9998

rmse: 0.1430



Practical 4

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

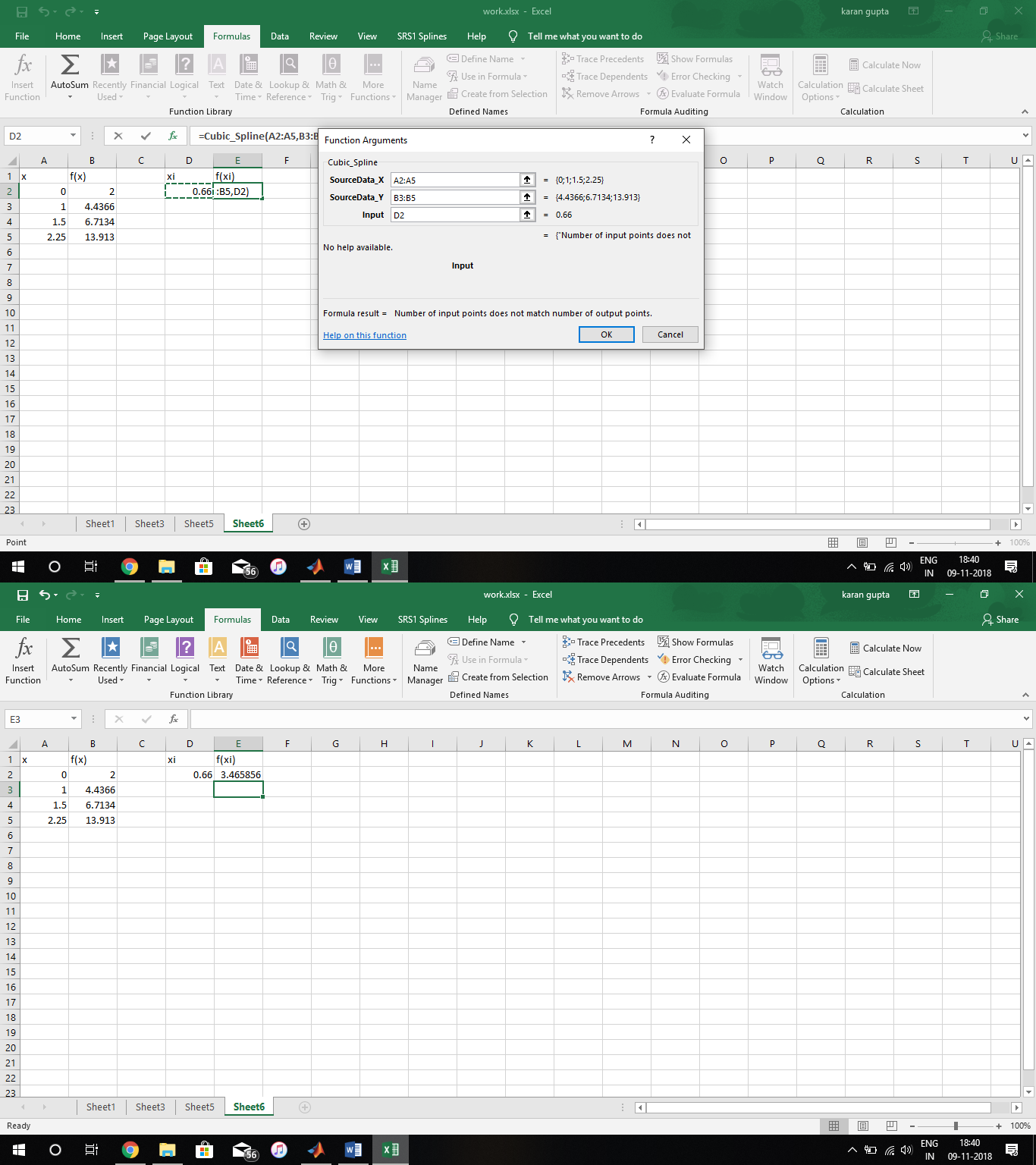


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | 0 | 1 | 1.5 | 2.25 |
| F(x) | 2 | 4.4366 | 6.7134 | 13.9130 |

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



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];

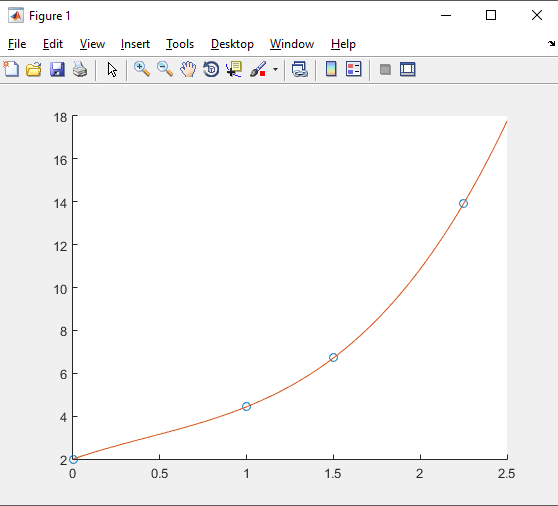
yplot = spline(x,y,xplot); h = scatter(x,y);

hold on

k = plot(xplot,yplot); hold off

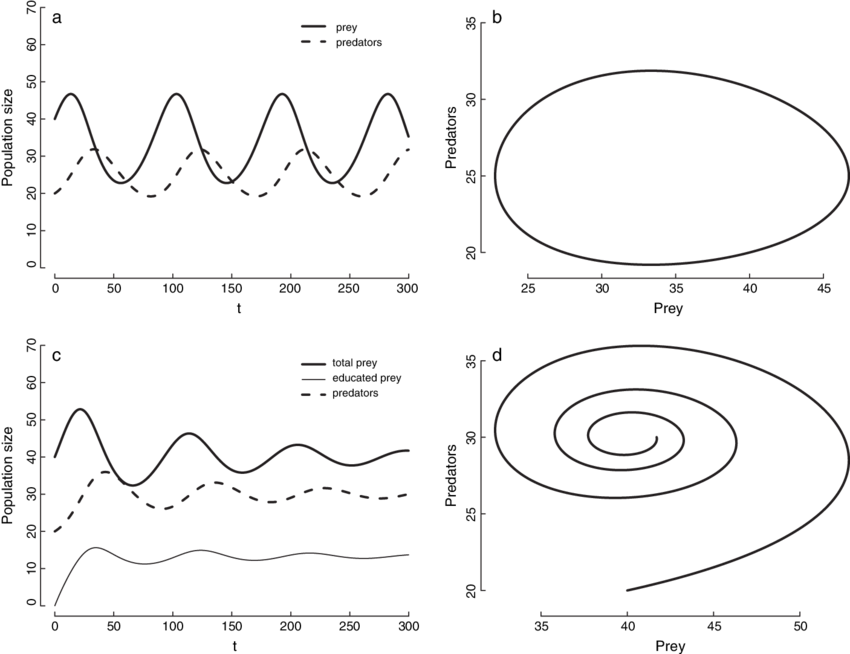
OUTPUT

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];

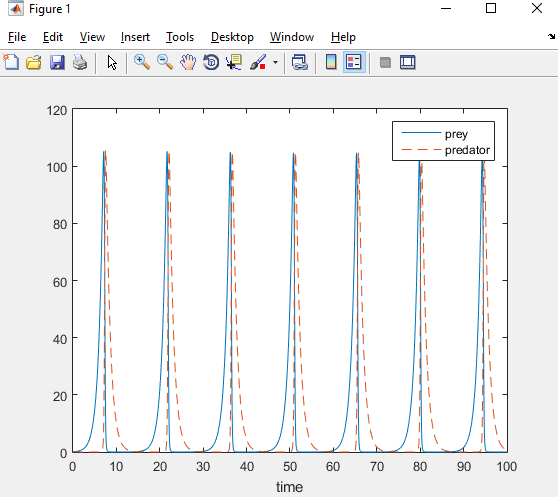
c0 = [0.1;0.1];

options = odeset('AbsTol',1e-20);

[t,c] = ode15s(@prey\_predator,[0 100],c0,options,a,b); plot (t,c(:,1)','-',t,c(:,2)','--');

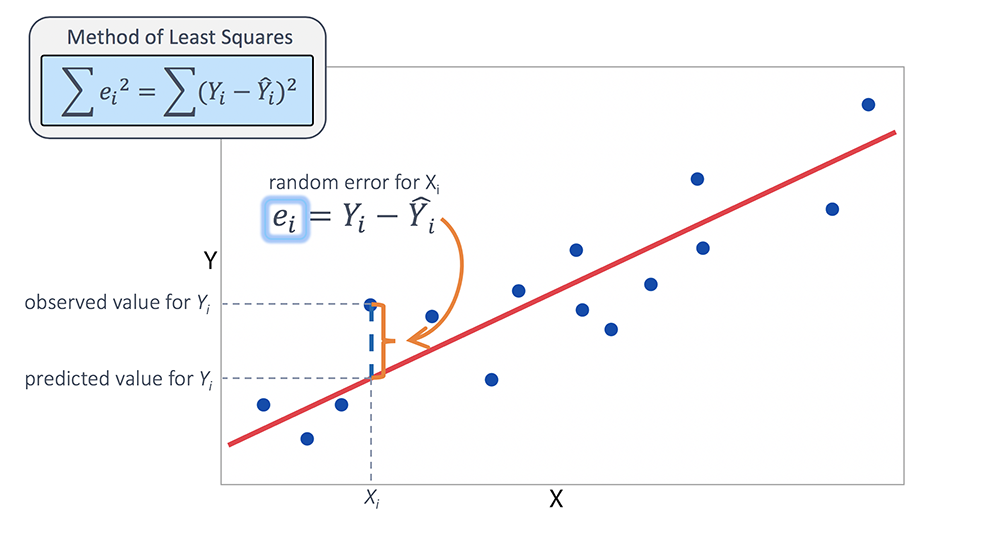
legend ('prey','predator'); xlabel ('time');

OUTPUT:



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**:

*# load in packages we'll use*

library(tidyverse) *# utility functions*

library(rpart) *# for regression trees*

library(randomForest) *# for random forests*

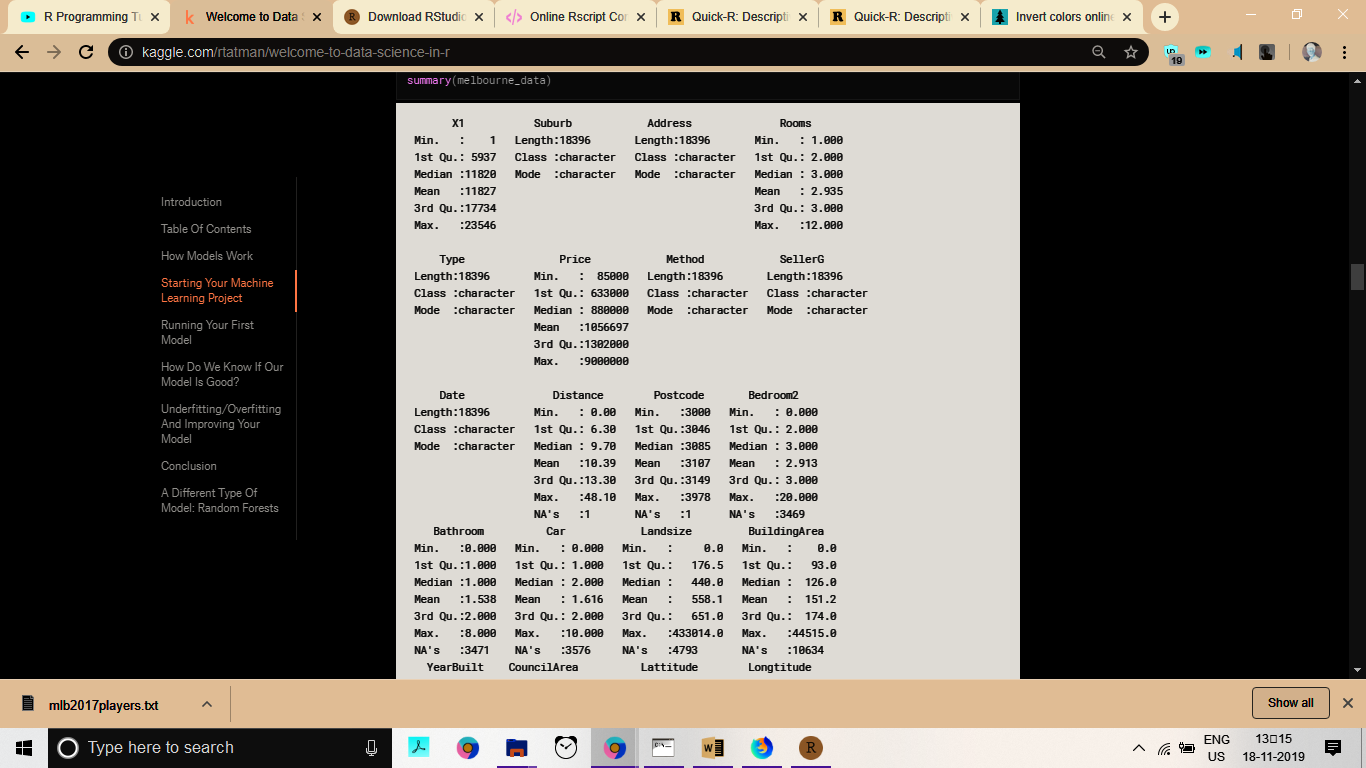
*# read the data and store data in DataFrame titled melbourne\_data*

melbourne\_data <- read\_csv("../input/melb\_data.csv")

*# print a summary of the data in Melbourne data*

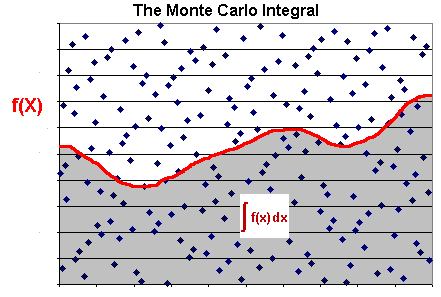
summary(melbourne\_data)

**Output:**



Practical 8

AIM**:** Write a program using Monte-Carlo method to find approximate integration of f(x) = x, x2, 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

end

value = ((b-a)/1000000)\*n;

## COMMAND:

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

49.9396

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

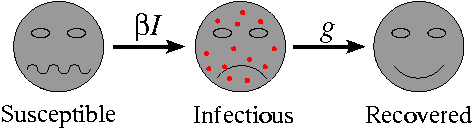
2.6703

>> 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')

**OUTPUT**

