DELHI TECHNOLOGICAL UNIVERSITY

MATHEMATICAL MODELLING AND SIMULATION

MC-407

Practical File

SUBMITTED BY:

Aiman Siddiqua

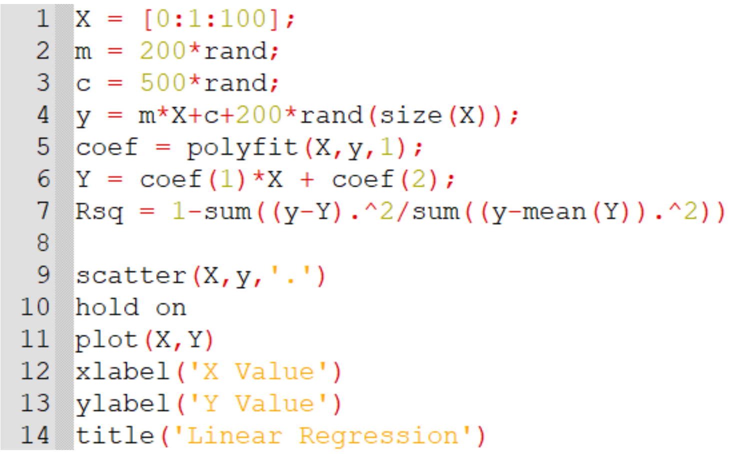
2K18/MC/008

INDEX

PRACTICAL – 1

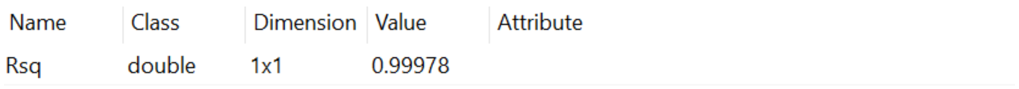
AIM: To write a program for linear fit of the curve using MATLAB/Octave.

CODE:

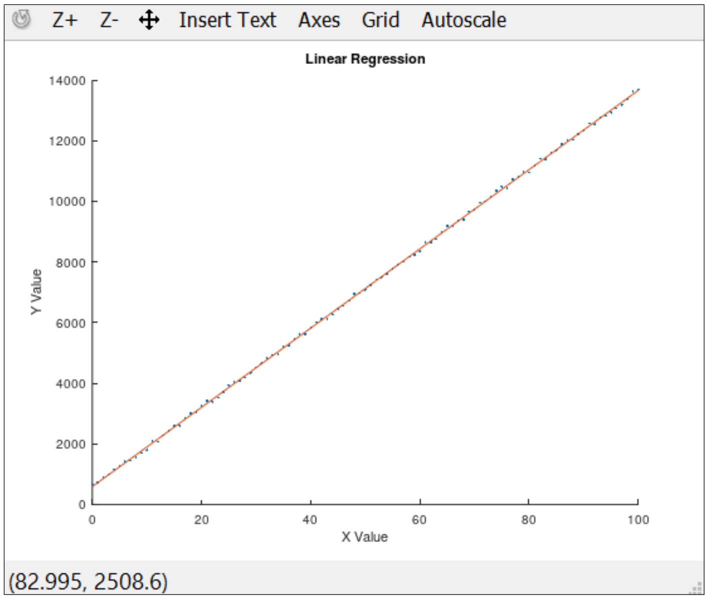


```
X = [0:1:100];
m = 200*rand;
c = 500*rand;
y = m*X+c+200*rand(size(X));
coef = polyfit(X,y,1);
Y = coef(1)*X + coef(2);
Rsq = 1-sum((y-Y).^2/sum((y-mean(Y)).^2))

scatter(X,y,'.')
hold on
plot(X,Y)
xlabel('X Value')
ylabel('Y Value')
title('Linear Regression')
```

OUTPUT:



Name	Class	Dimension	Value	Attribute
Rsq	double	1x1	0.99978	

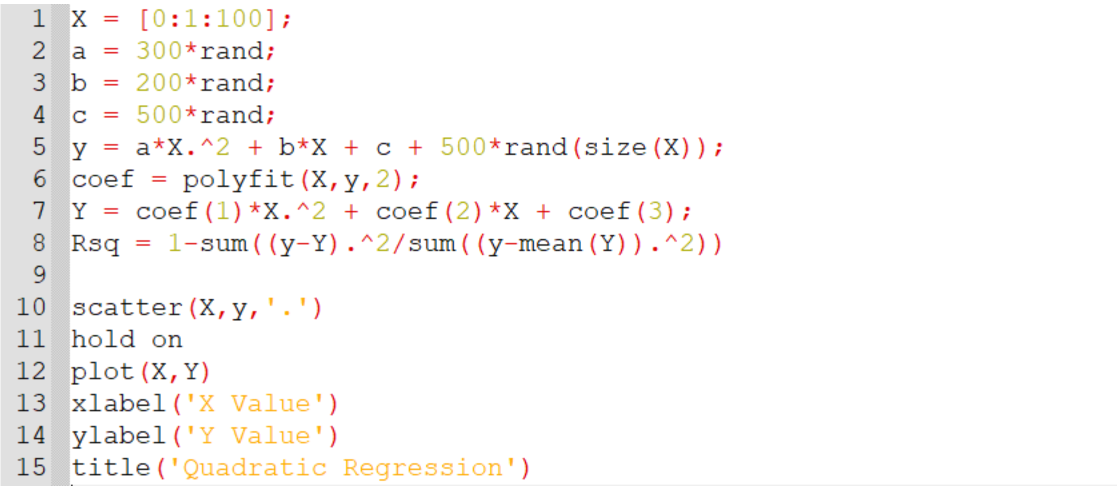


(82.995, 2508.6)

PRACTICAL – 2

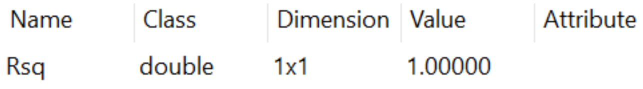
AIM: To write a program for quadratic fit of the curve using MATLAB/Octave.

CODE:

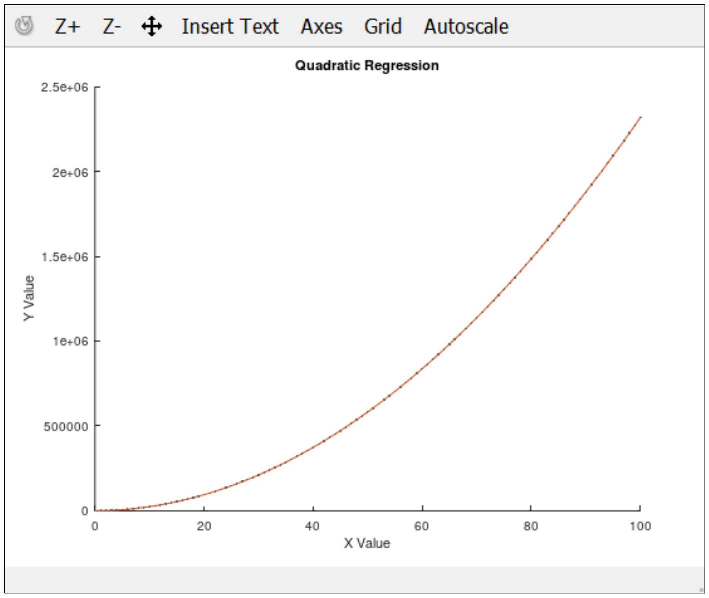


```
X = [0:1:100];
a = 300*rand;
b = 200*rand;
c = 500*rand;
y = a*X.^2 + b*X + c + 500*rand(size(X));
coef = polyfit(X,y,2);
Y = coef(1)*X.^2 + coef(2)*X + coef(3);
Rsq = 1-sum((y-Y).^2/sum((y-mean(Y)).^2))

scatter(X,y,'.')
hold on
plot(X,Y)
xlabel('X Value')
ylabel('Y Value')
title('Quadratic Regression')
```

OUTPUT:



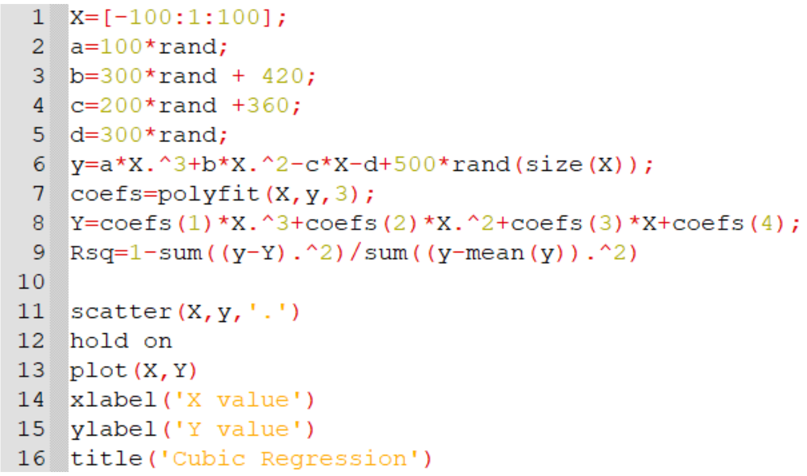
Name	Class	Dimension	Value	Attribute
Rsq	double	1x1	1.00000	



PRACTICAL – 3

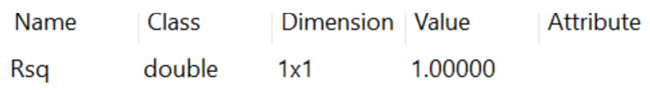
AIM: To write a program for cubic fit of the curve using MATLAB/Octave.

CODE:

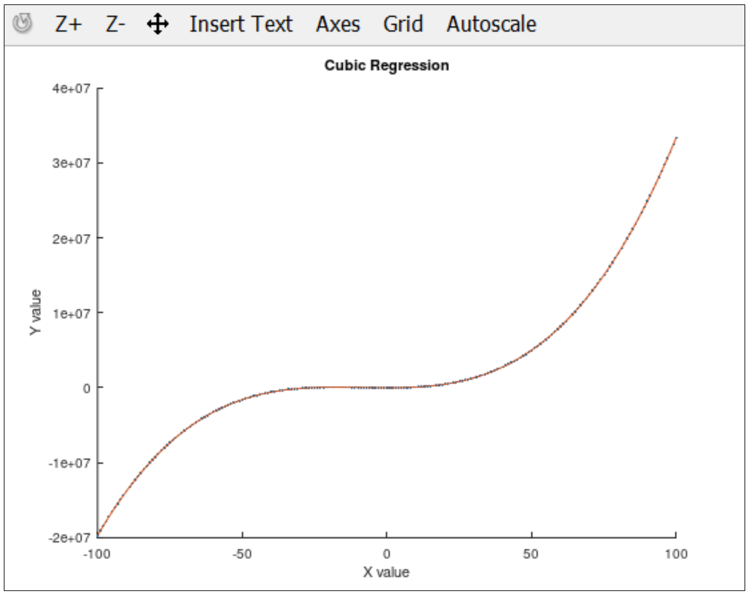


```
X=[-100:1:100];
a=100*rand;
b=300*rand + 420;
c=200*rand +360;
d=300*rand;
y=a*X.^3+b*X.^2-c*X-d+500*rand(size(X));
coefs=polyfit(X,y,3);
Y=coefs(1)*X.^3+coefs(2)*X.^2+coefs(3)*X+coefs(4);
Rsq=1-sum((y-Y).^2)/sum((y-mean(y)).^2)

scatter(X,y,'.')
hold on
plot(X,Y)
xlabel('X value')
ylabel('Y value')
title('Cubic Regression')
```

OUTPUT:



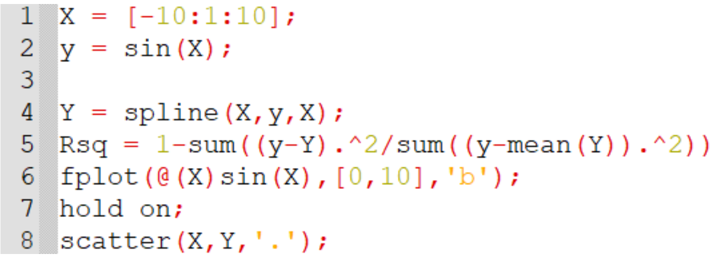
Name	Class	Dimension	Value	Attribute
Rsq	double	1x1	1.00000	



PRACTICAL – 4

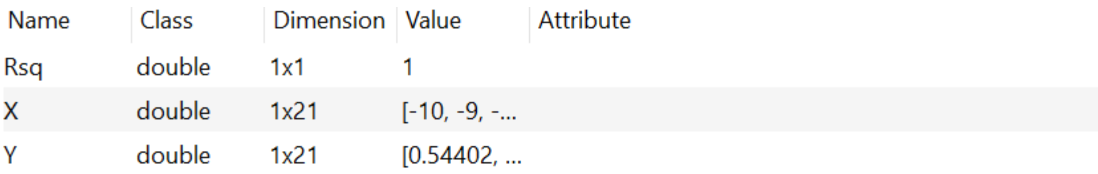
AIM: To write a program to find the cubic spline for a data and find its R2.

CODE:

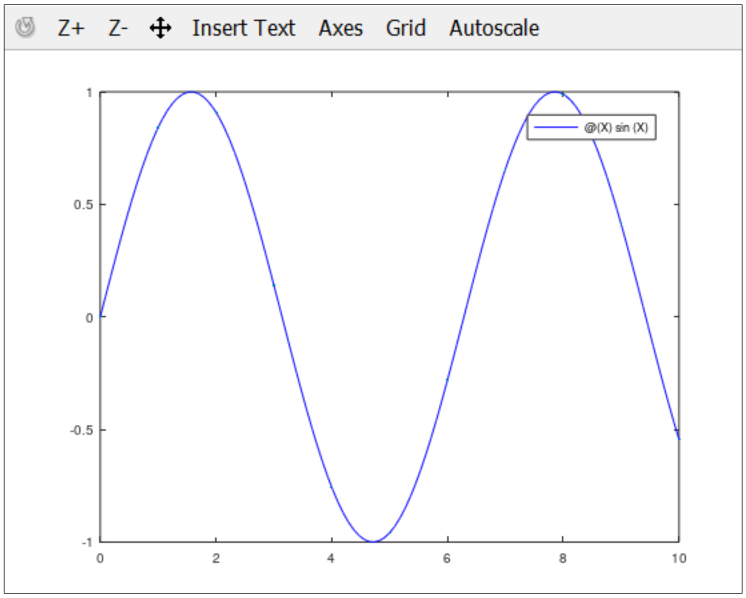


```
X = [-10:1:10];
y = sin(X);

Y = spline(X,y,X);
Rsq = 1-sum((y-Y).^2/sum((y-mean(Y)).^2))
fplot(@(X)sin(X),[0,10],'b');
hold on;
scatter(X,Y,'.');
```

OUTPUT:



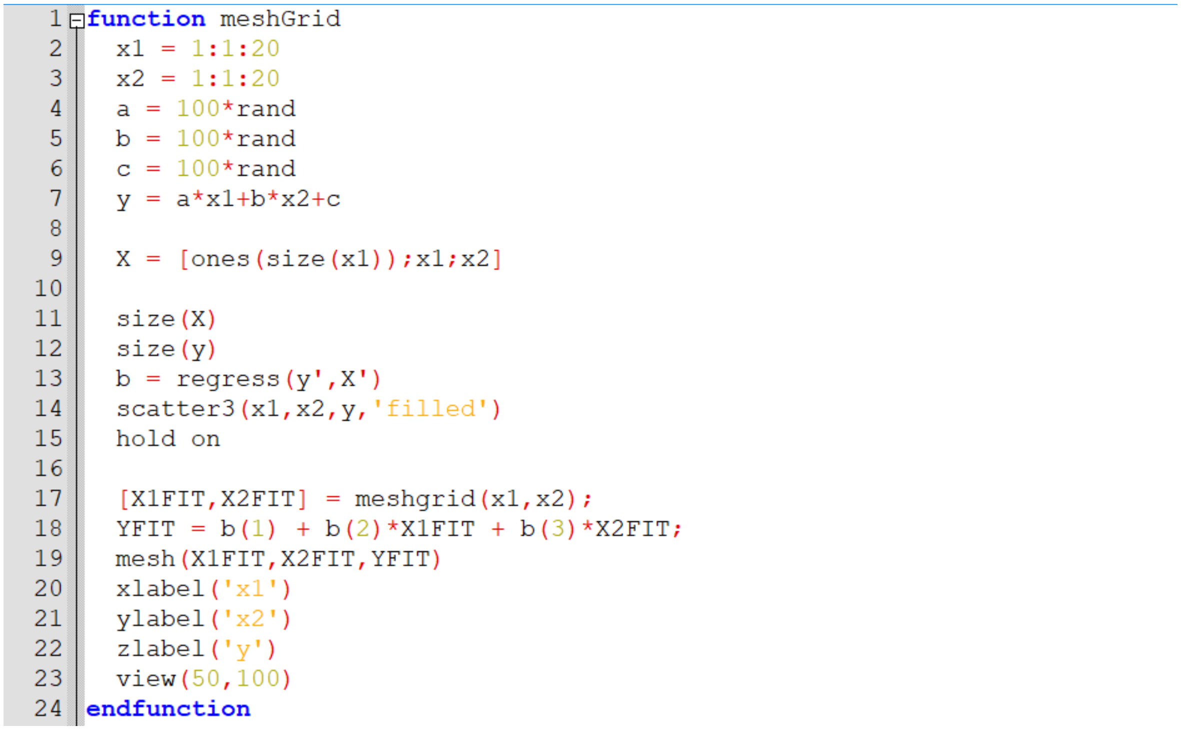
Name	Class	Dimension	Value	Attribute
Rsq	double	1x1	1	
X	double	1x21	[-10, -9, -...	
Y	double	1x21	[0.54402, ...	



PRACTICAL – 5

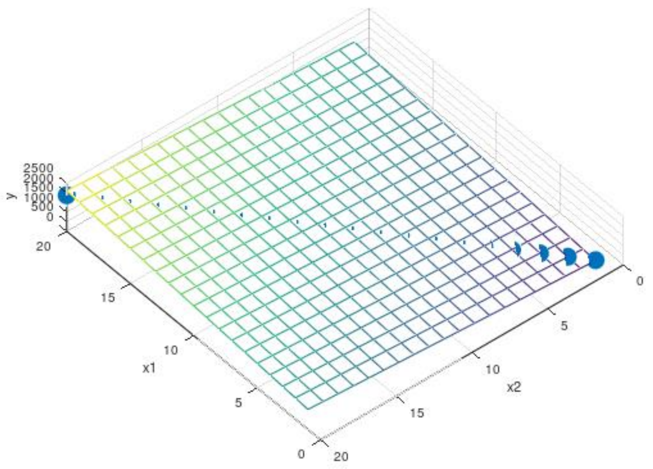
AIM: To write a program for multiple regression.

CODE:



```
function meshGrid
  x1 = 1:1:20
  x2 = 1:1:20
  a = 100*rand
  b = 100*rand
  c = 100*rand
  y = a*x1+b*x2+c

  X = [ones(size(x1));x1;x2]

  size(X)
  size(y)
  b = regress(y',X')
  scatter3(x1,x2,y,'filled')
  hold on

  [X1FIT,X2FIT] = meshgrid(x1,x2);
  YFIT = b(1) + b(2)*X1FIT + b(3)*X2FIT;
  mesh(X1FIT,X2FIT,YFIT)
  xlabel('x1')
  ylabel('x2')
  zlabel('y')
  view(50,100)
endfunction
```

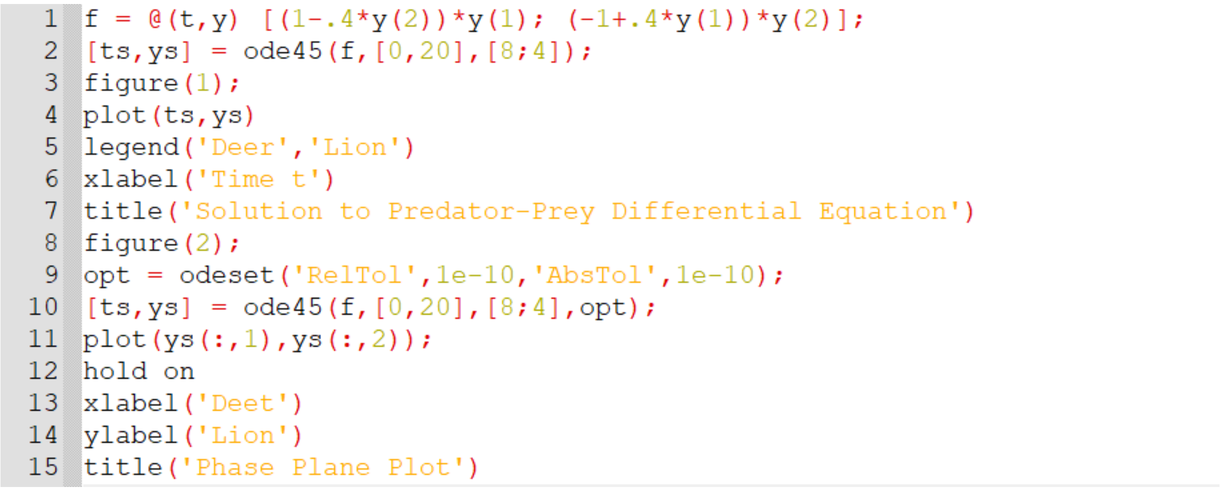
OUTPUT:



PRACTICAL – 6

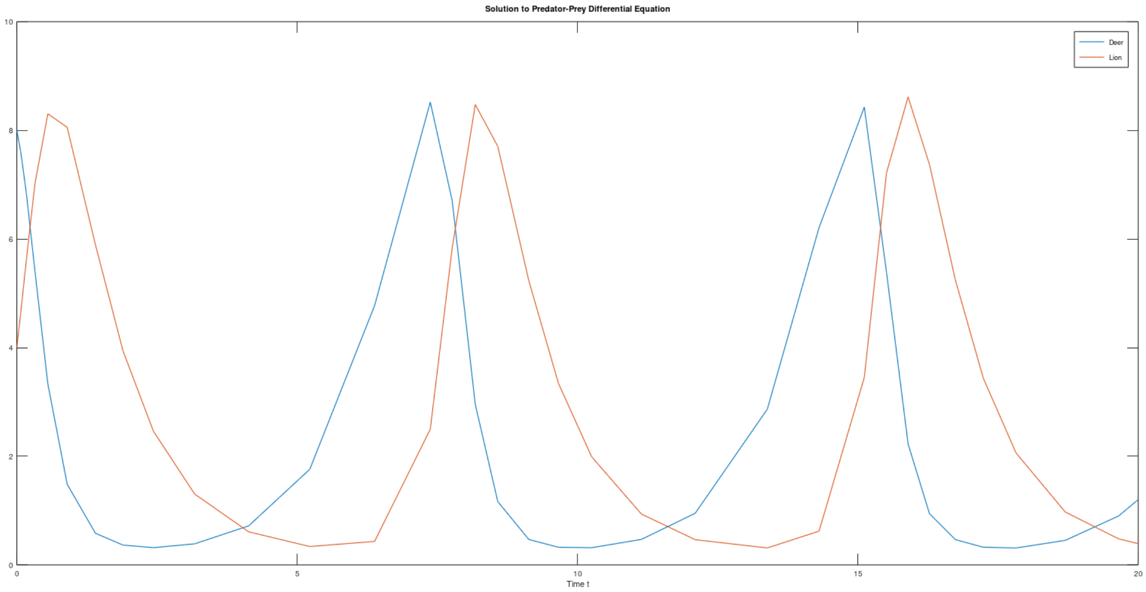
AIM: Draw some trajectories for predator-prey problem.

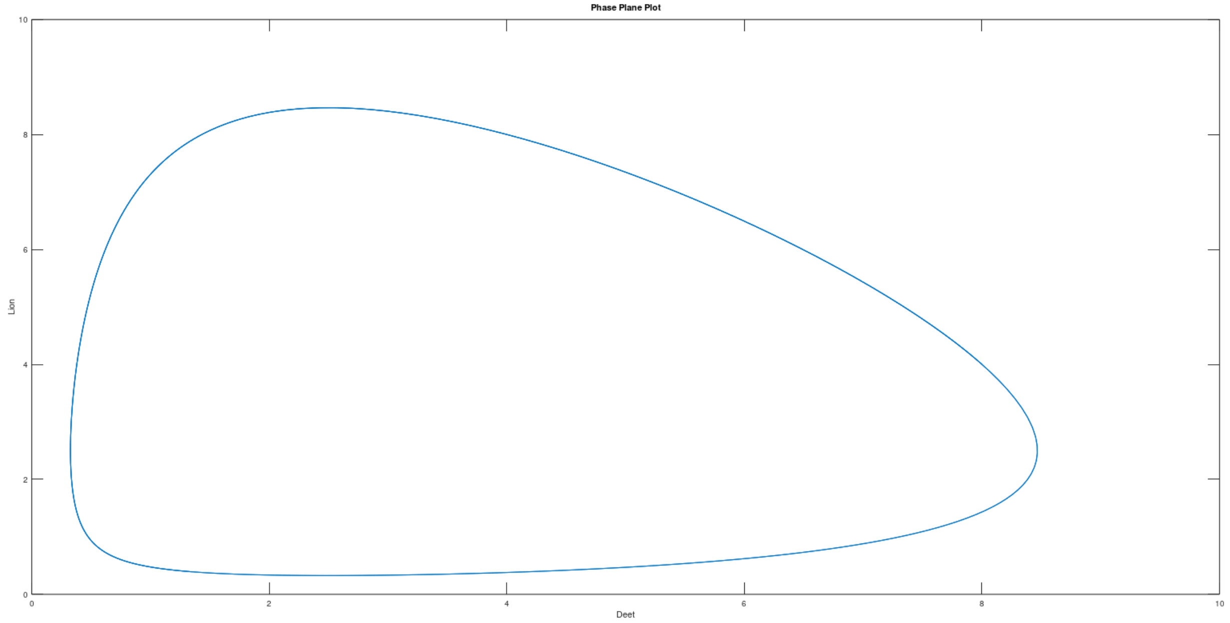
CODE:



```
f = @(t,y) [(1-.4*y(2))*y(1); (-1+.4*y(1))*y(2)];
[ts,ys] = ode45(f,[0,20],[8;4]);
figure(1);
plot(ts,ys)
legend('Deer','Lion')
xlabel('Time t')
title('Solution to Predator-Prey Differential Equation')
figure(2);
opt = odeset('RelTol',1e-10,'AbsTol',1e-10);
[ts,ys] = ode45(f,[0,20],[8;4],opt);
plot(ys(:,1),ys(:,2));
hold on
xlabel('Deet')
ylabel('Lion')
title('Phase Plane Plot')
```

OUTPUT:

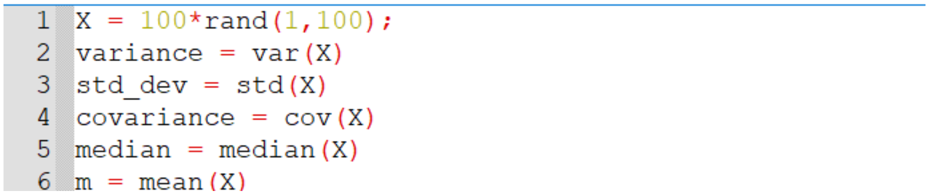



Phase Plane Plot
Lion
Deet
0
2
4
6
8
10
0
2
4
6
8
10

PRACTICAL – 7

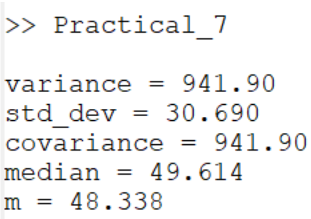
AIM: To write a program for statistical analysis of data.

CODE:



```
X = 100*rand(1,100);
variance = var(X)
std_dev = std(X)
covariance = cov(X)
median = median(X)
m = mean(X)
```

OUTPUT:

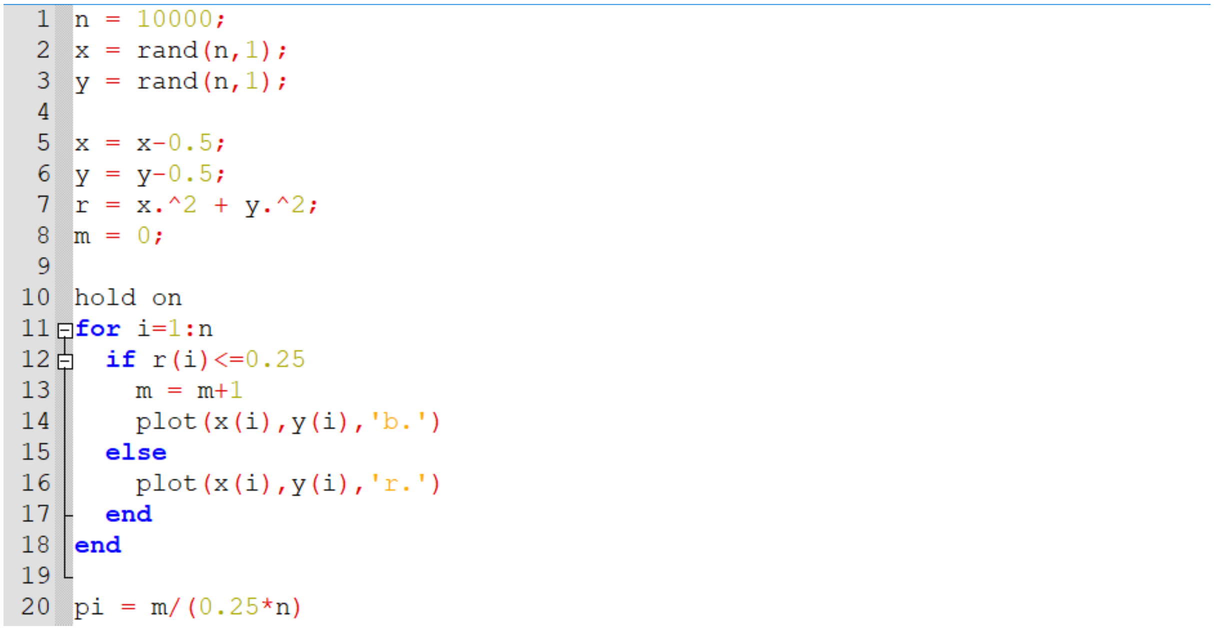


```
>> Practical_7

variance = 941.90
std_dev = 30.690
covariance = 941.90
median = 49.614
m = 48.338
```

PRACTICAL – 8

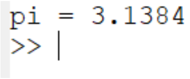
AIM: To write a program for monte carlo method.

CODE:

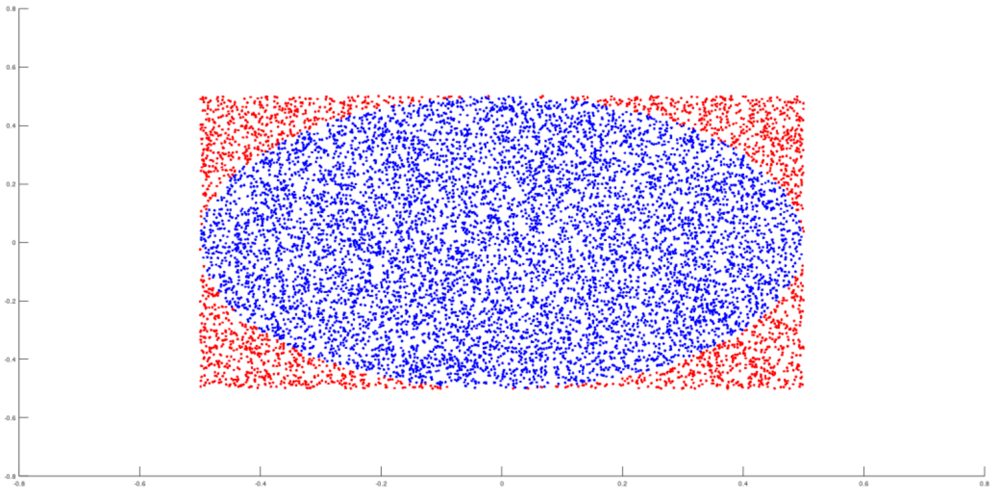


```
n = 10000;
x = rand(n,1);
y = rand(n,1);

x = x-0.5;
y = y-0.5;
r = x.^2 + y.^2;
m = 0;

hold on
for i=1:n
    if r(i)<=0.25
      m = m+1
      plot(x(i),y(i),'b.')
    else
      plot(x(i),y(i),'r.')
    end
end

pi = m/(0.25*n)
```

OUTPUT:



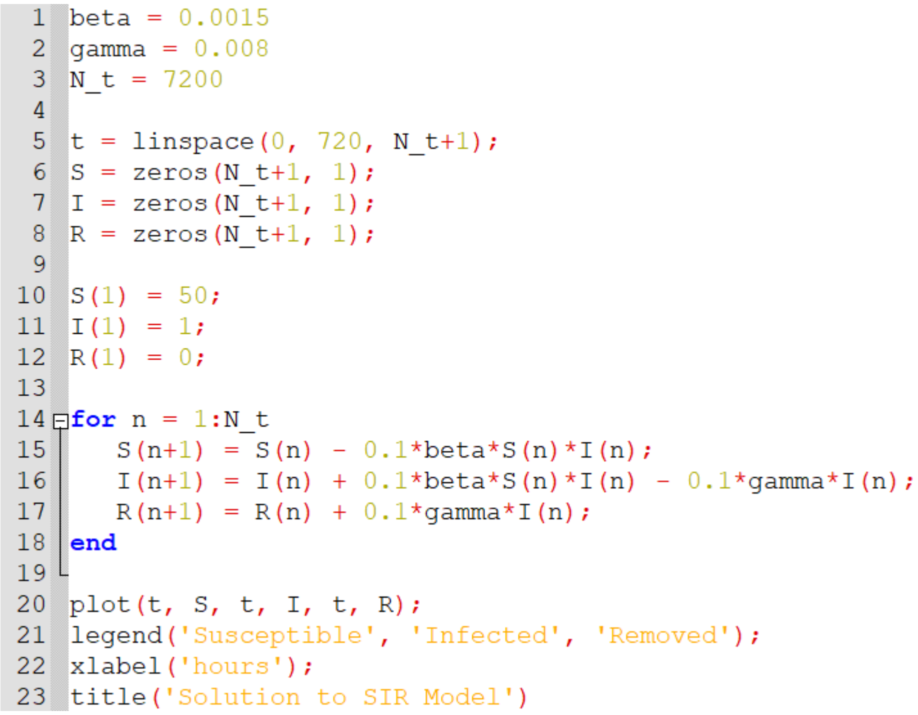
```
pi = 3.1384
>> |
```



PRACTICAL – 9

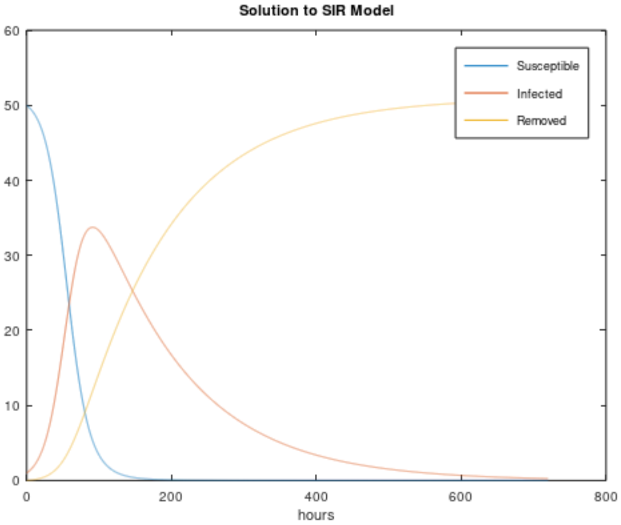
AIM: To write a program for epidemic model.

CODE:



```
beta = 0.0015
gamma = 0.008
N_t = 7200

t = linspace(0, 720, N_t+1);
S = zeros(N_t+1, 1);
I = zeros(N_t+1, 1);
R = zeros(N_t+1, 1);

S(1) = 50;
I(1) = 1;
R(1) = 0;

for n = 1:N_t
    S(n+1) = S(n) - 0.1*beta*S(n)*I(n);
    I(n+1) = I(n) + 0.1*beta*S(n)*I(n) - 0.1*gamma*I(n);
    R(n+1) = R(n) + 0.1*gamma*I(n);
end

plot(t, S, t, I, t, R);
legend('Susceptible', 'Infected', 'Removed');
xlabel('hours');
title('Solution to SIR Model')
```

OUTPUT:


Solution to SIR Model
hours
Susceptible
Infected
Removed