Vidyavardhaka Sangha®, Mysore
VIDYAVARDHAKA COLLEGE OF ENGINEERING
Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi
(Approved by AICTE, New Delhi & Government of Karnataka)
Accredited by NAAC with 'A' Grade
Department of Mathematics
Phone: +91 821-4276243 , Email: hodmaths@vvce.ac.in
Web: http://www.vvce.ac.in
VVCE
@vvceofficial

MATLAB

Manual

For IV Sem BE (Autonomous)

AY 2021-22

Prepared by:
Abhijeet Duggani and Vishnu M
Department of Mathematics
Vidyavardhaka College of Engineering
Mysuru

Contents

*****

Module – 1
Complex Analysis

1.CR Equations

Type 1:

Check whether the following functions satisfies CR equation.

$f(z) = u + \mathrm{i}v$ where $u = x + e^x \cos y$ and $v = y + e^x \sin y$.

Code

```
clear;

syms x y
u=@(x,y)x+exp(x)*cos(y)
v=@(x,y)y+exp(x)*sin(y)
u_x=diff(u,x)
u_y=diff(u,y)
v_x=diff(v,x)
v_y=diff(v,y)
if u_x==v_y && v_x==-u_y
    disp("f(z) satisfies CR equation")
else
    disp("f(z) doesn't satisfy CR equation and hence not analytic")
end
```

Output

```
u = function_handle with value:
    @(x,y)x+exp(x)*cos(y)

v = function_handle with value:
    @(x,y)y+exp(x)*sin(y)
```

u_x = $e^x \cos(y) + 1$

u_y = $-e^x \sin(y)$

v_x = $e^x \sin(y)$

v_y = $e^x \cos(y) + 1$

```
f(z) satisfies CR equation
```

Type 2:

Check whether the following functions satisfies CR equation (polar form).

$f(z) = \mathrm{rcos}(t) + \mathrm{irsin}(t)$

Code

```
clear;
syms r t
u=@(r,t)r*cos(t)
v=@(r,t)r*sin(t)
u_r=diff(u,r)
u_t=diff(u,t)
v_r=diff(v,r)
v_t=diff(v,t)
if u_r==(1/r)*v_t && v_r==(-1/r)*u_t
    disp("f(z) satisfies CR equation")
else
    disp("f(z) doesn't satisfy CR equation and hence not analytic")
end
```

Output

```
u = function_handle with value:
    @(r,t)r*cos(t)

v = function_handle with value:
    @(r,t)r*sin(t)
```

u_r = $\cos(t)$

u_t = $-r\sin(t)$

v_r = $\sin(t)$

v_t = $r\cos(t)$

```
f(z) satisfies CR equation
```

Type 3:

Check if the real and imaginary parts of the function $f(z)=|z|$ satisfies CR equations.

```
clear;
syms x y
z=x+i*y
f=abs(z)
u=real(f)
v=imag(f)
u_x=diff(u,x)
u_y=diff(u,y)
v_x=diff(v,x)
v_y=diff(v,y)
if u_x==v_y && v_x==-u_y
    disp("f(z) satisfies CR equation")
else
    disp("f(z) doesn't satisfy CR equation and hence not analytic")
end
```

Output

z = $x + y\,\mathrm{i}$

f = $|x + y\,\mathrm{i}|$

u = $|x + y\,\mathrm{i}|$

v = 0

u_x = $\mathrm{sign}(x + y\,\mathrm{i})$

u_y = $\mathrm{sign}(x + y\,\mathrm{i})\,\mathrm{i}$

v_x = 0

v_y = 0

f(z) doesn't satisfy CR equation and hence not analytic

3. Harmonic functions

(1)Check whether the given function is harmonic $e^x(xcosy - ysiny)$.

```
clear;
syms x y
u=exp(x)*(x*cos(y)-y*sin(y))
L=diff(u,x,2)+diff(u,y,2)
a=simplify(L)
if a==0
    disp("The given function is harmonic")
else
    disp("The given function is not harmonic")
end
```

Output

u = $e^x\,(x\cos(y) - y\sin(y))$

L = $e^x\,(x\cos(y) - y\sin(y)) + 2\,e^x\cos(y) - e^x\,(2\cos(y) + x\cos(y) - y\sin(y))$

a = 0

The given function is harmonic

(2) Check whether the given function is harmonic $u = \left(r + \frac{1}{r}\right)\cos(\theta)$

```
Code:

clear;
syms r t
u=(r+(1/r))*cos(t);
L=diff(u,r,2)+(1/r)*diff(u,r)+(1/r^2)*diff(u,t,2)
a=simplify(L)
if a==0
    disp("The function is harmonic")
else
    disp("The function is not harmonic")
end
```

Output

```
L =
```

$$\frac{2\cos(t)}{r^3} - \frac{\cos(t)\left(\frac{1}{r^2}-1\right)}{r} - \frac{\cos(t)\left(r+\frac{1}{r}\right)}{r^2}$$

```
a = 0

The function is harmonic
```

4. Milne Thomson method

1.Construct the analytic function whose real part is $u = \log\left(\sqrt{x^2+y^2}\right)$

```
clear;
syms x y z
u=log(sqrt(x^2+y^2));
u_x=diff(u,x);
u_y=diff(u,y);
F=u_x-i*u_y;
G=subs(F,x,z)%substituting z in place of x
H=subs(G,y,0)%substituting 0 in place of y
f=int(H,z)
```

A similar procedure can be applied when only imaginary part is given.

Output

```
G =
```

$$\frac{z}{y^2+z^2} - \frac{y\,\mathrm{i}}{y^2+z^2}$$

```
H =
```

$$\frac{1}{z}$$

f = $\log(z)$

A similar procedure can be applied when only imaginary part is given by replacing u by v and F by v_y+i *v_x.

(2) Construct the analytic function whose real part is $u = r^2\cos(2\theta)$

```
clear;
syms r t z
u=(r^2)*cos(2*t);
u_r=diff(u,r);
u_t=diff(u,t);
F=u_r-i*(1/r)*u_t;
G=subs(F,r,z)%substituting z in place of r
H=subs(G,t,0)%substituting 0 in place of t
f=int(H,z)
```

G = $2\,z\cos(2\,t) + 2\,z\sin(2\,t)\,\mathrm{i}$

H = $2\,z$

f = z^2

A similar procedure can be applied when only imaginary part is given by replacing u by v and F by (1/r)*v_t+i *v_r.

5. Poles and Residues

Find the poles and residues of the function $f(z) = \dfrac{z}{(z-1)^2(z-i)}$.

```
syms z
f(z)=z/((z-1)^2*(z-i));
%P=poles(f,z) %gives all the poles
%[P,N] = poles(f,z)%gives all the poles along with its order
[P,N,R] = poles(f,z)%gives all the poles along with its order and also the corresponding residues

% P = poles

% N = order of pole

% R = residue
```

Output

P =

$$\begin{pmatrix} i \\ 1 \end{pmatrix}$$

N =

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

R =

$$\begin{pmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{pmatrix}$$

Module – 2
Statistical Methods

1. Correlation and Regression lines

Calculate the correlation of coefficient between x and y also find regression lines.

Code

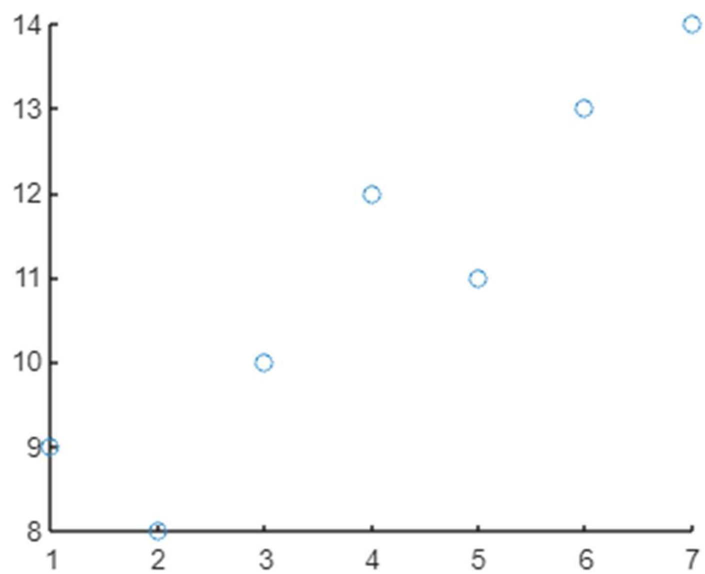
```
x=[1 2 3 4 5 6 7]
y=[9 8 10 12 11 13 14]
%coefficient of correlation
r=corrcoef(x,y)
%equation of the lines of regression
k=polyfit(x,y,1)
h=polyfit(y,x,1)
a=k(1)
b=k(2)
fprintf('the regression line y interm of x is Y=%f*x+%f',a,b)
c=h(1)
d=h(2)
fprintf('the regression line x interm of y is X=%f*y+%f',c,d)
scatter(x,y)
```

The value of the off-diagonal elements of r, which represents the correlation coefficient between X and Y.

Output

```
x =

     1     2     3     4     5     6     7


y =

     9     8    10    12    11    13    14


r =

    1.0000    0.9286
    0.9286    1.0000


k =
    0.9286    7.2857


h =
```

```
    0.9286   -6.2143

a =
    0.9286


b =
    7.2857

the regression line y interm of x is Y=0.928571*x+7.285714


c =
    0.9286


d =
   -6.2143


the regression line x interm of y is X=0.928571*y+-6.214286
```



2. Rank correlation

Calculate the Spearman's Rank Correlation between A and B

Code

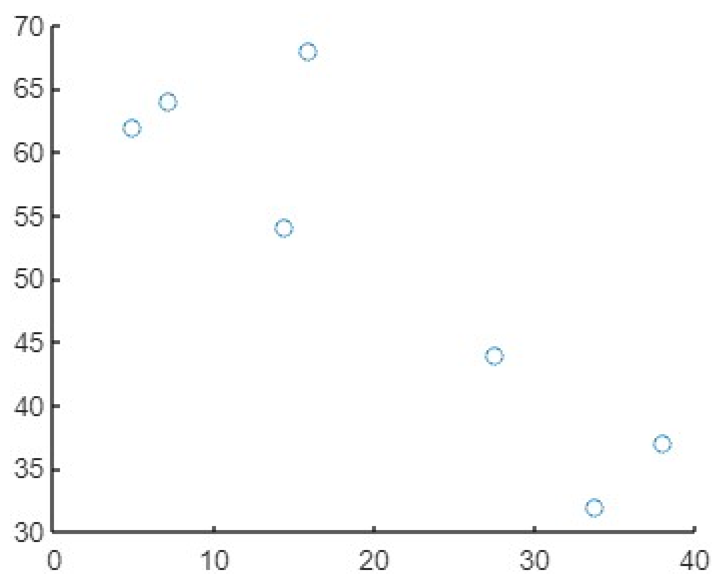
```
A = [14.4,7.2,27.5,33.8,38,15.9,4.9]

B = [54,64,44,32,37,68,62]
```

```
[RHO,PVAL]=corr(A',B','type','Spearman')

scatter(A,B)
```

NOTE ; The difference between corr(X,Y) and the MATLAB® function corrcoef(X,Y) is that corrcoef(X,Y) returns a matrix of correlation coefficients for two column vectors X and Y. If X and Y are not column vectors, corrcoef(X,Y) converts them to column vectors.

Output

```
A = 1×7
    14.4000    7.2000   27.5000   33.8000   38.0000   15.9000    4.9000

B = 1×7
    54    64    44    32    37    68    62

RHO = -0.7143
PVAL = 0.0881
```



Code

```
x= [1 2 3 4 5]

y = [2 5 3 8 7]

polyfit(x,y,1) %regression line y in term of x
```

```
polyfit(y,x,1) %regression line x in term of y
```

Output

```
x = 1×5
     1     2     3     4     5

y = 1×5
     2     5     3     8     7

ans = 1×2
    1.3000    1.1000

ans = 1×2
    0.5000    0.5000
```

3. Curve fitting

Code

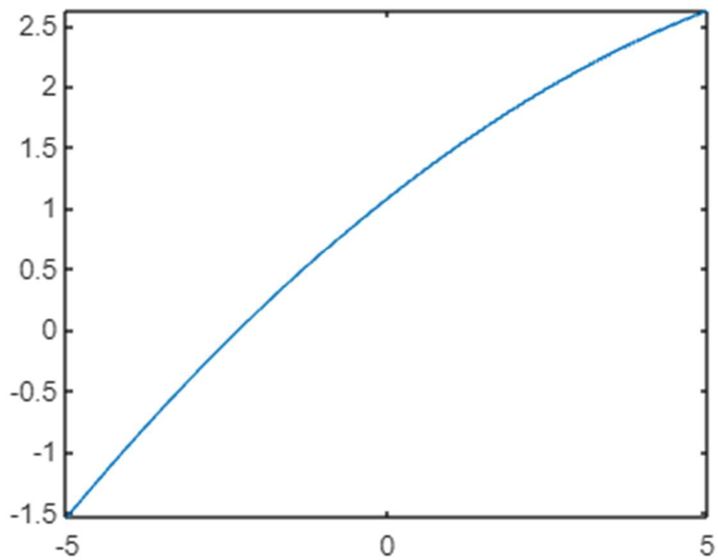
```
X = [0 1 2 3 4]

Y = [1 1.8 1.3 2.5 2.3]

k=polyfit(X,Y,2)

a=k(1);b=k(2);c=k(3);

syms x

fplot(a*x^2+b*x+c);
```

or

```
X = [0 1 2 3 4]

Y = [1 1.8 1.3 2.5 2.3]

scatter(X,Y)
```

Output

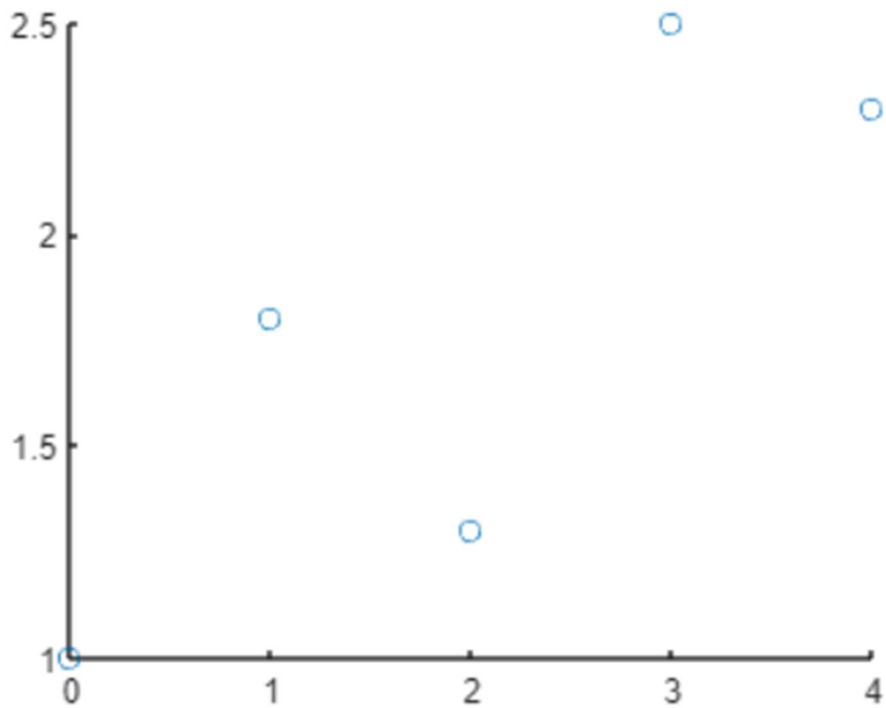
```
X = 1×5
      0     1     2     3     4

Y = 1×5
    1.0000    1.8000    1.3000    2.5000    2.3000

k = 1×3
   -0.0214    0.4157    1.0771
```

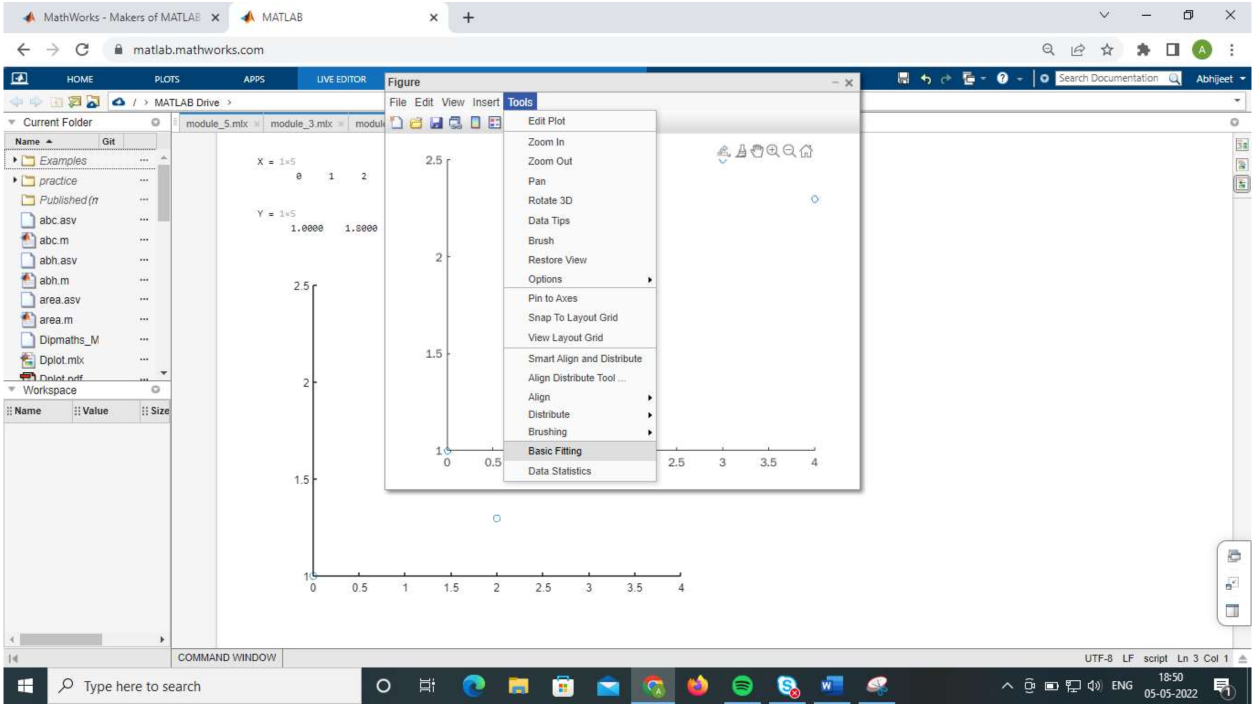


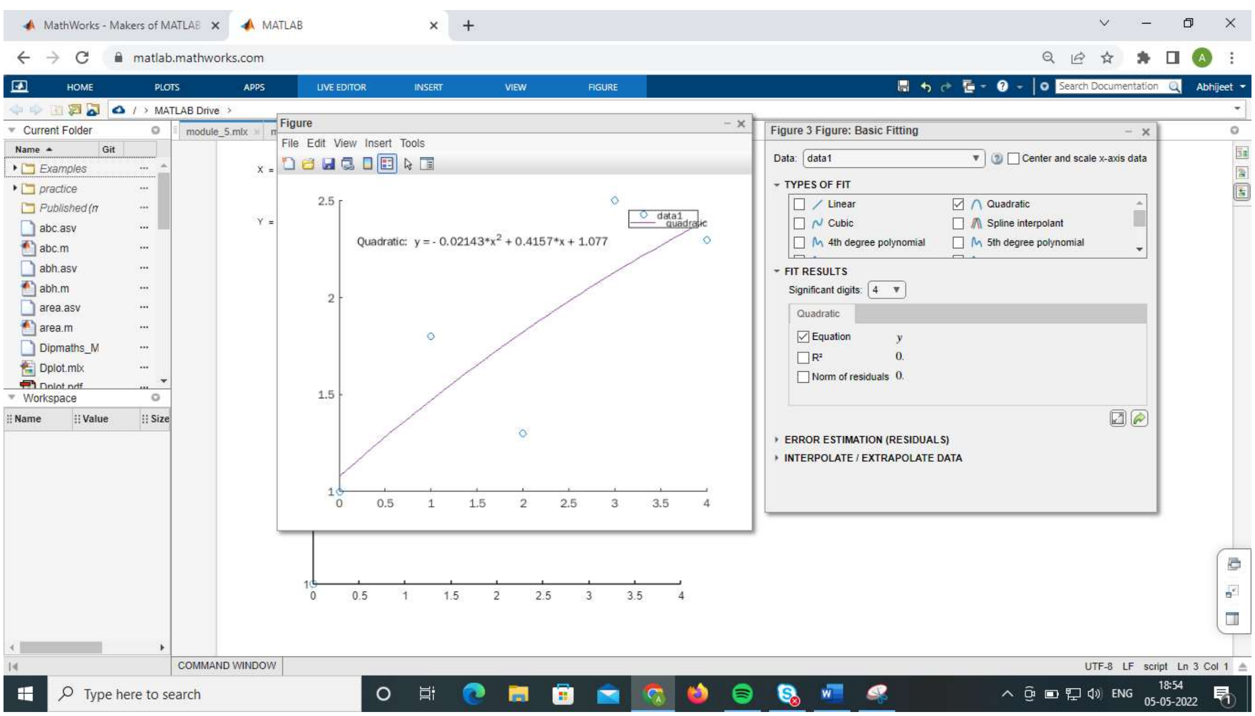
To fit a curve graphically follow the below steps:

1. Select the figure and click on right top corner
2. The figure will open in new tab
3. Select tools and then basic fitting
4. Select type of fit we need
5. After selecting required fit we get a curve along with equation

```
X = 1×5
     0     1     2     3     4

Y = 1×5
    1.0000    1.8000    1.3000    2.5000    2.3000
```




MathWorks - Makers of MATLAB
MATLAB
matlab.mathworks.com
Figure
File Edit View Insert Tools
Edit Plot
Zoom In
Zoom Out
Pan
Rotate 3D
Data Tips
Brush
Restore View
Options
Pin to Axes
Snap To Layout Grid
View Layout Grid
Smart Align and Distribute
Align Distribute Tool ...
Align
Distribute
Brushing
Basic Fitting
Data Statistics


MathWorks - Makers of MATLAB
MATLAB
matlab.mathworks.com
Figure
Quadratic: y = - 0.02143*x^2 + 0.4157*x + 1.077
Figure 3 Figure: Basic Fitting
Data: data1
Center and scale x-axis data
TYPES OF FIT
Linear
Quadratic
Cubic
Spline interpolant
4th degree polynomial
5th degree polynomial
FIT RESULTS
Significant digits: 4
Quadratic
Equation
Norm of residuals
ERROR ESTIMATION (RESIDUALS)
INTERPOLATE / EXTRAPOLATE DATA

Type 1

Fit a straight line y= a+b*x in the least square sense for the given data.

We can solve these type of problems graphically as we solved above or below code can be used .

Code

```
x=[1 3 4 6 8 9 11 14]
y=[1 2 4 4 5 7 8 9]
n = length(x)
syms a b
eq1=n*a+b*sum(x)==sum(y)
eq2= a*sum(x)+b*sum(x.*x)==sum(x.*y)


[A,B]= equationsToMatrix([eq1 eq2],[a,b])
X=linsolve(A,B)
a=double(X(1))
b=double(X(2))
fprintf('the curve fitting is given by Y=%f+%f*x',a,b)
```

Output

```
program to fit a straight line

 x = 1×8
       1     3     4     6     8     9    11    14


 y = 1×8
       1     2     4     4     5     7     8     9
```

```
n = 8

eq1 = 8a + 56b = 40

eq2 = 56a + 524b = 364

A =
```

$$\begin{pmatrix} 8 & 56 \\ 56 & 524 \end{pmatrix}$$

```
B =
```

$$\begin{pmatrix} 40 \\ 364 \end{pmatrix}$$

```
X =
```

$$\begin{pmatrix} \frac{6}{11} \\ \frac{7}{11} \end{pmatrix}$$

```
a = 0.5455

b = 0.6364

the curve fitting is given by Y=0.545455+0.636364*x
```

Type 2

Find the equation of the best fitting straight line y = a*x+b

Code

```
clc
clear
x=[5 10 15 20 25]
y=[16 19 23 26 30]
n = length(x)
syms a b
eq1=a*sum(x)+n*b==sum(y)
eq2=a*sum(x.*x)+b*sum(x)==sum(x.*y)


[A,B]= equationsToMatrix([eq1 eq2],[a,b])
X=linsolve(A,B)
a=double(X(1))
b=double(X(2))
fprintf('the curve fitting is given by Y=%f*x+%f',a,b)
```

Output

```
x = 1×5
     5    10    15    20    25

y = 1×5
    16    19    23    26    30

n = 5
```

eq1 = $75\,a + 5\,b = 114$

eq2 = $1375\,a + 75\,b = 1885$

A =

$$\begin{pmatrix} 75 & 5 \\ 1375 & 75 \end{pmatrix}$$

B =

$$\begin{pmatrix} 114 \\ 1885 \end{pmatrix}$$

X =

$$\begin{pmatrix} \frac{7}{10} \\ \frac{123}{10} \end{pmatrix}$$

```
a = 0.7000

b = 12.3000

the curve fitting is given by Y=0.700000*x+12.300000
```

Type 3

Similar to above we can fit a parabola of second degree y = a+b*x+c*x^2

Type 4

Fit a curve of the form y = a*x^(b) in the least square sence for the given data.

Code

```
clc
clear
x=[1 2 3 4 5]
y=[0.5 2 4.5 8 12.5]
n = length(x)
X=log(x);
Y=log(y);
syms a1 b1
eq1=n*a1+b1*sum(X)==sum(Y)
```

```
eq2=a1*sum(X)+b1*sum(X.*X)==sum(X.*Y)
[A,B]= equationsToMatrix([eq1 eq2],[a1,b1])
X=linsolve(A,B)
a1=double(X(1))
b1=double(X(2))
a=exp(a1)
b= b1
fprintf('the curve fitting is given by Y=%f*x^(%f)',a,b)
```

Output

```
program to fit a parabola

 x = 1×5
       1     2     3     4     5


 y = 1×5
      0.5000    2.0000    4.5000    8.0000   12.5000


n = 5

eq1 =
```

$$5\,a_1 + \frac{5390236507208137\,b_1}{1125899906842624} = \frac{6878401284312925}{1125899906842624}$$

```
eq2 =
```

$$\frac{5390236507208137\,a_1}{1125899906842624} + \frac{3490010726942091\,b_1}{562949953421312} = \frac{5111907835122879}{562949953421312}$$

```
A =
```

$$\begin{pmatrix} 5 & \frac{5390236507208137}{1125899906842624} \\ \frac{5390236507208137}{1125899906842624} & \frac{3490010726942091}{562949953421312} \end{pmatrix}$$

```
B =
```

$$\begin{pmatrix} \frac{6878401284312925}{1125899906842624} \\ \frac{5111907835122879}{562949953421312} \end{pmatrix}$$

```
X =
```

$$\begin{pmatrix} -\frac{709739593579658528609885168O496}{10239377919799210870464745857071} \\ \frac{2047875583959842736705353367423 5}{10239377919799210870464745857071} \end{pmatrix}$$

```
a1 = -0.6931

b1 = 2.0000

a = 0.5000

b = 2.0000

the curve fitting is given by Y=0.500000*x^(2.000000)
```

Module – 3

Probability Distribution and Joint Probability Distributions

I. DISCRETE PROBABILITY DISTRIBUTIONS.

Find the discrete probability distribution function, mean, variance and standard deviation for the given data:

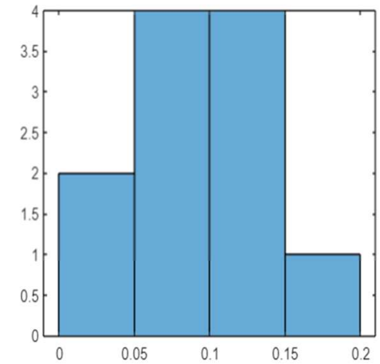
x_i	2	3	4	5	6	7	8	9	10	11	12
$p(x_i)$	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

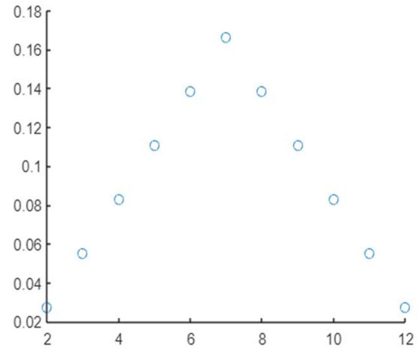
Also visualize the data in a histogram, scatterplot and boxplot.

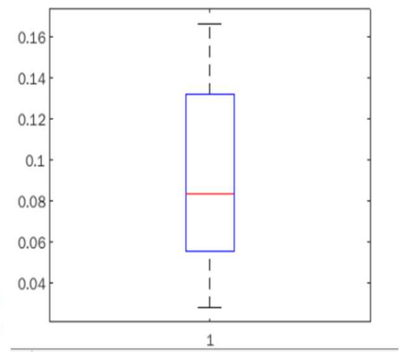
Code

```
clc;
clear;
x=[2 3 4 5 6 7 8 9 10 11 12]
p=[1/36 2/36 3/36 4/36 5/36 6/36 5/36 4/36 3/36 2/36 1/36]
sum(p)
i=1:length(p);
if all(p(:)>0) dand sum(p)==1;
    disp("p(x) is a discrete probability distribution function")
    mean=sum(x(i).*p(i))
    Variance=sum(((x(i)-mean).^2).*p(i))
    SD=sqrt(Variance)
else
    disp("p(x) is not a discrete probability distribution function")
end
histogram(p)
scatter(x,p)
boxplot(p)
```

```
x = 1×11
     2     3     4     5     6     7     8     9    10    11    12

p = 1×11
    0.0278    0.0556    0.0833    0.1111    0.1389    0.1667    0.1389    0.1111    0.0833    0.0556    0.0278

ans = 1.0000

i = 1×11
     1     2     3     4     5     6     7     8     9    10    11

p(x) is a discrete probability distribution function
mean = 7.0000
Variance = 5.8333
SD = 2.4152
```







II JOINT PROBABILITY DISTRIBUTIONS

The joint probability distribution of two random variables X and Y is as follows:

X \ Y	-2	-1	4	5
1	0.1	0.2	0.0	0.3
2	0.2	0.1	0.1	0.0

Compute: a] Expectation of X. b] Expectation of Y. c] σ_X and σ_Y d] COV(X,Y) e] $\rho(X,Y)$ f]E(XY)

Code

```
clear;
X=[1 2];
Y=[-2 -1 4 5];
J=[0.1 0.2 0 0.3;0.2 0.1 0.1 0];
m=2;
n=4;
x1=J(1,:)
x2=J(2,:)
y1=J(:,1)
y2=J(:,2)
y3=J(:,3)
y4=J(:,4)
f=[sum(x1) sum(x2)]
g=[sum(y1) sum(y2) sum(y3) sum(y4)]
S=sum(sum(J))%Sum should be equal to "1"
E_X=sum(X.*f)
E_Y=sum(Y.*g)
E_X2=sum((X.^2).*f)
E_Y2=sum((Y.^2).*g)
VX=E_X2-(E_X)^2
VarY=E_Y2-(E_Y)^2
SDX=sqrt(VarX)
```

```
SDY=sqrt(VarY)


for i=1:2
    for j=1:4
    E(i,j)=sum(X(i)*Y(j)*J(i,j)); %You will get a 2*3 matrix of the mentioned
form
    end
end
E_XY=sum(sum(E))
COV_XY= E_XY-(E_X*E_Y)
COR_XY= (COV_XY)/(SDX*SDY)
```

Output

```
x1 = 1x4
      0.1000    0.2000         0    0.3000


x2 = 1x4
      0.2000    0.1000    0.1000         0


y1 = 2x1
      0.1000
      0.2000


y2 = 2x1
      0.2000
      0.1000


y3 = 2x1
           0
      0.1000


y4 = 2x1
      0.3000
           0
```

```
f = 1x2
      0.6000    0.4000


g = 1x4
      0.3000    0.3000    0.1000    0.3000

S = 1

E_X = 1.4000

E_Y = 1.0000

E_X2 = 2.2000

E_Y2 = 10.6000

VarX = 0.2400

VarY = 9.6000

SDX = 0.4899

SDY = 3.0984
```

```
E_XY = 0.9000

COV_XY = -0.5000

COR_XY = -0.3294
```

Module – 4

Stochatic process and sampling theory

```
A=[1 2;3 4]
sum(A,2)
%You can create a sum matrix over rows by typing sum(matrixName, 2).
% This will return an array containing sum over rows.
```

```
A = 2×2
     1     2
     3     4

ans = 2×1
     3
     7
```

1.Check the given matrix is stochastic matrix or not

Code

```
B=[1/2 1/2 0;0 1/2 1/2;1/2 0 1/2]
n=length(B)
k=sum(B,2)
i=1:n
if k(i)==1
    disp('The matrix is stochastic matrix')
else
     disp('The matrix is not stochastic matrix')
end
```

output

```
B = 3×3
    0.5000    0.5000         0
         0    0.5000    0.5000
    0.5000         0    0.5000

n = 3

k = 3×1
     1
     1
     1

i = 1×3
     1     2     3

The matrix is stochastic matrix
```

2. Check the given matrix is regular stochastic or not
   Code

```
C=[0 1 0;0 0 1;1/2 1/2 0]
n = length(C)
i=1:n;
j=1:n;
for a=1:10
    if (C(i,j))^(a)>0
        %the value for which the matrix become regular
        disp(a)
        disp('matrix is regular stochastic ')
        break
    end
end
```

Output

2. Check the given matrix is regular stochastic or not

```
C = 3×3
         0    1.0000         0
         0         0    1.0000
    0.5000    0.5000         0

n = 3

     5
matrix is regular stochastic
```

3. Find unique stationary probability vector
   Code

```
syms x y z
P=[ 0 2/3 1/3;1/2 0 1/2;1/2 1/2 0]
X=[x y z]
K=X*P== X


eq1=K(1)
eq2=K(2)
eq3= x+y+z==1 %for 2*2 matrix x+y=1 and for 3*3 matrix x+y+z=1
sol = solve([eq1, eq2,eq3], [x, y,z])
```

```
P = 3×3
        0    0.6667    0.3333
   0.5000         0    0.5000
   0.5000    0.5000         0
```

X = $\begin{pmatrix} x & y & z \end{pmatrix}$

K =

$$\left(\frac{y}{2}+\frac{z}{2}=x \quad \frac{2\,x}{3}+\frac{z}{2}=y \quad \frac{x}{3}+\frac{y}{2}=z\right)$$

eq1 =

$$\frac{y}{2}+\frac{z}{2}=x$$

eq2 =

$$\frac{2\,x}{3}+\frac{z}{2}=y$$

eq3 = $x+y+z=1$

```
sol = struct with fields:
    x: 1/3
    y: 10/27
    z: 8/27
```

4. Sampling distribution

a) the mean and S.D of the population

code

```
clear all
A=[2 3 6 8 11]
N=length(A)
M=mean(A)


sum1=0
%to find variance
for i=1:N
    sum1=sum1+(A(i)-M)^(2); %standard formula
end
vari= sum1/N  %Variance
Sd = sqrt(vari)  % standard deviation
```

Output

```
A = 1×5
      2     3     6     8    11

N = 5
M = 6
sum1 = 0
vari = 10.8000
Sd = 3.2863
```

5.(i) t-test of a sample mean

Code

```
mu=3
x_1=3.1
n=25
s= 0.3
t=(x_1-mu)*sqrt(n)/s
```

Output

```
mu = 3

x_1 = 3.1000

n = 25

s = 0.3000

t = 1.6667
```

5.(ii) t-test of difference between sample means

Code

```
n1=10;  n2=10;
xb_1=500;  xb_2=560;
v_1=100;    v_2=121;
s_squar=(n1*v_1+n2*v_2)/(n1+n2-2)      %standard formula
s=sqrt(s_squar)
t=(xb_2-xb_1)/(s*sqrt(1/n1+1/n2))      %standard formula
```

Output

t-test of difference between sample means

```
s_squar = 122.7778

s = 11.0805

t = 12.1081
```

POISSON'S DISTRIBUTION

Fit a Poisson distribution for the following data and visualize the probability distribution of fit using scatterplot.

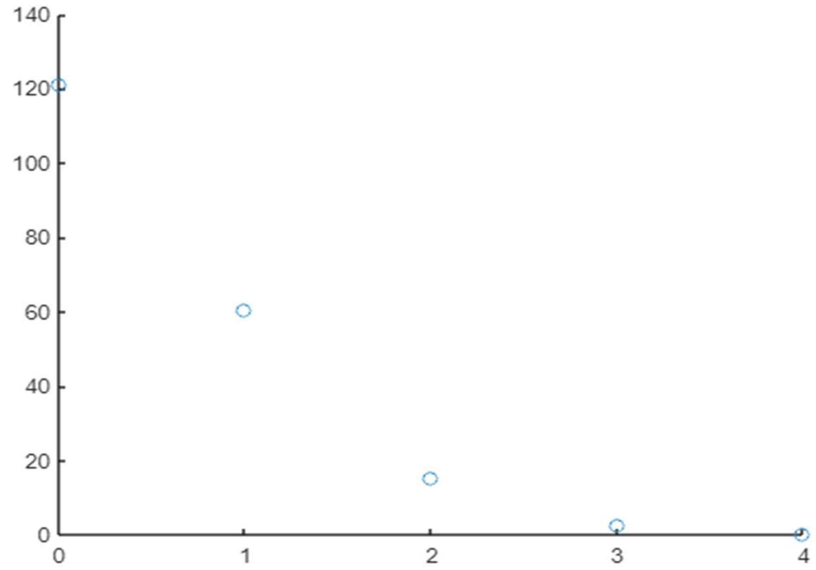
x	0	1	2	3	4
y	122	60	15	2	1

```
clear;
x=transpose([0 1 2 3 4]) %input must be in the form of a column
f=transpose([122 60 15 2 1])
N=200
mean=sum(f.*x)/sum(f)
Va=mean
pd=poissfit(x,mean)
F=N.*poisspdf(x,mean)
G=N.*poisscdf(x,mean)
scatter(x,F)
hold off
scatter(x,G)
```

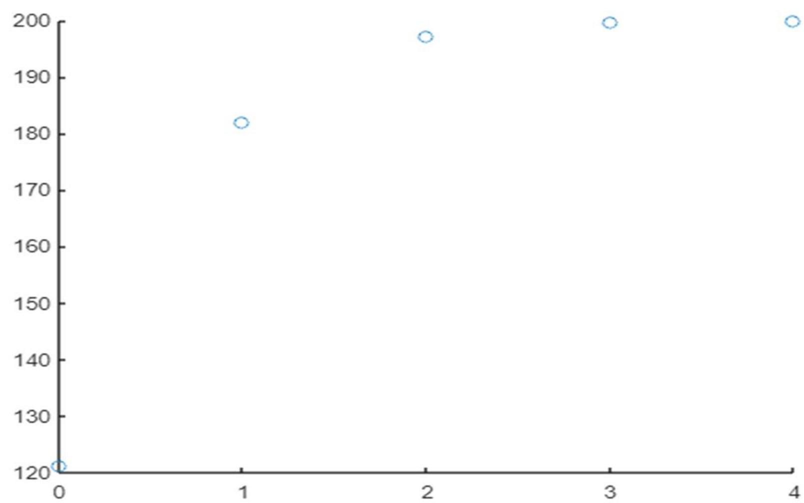
```
mean = 0.5000

Va = 0.5000

pd = 2

F = 5x1
```

	1
1	121.3061
2	60.6531
3	15.1633
4	2.5272
5	0.3159

```
G = 5x1
  121.3061
  181.9592
  197.1225
  199.6497
  199.9656
```


140
120
100
80
60
40
20
0
0
1
2
3
4

pdf


200
190
180
170
160
150
140
130
120
0
1
2
3
4

cdf

Module – 5 (Mech + civil)
Statistics, Queueing Theory and Reliability Engineering

Measures of central tendency

1.Mean and median of a set of observations

Code

```
q=[6 8 7 5 4 9 3]

mean(q)

median(q)
```

Output

t-test of difference between sample means

```
s_squar = 122.7778

s = 11.0805

t = 12.1081
```

2. (i) Mean of the grouped data giving the marks of 100 students

```
C=[1 10;11 20;21 30;31 40;41 50;51 60] %grouped data entered in this form
f=[3 16 26 31 16 8]  % given frequency
N=length(C)
k1=[]
% mean of each grouped data (x)
for i=1:N
    mean_x=(C(i,1)+C(i,2))/2
    k1=[k1 mean_x]
end
sumfx=sum(f.*k1) % summation of fx
mean_g= sumfx/sum(f)  % mean of grouped data
% cummulative frequency
c_f=0
k2=[]
for j=1:N
    c_f=c_f+f(j)
    k2=[k2 c_f]
end
```

Output

```
C = 6×2

     1    10
    11    20
    21    30
    31    40
    41    50
    51    60


f = 1×6

     3    16    26    31    16     8


N = 6

k1 =

     []

mean_x = 5.5000
k1 = 5.5000
mean_x = 15.5000
k1 = 1×2

    5.5000   15.5000

mean_x = 25.5000
k1 = 1×3

    5.5000   15.5000   25.5000

mean_x = 35.5000
k1 = 1×4

    5.5000   15.5000   25.5000   35.5000

mean_x = 45.5000
k1 = 1×5

    5.5000   15.5000   25.5000   35.5000   45.5000

mean_x = 55.5000
```

```
k1 = 1×6

    5.5000   15.5000   25.5000   35.5000   45.5000   55.5000


sumfx = 3200

mean_g = 32

c_f = 0

k2 =

     []

c_f = 3
k2 = 3
c_f = 19
k2 = 1×2

     3    19

c_f = 45
k2 = 1×3

     3    19    45

c_f = 76
k2 = 1×4

     3    19    45    76

c_f = 92
k2 = 1×5

     3    19    45    76    92

c_f = 100
k2 = 1×6

     3    19    45    76    92   100
```

2.(ii) Median of grouped data

```
%to find median class
N1=sum(f)
N_class=N1/2
% by observation we can say median class is 31 - 40
l= (30+31)/2
h=10
f_m=31
c=45
median_g=l+h/f_m*(N1/2-c) % standard formula for median
```

Output

2.(ii) Median of grouped data

```
N1 = 100

N_class = 50

l = 30.5000

h = 10

f_m = 31

c = 45

median_g = 32.1129
```

2.(iii) Mode of grouped data

Code

```
%to find Mode
%modal class is class of maximum frequency
fm= 31 %maximam frequency
f1= 26 %frequency just before modal class
f2= 16 %frequency just after modal class
h= 10  % width of the class
mode = l+((fm-f1)*h)/(2*fm-f1-f2) % standard formula for mode
```

Output

2.(iii) Mode of grouped data

```
fm = 31

f1 = 26

f2 = 16

h = 10

mode = 33
```

Measures of dispersion

We consider the above problem to find Quartile,mean , standard deviation

3.Quartile deviation

Code

```
% to find Quartile deviation

k2 % cumulative frequency

N1/4

% by observation we can select Q1 class that is 21 - 30

l1=(20+21)/2

c1=19

f1_q=26

h=10

Q1=l1+h/f1_q*((N1/4)-c1)    % standard formula of Q1


3*N1/4
% by observation we can select Q3 class that is 31 - 40

l2=(30+31)/2

c2=45

f2_q=31

h=10

Q3= l2+h/f2_q*((3*N1/4-c2))  % standard formula of Q3


Q_D=(Q3-Q1)/2    %Quartile deviation
```

Output

```
k2 = 1×6
     3    19    45    76    92   100

ans = 25

l1 = 20.5000

c1 = 19

f1_q = 26

h = 10

Q1 = 22.8077

ans = 75

l2 = 30.5000

c2 = 45

f2_q = 31

h = 10

Q3 = 40.1774

Q_D = 8.6849
```

5. Mean deviation

Code

```
r=abs(k1-mean_g)          % |x-mean|
t=f.*r                    % f*|x-mean|
M_D=sum(t)/N1              % Mean deviation formula
```

Output

```
r = 1×6
   26.5000   16.5000    6.5000    3.5000   13.5000   23.5000

t = 1×6
   79.5000  264.0000  169.0000  108.5000  216.0000  188.0000

M_D = 10.2500
```

6. standard deviation

Code

```
S_D= sqrt(sum(f.*r.^(2))/N1)
%Coefficient of variation
C_V=S_D/mean_g*100
```

Output

```
S_D = 12.3592

C_V = 38.6225
```

6.Reliability Engineering: Mean time to failure(MTTF)

Code

```
% compute mean time to failure
t =[0 10;10 20;20 30;30 40;40 50;50 60]  %time interval
ff=[3 16 26 31 16 8]                     %No of failures
N2=length(ff)
sumf=0
for i=1:N2
    sumf=sumf+t(i,2)*ff(i);
end
MTTF=sumf/sum(ff)
```

Output

```
t = 6×2
     0    10
    10    20
    20    30
    30    40
    40    50
    50    60

ff = 1×6
     3    16    26    31    16     8

N2 = 6

sumf = 0

MTTF = 36.5000
```

Module – 5(CSE+ISE+AIML+ECE+EEE)

Optimization Techniques

I Solving an optimization problem (Maximization)

Write a matlab code to maximize $z = x + 1.5y$ given $x \geq 0\,, y \geq 0$ subject to the constraints $x + 2y \leq 160, 3x + 2y \leq 240$.

Code

```
format short
clear;
prob=optimproblem("Description","Maximization","ObjectiveSense","maximize");
x=optimvar("x","LowerBound",0);
y=optimvar("y","LowerBound",0);
z=x+1.5*y;
prob.Objective=z %Objective function
c1=x+2*y;
prob.Constraints.one=c1<=160 %Constraint 1
c2=3*x+2*y;
prob.Constraints.two=c2<=240 %Constraint 2
sol=solve(prob)
Optivalue=evaluate(z,sol)
```

II Visualizing graphically

Visualize the above linear programming problem by plotting the lines and identify the corner points.

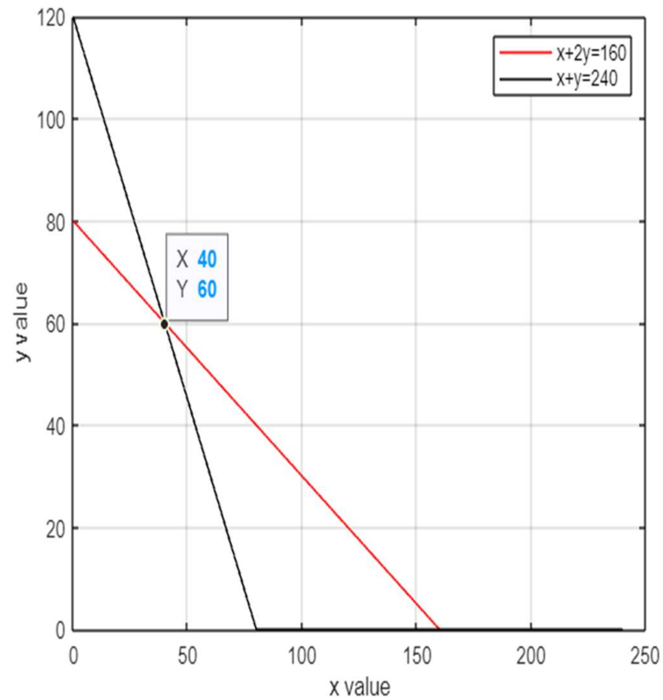
Code

```
clear;
%Input parameters
C=[1 1.5];%coefficients present in objective function
A=[1 2;3 2];%coefficients in the constaints
b=[160;240];%RHS(Constraints)


%Plotting the graphs
x=0:1:max(b);
y1=(b(1)-A(1,1).*x)./A(1,2);
y2=(b(2)-A(2,1).*x)./A(2,2);


y1=max(0,y1)
y2=max(0,y2)


plot(x,y1,'r',x,y2,'k')
xlabel('x value')
ylabel('y value')
```

```
legend('x+2y=160','x+y=240')
grid on

%Identify the feasible region and the corner points
```

Output

```
  OptimizationProblem : Maximization

	Solve for:
       x, y

	maximize :
       x + 1.5*y


	subject to one:
       x + 2*y <= 160

	subject to two:
       3*x + 2*y <= 240

	variable bounds:
       0 <= x

       0 <= y

Solving problem using linprog.

Optimal solution found.
sol = struct with fields:
    x: 40
    y: 60.0000

Optivalue = 130
```



II Solving an optimization problem (Maximization)

Write a matlab code to minimize $z = 5x + 4y$ given $x \geq 0, y \geq 0$ subject to the constraints $x + 2y \geq 10, x + y \geq 8, 2x + y \geq 12.$

Code

```
format short
clear;
prob=optimproblem("Description","Minimization");
x=optimvar("x","LowerBound",0);
y=optimvar("y","LowerBound",0);
z=5*x+4*y;
prob.Objective=z;
c1=x+2*y;
prob.Constraints.one=c1>=10;
c2=x+y;
prob.Constraints.two=c2>=8;
c3=2*x+y;
prob.Constraints.three=c3>=12;

show(prob)
sol=solve(prob)
```

```
Optimumvalue=evaluate(z,sol)
```

Output

```
  OptimizationProblem : Minimization

        Solve for:
       x, y

        minimize :
       5*x + 4*y


        subject to one:
       x + 2*y >= 10

        subject to two:
       x + y >= 8

        subject to three:
       2*x + y >= 12

        variable bounds:
       0 <= x

       0 <= y

Solving problem using linprog.

Optimal solution found.
sol = struct with fields:
    x: 4.0000
    y: 4.0000

Optimumvalue = 36
```

Visualize the minimization problem as given in problem I by plotting the graph using a similar code given above.

*****