Date 21-3-24

Expt. No. 8. Page No.

PRACTICAL - 8

AIM: WAP to plot Normal and Exponential Distribution.

SOFTWARE USED - SciLab

SOFTWAR

SOURCE CODE:

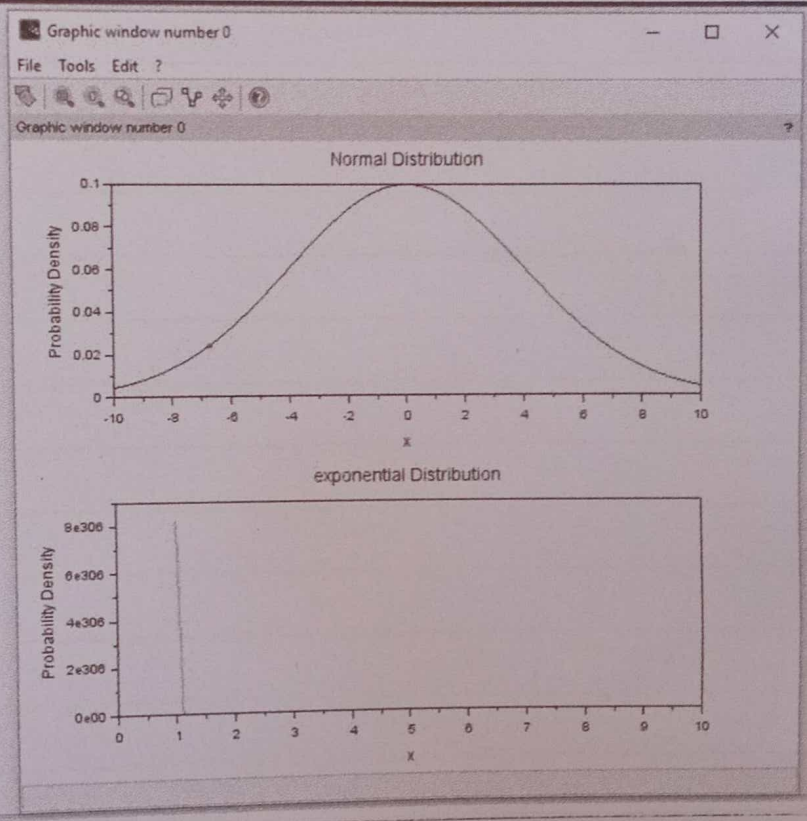
```
clc
mean=input("Enter mean:")
sigma=input("Enter sigma:")
lambda=input("Ener the lambda:")
x=linspace(-10,10,100)
normal=(1/(sigma*sqrt(2*%pi)))*exp(-0.5*((x-mean)/sigma).^2)
exponential=lambda*exp(-lambda*x)
clf();
subplot(2,1,1)
plot(x,normal,'-b')
xlabel('x')
ylabel('Probability Density')
title('Normal Distribution')
x=linspace(0,10,100)
subplot(2,1,2)
plot(x,exponential,"-g")
xlabel('x')
ylabel('Probability Density')
title('exponential Distribution');
```

Teacher's Signature

Expt. No. _____ Page No. _____

OUTPUT :

```
Enter mean:0

Enter sigma:4

Ener the lambda:88

exec: Wrong number of output argument(s): 0 expected.

-->
```



Date 25.4.24

Expt. No. 9.

Page No.

PRACTICAL - 9

AIM : Fitting of Linear regression line to the data set given in theory.

SOFTWARE USED : SCILAB

SOURCE CODE :

```
clc
x=[1 2 3 4 5 6 7];
disp("data points")
disp(x)
y=[1.5 3.8 6.7 9.0 11.2 13.6 16]
disp("data values")
disp(y)
n=length(x);
sum_x=0;
sum_y=0;
sum_xy=0;
sum_xx=0;
for i=1:n
    sum_x=sum_x+x(i);
    sum_y=sum_y+y(i);
    sum_xy=sum_xy+x(i)*y(i);
    sum_xx=sum_xx+x(i)*x(i);

end
slope=(n*sum_xy-sum_x*sum_y)/(n*sum_xx-sum_x*sum_x)
intercept = (sum_y-slope*sum_x)/n;
disp("slope")
disp(slope)
disp("intercept")
disp(intercept)
y_pred=slope*x+intercept;
clf;
plot(x,y,'bo','MarkerSize',8,'LineWidth',1.5);
plot(x,y_pred,'r-','LineWidth',1.5);
xlabel('x');
ylabel('y');
title('Linear Regression');
legend('data points','linear regression line');
```

Teacher's Signature

THEORY :

x	1	2	3	4	5	6	7
y	1.5	3.8	6.7	9.0	11.2	13	16

x	y	xy	x^2
1	1.5	1.5	1
2	3.8	7.6	4
3	6.7	20.1	9
4	9.0	36	16
5	11.2	56	25
6	13.6	81.6	36
7	16	11.2	49

$y = mx + k$

where, m = slope of line

$\hat{m}$ = y-intercept

i) Calculating m, i.e, slope

$$m = \frac{n\Sigma xy - \Sigma x \Sigma y}{n\Sigma x^2 - (\Sigma x)^2} = \underline{2.41486}$$

ii) Calculate :

$$\theta = \frac{\Sigma y - m.\Sigma x}{n} = \frac{81.8 - (2.414)(28)}{7} = \underline{0.82857}$$

iii) Final regression line w.r.t m & θ $\Rightarrow$ $y = mx + c$

$y = 2.4m + (-0.83)$

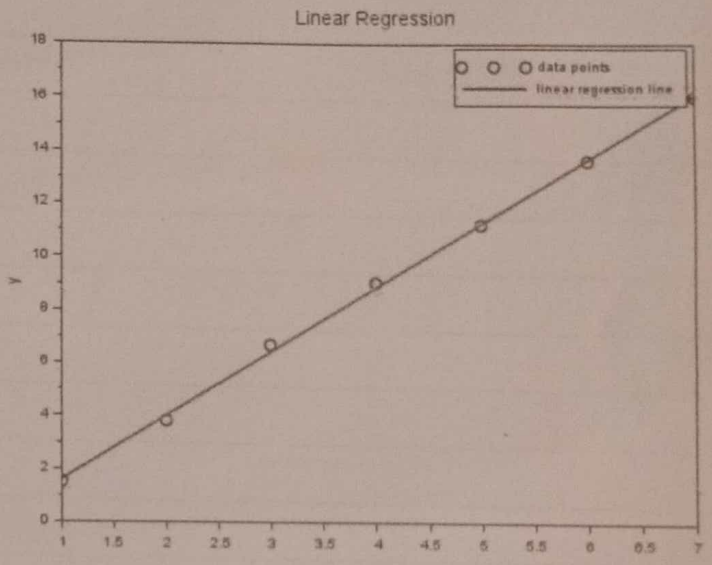
$\therefore$ $\underline{y = 2.91x - 0.83}$

Date _____

Expt. No. _____ Page No. _____

OUTPUT :

```
"data points"

 1.   2.   3.   4.   5.   6.   7.

"data values"

 1.5   3.8   6.7   9.   11.2   13.6   16.

"slope"

 2.4142857

"intercept"

-0.8285714
```



Expt. No. 10.

Date 2.5.24

Page No.

PRACTICAL - 10

AIM : WAP in Scilab to solve assignment problem in Linear Program

SOFTWARE USED : Scilab

SOURCE CODE :

```
clc
clear
n=input("how many workers and job do you have  :");

for i=1:n
        for j=1:n
        printf("worker %d  job%d :",i,j);
        T(i,j)=input("");
        end
end

printf("\n\nDATA YOU ENTERED IS      :\n ");
for i=1:n
  printf("\t\tjob%d ",i);
end

for i=1:n
  printf("\nworker%d  ",i);
  for j=1:n
    printf("\t\t%d ",T(i,j));
  end
end

minim=[1000,1000,1000,1000,1000,1000,1000,1000,1000,1000];

for i=1:n
  for j=1:n
    if(T(i,j)<=minim(i)) then
      minim(i)=T(i,j);
```

Teacher's Signature

Date _____

Expt. No. _____ Page No. _____

```
for i=1:n
  printf("\nworker%d ",i);
  for j=1:n
     printf("\t\t%d ",T(i,j));
  end
end

zerr=[1000,1000,1000,1000,1000,1000];
zerc=[1000,1000,1000,1000,1000,1000];

for i=1:n
  for j=1:n
     if(T(i,j)==0) then
        zerr(i)=0;
        zerc(j)=0;
     end
  end
end

f=0; y=0;

for i=1:n
  if(zerr(i)) then
     f=1;
        end
end

for i=1:n
  if(zerc(i)) then
     y=1;
        end
end
```

```
if f y then

        mn=[1000,1000,1000,1000,1000,1000,1000,1000,1000,1000];

        for j=1:n
        for i=1:n
        if(T(i,j)<=mn(j)) then
        mn(j)=T(i,j);
       end
                end
        end

  for j=1:n
        for i=1:n
                T(i,j)=T(i,j)-mn(j);
        end
  end

        printf("\n\n**********data after column minimum decrement is**********\n\n")

        for i=1:n
                printf("\t\tjob%d ",i);
        end

for i=1:n
  printf("\nworker%d ",i);
  for j=1:n
```

Signature _____

Expt. No. _____ Date _____

Page No. _____

OUTPUT

```
How many workers and job do you have

DATA YOU ENTERED IS :
        job1        job2        job3
worker1
worker2
worker3
**************data after row minimum decrement is**************
        job1        job2        job3
worker1
worker2
worker3
**************data after column minimum decrement is**************
        job1        job2        job3
worker1
worker2
worker3
**************final job assignment is**************
assign job    to worker
assign job    to worker
assign job    to worker
```

```
      printf("\t%d ",T(i,j));
    end
  end

  printf("\n\n\n**************Final job assignment is**************\n\n");

  for j=1:n
    for i=1:n
      if(T(i,j)==0) then
        printf("\n assign job %d to worker %d",j,i);
        for z=1:n
          if (z==i) then
            continue;
          end

          if(T(z,j)==0) then
            T(z,j)=1000;
          end
          end

          for l=1:n
            if(l==j) then
              continue;
            end

            if(T(i,l)==0) then
              T(i,l)=1000;
            end
          end
        end
      end
    end
  end
  end
  printf("\n");
```

Date 9.5.24

Expt. No. 11.

Page No.

CONTENT BEYOND SYLLABUS

PRACTICAL - II

AIM : To find the Eigen values and Eigen vectors in Scilab

SOFTWARE USED : Scilab

OUTPUT :

```
 "enter the Matrix"
\5

\4

\1

\2


 "The characteristic equation is:"

 " e^2 + "  "-7"  "*e + "  "6"  " = 0"

 "First Eigen value is:"

  6.

 "First Eigen vector is:"

  4.
  1.

 "Second Eigen value is:"

  1.

 "Second Eigen vector is:"

  4.
 -4.
```

Teacher's Signature

SOURCE CODE :

File Edit Format Options Window Execute ?

eigenval.sci | eigenvector.sci

```
clc
disp('enter the Matrix')
for i=1:2
 for j=1:2
 A(i,j)=input('\');
 end
end
b=A(1,1)+A(2,2);
c=A(1,1)*A(2,2)-A(1,2)*A(2,1);
disp('The characteristic equation is:')
disp([' e^2 + ' string(-b) '*e + ' string(c) ' = 0'])
e1=(b+sqrt(b^2-4*c))/2;
e2=(b-sqrt(b^2-4*c))/2;
if A(1,2) ~= 0
 v1 = [A(1,2); e1-A(1,1)];
 v2 = [A(1,2); e2-A(1,1)];
elseif A(2,1) ~= 0
 v1 = [e1-A(2,2); A(2,1)];
 v2 = [e2-A(2,2); A(2,1)];
else
 v1 = [1; 0];
 v2 = [0; 1];
end
 disp('First Eigen value is:');
 disp(e1)
 disp('First Eigen vector is:');
 disp (v1)
 disp('Second Eigen value is:');
 disp(e2)
 disp('Second Eigen vector is:');
 disp(v2)
```