KARPAGAM ACADEMY OF HIGHER EDUCATION

KARPAGAM UNIVERSITY

(Deemed University Under Section 3 of UGC Act 1956)

Coimbatore -21.

DEPARTEMENT OF COMPUTER SCIENCE

SYLLABUS

Semester-I

17CSP112 DATA MINING LAB USING MATLAB– PRACTICAL 4H – 2C

Instruction Hours / week: L: 0 T: 0 P: 4 Marks: Int : **40** Ext : **60** Total: **100**

1. Write a MATLAB program to do all basic matrix operations in MATLAB for a multidimensional array.

2. Write a MATLAB code to compares and contrasts some similarity and distance measures for the following .

(a) Compute the Hamming distance and the Jaccard similarity between the following two binary vectors.

x = 0101010001

y = 0100011000

3. a. Plot the graph of f (x) = $\exp(-2x^2 - 3y^2)$. Choose appropriate intervals for x and y.

b. Plot the graph of f (x) = cos(x) sin(y). Choose appropriate intervals for x and y.

4. The number of children for different patients in a database is given with a vector

c= {31027634-20010156}

. Find the outliers in the set C using standard statistical parameters mean and variance.

a. If the threshold value is changed from ±3 standard deviations to ±2 standard deviations what additional outliers are found?

5. For a given data set X of three-dimensional samples

X=[{120}{314}{215}{016}{243}{442}{521}{777}{000}{333}]

a) find the outliers using the distance-based technique if

i) the threshold distance is 4 and threshold fraction p for non-neighbor samples is 3.

ii) the threshold distance is 6 and threshold fraction p for non-neighbor samples is 2.

6. Given the data set X with three input features and one output feature representing the classification of samples

X:	**I_1**	**I_2**	**I_3**	**O**
	2.5	1.6	5.9	0
	7.2	4.3	2.1	1
	3.4	5.8	1.6	1
	5.6	3.6	6.8	0
	4.8	7.2	3.1	1
	8.1	4.9	8.3	0
	6.3	4.8	2.4	1

Rank the features using a comparison of means and variances

7. A data set for analysis includes only one attribute X:

X={7125185913121971212133451387 6}

a) What is the mean of the data set X?

b) What is the median?

c) What is the mode and what is the modality of the data set X?

d) Find the standard deviation for X.

e) Give a graphical summarization of the data set X using a boxplot representation.

f) Find outliers in the data set X.

8. Given a data set with two dimensions X and Y.

X	**Y**
1	5
4	2.75
3	3
5	2.5

a) Use a linear-regression method to calculate the parameters α and β where $y= \alpha + \beta x$.

b) Estimate the quality of the model obtained in a) Using the correlation coefficient r.

9. The following is the data set X:

X:	**Year**	**A**	**B**
	1996	7	100
	1997	5	150
	1998	7	120
	1999	9	150
	2000	5	130
	2001	7	150

Create 2D Presentations:

a) Show a bar chart for the variable A

b) Show a histogram for the variable B.

c) Show a line chart for the variable B

d) Show a pie chart for the variable A

10. Create a MATLAB function to count the number of lines in a text file.

11. Create a structure array for student mark details and print a plot for the marks of the students.

12. The test scores for the three students are given in the following table:

	RDBMS	**OracleDBA**	**WebDesigning**	**AI**
Smith	66	91	95	83
Sam	91	88	80	73
John	80	88	80	78

Find the best student using multifactorial evaluation if the weight factors for the subjects are given as the vector W= [0.3 0.2 0.1 0.4]

BASIC MATRIX OPERATIONS

(Multidimensional Array)

EX.NO:1

DATE: 26.07.17

AIM:

To write a MATLAB program to do all basic operations in mat lab for a multidimensional array.

ALGORITHMS:

Step1: Start the program.

Step2: Declare a matrix. a=[1 2 3;5 4 3;1 2 6];

Step3: The matrix b can be included as second element of a three dimensional matrix by extending the two dimensional matrix a.

B=[2 4 6;1 3 6;3 6 9];

A(:,:,)=b

Step4: The basic matrix operations for a multidimensional are transpose, addition, subtraction, multiplication, right division, left division, exponentiation,diagonal matrix, determinant, rank.

Step5: It is performed using the statements. B=b',b+c,b-c,b*c,b./c,b.\c,b.^2, diag(c), trace(c),det(c),rank(c).

Step6: Save and execute program.

Step7: Stop the program.

PROGRAM:

```
disp(‘ BASIC MATRIX OPERATIONS’);
disp(‘ EXTENDING THE MATRIX DIMENSIONS’);
a=[1 2 3;4 5 6;7 8 9];
b=[2 1 4;3 6 5;9 8 7];
a(:,:,2)=b
c=[9 8 7;6 5 4;3 2 1 ];
disp(‘ transpose of a matrix’);
b=b’
disp(‘ addition of a matrix’);
add=b+c
disp(‘ subtraction of a matrix’);
sub=b-c
disp(‘ multiplication of a matrix’);
mul=b*c
disp(‘ right division of a matrix’);
rdiv=b./c
disp(‘ left division of a matrix’);
ldiv=b.\c
disp(‘exponentation of a matrix’);
exp=b.^c
disp(‘diagonals of a matrix’);
diag(c)
```

OUTPUT:

```
1.3333    2.5000    7.0000

left division of a matrix

ldiv=

    4.5000    2.6667    0.7778

    6.0000    0.8333    0.5000

    0.7500    0.4000    0.1429


expenentiation of a matrix

exp=

    512   6561   4782969

      1   7776      4096

     64     25         7


diagonals of a matrix

ans=

     9

     5

     1
```

RESULT:

The above program has been executed successfully and the output was verified.

SIMULARITY AND DISTANCE MEASURE

EX.NO:2

DATE: 26.07.17

AIM:

Write a MATLAB code to compares and contrasts some similarity and distance measures for the following.

(a) For binary data, the L1 distance corresponds to the Hamming distance; that is, the number of bits that are different between two binary vectors. The Jaccard similarity is a measure of the similarity between two binary vectors. Compute the Hamming distance and the Jaccard similarity between the following two binary vectors.

x = 0101010001
y = 0100011000

ALGOTITHM:

Step1: start the program.

Step2: Let x=[0 1 0 1 0 1 0 0 0 1]

Let y=[0 1 0 0 0 1 1 0 0 0]

Step3: Compute the length of the vector n=length(x).

Step4: With the help of for loop and if else statement to measure the number of Disagreement between two vectors.

Step5: Compute jaccard similarity using the formula

j=m11/(m01+m10+m11)

Step6: Display hamming distance and jaccard similarity.

Step7: Save and execute the program.

Step8: Stop the program.

PROGRAM:

```
x=[0 1 0 1 0 1 0 0 0 1]
y=[0 1 0 0 0 1 1 0 0 0]
n=length(x)

%FINDING HAMMING DISTANCE%
p=0;
q=0;
for i=1:n
if x(i)==y(i)
p=p+1;
else
q=q+1;
end
end
disp("Hamming Distance is:"+string(q));
```

OUTPUT:

```
x =
    0    1    0    1    0    1    0    0    0    1

y =
    0    1    0    0    0    1    1    0    0    0

n =
   10

hamming distance is 3
```

RESULT:

The above program has been executed successfully and the output was verified.

PLOT THE GRAPH

EX.NO:3

DATE: 29.07.17

AIM:

To plot the graph.

i) F(x)=exp(-2x-3y2)
ii) F(x)=cos(x)sin(y)and choose appropriate interates for x and y.

ALGORITHM:

Step1: Start the program.

Step2: To plot the graph.

i) F(x)=exp(-2x-3y2)

Step3: Declare x=[0:0.05:5]; y=[0:0.05:5];

Step4: Assign x1=exp(-2*x.^2-3*y.^2) and plot the graph exp pot(x,x1)

Step5: Give title, x label & y label to the graph.

Step6: To plot the graph.

ii) F(x)=cos(x)sin(y)

Step7: Declare x[0:0.02:10]; y[0:0.02:10];

Step8: Assign x1=cos(x), y1=sin(y) and put the graph using plot(x1,y1)

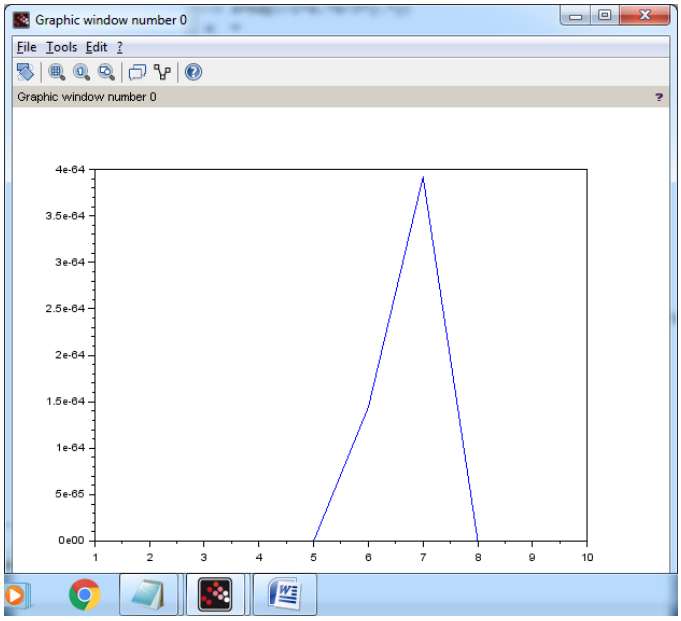
Step9: Give the title, x label, y label, grid on and to the graph.

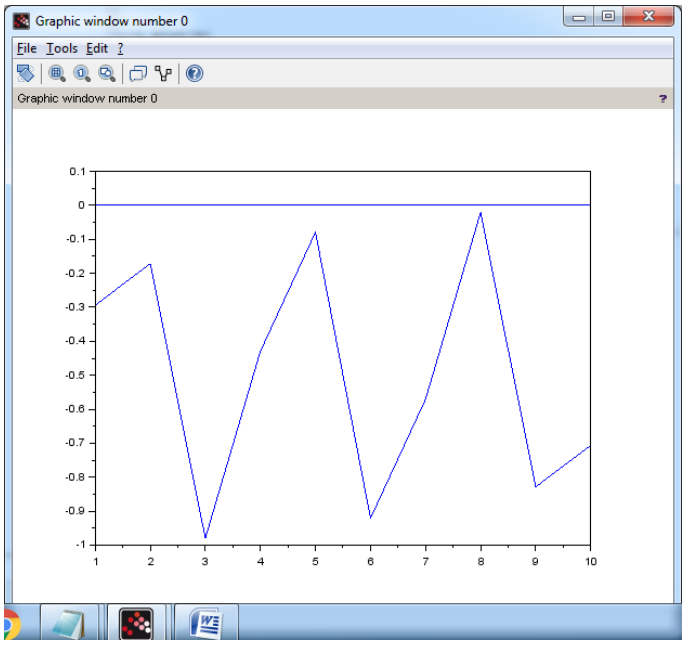
Step10: Stop the process.

PROGRAM:

```
x=[1,2,3,4,5,6,7,8,9,10]
x  =
  1.   2.   3.   4.   5.   6.   7.   8.   9.   10.
--> y=[10,9,8,7,6,5,4,3,2,1]
y  =
  10.   9.   8.   7.   6.   5.   4.   3.   2.   1.
--> a=(-2*x.*x-3*y.*y)
a  =
  -302.  -251.  -210.  -179.  -158.  -147.  -146.  -155.  -174.  -203.
--> a=exp(-2*x.*x-3*y.*y)
a  =
   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
-->plot(a)
-->plot(a)
--> b=cos(x).*sin(y)
b  =
column 1 to 6
  -0.2939359  -0.1715018  -0.9794572  -0.4294351  -0.0792596  -0.9207306
column 7 to 10
  -0.5705551  -0.020533  -0.8284884  -0.7060543
-->plot(b)
```

OUTPUT:





RESULT:

The above program has been executed successfully and the output was verified.

OUTLIERS USING STANDARD STATISTICAL PARAMETERS

(Mean and Variance)

EX.NO:4

DATE: 02.08.17

AIM:

The number of children for different patients in a database is given with a vector.

C={3,1,0,7,6,3,4,-2,0,0,10,15,6}

a. To find the outliers in the set C using statistical parameters mean and variance.
b. If the threshold value is changed from±3 standard deviation to ±2 standard deviation. What additional outliers are found.

ALGORITHM:

Step1: Start the process.

Step2: Define the mean and variance in the given set c.

Step3: Declare the standard deviation.

Step4: Find the number of rows with outliers greater than two standard deviation.

Step5: Save and run the program.

Step6: Stop the process.

PROGRAM:

```
C=[3 1 0; 2 7 6; 3 4 -2; 0 0 1; 0 1 5 6]

mu=mean(C)

vr=var(C)

sigma=std(C)

[n,p]=size(C)

outliers=abs(C -mu(ones(n,1),:))>2*sigma(ones(n,1),:)

nout=sum(outliers)

C(any(outliers'),:)=[]
```

OUTPUT:

C= 3 1 0

2 7 6

3 4 -2

0 0 1

0 15 6

mu= 1.6000 5.4000 2.2000

vr= 2.3000 36.3000 13.2000

sigma= 1.1566 6.0249 3.6332

n= 5

P= 3

Outliers=

0 0 0

0 0 0

0 0 0

0 0 0

0 0 0

nout=

0 0 0

RESULT:

The above program has been executed successfully and the output was verified.

OUTLIERS USING DISTANCE-BASED TECHNIQUE

(3-DIMENSIONALS)

EX.NO:5

DATE: 09.08.17

AIM: For a given data set X of three-dimensional samples.

X=[{1,2,0},{3,1,4},{2,1,5},{0,1,6},{2,4,3},{4,4,2},{5,2,1},{7,7,7},{0,0,0},{3, 3,3}]

a) Find the outliers using the distance- based technique if

I) The threshold distance is 4, and threshold fraction p for non- neighbor Sample is 3.

II) The threshold distance is 6, and threshold fraction p for non-neighbor Sample is 2.

ALGORITHM:

Step1: Start the program.

Step2: Calculate Euclidean distance between each pair of data using the formula as follows:

Euclidean distance=[(x1-x2)^2+(y1-y2)^2+(z1-z2)^2] and tabulate the result as a 10 x 10 matrix 'd'

Step3: Find out the outliers for threshold distance 4 and 6 by using logical array d>= 4 and d>=6.

Step 4: Column wise sum of the above logical array gives the vector with threshold distance 4 and 6 respectively.

Step 5: From the vectors find the outliers which have threshold fraction above than 3 and 3 respectively and display the results

Step6: Save and execute the program.

Step7: Stop the program.

PROGRAM

```
s=[1 2 0;3 1 4;2 1 5;0 1 6;2 4 3; 4 4 2;5 2 1;7 7 7;0 0 0;3 3 3;]
 n=length(s);
 t4=zeros(n,n);
 t6=zeros(n,n);
 fprintf('\n euclidean distance of every pair of data')
 x=pdist(s,'euclid');
 d=squareform(x)
 for p=1:n
    for q=1:n
        if d(p,q)>=4
            t4(p,q)=d(p,q);
        end
        if d(p,q)>=6
            t6(p,q)=d(p,q);
        end
    end
 end
 fprintf('\n')
 disp('threshold distance >=4')
 disp(t4);
 fprintf('\n')
 fprintf('\n clustering of data based on euclidean distance \n')
 z=linkage(x)
```

```
%plotting the dendrogram of the data with threshold>2
subplot(2,1,1)
[h1,t1,p1]=dendrogram(z,'colorthreshold',2);
fprintf('\n permutation of data based on their threshold distance \n');
disp(p1)
disp('outliers with non neighbouring threshold fraction >2')
mask=z(:,3)>2;
mask1=[0,mask'];
disp(p1(logical (mask1)));
fprintf('\n')
fprintf('\n')
disp('threshold distance >=6')
disp(t6);
%plotting the dendrogram of the data with threshold >3
subplot(2,1,2)
[h2,t2,p2]=dendrogram(z,'colorthreshold',3);
msk=z(:,3)>3;
 mask2=[0,msk'];
disp(p2(logical (mask2)));
fprintf('\n')
```

OUTPUT:

s =

1	2	0
3	1	4
2	1	5
0	1	6
2	4	3
4	4	2
5	2	1
7	7	7
0	0	0
3	3	3

euclidean distance of every pair of data

d = Columns 1 through 6

0	4.5826	5.1962	6.1644	3.7417	4.1231
4.5826	0	1.4142	3.6056	3.3166	3.7417
5.1962	1.4142	0	2.2361	3.6056	4.6904
6.1644	3.6056	2.2361	0	4.6904	6.4031
3.7417	3.3166	3.6056	4.6904	0	2.2361
4.1231	3.7417	4.6904	6.4031	2.2361	0
4.1231	3.7417	5.0990	7.1414	4.1231	2.4495
10.4881	7.8102	8.0623	9.2736	7.0711	6.5574
2.2361	5.0990	5.4772	6.0828	5.3852	6.0000
3.7417	2.2361	3.0000	4.6904	1.4142	1.7321

Columns 7 through 10

4.1231	10.4881	2.2361	3.7417
3.7417	7.8102	5.0990	2.2361
5.0990	8.0623	5.4772	3.0000
7.1414	9.2736	6.0828	4.6904
4.1231	7.0711	5.3852	1.4142
2.4495	6.5574	6.0000	1.7321
0	8.0623	5.4772	3.0000
8.0623	0	12.1244	6.9282
5.4772	12.1244	0	5.1962
3.0000	6.9282	5.1962	0

threshold distance >=4

Columns 1 through 6

0	4.5826	5.1962	6.1644	0	4.1231
4.5826	0	0	0	0	0
5.1962	0	0	0	0	4.6904
6.1644	0	0	0	4.6904	6.4031
0	0	0	4.6904	0	0
4.1231	0	4.6904	6.4031	0	0
4.1231	0	5.0990	7.1414	4.1231	0
10.4881	7.8102	8.0623	9.2736	7.0711	6.5574
0	5.0990	5.4772	6.0828	5.3852	6.0000
0	0	0	4.6904	0	0

Columns 7 through 10

4.1231	10.4881	0	0
0	7.8102	5.0990	0
5.0990	8.0623	5.4772	0
7.1414	9.2736	6.0828	4.6904
4.1231	7.0711	5.3852	0
0	6.5574	6.0000	0
0	8.0623	5.4772	0
8.0623	0	12.1244	6.9282
5.4772	12.1244	0	5.1962
0	6.9282	5.1962	0

clustering of data based on euclidean distance

z =

5.0000	10.0000	1.4142
2.0000	3.0000	1.4142
6.0000	11.0000	1.7321
12.0000	13.0000	2.2361
4.0000	14.0000	2.2361
1.0000	9.0000	2.2361
7.0000	15.0000	2.4495
16.0000	17.0000	3.7417
8.0000	18.0000	6.5574

permutation of data based on their threshold distance

5 10 6 2 3 4 7 1 9 8

outliers with non neighbouring threshold fraction >2

3 4 7 1 9 8

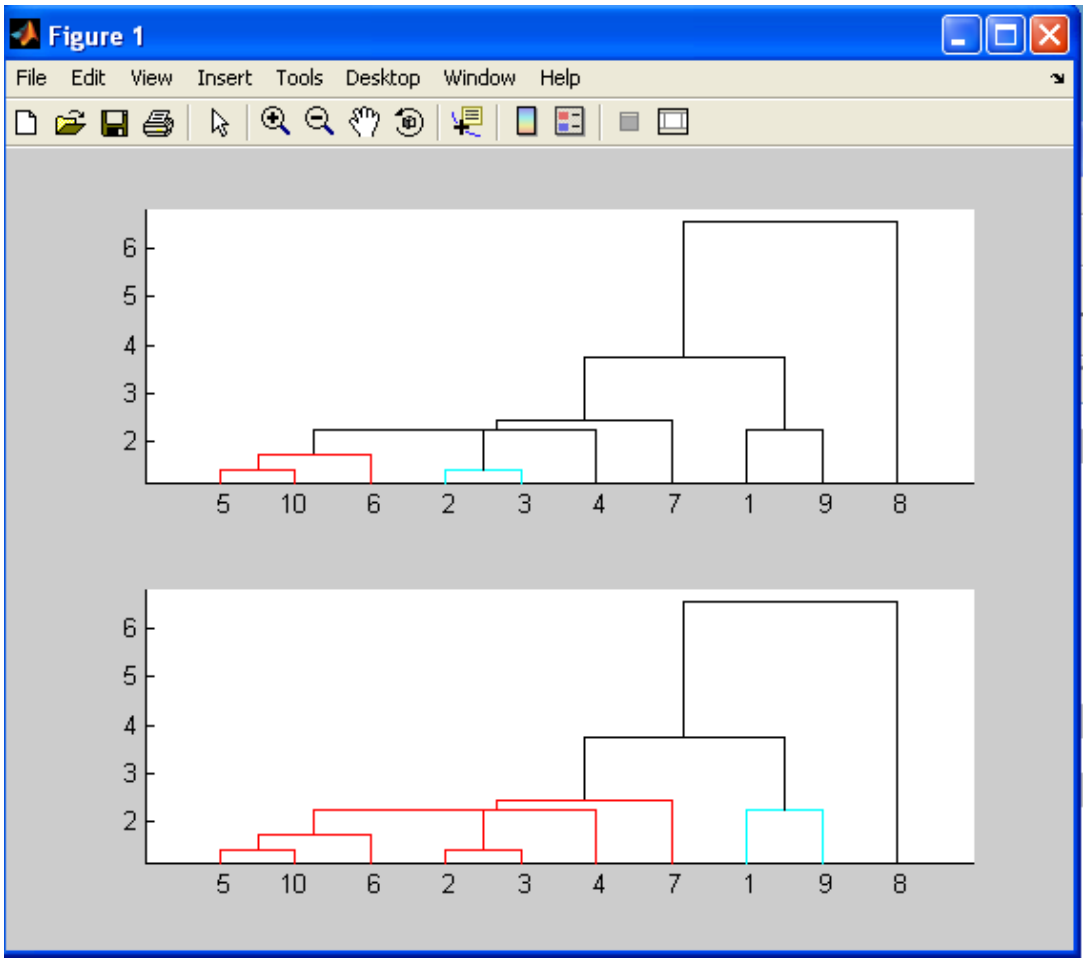
threshold distance >=6

Columns 1 through 6

0	0	0	6.1644	0	0
0	0	0	0	0	0
0	0	0	0	0	0
6.1644	0	0	0	0	6.4031
0	0	0	0	0	0
0	0	0	6.4031	0	0
0	0	0	7.1414	0	0
10.4881	7.8102	8.0623	9.2736	7.0711	6.5574
0	0	0	6.0828	0	6.0000
0	0	0	0	0	0

Columns 7 through 10

0	10.4881	0	0
0	7.8102	0	0
0	8.0623	0	0
7.1414	9.2736	6.0828	0
0	7.0711	0	0
0	6.5574	6.0000	0
0	8.0623	0	0
8.0623	0	12.1244	6.9282
0	12.1244	0	0
0	6.9282	0	0



RESULT:

The above program has been executed successfully and the output was verified.

MEAN AND VARIANCE

EX.NO:6

DATE: 16.08.17

AIM:

Given the data set X with three input features and one output feature representing the classification of samples

X:	I_1	I_2	I_3	O
	2.5	1.6	5.9	0
	7.2	4.3	2.1	1
	3.4	5.8	1.6	1
	5.6	3.6	6.8	0
	4.8	7.2	3.1	1
	8.1	4.9	8.3	0
	6.3	4.8	2.4	1

Rank the features using a comparison of means and variances

ALGORITHM:

Step1: Start the program.

Step2: Assign a1=[2.5 1.6 5.9],a2=[7.2 4.3 2.1],a3=[3.4 5.8 1.6],

A4=[5.6 3.6 6.8],a5=[4.8 7.2 3.1],a6=[8.1 4.9 8.3],a7=[6.3 4.8 2.4].

Step3: Calculate mean and variance by using the function.

Step4: Compare the variance of mean and variance.

Step5: Rank the features based on corresponding values.

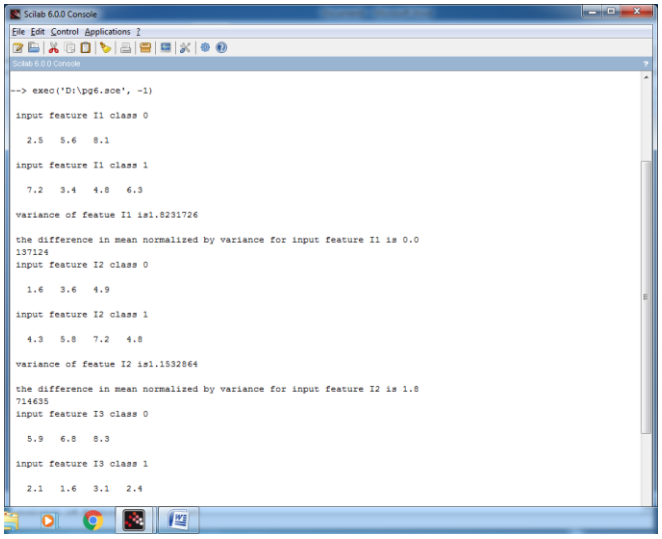
Step6: Save and execute the program.

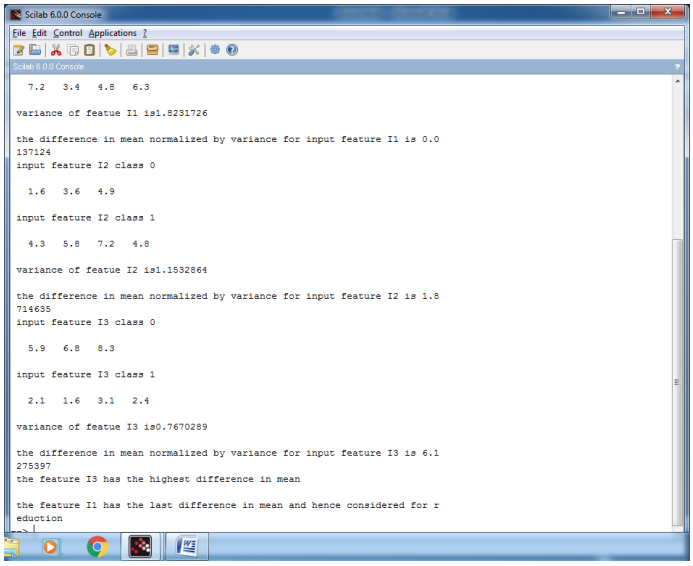
Step7: Stop the program.

Coding:

```
disp('input feature I1 class 0');
I10=[2.5,5.6,8.1]
disp(I10)
disp('input feature I1 class 1');
I11=[7.2,3.4,4.8,6.3]
disp(I11)
SE_I1=sqrt(variance(I10)/length(I10)+variance(I11)/length(I11));
disp('variance of featue I1 is'+string(SE_I1));
tI1=abs(mean(I10)-mean(I11))/SE_I1
disp('the difference in mean normalized by variance for input feature I1 is
'+string(tI1));
disp('input feature I2 class 0');
I20=[1.6,3.6,4.9]
disp(I20)
disp('input feature I2 class 1');
I21=[4.3,5.8,7.2,4.8]
disp(I21)
SE_I2=sqrt(variance(I20)/length(I20)+variance(I21)/length(I21));
disp('variance of featue I2 is'+string(SE_I2));
tI2=abs(mean(I20)-mean(I21))/SE_I2
disp('the difference in mean normalized by variance for input feature I2 is
'+string(tI2));
disp('input feature I3 class 0');
I30=[5.9,6.8,8.3]
disp(I30)
disp('input feature I3 class 1');
I31=[2.1,1.6,3.1,2.4]
disp(I31)
SE_I3=sqrt(variance(I30)/length(I30)+variance(I31)/length(I31));
disp('variance of featue I3 is'+string(SE_I3));
tI3=abs(mean(I30)-mean(I31))/SE_I3
disp('the difference in mean normalized by variance for input feature I3 is
'+string(tI3));
disp('the feature I3 has the highest difference in mean');
disp('the feature I1 has the last difference in mean and hence considered for
reduction');
```

OUTPUT:





RESULT:

The above program has been executed successfully and the output was verified.

MEAN, MEDIAN, MODE, STANDARD DEVIATION AND BOX PLOT

EX.NO:7

DATE: 19.08.17

AIM:

A data set for analysis includes only one attribute X:

X={7,12,5,18,5,9,13,12,19,7,12,12,13,3,4,5,13,8,7,6}

a) What is the mean of the data set X?

b) What is the median?

c) What is the mode, and what is the modality of the data set X?

d) Find the standard deviation for X.

e) Give a graphical summarization of the data set X using a box plot representation.

f) Find outliers in the data set X.

ALGORITHM:

Step1: Start the program.

Step2: Let X= [7 12 5 18 5 9 13 12 19 7 12 12 13 3 4 5 13 8 7 6].

Step3: Compute the length of the vector and the values are sorted using sort(x) and length(x) function.

Step4: Calculate mean, median, mode, standard deviation, and outliers using the Formulas.

Step5: Show the boxplot representation using the graphical method.

Step6: Save and execute the program.

Step7: Stop the program.

PROGRAM:

```
x= [7 12 5 18 5 9 13 12 19 7 12 12 13 3 4 5 13 8 7 6]
s=gsort(x,’g’,’i’)
n=length(x)
a=max(s)
b=min(s)
c=(a-b)/2
mode=x(c)
m=mean(x)
m1=median(x)
s=std(x)
t=m+2*s
for i=1:n
   if(x(i)>t)
disp('OUTLIER ‘)
   end
end
```

OUTPUT:

```
x =

  Columns 1 through 13

    7    12     5    18     5     9    13    12    19     7    12    12    13

  Columns 14 through 20

    3     4     5    13     8     7     6

s =

  Columns 1 through 13

    3     4     5     5     5     6     7     7     7     8     9    12    12

  Columns 14 through 20

   12    12    13    13    13    18    19


n =    20
a =    19
b =     3
c =     8
mode =    12
m =    9.5000
m1 =    8.5000
s =    4.5364
t =   18.5728

OUTLIER 19
```

RESULT:

The above program has been executed successfully and the output was verified.

LINEAR REGRESSION & ESTIMATE THE MODEL USING CORRELATION COEFFICIENT

EX NO : 8

DATE: 23.08.17

AIM:

Given a data set with two dimensions X and Y.

X	Y
1	5
4	2.75
3	3
5	2.5

a) Use a linear-regression method to calculate the parameters α and β where y= α + βx.

b) Estimate the quality of the model obtained in a) Using the correlation coefficient r.

ALGORITHM:

Step 1: Start the program.

Step 2: Assign **x= [1 4 3 5]** and **y= [5 2.75 3 2.5].**

Step 3: Calculate the parameter values using Linear-Regression method.

$$\beta = \left[\sum_{i=1}^{n} (xi\text{-}meanx)\text{-}(yi\ meany) \right] / \left[\sum_{i=1}^{n} (xi\text{-}meanx)\ \text{^}2 \right]$$

α = meany- βmeanx

y= α+ β.x

Step 4: Calculate the correlation co-efficient value using the Formula

$$r = Sxy / \sqrt{(Sxx \cdot Syy)}$$

$$Sxx = \left[\sum_{i=1}^{n} (xi - meanx) \right] \wedge 2$$

$$Syy = \left[\sum_{i=1}^{n} (yi - meany) \right] \wedge 2$$

$$Sxy = \left[\sum_{i=1}^{n} (xi - meanx)(yi - meany) \right]$$

Step 5: Save and execute the program.

Step 6: Stop the program.

PROGRAM:

```
x=[1 4 3 5]
y=[5 2.75 3 2.5]
m=length(x)
n=length(y)
meanx=mean(x)
meany=mean(y)
for i=1:n
   xx=sum(x(i)-meanx)^2
  yy=sum(y(i)-meany)^2
  a=(sum(x(i)-meanx)*(y(i)-meany))
  b=(sum(x(i)-meanx)^2)
end
beta=a/b
alpha=(mean(y)-beta*mean(x))
disp('alpha'=+string(alpha)+'.beta='+string(beta))
y=(alpha+(beta*x))
r=a/sqrt(xx*yy)
disp('correlation='+string(r))
```

OUTPUT:

x = 1 4 3 5

y = 5.0000 2.7500 3.0000 2.5000

m = 4

n = 4

meanx = 3.2500

meany = 3.3125

xx = 5.0625

yy = 2.8477

a = -3.7969

b = 5.0625

xx = 0.5625

yy = 0.3164

a = -0.4219

b = 0.5625

xx = 0.0625

yy = 0.0977

a = 0.0781

b = 0.0625

xx = 3.0625

yy = 0.6602

a = -1.4219

b = 3.0625

beta = -0.4643

```
alpha =    4.8214

y =    4.3571    2.9643    3.4286    2.5000

r =    -1
```

RESULT:

The above program has been executed successfully and the output was verified.

GRAPHICAL REPRESENTATION OF DATA

EX.NO:9

DATE: 06.09.17

AIM:

The following is the data set X:

X:	Year	A	B
	1996	7	100
	1997	5	150
	1998	7	120
	1999	9	150
	2000	5	130
	2001	7	150

Create 2D Presentations:

a) Show a bar chart for the variable A

b) Show a histogram for the variable B.

c) Show a line chart for the variable B

d) Show a pie chart for the variable A

ALGORITHM:

Step1: Start the program.

Step2: Assign year= [1996 1997 1998 1999 2000 2001]

A= [7 5 7 9 5 7]

B= [100 150 120 150 130 150]

Step3: Using the command subplot to specify the position.

Step4: Use bar() to display bar chart, pie() to display pie chart, line() to Display Line Chart and hist() to display histogram.

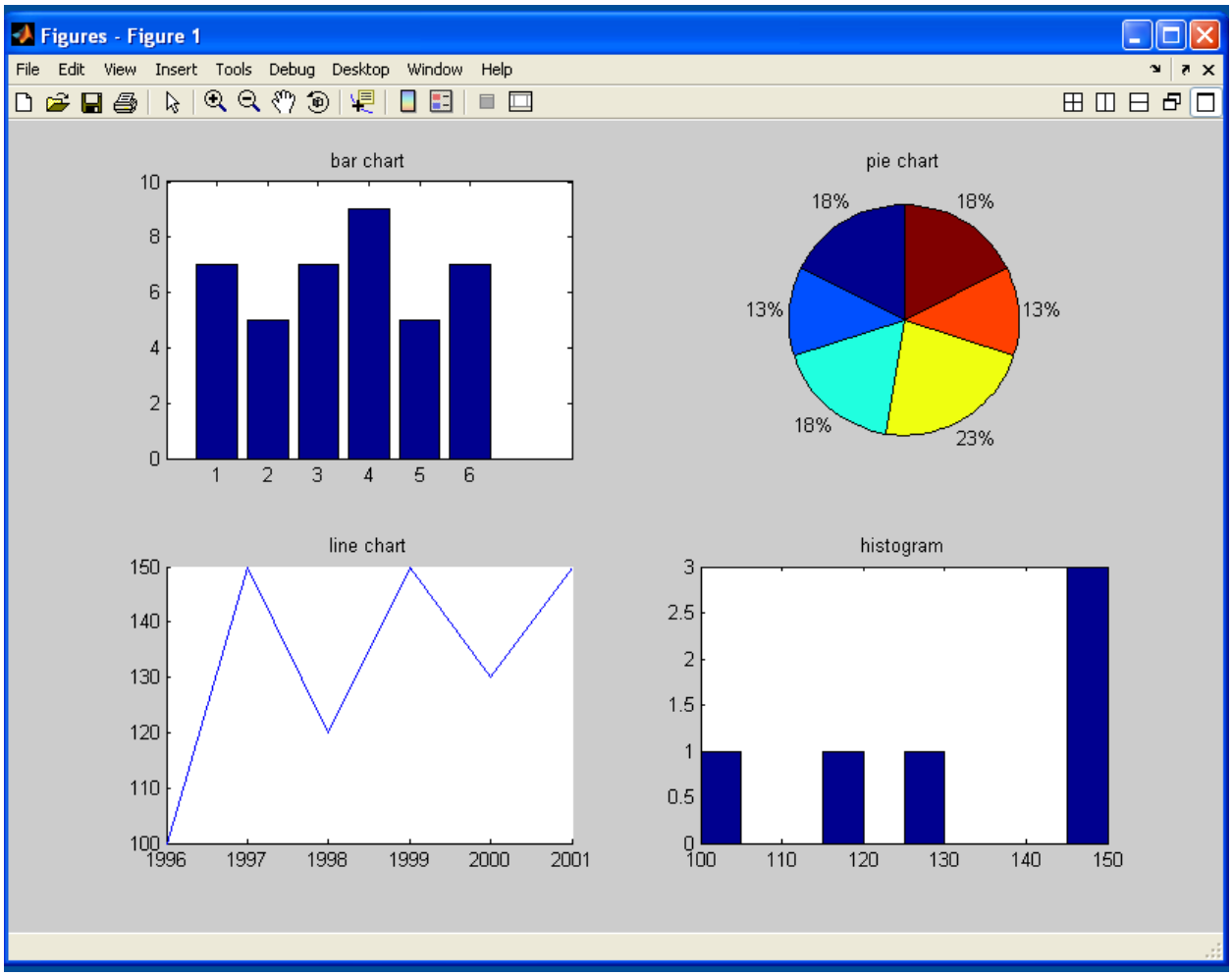
Step5: Save and execute the program.

Step6: Stop the program.

PROGRAM:

```
Year= [1996 1997 1998 1999 2000 2001];
a= [7 5 7 9 5 7];
b= [100 150 120 150 130 150];
subplot(2,2,1);
bar(a);
title('bar chart');
subplot(2,2,2);
pie(a);
title('pie chart');
subplot(2,2,3);
line(year,b);
title('line chart');
subplot(2,2,4);
hist(b);
title('histogram');
```

OUTPUT:



RESULT:

The above program has been executed successfully and the output was verified.

COUNT THE NUMBER OF LINES

EX.NO:10

DATE: 08.09.17

AIM:

To create a MATLAB function to count the number of lines in a text file.

ALGORITHM:

Step1: Start the process.

Step2: Create a file in notepad with some text and save with .txt extension.

Step3: Open the text file to read.

Step4: using while and if statement count the number of lines in a text file.

Step5: Save and run the program.

Step6: Stop the program.

PROGRAM:

```
function[l]=readtext()
fd=mopen('D:/I M.Sc.txt','r')
t=mgete(fd,-1)
[l,r]=size[l]
mclose(ld)
end function
no=readtext()
disp("THE NO.OF LINES IN THE TEXT FILE IS:"+string(no))
```

OUTPUT:

THE NO.OF LINES IN THE TEXT FILE IS:13

RESULT:

The above program has been executed successfully and the output was verified.

STRUCTURED ARRAY FOR STUDENT MARK DETAILS

EX.NO:11

DATE: 15.09.17

AIM:

To create a structure array for student mark details and print a plot for the marks of the students.

ALGORITHM:

Step1: Start the process.

Step2: A simple 1 by 1 structure array student is created with the fields as name, rollno, courses, marks and year.

Step3: Data is assigned to individual fields array.

Step4: To increase the size of the structure array. Define the second structure element of array using index values.

Step5: The structure is further expanded using structure array of size 1 by 5.

Step6: The name of me structure student is typed and the result is obtained.

Step7: The fields can be added and removed in me structure array.

Step8: A bar graph is created for a student marks using me statement.bar (student(2).marks).

Step9: The title,x label and y label are added to me bar graph.

Step10: Stop the process.

PROGRAM:

```
Student(1).name=’sugu’;
Student(1).rollno=67;
Student(1).marks=[87 78 67;68 79 90;45 54 65];
Student
Student(2).name=’harsha’;
Student(2).rollno=23;
Student(2).marks=[98 99 97;95 96 94;93 92 91];
student
Student(3).name=’leens’;
Student(3).rollno=45;
Student(3).marks=78 77 65;45 65 64;63 53 54];
student
Student(4).name=’gomz’;
Student(4).rollno=12;
Student(4).marks=[78 77 86;65 67 69;87 86 85];
student
Student(5).name=’manju’;
Student(5).rollno=54;
Student(5).marks=[67 68 69;75 74 85;73 76 72];
student
disp(‘ADD A FIELD’);
Student(5).branch_name=’computer science’;
Student(5).year=’2014’
```

```
Student(5).semester='1';
Student
disp('REMOVE A FIELD');
student=rmfield(student,'branch_name');
student
bar(student(1).marks)
title('sugu mark details');
xlabel('marks of sugu');ylabel('range of sugu');
```

OUTPUT:

Student=

Name:’sugu’

Rollno:67

Marks:[3x3 double]

Student=

1x2 struct array with fields:

Name

Rollno

Marks

Student=

1x3 struct array with fields:

Name

Rollno

Marks

Student=

1x4 struct array with fields:

Name

Rollno

Marks

Student=

1x5 struct array with fields:

Name

Rollno

Marks

ADD A FIELD

Student=

1x5 struct array with fields:

Name

Rollno

Marks

Branch_name

year

Student=

1x5 struct array with fields:

Name

Rollno

Marks

Branch_name

year

semester

REMOVE A FIELD

Student=

1x5 struct array with fields:

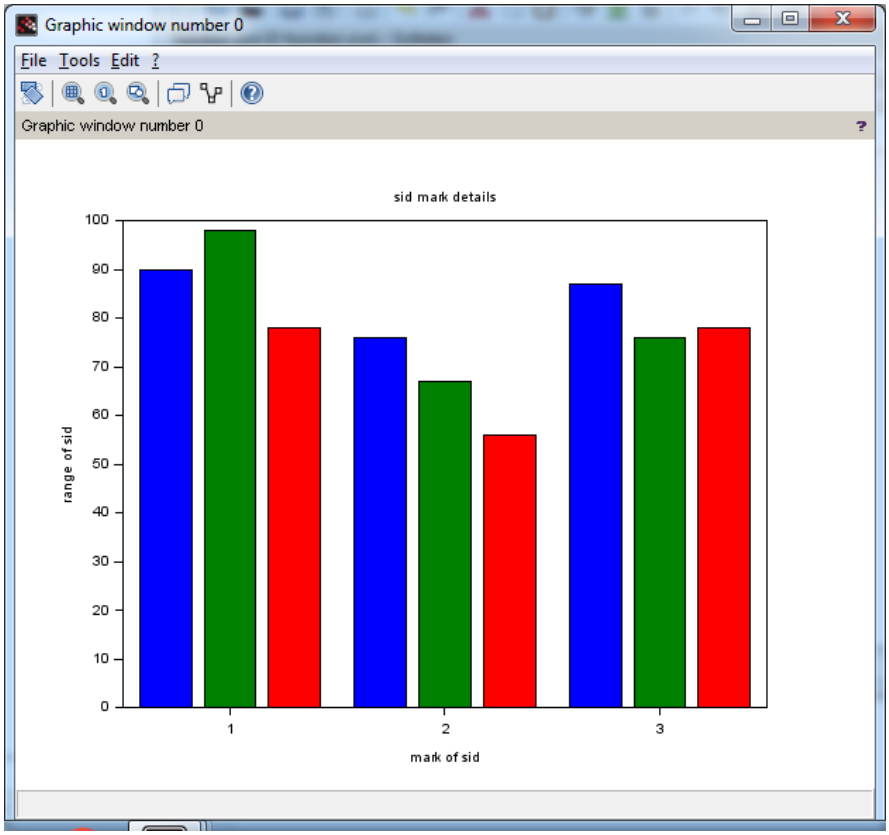
Name

Rollno

Marks

year

semester



RESULT:

The above program has been executed successfully and the output was verified.

MULTIFACTORIAL EVALUATION

EX.NO:12

DATE: 22.09.17

AIM:

The test scores for the three students are given in the following table:

	RDBMS	OracleDBA	WebDesigning	AI
Smith	66	91	95	83
Sam	91	88	80	73
John	80	88	80	78

Find the best student using multifactorial evaluation, if the weight factors for the subjects are given as the vector W= [0.3, 0.2, 0.1, 0.4].

ALGORITHM:

Step1: Start the program.

Step2: To find the length of the students and using for loop find the student

Scores.

Step3: To find the best student by using weight factors W.

Step4: Save and execute the program.

Step5: Stop the program.

PROGRAM:

```
smith=[66 91 95 83]
sam=[91 88 80 73]
john=[80 88 80 78]
w=[0.3,0.2,0.1,0.4]
a=cat(1,smith,sam,john)
n=length(sam)
for i=1:n
s1=smith(1)*0.3+smith(2)*0.2+smith(3)*0.1+smith(4)*0.4
s2=sam(1)*0.3+sam(2)*0.2+sam(3)*0.1+sam(4)*0.4
s3=john(1)*0.3+john(2)*0.2+john(3)*0.1+john(4)*0.4
end
disp('\n smith score %f',s1)
disp ('\n sam score %f',s2)
disp ('\n john score %f',s3)
b=w*a'
c=max(b)
if(c==b(1))
disp ('\n the best student is smith,mark is %f',c);
elseif(c==b(2))
disp ('\n the best student is sam,mark is %f',c);
else
disp ('\n the best student is john,mark is %f',c);
end
```

OUTPUT:

```
smith =    66    91    95    83

sam =    91    88    80    73

john =    80    88    80    78
w =    0.3000    0.2000    0.1000    0.4000
a =
   66    91    95    83
   91    88    80    73
   80    88    80    78
n =     4
s1 =   80.7000
s2 =   82.1000
s3 =   80.8000
 smith score 80.700000
 sam score 82.100000
 john score 80.800000

b =   80.7000   82.1000   80.8000

c =   82.1000

 the best student is sam,mark is 82.100000>>
```

RESULT:

The above program has been executed successfully and the output was verified.