1)
```
import java.io.*;
import java.util.*;

class Records
{
	int index,attr1;
	String attr2;
}

public class prog1
{
	static Records[] rc=new Records[10];
	public static void main(String args[]) throws
	FileNotFoundException,IOException
	{
		BufferedReader CSV=new BufferedReader(new FileReader(new
		File("/home/7c/data.csv")));
		String data=CSV.readLine();
		int i=0,min = Integer.MAX_VALUE,max = Integer.MIN_VALUE;
		System.out.println("Dataset:");
		while(data!=null)
		{
			rc[i]=new Records();
			String[] dataArray=data.split(",");
			rc[i].index=Integer.parseInt(dataArray[0]);
			rc[i].attr1=Integer.parseInt(dataArray[1]);
			rc[i].attr2=dataArray[2];
			if(rc[i].attr1 > max)
				max=rc[i].attr1;
			if(rc[i].attr1 < min)
				min=rc[i].attr1;

			System.out.println(rc[i].index+" "+rc[i].attr1+"
					"+rc[i].attr2);
			data=CSV.readLine();
			i++;
		}

		//finding aggregate for numeric attribute
		int avg =0;
		for(int j=0;j<i;j++)
			avg += rc[j].attr1;
		avg=avg/i;
		System.out.println("max value :"+max+"\tmin value:"+min);
		System.out.println("Average value is: "+avg);

		//performing discretization for numeric attribute
		 int mean = (min + max) / 2;
		 int mid1 = (min + mean) / 2;
		 int mid2 = (mean + max) / 2;
		 int sampling[] = new int[4];
		 for(int j=0;j<i;j++)
		 {
			System.out.print(rc[j].index+" "+rc[j].attr1+" "+rc[j].attr2);
			if(rc[j].attr1 >= min && rc[j].attr1 < mid1)
			{
				System.out.println(" ["+min+"-"+mid1+")");
				sampling[0]=rc[j].attr1;
			}
			else if(rc[j].attr1 >= mid1 && rc[j].attr1 < mean)
			{
				System.out.println(" ["+mid1+"-"+mean+")");
```

```
                        sampling[1]=rc[j].attr1;
                  }
                  else if(rc[j].attr1 >= mean && rc[j].attr1 < mid2)
                  {
                        System.out.println(" ["+mean+"-"+mid2+")");
                        sampling[2]=rc[j].attr1;
                  }
                  else if(rc[j].attr1 >= mid2 && rc[j].attr1 <= max)
                  {
                        System.out.println(" ["+mid2+"-"+max+")");
                        sampling[3]=rc[j].attr1;
                  }
            }
            //sampling result stored in array sampling---will return 0 if none
                  elements are present in the range
            System.out.println("----sampling-------");
            System.out.println(" ["+min+"-"+mid1+") -"+sampling[0]);
            System.out.println(" ["+mid1+"-"+mean+") -"+sampling[1]);
            System.out.println(" ["+mean+"-"+mid2+") -"+sampling[2]);
            System.out.println(" ["+mid2+"-"+max+") -"+sampling[3]);
      }

}
```

1 50 hello
2 30 ajay
3 23 ager
4 76 agre
5 87 wert
6 65 wefg
7 12 fgdb

---

2)

```
import java.io.*;
import java.util.*;

public class prog2
{
      static ArrayList<String[]> data = new ArrayList<>();
      public static void replace(int x,String y)
      {
            for(String[] line:data)
            {
                  if(x < line.length && line[x].isEmpty())
                  {
                        line[x] = y;
                  }
            }
      }
      public static void main(String args[]) throws
                  FileNotFoundException,IOException
      {
            BufferedReader csv=new BufferedReader(new FileReader(new
                        File("/home/7c/data2.csv")));
            String lines;
            Scanner sc = new Scanner(System.in);
            while ((lines = csv.readLine()) !=null)
            {
                        String[] singleLine = lines.split(",",-1);
                        data.add(singleLine);
            }
            while(true)
            {
```

```
		System.out.println("Choice\n1.Avg of columns\n2.Mode of
				Column\n3.Print CSV\n4.Exit");
		int option = sc.nextInt();
		switch(option)
		{
			case 1: System.out.println("enter the column number\n");
				int col=sc.nextInt();
				double avg=0,total=0;
				for (String[] line:data)
					if(col<line.length &&
						! line[col].isEmpty())
					{

					avg=avg+Double.parseDouble(line[col]);
						total++;
					}
				replace(col,Double.toString(avg/total));
				break;
			case 2: System.out.println("enter the column number\n");
				int col1=sc.nextInt();
				HashMap<String,Integer> al = new
							HashMap<>();
				for(String[] line:data)
				{
					if(col1<line.length &&
					 !line[col1].isEmpty())
					{
						if(al.containsKey(line[col1]))
						{
							int temp =
							al.get(line[col1]);
							al.put(line[col1],temp+1);
						}
						else
							al.put(line[col1],1);
					}
				}
				String mode="";
				if(!al.isEmpty()){
					int max=0;
					for(String key: al.keySet()){
						if(al.get(key)>max){
							max=al.get(key);
							mode=key;
						}
					}
				}
				replace(col1,mode);
				break;
			case 3: for (String[] line:data)
				{
					for(String word:line)
					{
						System.out.print(word + "\t\t");
					}
					System.out.println();
				}
				break;
			case 4: System.exit(0);
			default:System.out.println("Wrong choice");
		}
	}
```

```
	}
}
```

hello	1	name	Is	shoaib	hello
hi	2	is	my	computer	
now	3	this	on	the console	
hello	4				hi
hello					hellow

---

3)
```
import java.io.*;
import java.util.*;

class p3
{
	static boolean check(String x1,String x2)
	{
	x2 = x2.replace("", ".*");
	if(x1.matches(x2))
		return true;
	else
		return false;
	}
	public static void main(String[] args) throws
	IOException,FileNotFoundException
	{
		BufferedReader csv = new BufferedReader(new FileReader(new
					File("/home/7c/data3.csv")));
		String data = csv.readLine();
		HashSet<String> hs = new HashSet<>();
		ArrayList<String> al = new ArrayList<>();
		ArrayList<String> bl = new ArrayList<>();
		ArrayList<String> cl = new ArrayList<>();
		double support = 0.4,confidence=0.5;
		while(data != null)
		{
			String dataarray[] = data.split(",");
			String temp1="";
			for(String x:dataarray)
			{
				hs.add(x);
				temp1=temp1+x;
			}
			bl.add(temp1);
			data = csv.readLine();
		}
		String d[] = hs.toArray(new String[hs.size()]);
		int n = d.length;
		// generate all possible subset
		for(int i=0;i < (1<<n); i++)
		{
			String temp="";
			for(int j=0;j<n;j++)
				if(( i & (1<<j))>0)
					temp = temp+d[j];
			al.add(temp);
		}
		// generate frequent itemset
		for(int i=1;i<=4;i++)
		{
			System.out.println("\nFrequent "+i+"-itemset");
```

```
                  for(String y:al)
                        if(i == y.length())
                              {
                                    double count = 0;
                                    for(String x:bl)
                                          if(check(x,y))
                                                count++;
                                    if(count/bl.size() >= support)
                                    {
                                          if(i == 4)
                                                cl.add(y);
                                          System.out.println(y + " ->
                                                "+count/bl.size());
                                    }
                              }
            }
            System.out.println("\n----------Strong rules-----------\n");
            //generate rules
            for(String p:cl)
            {
            System.out.println("\n-----For string "+p+"----------");
                  char[] c = p.toCharArray();
                  n = c.length;
                  for(int i=0;i < (1<<n); i++)  //generate all subset
                  {
                        String temp3="",temp4="";
                        for(int j=0;j<n;j++)
                        {
                              if(( i & (1<<j))>0)
                                    temp3 = temp3+c[j];
                              else
                                    temp4 = temp4+c[j];
                        }
                        if(temp3.length() !=0 && temp3.length() != 4)
                              {
                              double count1=0,count2=0;
                              for(String x:bl)
                                    {
                                          if(check(x,p))
                                                count1++;
                                          if(check(x,temp3))
                                                count2++;
                                    }
                                    if(count2 > 0 && (count1/count2) >=
                                                            confidence)
                                          System.out.println(temp3+"->"+temp4+"
                                                confidence: "+count1/count2);
}}}}}
```

```
a b c
a b c d e
a c d
a c d e
a b c d
```

---

4)

```
import java.io.*;
import java.util.*;

class record
```

```
{
	int attr[] = new int[7];
}

class p4
{
	static record[] rc = new record[2];
	public static void main(String args[]) throws FileNotFoundException,
IOException
	{
		BufferedReader csv = new BufferedReader(new FileReader(new
File("/home/7c/data4.csv")));
		String data = csv.readLine();
		int i=0;
		while(data != null)
		{
			rc[i] = new record();
			String[] dataarray = data.split(",");
			for(int j=0;j<7;j++)
				rc[i].attr[j] = Integer.parseInt(dataarray[j]);
			data = csv.readLine();
			i++;
		}
		int totalOfCol[] = new int[7];
		double giniOfCol[] = new double[7];
		double entropyOfCol[] = new double[7];
		double gini[] = new double[3];
		double entropy[] = new double[3];
		double gain[] = new double[3];

		for(i=0;i<rc.length;i++)
			for(int j=0;j<totalOfCol.length;j++)
				totalOfCol[j] += rc[i].attr[j];

		for(i=0;i<rc.length;i++)
			for(int j=0;j<totalOfCol.length;j++)
			{
				double temp = rc[i].attr[j]/(totalOfCol[j] * 1.0);
				giniOfCol[j] += Math.pow(temp,2);
				entropyOfCol[j] += temp * Math.log(temp) / Math.log(2);
			}

			for(int j=0;j<totalOfCol.length;j++)
			{
				giniOfCol[j] = 1 - giniOfCol[j];
				entropyOfCol[j] = -1 * entropyOfCol[j];
			}

		gini[0] = ( giniOfCol[0]*totalOfCol[0] +
			giniOfCol[1]*totalOfCol[1] )/(totalOfCol[0]+totalOfCol[1]);
		gini[1] = ( giniOfCol[2]*totalOfCol[2] +
			giniOfCol[3]*totalOfCol[3] )/(totalOfCol[2]+totalOfCol[3]);
		gini[2] = ( giniOfCol[4]*totalOfCol[4] + giniOfCol[5]*totalOfCol[5]
				+ giniOfCol[6]*totalOfCol[6])/
			       (totalOfCol[4]+totalOfCol[5]+totalOfCol[6]);

		entropy[0] = ( entropyOfCol[0]*totalOfCol[0] +
			entropyOfCol[1]*totalOfCol[1] )/(totalOfCol[0]+totalOfCol[1]);
		entropy[1] = ( entropyOfCol[2]*totalOfCol[2] +
			entropyOfCol[3]*totalOfCol[3] )/(totalOfCol[2]+totalOfCol[3]);
		entropy[2] = ( entropyOfCol[4]*totalOfCol[4] +
			entropyOfCol[5]*totalOfCol[5] +
```

```
                  entropyOfCol[6]*totalOfCol[6])/
                  (totalOfCol[4]+totalOfCol[5]+totalOfCol[6]);

            double parententropy = 1.0;
            double parentgini = 0.5;
            for(i=0;i<gini.length;i++)
                  gain[i] = parententropy - entropy[i];

            System.out.println("entrpoy="+entropy[0]);
            System.out.println("gini="+gini[0]);
            System.out.println("gain="+gain[0]);

            System.out.println("entrpoy="+entropy[1]);
            System.out.println("gini="+gini[1]);
            System.out.println("gain="+gain[1]);

            System.out.println("entrpoy="+entropy[2]);
            System.out.println("gini="+gini[2]);
            System.out.println("gain="+gain[2]);
      }
}

 7  3  8  2  8  1  1  0
 2  8  7  3  1  2  7  1
```

R1)
```
mydata <- read.csv("/home/7c/data.csv",header = TRUE,sep=",")
print(mydata)
write.csv(mydata,"/home/7c/data_1.csv",quote = T,append = F,row.names =
T,col.names = T)
```

R2)
```
library(datasets)

hist(iris$Sepal.Length[1:5])
gra<-density(iris$Sepal.Length[1:5])
plot(gra)
pie(table(iris$Species))
```

R3)
```
library(arules)
library(arulesViz)
patterns = random.patterns(nItems = 1000)
trans = random.transactions(nItems = 1000, nTrans = 1000, method = "agrawal",
patterns = patterns)
rules = apriori(trans, parameter=list(support=0.01, confidence=0.5))
inspect(rules)
plot(rules, method="grouped")
```
