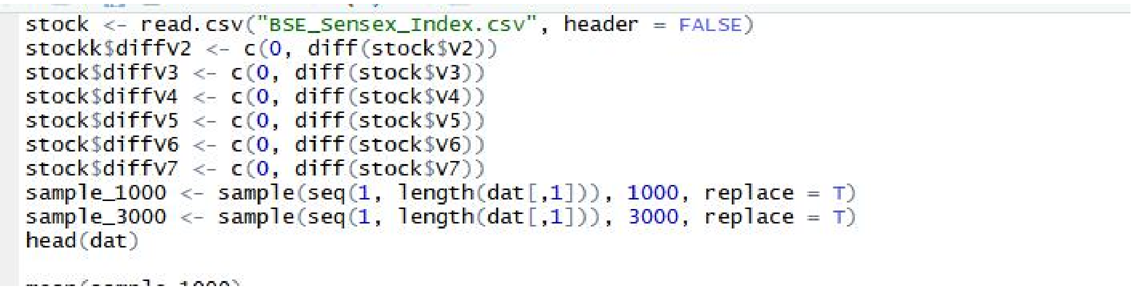
Data Mining- Lab Exam

Time: 24 hours
Marks:100

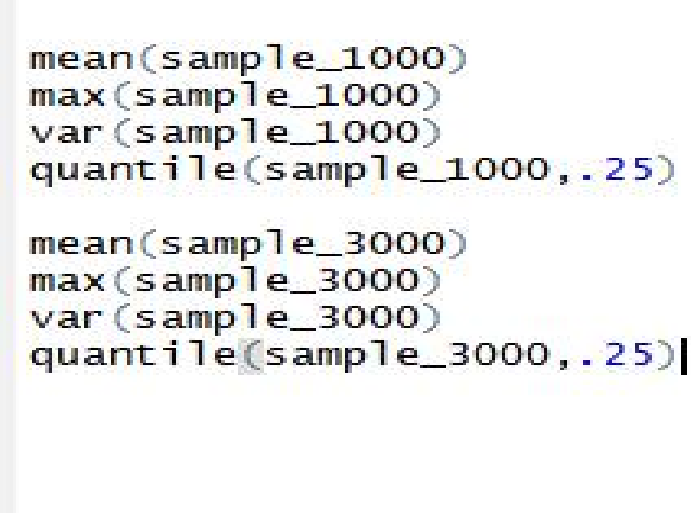
Open a document and update document with your answers for each question and submit it.

1. a) For the dataset BSE_Sensex_Index.csv, create an extra column of successive differences for each column of numeric values in this data file. Extract two simple random samples with replacement of 1000 and 3000 observations (rows). Show your R commands for doing this.
   Do the same thing by using Excel. Show your Excel commands.
   Note: Successive difference for date d1= (date d1 value-immediate available previous date of d1 value)/immediate available previous date of d1. For the last row fill up values with mean of its immediate three previous row values.

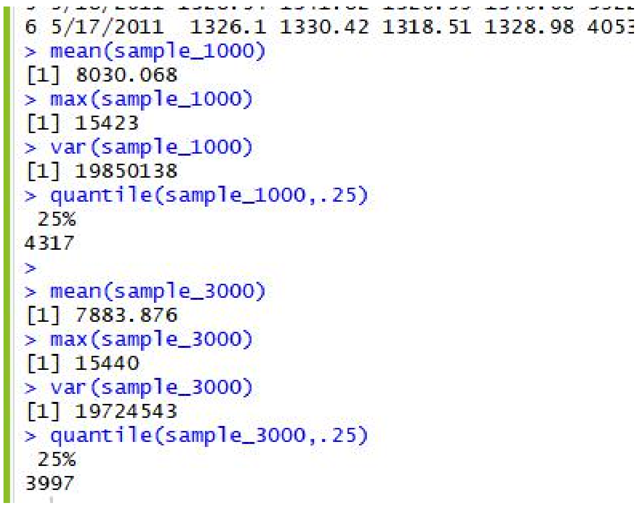


```
stock <- read.csv("BSE_Sensex_Index.csv", header = FALSE)
stockk$diffv2 <- c(0, diff(stock$V2))
stock$diffv3 <- c(0, diff(stock$V3))
stock$diffv4 <- c(0, diff(stock$V4))
stock$diffv5 <- c(0, diff(stock$V5))
stock$diffv6 <- c(0, diff(stock$V6))
stock$diffv7 <- c(0, diff(stock$V7))
sample_1000 <- sample(seq(1, length(dat[,1])), 1000, replace = T)
sample_3000 <- sample(seq(1, length(dat[,1])), 3000, replace = T)
head(dat)
```

   b) For your samples, use the functions mean(), max(), var() and quartile(,.25) to compute the mean, maximum, variance and 1st quartile respectively for each column which has successive differences. Show your R code and the resulting values.
   Do the same thing by using Excel. Show your Excel commands.



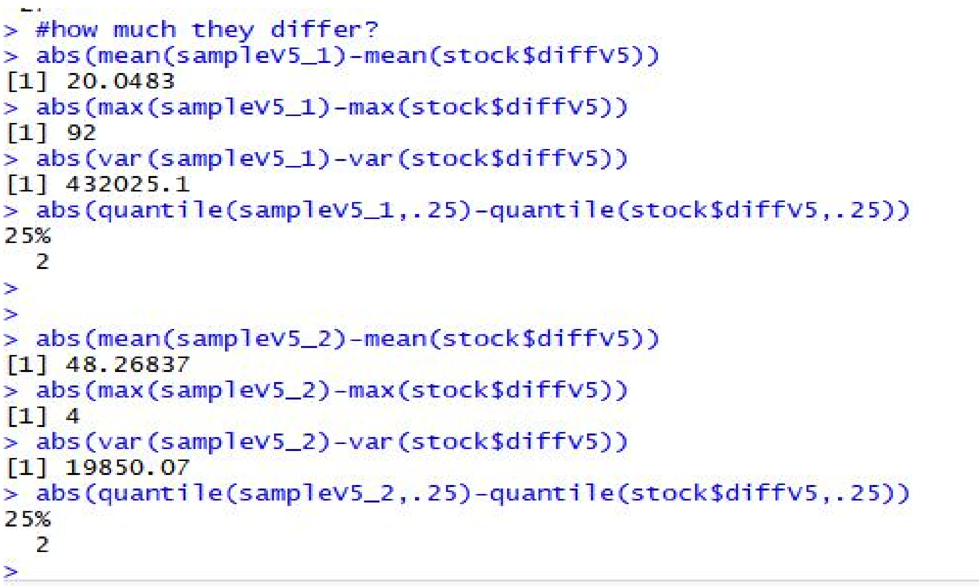
```
mean(sample_1000)
max(sample_1000)
var(sample_1000)
quantile(sample_1000,.25)

mean(sample_3000)
max(sample_3000)
var(sample_3000)
quantile(sample_3000,.25)
```



```
6 5/17/2011  1326.1 1330.42 1318.51 1328.98 4053
> mean(sample_1000)
[1] 8030.068
> max(sample_1000)
[1] 15423
> var(sample_1000)
[1] 19850138
> quantile(sample_1000,.25)
 25%
4317
>
> mean(sample_3000)
[1] 7883.876
> max(sample_3000)
[1] 15440
> var(sample_3000)
[1] 19724543
> quantile(sample_3000,.25)
 25%
3997
```

c) Compute the same quantities in part b on the entire data set and show your answers. How much do they differ from your answers in part b? Do you find any significant difference between two sample values like mean in comparison with entire data? If so what explanation you can give for that?

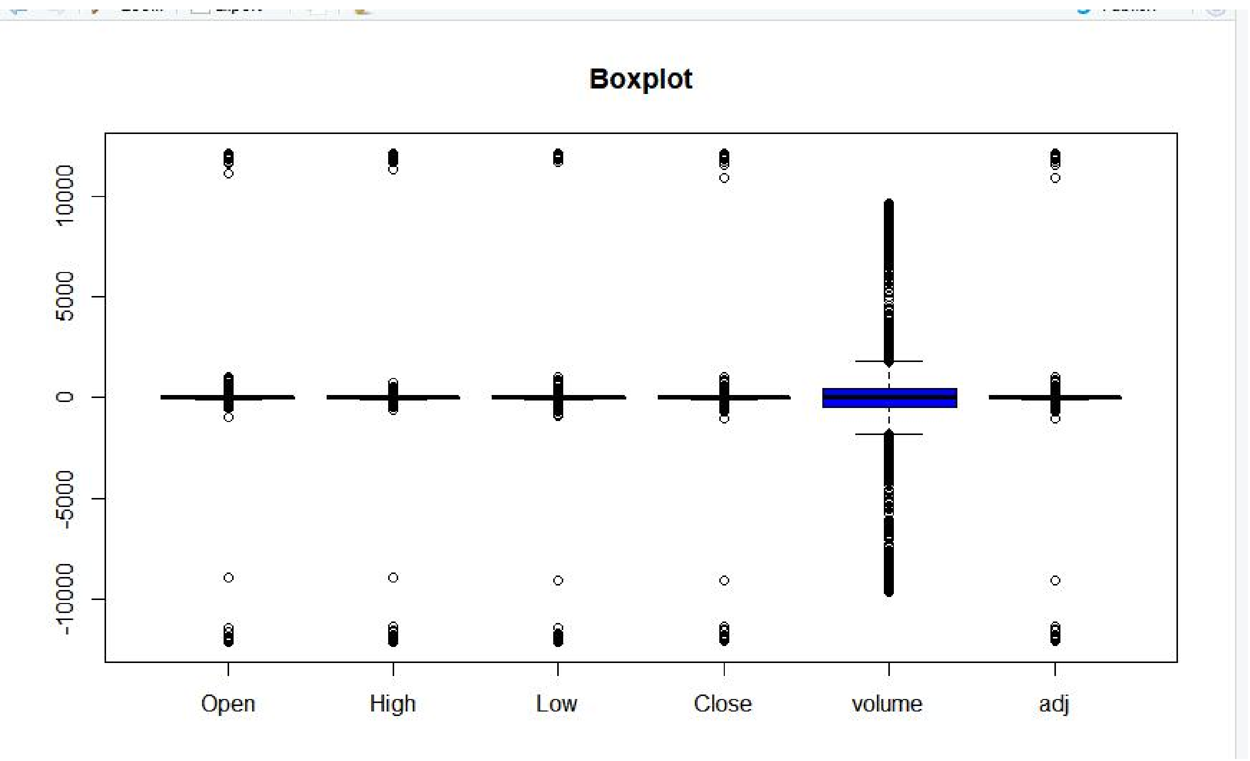
Do the same thing by using Excel. Show your Excel commands.



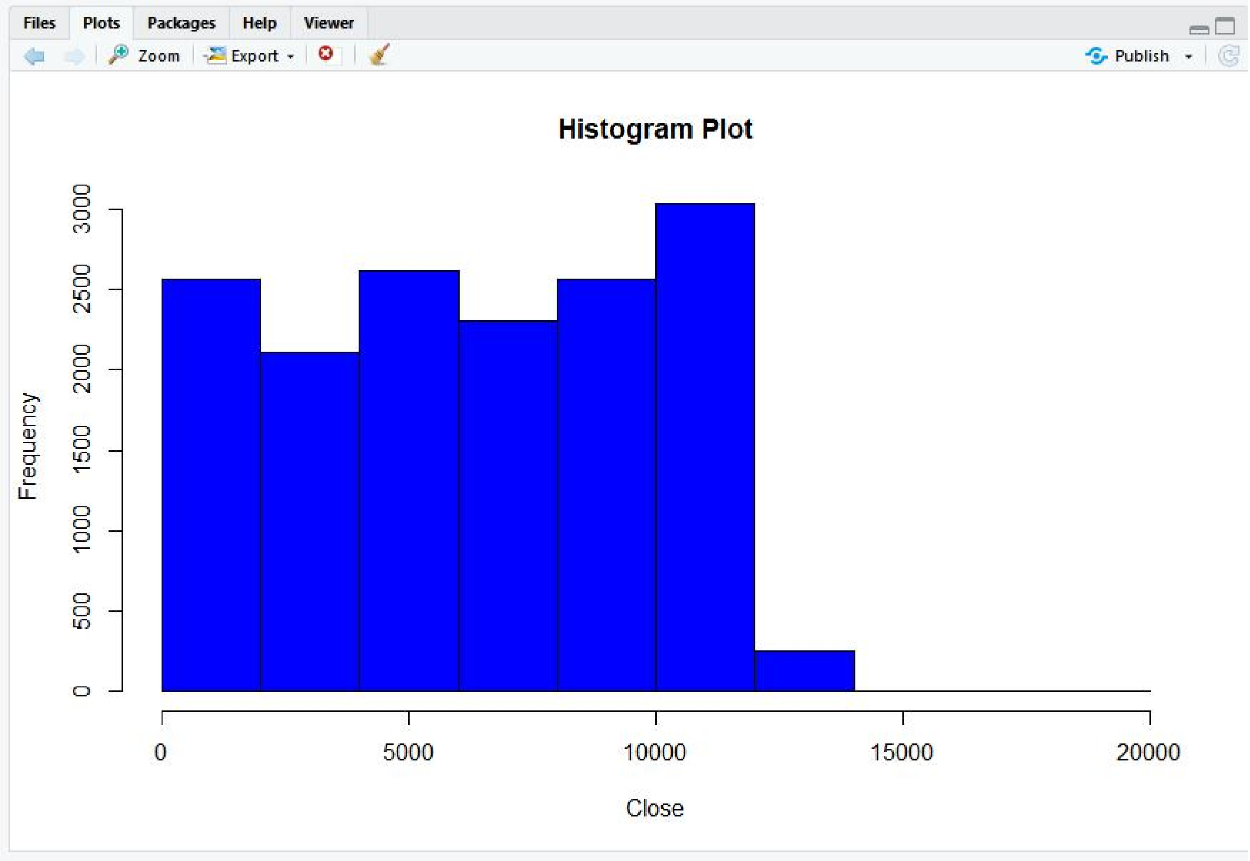
```
> #how much they differ?
> abs(mean(samplev5_1)-mean(stock$diffv5))
[1] 20.0483
> abs(max(samplev5_1)-max(stock$diffv5))
[1] 92
> abs(var(samplev5_1)-var(stock$diffv5))
[1] 432025.1
> abs(quantile(samplev5_1,.25)-quantile(stock$diffv5,.25))
25%
  2
>
>
> abs(mean(samplev5_2)-mean(stock$diffv5))
[1] 48.26837
> abs(max(samplev5_2)-max(stock$diffv5))
[1] 4
> abs(var(samplev5_2)-var(stock$diffv5))
[1] 19850.07
> abs(quantile(samplev5_2,.25)-quantile(stock$diffv5,.25))
25%
  2
>
```

d) Use R to produce a single graph displaying a boxplot for open, close, high and low. Include the R commands and the plot.

Do the same thing by using Excel. Show your Excel commands



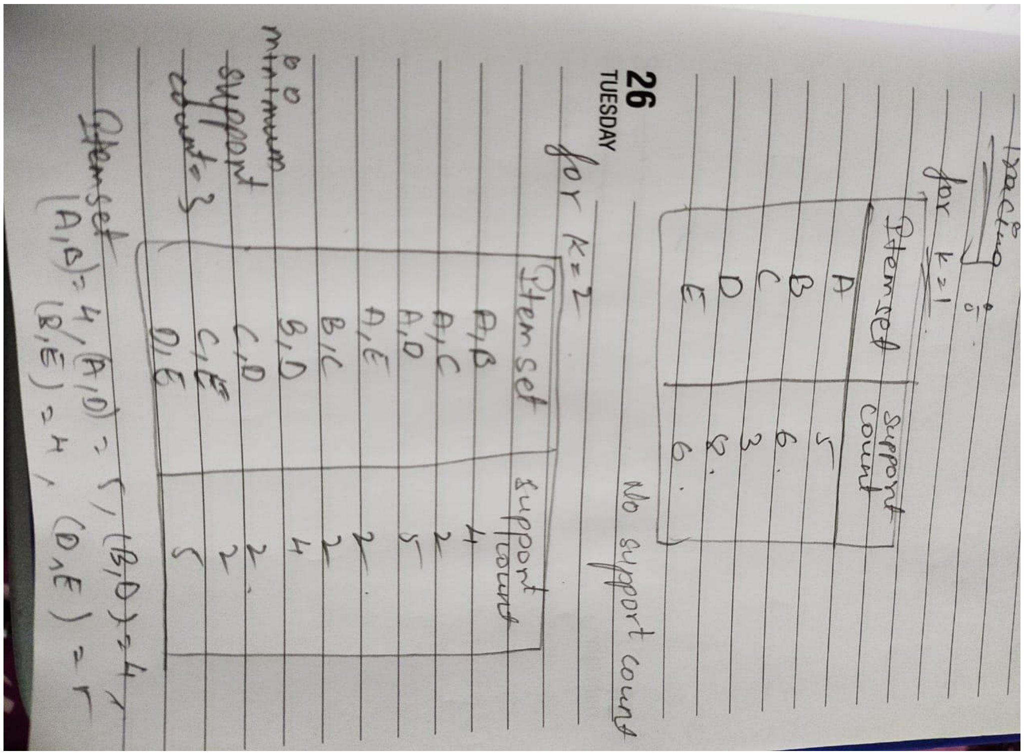
e) Use R to produce a frequency histogram for Close values. Use intervals of width 2000 beginning at 0. Include the R commands and the plot.
Do the same thing by using Excel. Show your Excel commands.   (10+10=20M)



2. Implement Apriori Algorithm or use built in packages to find out the frequent itemsets and generate rules for frequent itemsets. Trace and submit the program output for the following given dataset of transactions with a minimum support of 3. (10M)

```
TID, Items
101, A,B,C,D,E
102, A,C,D
103, D,E
104, B,C,E
105, A,B,D,E
106, A,B
107, B,D,E
108, A,B,D
109, A,D
110, D,E
```

Tracing:



Tracing :

for k=1

Itemset	support count
A	5
B	6.
C	3
D	8.
E	6.

No support count

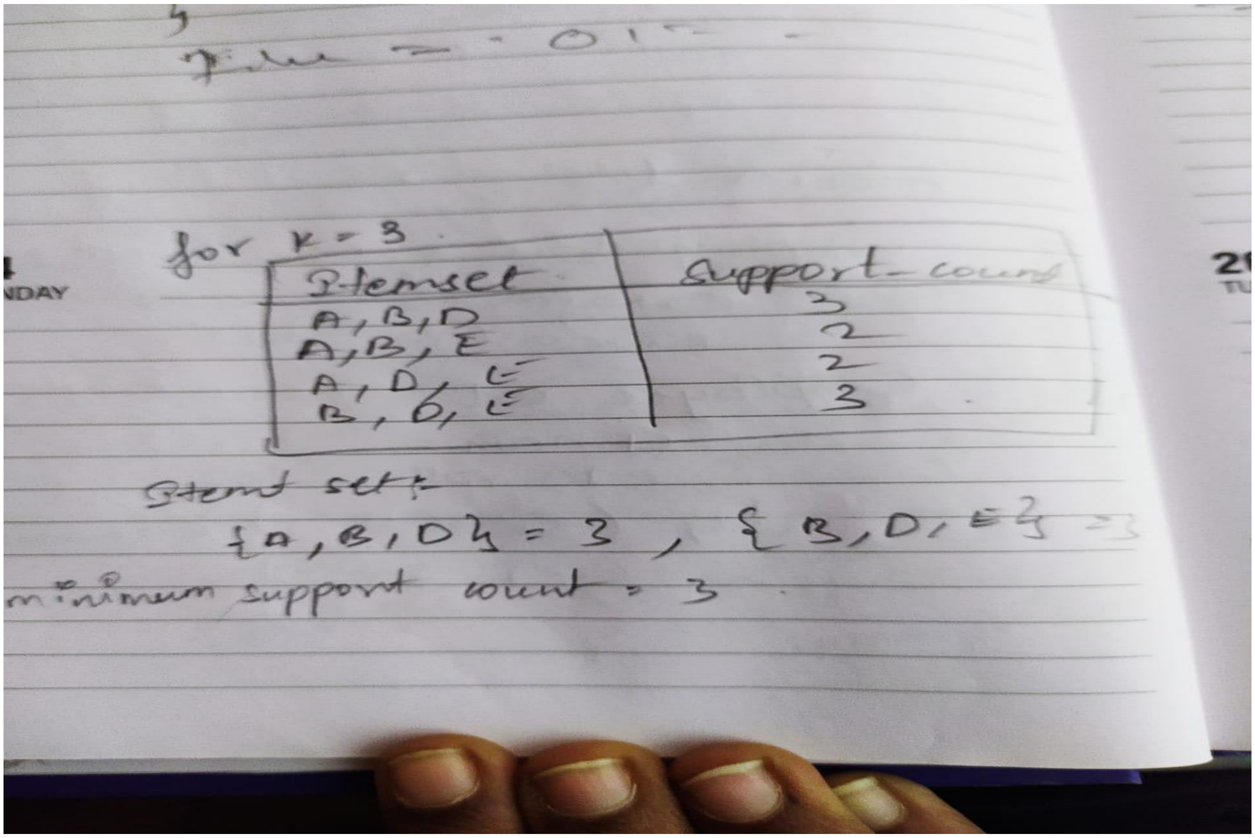
26
TUESDAY

for k=2

Itemset	support count
A,B	4
A,C	2
A,D	5
A,E	2
B,C	2
B,D	4
C,D	2.
C,E	2
D,E	5

minimum support count=3

Itemset
(A,B)=4, (A,D) = 5, (B,D)=4,
(B,E) = 4, (D,E) = 5



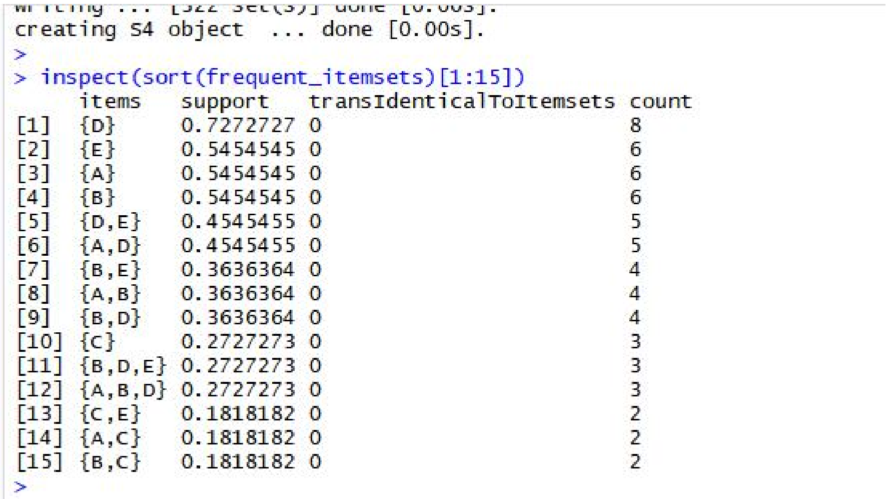
[illegible] = .01

for k = 3.

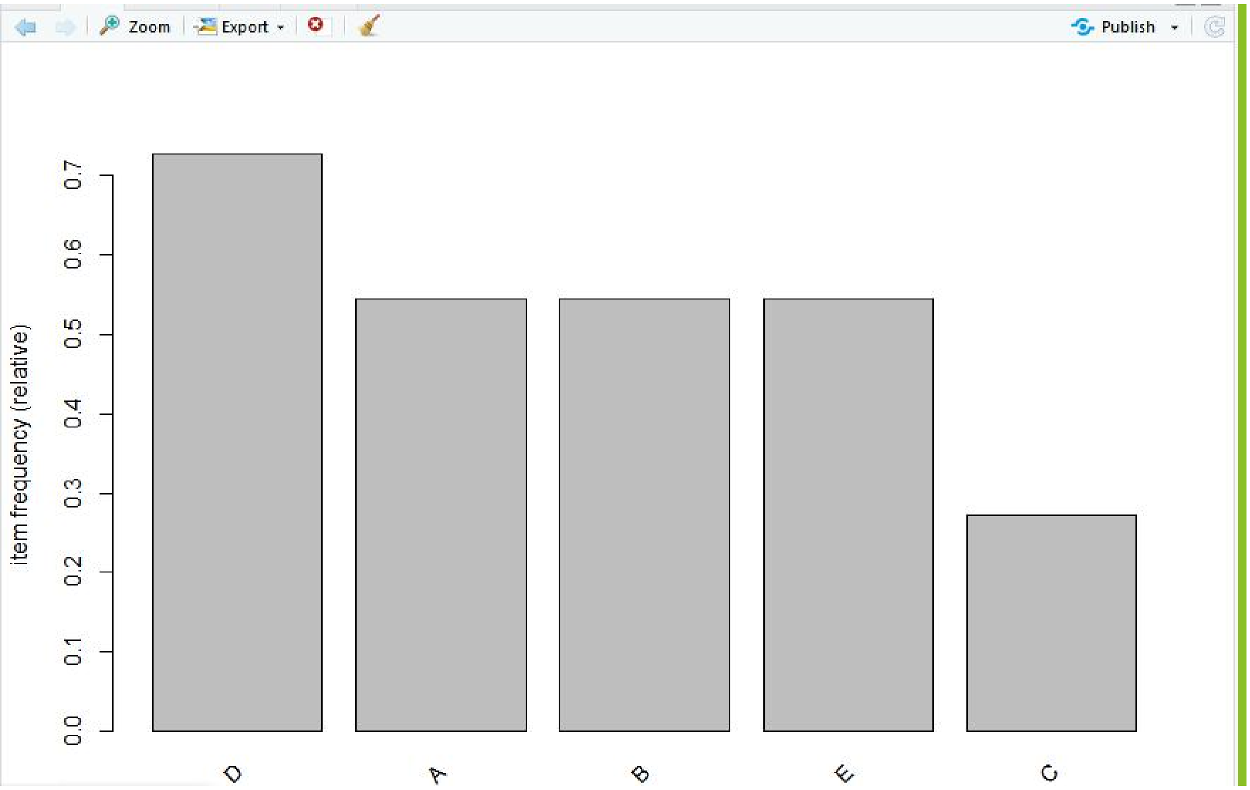
Itemset	support-count
A,B,D	3
A,B,E	2
A,D,E	2
B,D,E	3

Itemset:-
{A,B,D} = 3 , {B,D,E} = 3

minimum support count = 3

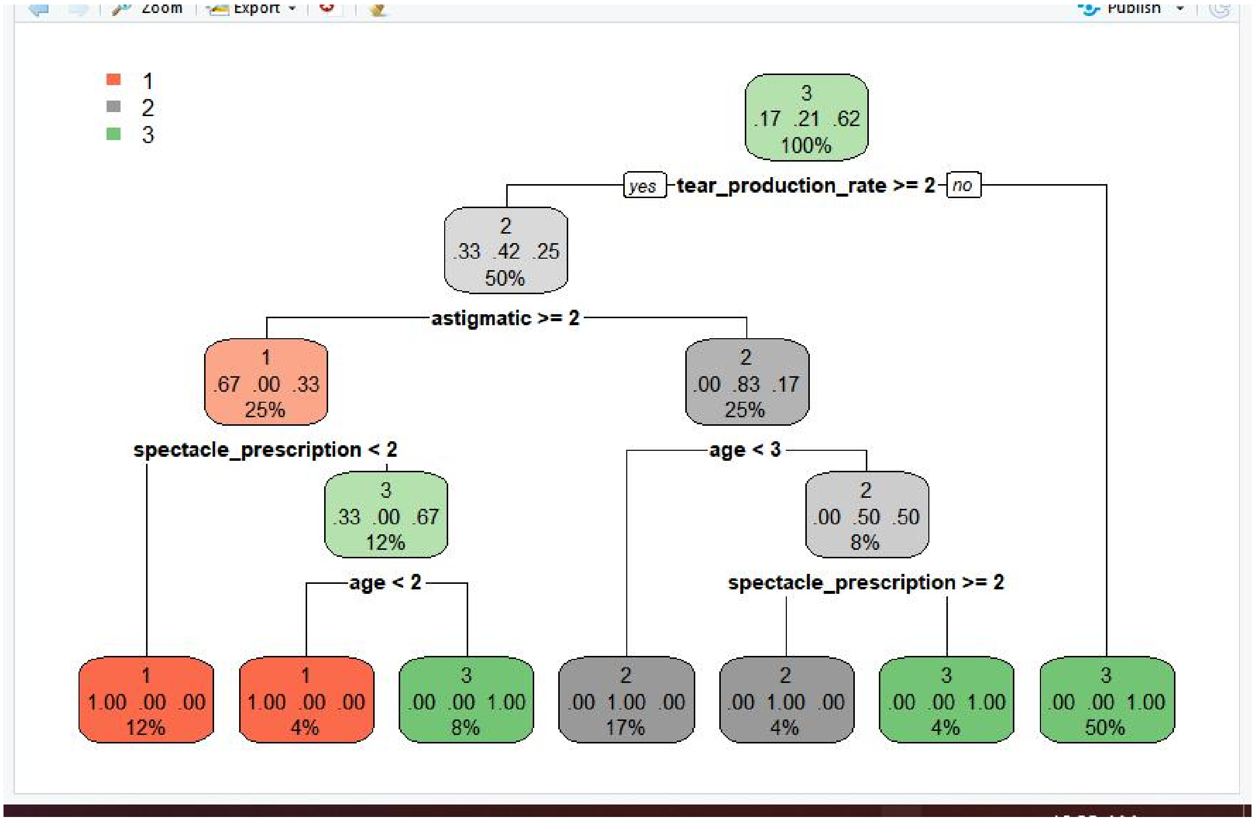


```
writing ... [322 set(s)] done [0.00s].
creating S4 object  ... done [0.00s].
>
> inspect(sort(frequent_itemsets)[1:15])
     items   support   transIdenticalToItemsets count
[1]  {D}     0.7272727 0                        8
[2]  {E}     0.5454545 0                        6
[3]  {A}     0.5454545 0                        6
[4]  {B}     0.5454545 0                        6
[5]  {D,E}   0.4545455 0                        5
[6]  {A,D}   0.4545455 0                        5
[7]  {B,E}   0.3636364 0                        4
[8]  {A,B}   0.3636364 0                        4
[9]  {B,D}   0.3636364 0                        4
[10] {C}     0.2727273 0                        3
[11] {B,D,E} 0.2727273 0                        3
[12] {A,B,D} 0.2727273 0                        3
[13] {C,E}   0.1818182 0                        2
[14] {A,C}   0.1818182 0                        2
[15] {B,C}   0.1818182 0                        2
>
```

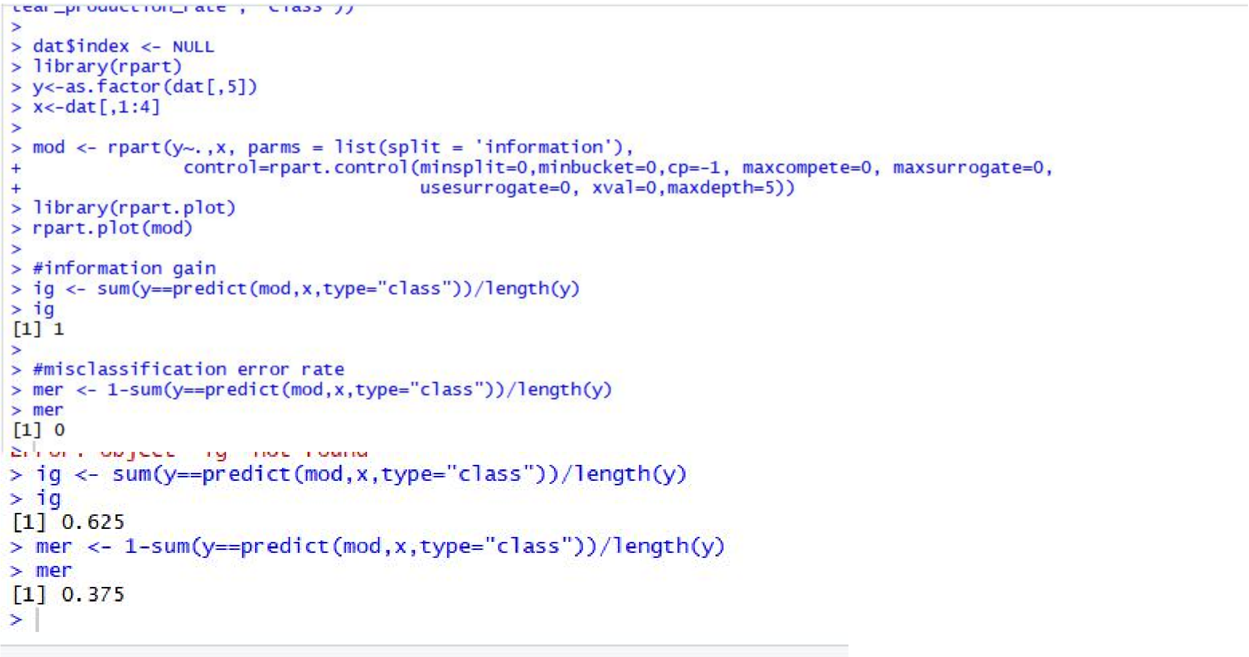


3. Build Decision Trees by using i) information gain and ii) misclassification error rate for Lenses Data Set provided at http://archive.ics.uci.edu/ml/datasets/Lenses. In terms of tree size what do you conclude comparing these two? (10M)

Decision Tree



Information gain and missclassification error rate

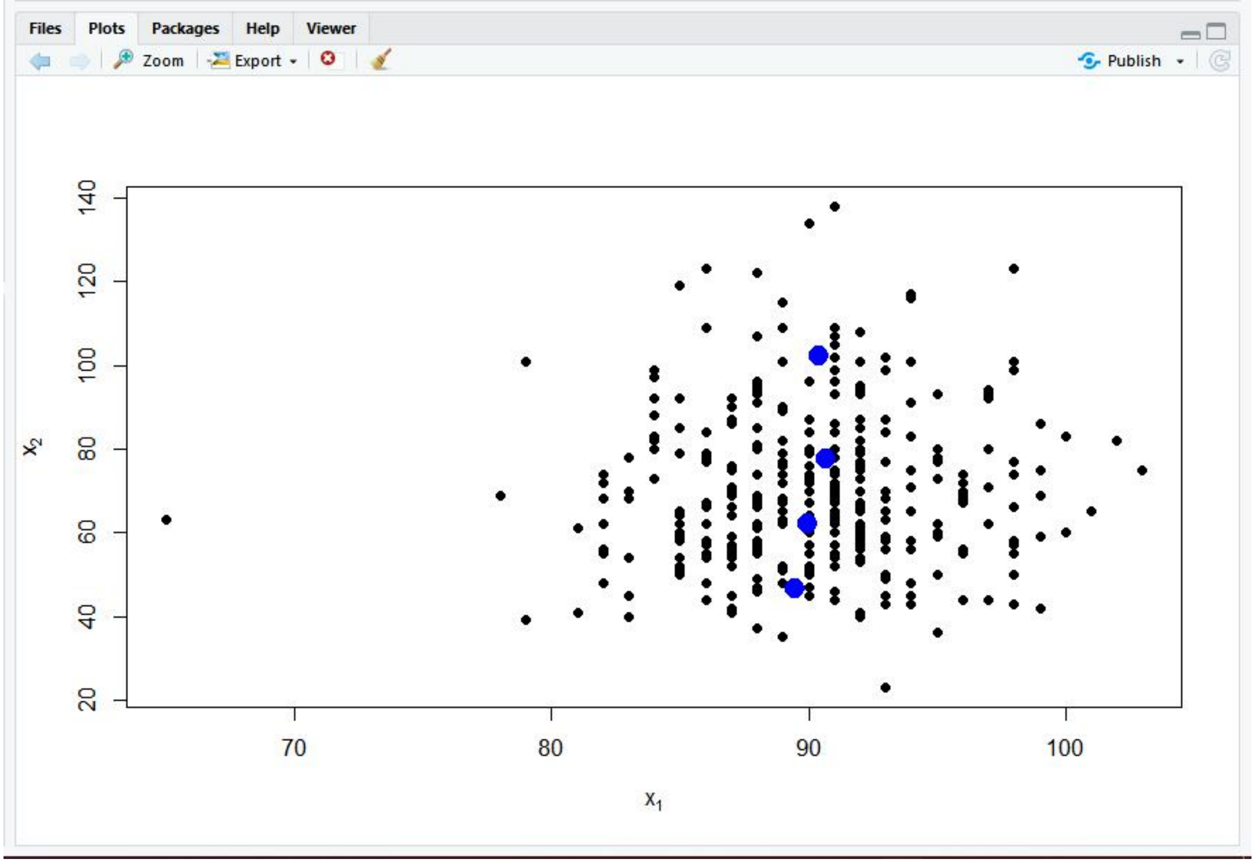


```
tear_production_rate", "class"))
>
> dat$index <- NULL
> library(rpart)
> y<-as.factor(dat[,5])
> x<-dat[,1:4]
>
> mod <- rpart(y~.,x, parms = list(split = 'information'),
+              control=rpart.control(minsplit=0,minbucket=0,cp=-1, maxcompete=0, maxsurrogate=0,
+                                    usesurrogate=0, xval=0,maxdepth=5))
> library(rpart.plot)
> rpart.plot(mod)
>
> #information gain
> ig <- sum(y==predict(mod,x,type="class"))/length(y)
> ig
[1] 1
>
> #misclassification error rate
> mer <- 1-sum(y==predict(mod,x,type="class"))/length(y)
> mer
[1] 0
Error: object 'ig' not found
> ig <- sum(y==predict(mod,x,type="class"))/length(y)
> ig
[1] 0.625
> mer <- 1-sum(y==predict(mod,x,type="class"))/length(y)
> mer
[1] 0.375
>
```

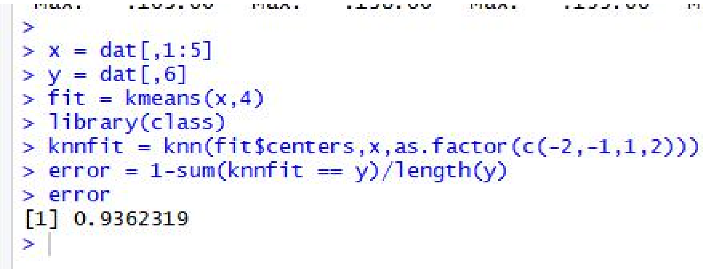
4. Fit 1, 2 and 3-nearest-neighbor classifiers to the Liver Disorders Data Set at http://archive.ics.uci.edu/ml/datasets/Liver+Disorders for measures Euclidean and cosine. Last but one column is a decision attribute. Replace decision values in to 4 classes (0<=c1<5, 5<=c2<10, 10<=c3<15, 15<=c4<=20). Last column is a data split column in to training and test sets. 1 means the object is used for training. 2 means the object is used for testing. Explain the input parameters you provided for the classifier. Compute the misclassification error on the training data and also on the test data. Annotate your program. (10M)

5. Use Support Vector machine for above problem. And compare the performance of both. Explain the input parameters you provided for the classifier. (10M)

6. Create k-means clusters for k=4 for the Liver Disorders Data Set at http://archive.ics.uci.edu/ml/datasets/Liver+Disorders . Explain the input parameters you provided for the clustering algorithm. Plot the fitted cluster centers using a different color. Finally assign the cluster membership for the points to the nearest cluster center. Color the points according to their cluster membership. (10+10=20M)

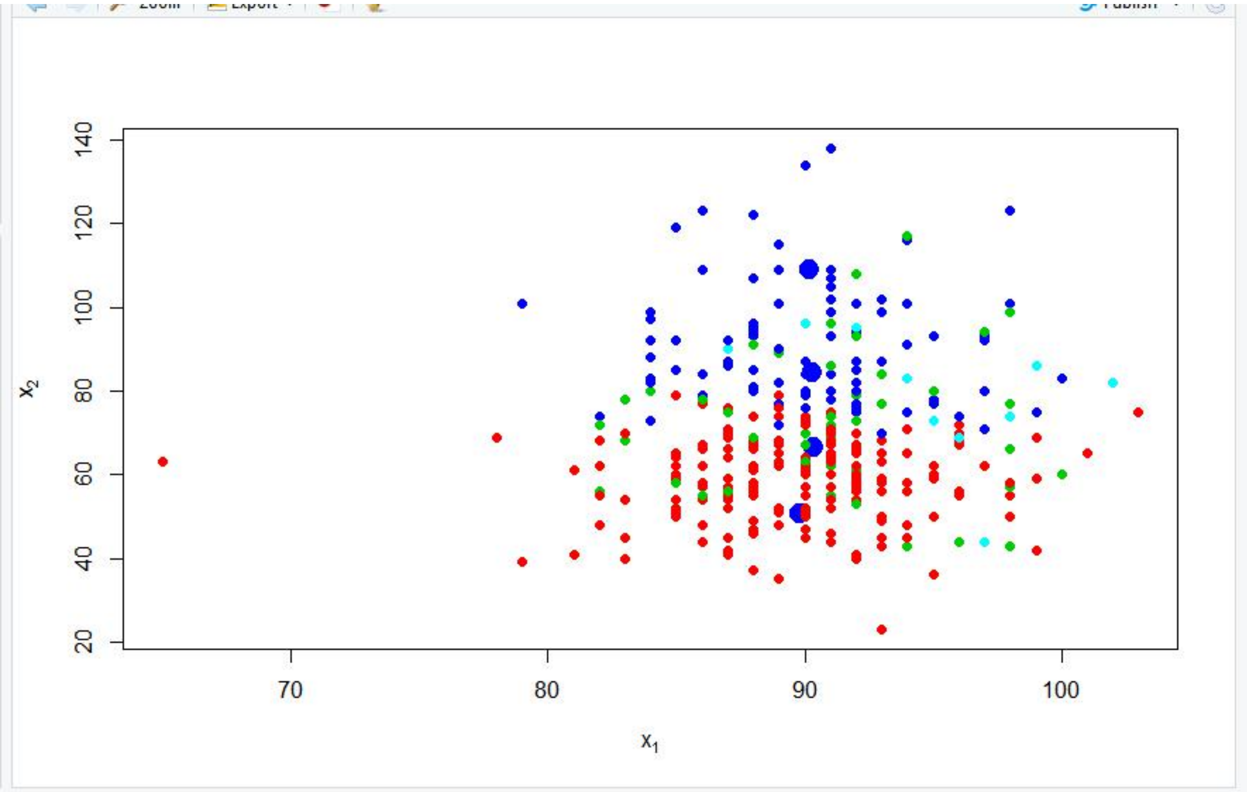


7. Compute the misclassification error that would result if you used your clustering rule to classify the data by assigning the majority class of the cluster. (10M)



```
>
> x = dat[,1:5]
> y = dat[,6]
> fit = kmeans(x,4)
> library(class)
> knnfit = knn(fit$centers,x,as.factor(c(-2,-1,1,2)))
> error = 1-sum(knnfit == y)/length(y)
> error
[1] 0.9362319
> 
```

Misclassification error = 0.936231



8. Consider the dataset BSE_Sensex_Index.csv. Create an extra column of successive growth rate for column close where the successive growth rate is defined as (value of day x- value of day x-1)/value of day x-1. Use a z score cut off of 3 to identify any outliers. List the respective dates from the csv file on which day these outliers fall. (10M)