

Data Mining- Lab Exam

Time: 24 hours

Marks:100

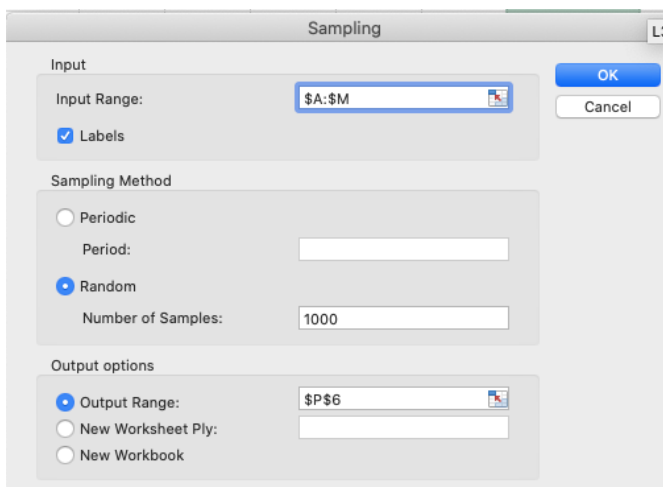
Open a document and update document with your answers for each question and submit it.

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.

```
> Open_diff <- (bse_datasets$Open[seq(1,nrow(bse_datasets)-1)] - bse_datasets$Open[seq(2,nrow(bse_datasets))]) /  
bse_datasets$Open[seq(2,nrow(bse_datasets))])  
> avg <- mean(tail(Open_diff[1:length(Open_diff)-1],3))  
> Open_diff <- append(Open_diff, avg)  
> bse_datasets <- cbind(bse_datasets, Open_diff=Open_diff)  
.  
.  
> bse_datasets[1:8,]  
  Date      Open      High      Low      Close      Volume Adj.Close      Open_diff      High_diff      Low_diff      Close_diff      Volume_diff  
1 5/23/2011 1333.07 1333.07 1312.88 1317.37 3255580000 1317.37 -0.0066542474 -0.0066542474 -0.013369205 -0.0119255665 -0.19932022  
2 5/20/2011 1342.00 1342.00 1330.67 1333.27 4066020000 1333.27 -0.0002979738 -0.0035788004 -0.004257835 -0.0076883001 0.12131733  
3 5/19/2011 1342.40 1346.82 1336.36 1343.60 3626110000 1343.60 0.0104325049 0.0037262822 0.007364747 0.0021779992 -0.07545072  
4 5/18/2011 1328.54 1341.82 1326.59 1340.68 3922030000 1340.68 0.0018399819 0.0085687227 0.006128129 0.0088037442 -0.03254587  
5 5/17/2011 1326.10 1330.42 1318.51 1328.98 4053970000 1328.98 -0.0064955011 -0.0096104457 -0.006637435 -0.0003685679 0.05400585  
6 5/16/2011 1334.77 1343.33 1327.32 1329.47 3846250000 1329.47 -0.0103211264 -0.0052870482 -0.004529909 -0.0062043550 0.12244868  
7 5/13/2011 1348.69 1350.47 1333.36 1337.77 3426660000 1337.77 0.0069434593 -0.0004292957 0.000998476 -0.0080673266 -0.09280660  
8 5/12/2011 1339.39 1351.05 1332.03 1348.65 3777210000 1348.65 -0.0111627083 -0.0025544293 -0.003240145 0.0048953863 -0.01794995  
Adj.Close_diff  
1 -0.0119255665  
2 -0.0076883001  
3 0.0021779992  
4 0.0088037442  
5 -0.0003685679  
6 -0.0062043550  
7 -0.0080673266  
8 0.0048953863  
.  
.  
> sample_1000 <- bse_datasets[sample(nrow(bse_datasets), 1000),]  
> sample_3000 <- bse_datasets[sample(nrow(bse_datasets), 3000),]  
.
```



The image shows the 'Sampling' dialog box in Microsoft Excel. The 'Input Range' is set to '\$A:\$M'. The 'Labels' checkbox is checked. Under 'Sampling Method', the 'Random' radio button is selected, and the 'Number of Samples' is set to 1000. Under 'Output options', the 'Output Range' radio button is selected, and the range '\$P\$6' is entered. The 'OK' button is highlighted in blue.

b) For your samples, use the functions `mean()`, `max()`, `var()` and `quantile(.,.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.

```
> samp <- sample(bse_datasets$Open_diff, 1000, replace=TRUE)
> mean(samp)
[1] -9.719929e-05
> max(samp)
[1] 0.04202953
> var(samp)
[1] 8.428637e-05
> quantile(samp, .25)
      25%
-0.004561395

~
> samp <- sample(bse_datasets$High_diff, 1000, replace=TRUE)
> mean(samp)
[1] 0.0004403912
> max(samp)
[1] 0.0488628
> var(samp)
[1] 6.93137e-05
> quantile(samp, .25)
      25%
-0.003914968

~
> samp <- sample(bse_datasets$Low_diff, 1000, replace=TRUE)
> max(samp)
[1] 0.07856698
> mean(samp)
[1] 0.000603228
> var(samp)
[1] 6.681119e-05
> quantile(samp, .25)
      25%
-0.003729326
```

```

> samp <- sample(bse_datasets$Close_diff, 1000, replace=TRUE)
> var(samp)
[1] 9.68133e-05
> quantile(samp,.25)
      25%
-0.0042079
> mean(samp)
[1] 8.4484e-05
> max(samp)
[1] 0.09099354

> samp <- sample(bse_datasets$Volume_diff, 1000, replace=TRUE)
> max(samp)
[1] 26.51968
> mean(samp)
[1] 0.0506174
> quantile(samp,.25)
      25%
-0.08800347
> var(samp)
[1] 0.7435702

-
> samp <- sample(bse_datasets$Adj.Close_diff, 1000, replace=TRUE)
> var(samp)
[1] 0.0001010096
> quantile(samp,.25)
      25%
-0.003994444
> mean(samp)
[1] 0.0003046662
> max(samp)
[1] 0.1158004

```

	samp_Open_diff	samp_High_Diff	sample_LowDiff	Samp_CloseDiff	Sampe_VolumeDiff	Samp_AdjCloseDiff
Average	0.000258822	0.000254861	0.000111912	0.000668447	0.025855988	0.000519359
max	0.060049465	0.05228356	0.094105146	0.051360282	3.234604878	0.05332681
var	0.000128361	7.00466E-05	8.6496E-05	8.63585E-05	0.048704956	8.36347E-05
quantile	-0.00434392	-0.003877694	-0.004365708	-0.004070566	-0.08493724	-0.003647843

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.

```

> mean(bse_datasets$Open_diff)
[1] 0.000329409
> max(bse_datasets$Open_diff)
[1] 0.1067121
> var(bse_datasets$Open_diff)
[1] 9.027371e-05
> quantile(bse_datasets$Open_diff, .25)
      25%
-0.004110794

```

```

> mean(bse_datasets$High_diff)
[1] 0.0003187801
> max(bse_datasets$High_diff)
[1] 0.08037943
> var(bse_datasets$High_diff)
[1] 6.939792e-05
> quantile(bse_datasets$High_diff)
      0%      25%      50%      75%     100%
-0.1311637779 -0.0037729124  0.0004030633  0.0044944574  0.0803794309
> quantile(bse_datasets$High_diff, .25)
      25%
-0.003772912

> mean(bse_datasets$Low_diff)
[1] 0.0003265001
> max(bse_datasets$Low_diff)
[1] 0.1067194
> var(bse_datasets$Low_diff)
[1] 8.646352e-05
> quantile(bse_datasets$Low_diff, .25)
      25%
-0.003996406

> quantile(bse_datasets$Close_diff, .25)
      25%
-0.004121264
> max(bse_datasets$Close_diff)
[1] 0.1158004
> mean(bse_datasets$Close_diff)
[1] 0.0003302519
> var(bse_datasets$Close_diff)
[1] 9.350225e-05

> var(bse_datasets$Volume_diff)
[1] 0.09080578
> max(bse_datasets$Volume_diff)
[1] 26.51968
> mean(bse_datasets$Volume_diff)
[1] 0.02062343
> quantile(bse_datasets$Volume_diff, .25)
      25%
-0.09553922

> quantile(bse_datasets$Adj.Close_diff, .25)
      25%
-0.004121264
> max(bse_datasets$Adj.Close_diff)
[1] 0.1158004
> mean(bse_datasets$Adj.Close_diff)
[1] 0.0003302519
> var(bse_datasets$Adj.Close_diff)
[1] 9.350225e-05

```

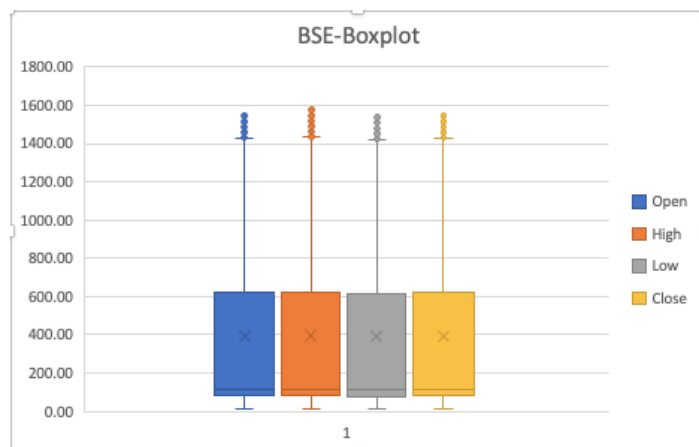
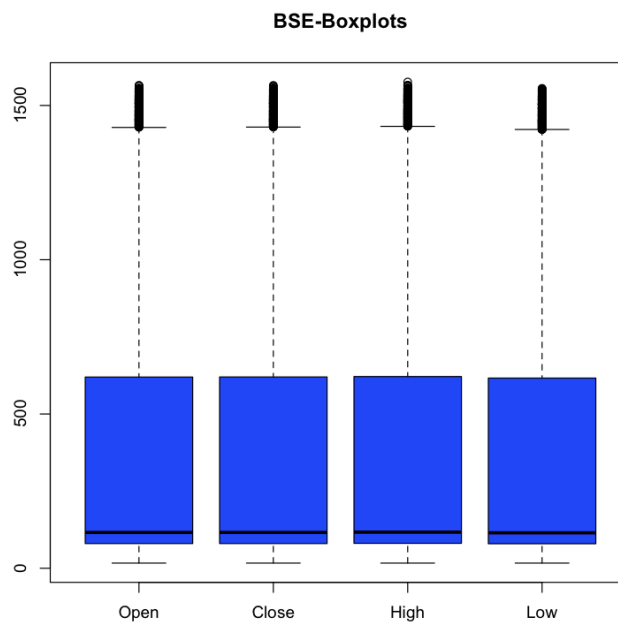
	Open_diff	High_Diff	LowDiff	CloseDiff	VolumeDiff	AdjCloseDiff	
Average	0.000329409	0.00031878	0.0003265	0.000330252	0.020623425	0.000330252	
max	0.106712058	0.080379431	0.106719368	0.11580036	26.51967979	0.11580036	
var	9.02737E-05	6.93979E-05	8.64635E-05	9.35022E-05	0.090805782	9.35022E-05	
quartile	-0.004110794	-0.003772912	-0.003996406	-0.004121264	-0.095539224	-0.004121264	

The mean computed from the sample of an attribute 'Open_diff' was around -0.000971 and whereas the mean with entire data (~0.00032) differs with some significant differences. One of the reason could be that as the mean is more sensitive towards extreme or outliers values.

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

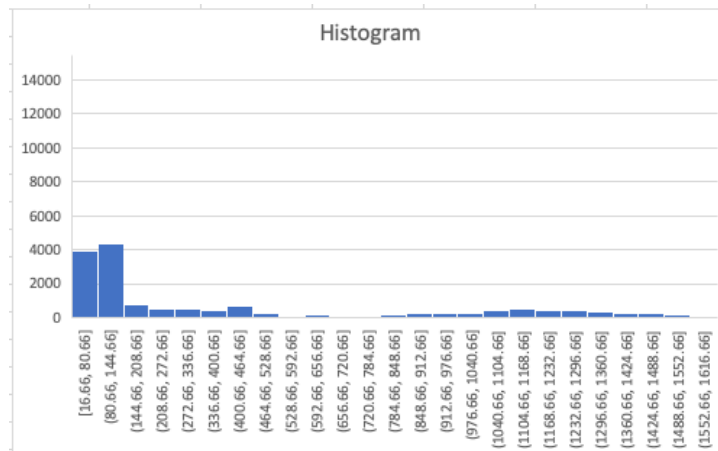
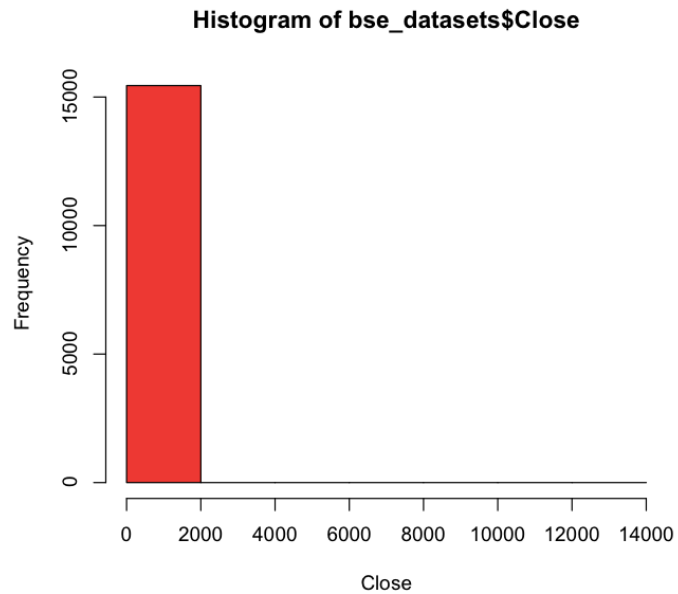
```
> boxplot(bse_datasets$Open, bse_datasets$Close, bse_datasets$High, bse_datasets$Low,
col="blue", main="BSE-Boxplots", names=c("Open", "Close", "High", "Low"))
~ |
```



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)

```
> hist(bse_datasets$Close, breaks=seq(0,15447,by=2000), col="red", xlab="Close",  
ylab="Frequency")
```



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

```
> inspect(itemsets)
```

	items	support	transIdenticalToItemsets	count
[1]	{C}	0.3	0.0	3
[2]	{A}	0.6	0.0	6
[3]	{E}	0.6	0.0	6
[4]	{B}	0.6	0.0	6
[5]	{D}	0.8	0.0	8
[6]	{A, B}	0.4	0.1	4
[7]	{A, D}	0.5	0.1	5
[8]	{B, E}	0.4	0.0	4
[9]	{D, E}	0.5	0.2	5
[10]	{B, D}	0.4	0.0	4
[11]	{A, B, D}	0.3	0.1	3
[12]	{B, D, E}	0.3	0.1	3

```

> rules <- apriori(trans, parameter = list(supp = .3, target="rules"))
Apriori

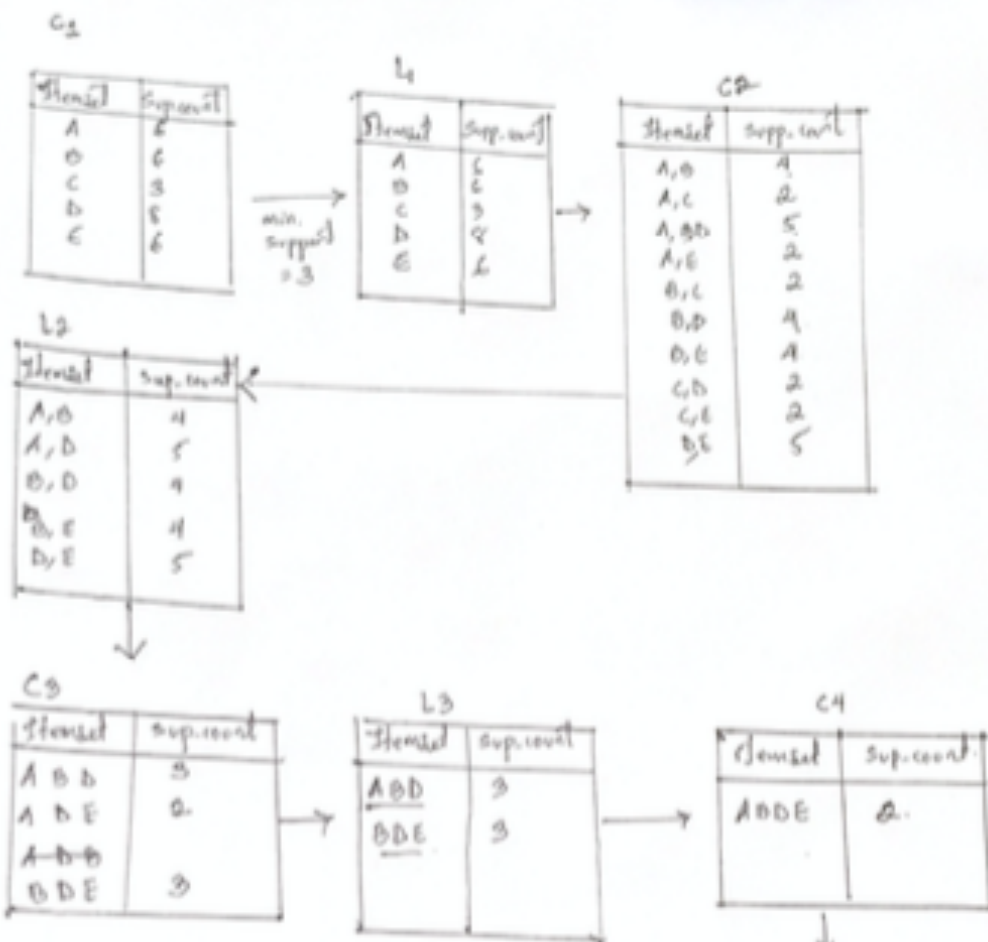
Parameter specification:
confidence minval smax arem aval originalSupport maxtime support minlen
      0.8      0.1    1 none FALSE              TRUE        5      0.3      1
maxlen target  ext
      10  rules TRUE

Algorithmic control:
filter tree heap memopt load sort verbose
  0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 3

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[5 item(s), 10 transaction(s)] done [0.00s].
sorting and recoding items ... [5 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 done [0.00s].
writing ... [3 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
> inspect(head(rules))
   lhs rhs support confidence coverage lift    count
[1] {} => {D} 0.8      0.8000000 1.0      1.000000 8
[2] {A} => {D} 0.5      0.8333333 0.6      1.041667 5
[3] {E} => {D} 0.5      0.8333333 0.6      1.041667 5

```

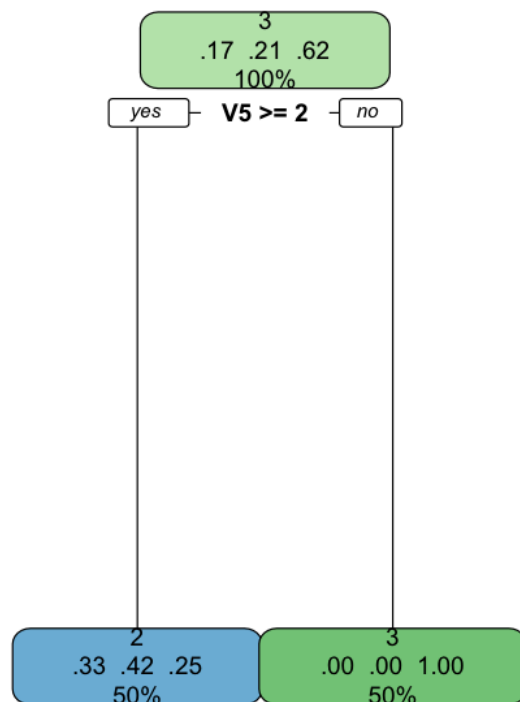
frequent Itemset is $L_3 = \{A, B, D\}, \{B, D, E\}$.

→ $\{A, B, D\}$
Association Rules
 $\{A, B\} \rightarrow \{D\}$
 $\{A, D\} \rightarrow \{B\}$
 $\{B, D\} \rightarrow \{A\}$
 $\{A\} \rightarrow \{B, D\}$
 $\{B\} \rightarrow \{A, D\}$
 $\{D\} \rightarrow \{A, B\}$

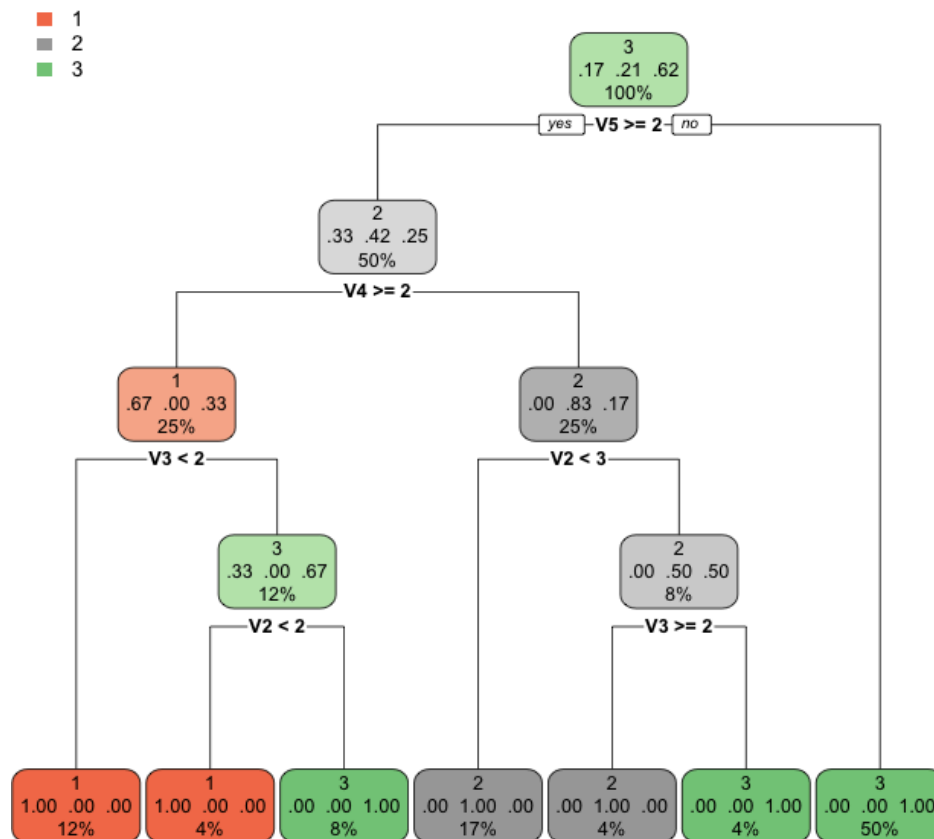
Association Rules for $\{B, D, E\}$
 $\{B, D\} \rightarrow \{E\}$
 $\{B, E\} \rightarrow \{D\}$
 $\{D, E\} \rightarrow \{B\}$
 $\{B\} \rightarrow \{D, E\}$
 $\{D\} \rightarrow \{B, E\}$
 $\{E\} \rightarrow \{B, D\}$

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)

```
> x <- lenses[,2:5]
> y <- as.factor(lenses[,6])
> fit <- rpart(y~., data=x, method="class")
```



```
> fit <- rpart(y~., data=x, control=rpart.control(minspit=0, minbucket=0, cp=-1, maxcompete=0,
maxsurrogate=0, usesurrogate=0, xval=0, maxdepth=4))
> rpart.plot(fit)
>
```



```

> 1 - sum(y==predict(fit, x, type = 'class')) / length(y)
[1] 0.2916667
> |

```

```

> information.gain(y~., lenses)
attr_importance
V1      0.0000000
V2      0.0000000
V3      0.0000000
V4      0.2613201
V5      0.3803957
V6      0.9191738
> sum(y==predict(fit, x, type='class')) / length(y)
[1] 0.7083333
> |

```

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 ($0 \leq c_1 < 5$, $5 \leq c_2 < 10$, $10 \leq c_3 < 15$, $15 \leq c_4 \leq 20$). 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)

Parameters:

x_train: data frame of training sets

x_test: data frame of testing sets

y_train: factor of true classification of training sets

```

>
> datasets <- split(liver_datasets, liver_datasets$V7) #splitting the data into
train and test sets based on the last attribute.
> train <- datasets[[1]]
> test <- datasets[[2]]
> x_train <- train[,1:5] #Independent variables
> y_train <- cut(train$V6, breaks=c(-Inf,0,5,10,15,20,Inf),
labels=c("na","c1","c2","c3","c4","c4"), right=FALSE) #dependent variable:
replaced the decision values into 4 classes (0<=c1<5, 5<=c2<10, 10<=c3<15,
15<=c4<=20)
> x_test <- test[,1:5]
> y_test <- cut(test$V6, breaks=c(-Inf,0,5,10,15,20,Inf),
labels=c("na","c1","c2","c3","c4","c4"), right=FALSE)
>
> library(class)
> fit_train_set <- knn(x_train, x_train, y_train) #fitting knn model with default
one nearest-neighbor and with default measure euclidean distance
> 1 - sum(y_train == fit_train_set) / length(y_train) #compute misclassification
error on the training data fit with 1-nearest neighbor
[1] 0
> fit_train_set <- knn(x_train, x_train, y_train, k=2) #fitting knn model with 2
nearest-neighbor and with default measure euclidean distance
> 1 - sum(y_train == fit_train_set) / length(y_train) #misclassification error on
the training data fit with 2-nearest neighbor
[1] 0.1586207
> fit_train_set <- knn(x_train, x_train, y_train, k=3) #fit knn model with 3
nearest-neighbor and with default measure euclidean distance
> 1 - sum(y_train == fit_train_set) / length(y_train) #misclassification error on
the training data fit with 3-nearest neighbor
[1] 0.2206897

>
> fit_test_set <- knn(x_train, x_test, y_train) #fitting knn model with default
one nearest-neighbor and with default measure euclidean distance
> 1 - sum(y_test == fit_test_set) / length(y_test)
[1] 0.445
> fit_test_set <- knn(x_train, x_test, y_train, k=2) #fit knn model with 2
nearest-neighbor and with default measure euclidean distance
> 1 - sum(y_test == fit_test_set) / length(y_test)
[1] 0.445
> fit_test_set <- knn(x_train, x_test, y_train, k=3) #fit knn model with 3
nearest-neighbor and with default measure euclidean distance
> 1 - sum(y_test == fit_test_set) / length(y_test)
[1] 0.4
>

```

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

parameters:

x_train: data frame of training set (Independent variables)

y_train: data frame of training set (Dependent variable)

```
> library(e1071)
> fit_train <- svm(x_train,y_train)
> 1 - sum(y_train==predict(fit_train,x_train)) / length(y_train)
[1] 0.2137931
> 1 - sum(y_test==predict(fit_train, x_test)) / length(y_test)
[1] 0.285
```

SVM is better with classification than the nearest neighbor.

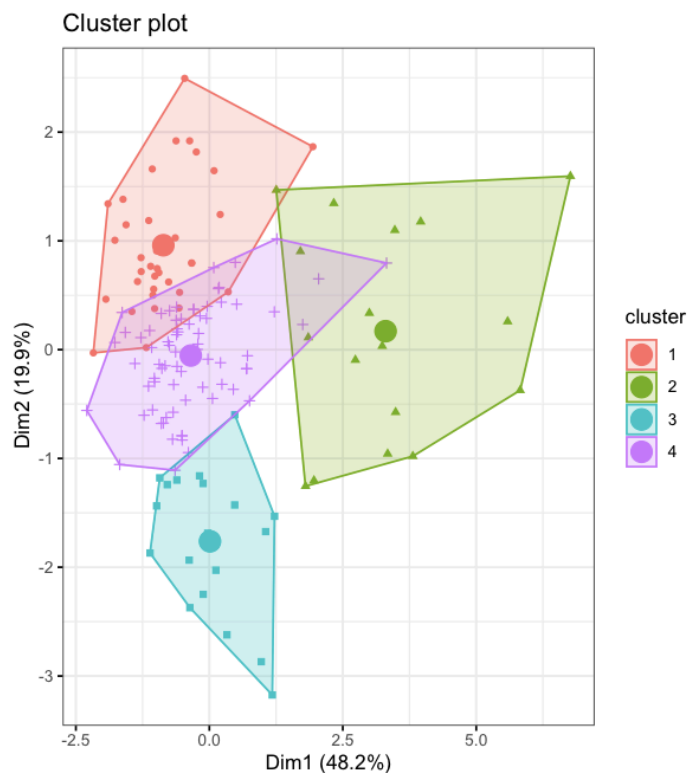
6. Create k-means clusters for k=4 for the Liver Disorders Data Set at <http://archive.ics.uci.edu/ml/datasets/Liver+Disorders> . Explain the input parameters you provided for the clustering algorithm. Plot the fitted cluster centers using a different color. Finally assign the cluster membership for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

parameters:

x_train: data frame of training set

4: (k value) number of clusters

```
> kmeans_model <- kmeans(x_train,4)
> library(factoextra)
> library(ggpubr)
> fviz_cluster(kmeans_model, data = x_train, geom="point", ellipse.type = "convex",
ggtheme=theme_bw()) + stat_mean(aes(color=cluster), size=5)
```



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)

```
y <- liver_datasets[,6]
knn_fit <- knn(kmeans_model$centers, x_train, as.factor(c(1,2,3,4)))
1 - sum(y==knn_fit) / length(y)
[1] 0.9362319
```

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

$(\text{value of day } x - \text{value of day } x-1) / \text{value of day } x-1$. 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)

```
> Close_growth <- (bse_datasets$Close[seq(2,nrow(bse_datasets))] -
bse_datasets$Close[seq(1,nrow(bse_datasets)-1)]) /
bse_datasets$Close[seq(1,nrow(bse_datasets)-1)]
>
> growth_rate <- c(NA)
[1] NA
> growth_rate <- append(growth_rate, Close_growth)
[1] NA
>
> bse_datasets <- cbind(bse_datasets, close_growth_rate=growth_rate)
```



```
> bse_datasets[1:4,]
```

	Date	Open	High	Low	Close	Volume	Adj.Close	Open_diff
1	5/23/2011	1333.07	1333.07	1312.88	1317.37	3255580000	1317.37	-0.0066542474
2	5/20/2011	1342.00	1342.00	1330.67	1333.27	4066020000	1333.27	-0.0002979738
3	5/19/2011	1342.40	1346.82	1336.36	1343.60	3626110000	1343.60	0.0104325049
4	5/18/2011	1328.54	1341.82	1326.59	1340.68	3922030000	1340.68	0.0018399819

	High_diff	Low_diff	Close_diff	Volume_diff	Adj.Close_diff
1	-0.006654247	-0.013369205	-0.011925566	-0.19932022	-0.011925566
2	-0.003578800	-0.004257835	-0.007688300	0.12131733	-0.007688300
3	0.003726282	0.007364747	0.002177999	-0.07545072	0.002177999
4	0.008568723	0.006128129	0.008803744	-0.03254587	0.008803744

	close_growth_rate
1	NA
2	0.012069502
3	0.007747868
4	-0.002173266

```
> z_score = (bse_datasets$close_growth_rate - mean(bse_datasets$close_growth_rate, na.rm=T)) /  
sd(bse_datasets$close_growth_rate, na.rm=T)
```

	Date	Open	High	Low	Close	Volume	Adj.Close	Open_diff	High_diff
1	5/23/2011	1333.07	1333.07	1312.88	1317.37	3255580000	1317.37	-0.0066542474	-0.006654247
2	5/20/2011	1342.00	1342.00	1330.67	1333.27	4066020000	1333.27	-0.0002979738	-0.003578800
3	5/19/2011	1342.40	1346.82	1336.36	1343.60	3626110000	1343.60	0.0104325049	0.003726282
4	5/18/2011	1328.54	1341.82	1326.59	1340.68	3922030000	1340.68	0.0018399819	0.008568723

	Low_diff	Close_diff	Volume_diff	Adj.Close_diff	close_growth_rate	z_score
1	-0.013369205	-0.011925566	-0.19932022	-0.011925566	NA	NA
2	-0.004257835	-0.007688300	0.12131733	-0.007688300	0.012069502	1.2601205
3	0.007364747	0.002177999	-0.07545072	0.002177999	0.007747868	0.8175583
4	0.006128129	0.008803744	-0.03254587	0.008803744	-0.002173266	-0.1984275

```

> bse_datasets$Date[bse_datasets$z_score > 3 | bse_datasets$z_score < -3]
[1] NA "7/15/2010" "7/6/2010" "6/28/2010" "6/3/2010" "5/26/2010"
[7] "5/19/2010" "5/7/2010" "5/5/2010" "2/3/2010" "7/1/2009" "6/19/2009"
[13] "5/1/2009" "4/17/2009" "4/8/2009" "3/27/2009" "3/20/2009" "3/16/2009"
[19] "3/11/2009" "3/9/2009" "3/4/2009" "2/27/2009" "2/23/2009" "2/20/2009"
[25] "2/13/2009" "2/9/2009" "1/28/2009" "1/27/2009" "1/20/2009" "1/16/2009"
[31] "1/13/2009" "1/6/2009" "12/31/2008" "12/15/2008" "12/10/2008" "12/5/2008"
[37] "12/4/2008" "12/3/2008" "12/1/2008" "11/28/2008" "11/25/2008" "11/21/2008"
[43] "11/20/2008" "11/19/2008" "11/18/2008" "11/13/2008" "11/12/2008" "11/11/2008"
[49] "11/5/2008" "11/4/2008" "11/3/2008" "10/27/2008" "10/24/2008" "10/23/2008"
[55] "10/21/2008" "10/20/2008" "10/17/2008" "10/15/2008" "10/14/2008" "10/10/2008"
[61] "10/8/2008" "10/6/2008" "10/3/2008" "10/1/2008" "9/29/2008" "9/26/2008"
[67] "9/19/2008" "9/18/2008" "9/17/2008" "9/16/2008" "9/12/2008" "9/8/2008"
[73] "9/3/2008" "6/25/2008" "6/5/2008" "3/31/2008" "3/17/2008" "3/10/2008"
[79] "2/4/2008" "1/16/2008" "11/6/2007" "8/8/2007" "2/26/2007" "3/21/2003"
[85] "3/14/2003" "3/12/2003" "1/23/2003" "12/31/2002" "10/14/2002" "10/10/2002"
[91] "10/9/2002" "9/30/2002" "9/26/2002" "9/18/2002" "8/30/2002" "8/13/2002"
[97] "8/7/2002" "8/2/2002" "7/31/2002" "7/26/2002" "7/23/2002" "7/19/2002"
[103] "7/18/2002" "7/9/2002" "7/3/2002" "5/7/2002" "1/28/2002" "9/21/2001"
[109] "9/19/2001" "9/10/2001" "4/17/2001" "4/4/2001" "4/2/2001" "3/9/2001"
[115] "1/2/2001" "12/10/2000" "12/4/2000" "10/10/2000" "10/12/2000" "5/26/2000"

```