1st solution:

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 the mean of its immediate three previous row values.

```
C:/Users/HARIKA/Desktop/FinalExam/
      Open            High             Low             Close           Volume
 Min.   :  17.08   Min.   :  17.08   Min.   :  17.08   Min.   :  17.08   Min.   :7.800e+05
 1st Qu.:  83.43   1st Qu.:  84.07   1st Qu.:  82.50   1st Qu.:  83.17   1st Qu.:9.030e+06
 Median : 116.45   Median : 117.59   Median : 115.03   Median : 116.34   Median :4.390e+07
 Mean   : 398.28   Mean   : 401.03   Mean   : 395.52   Mean   : 398.50   Mean   :5.964e+08
 3rd Qu.: 650.67   3rd Qu.: 654.12   3rd Qu.: 644.93   3rd Qu.: 648.62   3rd Qu.:4.035e+08
 Max.   :1522.19   Max.   :1526.45   Max.   :1500.74   Max.   :1506.34   Max.   :8.926e+09
   Adj.Close          open_new            high_new            low_new
 Min.   :  17.08   Min.   :-0.0582780   Min.   :-0.0432817   Min.   :-0.0474458
 1st Qu.:  83.17   1st Qu.:-0.0039618   1st Qu.:-0.0034432   1st Qu.:-0.0038973
 Median : 116.34   Median : 0.0005554   Median : 0.0003948   Median : 0.0008122
 Mean   : 398.50   Mean   : 0.0005955   Mean   : 0.0004185   Mean   : 0.0005022
 3rd Qu.: 648.62   3rd Qu.: 0.0050955   3rd Qu.: 0.0045302   3rd Qu.: 0.0047861
 Max.   :1506.34   Max.   : 0.1067121   Max.   : 0.0343908   Max.   : 0.0910833
   close_new           volume_new         Adj.close_new
 Min.   :-0.0402908   Min.   :-0.718888   Min.   :-0.0402908
 1st Qu.:-0.0042513   1st Qu.:-0.105633   1st Qu.:-0.0042513
 Median : 0.0003301   Median :-0.002597   Median : 0.0003301
 Mean   : 0.0003370   Mean   : 0.007552   Mean   : 0.0003370
 3rd Qu.: 0.0048696   3rd Qu.: 0.103772   3rd Qu.: 0.0048696
 Max.   : 0.0573273   Max.   : 1.677175   Max.   : 0.0573273
> data_3000 = randomRows(data, 3000)
> summary(data_3000)
      Open            High             Low             Close           Volume
 Min.   :  16.72   Min.   :  16.72   Min.   :  16.72   Min.   :  16.72   Min.   :7.400e+05
 1st Qu.:  79.61   1st Qu.:  80.10   1st Qu.:  78.94   1st Qu.:  79.42   1st Qu.:5.972e+06
 Median : 113.11   Median : 114.21   Median : 111.98   Median : 112.88   Median :4.013e+07
 Mean   : 379.96   Mean   : 382.57   Mean   : 377.36   Mean   : 380.19   Mean   :5.449e+08
 3rd Qu.: 495.77   3rd Qu.: 497.82   3rd Qu.: 494.57   3rd Qu.: 497.14   3rd Qu.:3.181e+08
 Max.   :1556.51   Max.   :1563.03   Max.   :1554.09   Max.   :1561.80   Max.   :1.146e+10
   Adj.Close          open_new            high_new            low_new
 Min.   :  16.72   Min.   :-0.0871188   Min.   :-0.0685302   Min.   :-0.0821116
 1st Qu.:  79.42   1st Qu.:-0.0039658   1st Qu.:-0.0039459   1st Qu.:-0.0041704
 Median : 112.88   Median : 0.0005062   Median : 0.0004148   Median : 0.0005606
 Mean   : 380.19   Mean   : 0.0003592   Mean   : 0.0003885   Mean   : 0.0004167
 3rd Qu.: 497.14   3rd Qu.: 0.0049885   3rd Qu.: 0.0046277   3rd Qu.: 0.0047436
 Max.   :1561.80   Max.   : 0.0594595   Max.   : 0.0540658   Max.   : 0.1067194
   close_new           volume_new         Adj.close_new
 Min.   :-0.0680141   Min.   :-0.754927   Min.   :-0.0680141
 1st Qu.:-0.0044001   1st Qu.:-0.092642   1st Qu.:-0.0044001
```

```
> data_3000 = randomRows(data, 3000)
> summary(data_3000)
      Open              High              Low               Close              Volume
 Min.   :  16.72   Min.   :  16.72   Min.   :  16.72   Min.   :  16.72   Min.   :7.400e+05
 1st Qu.:  79.61   1st Qu.:  80.10   1st Qu.:  78.94   1st Qu.:  79.42   1st Qu.:5.972e+06
 Median : 113.11   Median : 114.21   Median : 111.98   Median : 112.88   Median :4.013e+07
 Mean   : 379.96   Mean   : 382.57   Mean   : 377.36   Mean   : 380.19   Mean   :5.449e+08
 3rd Qu.: 495.77   3rd Qu.: 497.82   3rd Qu.: 494.57   3rd Qu.: 497.14   3rd Qu.:3.181e+08
 Max.   :1556.51   Max.   :1563.03   Max.   :1554.09   Max.   :1561.80   Max.   :1.146e+10
   Adj.Close          open_new             high_new             low_new
 Min.   :  16.72   Min.   :-0.0871188   Min.   :-0.0685302   Min.   :-0.0821116
 1st Qu.:  79.42   1st Qu.:-0.0039658   1st Qu.:-0.0039459   1st Qu.:-0.0041704
 Median : 112.88   Median : 0.0005062   Median : 0.0004148   Median : 0.0005606
 Mean   : 380.19   Mean   : 0.0003592   Mean   : 0.0003885   Mean   : 0.0004167
 3rd Qu.: 497.14   3rd Qu.: 0.0049885   3rd Qu.: 0.0046277   3rd Qu.: 0.0047436
 Max.   :1561.80   Max.   : 0.0594595   Max.   : 0.0540658   Max.   : 0.1067194
   close_new            volume_new         Adj.close_new
 Min.   :-0.0680141   Min.   :-0.754927   Min.   :-0.0680141
 1st Qu.:-0.0044001   1st Qu.:-0.092642   1st Qu.:-0.0044001
 Median : 0.0004455   Median : 0.004051   Median : 0.0004455
 Mean   : 0.0004045   Mean   : 0.017172   Mean   : 0.0004045
 3rd Qu.: 0.0050338   3rd Qu.: 0.109569   3rd Qu.: 0.0050338
 Max.   : 0.1078900   Max.   : 2.996867   Max.   : 0.1078900
>
```

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(data_1000$open_new)
[1] 0.0005955025
> mean(data_1000$high_new)
[1] 0.0004184797
> mean(data_1000$low_new)
[1] 0.0005022487
> mean(data_1000$close_new)
[1] 0.0003369592
> mean(data_1000$volume_new)
[1] 0.007551912
> mean(data_1000$Adj.close_new)
[1] 0.0003369592
>
> var(data_1000$open_new)
[1] 8.714339e-05
> var(data_1000$high_new)
[1] 6.119132e-05
> var(data_1000$low_new)
[1] 8.313995e-05
> var(data_1000$close_new)
[1] 7.637739e-05
> var(data_1000$volume_new)
[1] 0.0327711
> var(data_1000$Adj.close_new)
[1] 7.637739e-05
>
> max(data_1000$open_new)
[1] 0.1067121
> max(data_1000$high_new)
[1] 0.03439077
> max(data_1000$low_new)
[1] 0.09108332
> max(data_1000$close_new)
[1] 0.05732732
> max(data_1000$volume_new)
[1] 1.677175
> max(data_1000$Adj.close_new)
[1] 0.05732732
>
> quantile(data_1000$open_new,0.25)
```

Console Terminal Jobs

C:/Users/HARIKA/Desktop/FinalExam/

```
[1] 8.768766e-05
> var(data_3000$close_new)
[1] 8.588174e-05
> var(data_3000$volume_new)
[1] 0.03939109
> var(data_3000$Adj.close_new)
[1] 8.588174e-05
>
> max(data_3000$open_new)
[1] 0.05945946
> max(data_3000$high_new)
[1] 0.05406578
> max(data_3000$low_new)
[1] 0.1067194
> max(data_3000$close_new)
[1] 0.10789
> max(data_3000$volume_new)
[1] 2.996867
> max(data_3000$Adj.close_new)
[1] 0.10789
>
> quantile(data_3000$open_new,0.25)
         25%
-0.003965834
> quantile(data_3000$high_new,0.25)
         25%
-0.003945885
> quantile(data_3000$low_new,0.25)
         25%
-0.004170403
> quantile(data_3000$close_new,0.25)
        25%
-0.00440009
> quantile(data_3000$volume_new,0.25)
        25%
-0.09264194
> quantile(data_3000$Adj.close_new,0.25)
        25%
-0.00440009
>
```

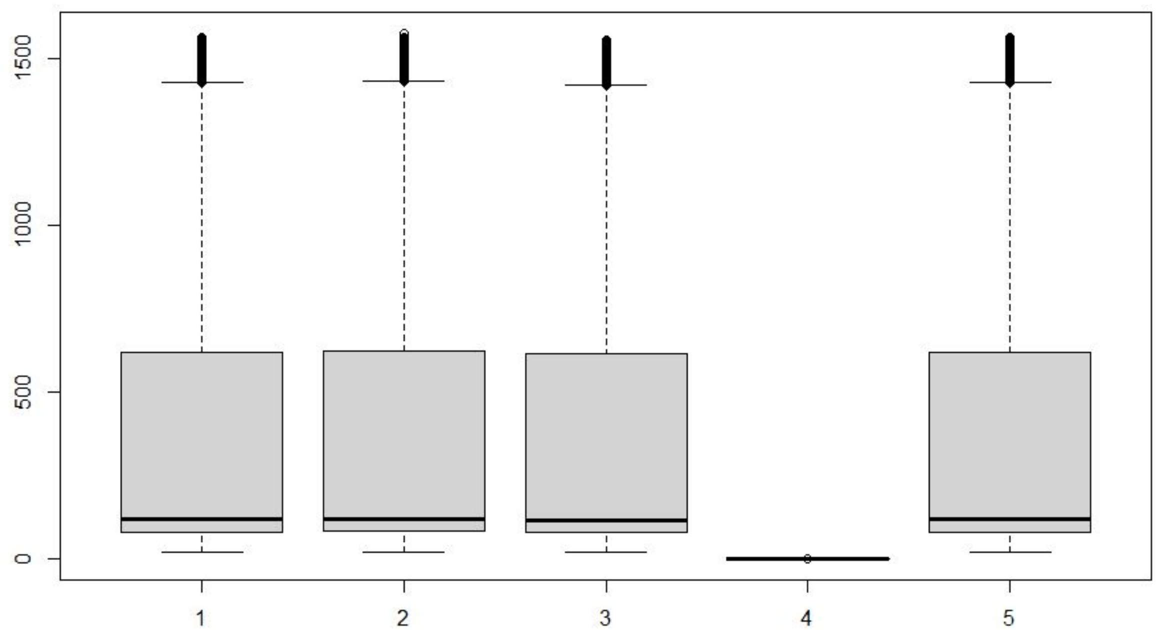
c) Compute the same quantities in part b on the entire data set and show your answers. How much do they differ from your answers in part b? Do you find any significant difference between two sample values like mean in comparison with entire data? If so what explanation you can give for that?
Do the same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

```
> mean(data$open_new)
[1] 0.000329528
> mean(data$high_new)
[1] 0.0003188991
> mean(data$low_new)
[1] 0.0003266191
> mean(data$close_new)
[1] 0.0003303709
> mean(data$volume_new)
[1] 0.02062874
> mean(data$Adj.close_new)
[1] 0.0003303709
>
> var(data$open_new)
[1] 9.027493e-05
> var(data$high_new)
[1] 6.939914e-05
> var(data$low_new)
[1] 8.646474e-05
> var(data$close_new)
[1] 9.350347e-05
> var(data$volume_new)
[1] 0.09080738
> var(data$Adj.close_new)
[1] 9.350347e-05
>
> max(data$open_new)
[1] 0.1067121
> max(data$high_new)
[1] 0.08037943
> max(data$low_new)
[1] 0.1067194
> max(data$close_new)
[1] 0.1158004
> max(data$volume_new)
[1] 26.51968
> max(data$Adj.close_new)
[1] 0.1158004
>
> quantile(data$open_new,0.25)
        25%
```

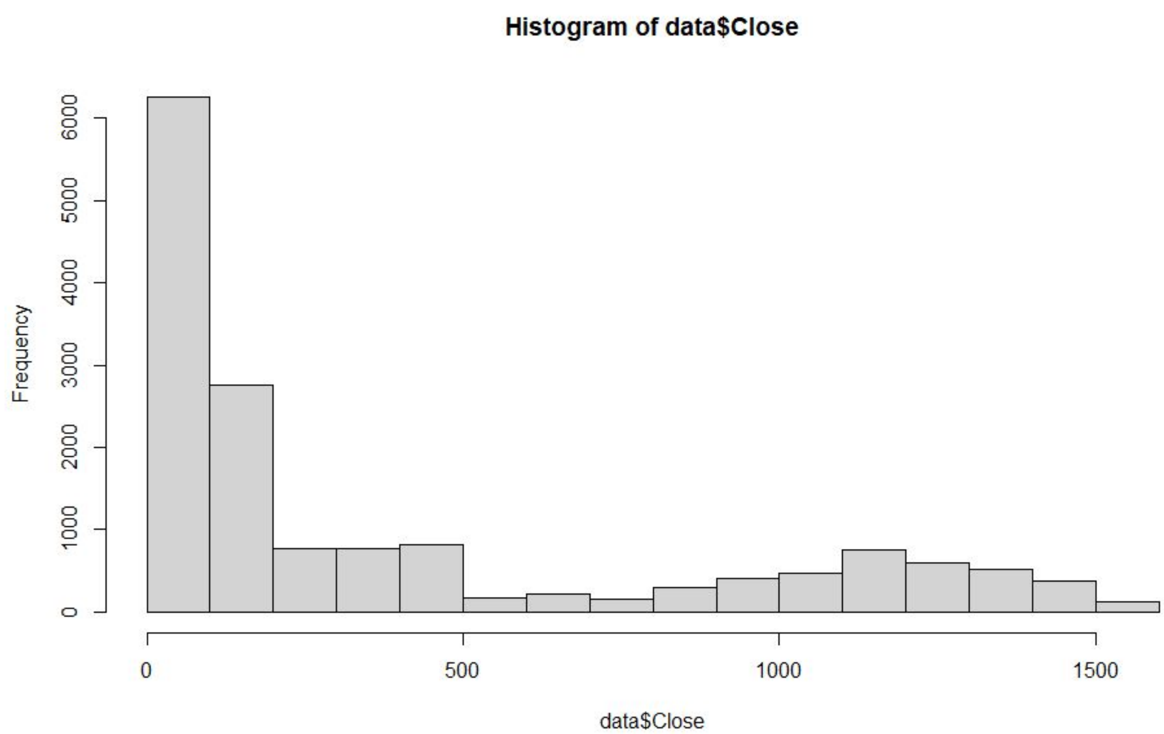
```
        25%
-0.004110794
> quantile(data$high_new,0.25)
        25%
-0.003772912
> quantile(data$low_new,0.25)
        25%
-0.003996406
> quantile(data$close_new,0.25)
        25%
-0.004121264
> quantile(data$volume_new,0.25)
       25%
-0.09553922
> quantile(data$Adj.close_new,0.25)
        25%
-0.004121264
> 
```

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)



0
500
1000
1500
1
2
3
4
5



Histogram of data$Close
Frequency
0
1000
2000
3000
4000
5000
6000
500
1000
1500
data$Close

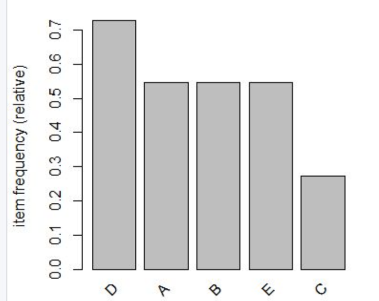
2.Implement Apriori Algorithm or use built in packages to find out the frequent itemsets and generate rules for frequent itemsets. Trace and submit the program output for the following given dataset of transactions with a minimum support of 3. (10M)

```
C:/Users/HARIKA/Desktop/FinalExam/
> setwd("C:\\Users\\HARIKA\\Desktop\\FinalExam")
> data = read.csv("Apriori-Data.csv", header = TRUE);
>
> View(data)
> data$TID <- NULL
> library(arules)
>
> write.csv(data, "Frequent-Itemsets.csv", quote = FALSE, row.names = TRUE)
> transactions = read.transactions("ItemList.csv", sep=',', rm.duplicates = TRUE)
> inspect(transactions)
     items
[1]  {Items,X}
[2]  {1,A,B,C,D,E}
[3]  {2,A,C,D}
[4]  {3,D,E}
[5]  {4,B,C,E}
[6]  {5,A,B,D,E}
[7]  {6,A,B}
[8]  {7,B,D,E}
[9]  {8,A,B,D}
[10] {9,A,D}
[11] {10,D,E}
>
> frequent_itemsets <- apriori(transactions, parameter = list(sup = 0.03, conf = 0.5,target="frequent ite
msets"))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen            target
         NA    0.1    1 none FALSE            TRUE       5    0.03      1     10 frequent itemsets
  ext
 TRUE

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

Absolute minimum support count: 0
```

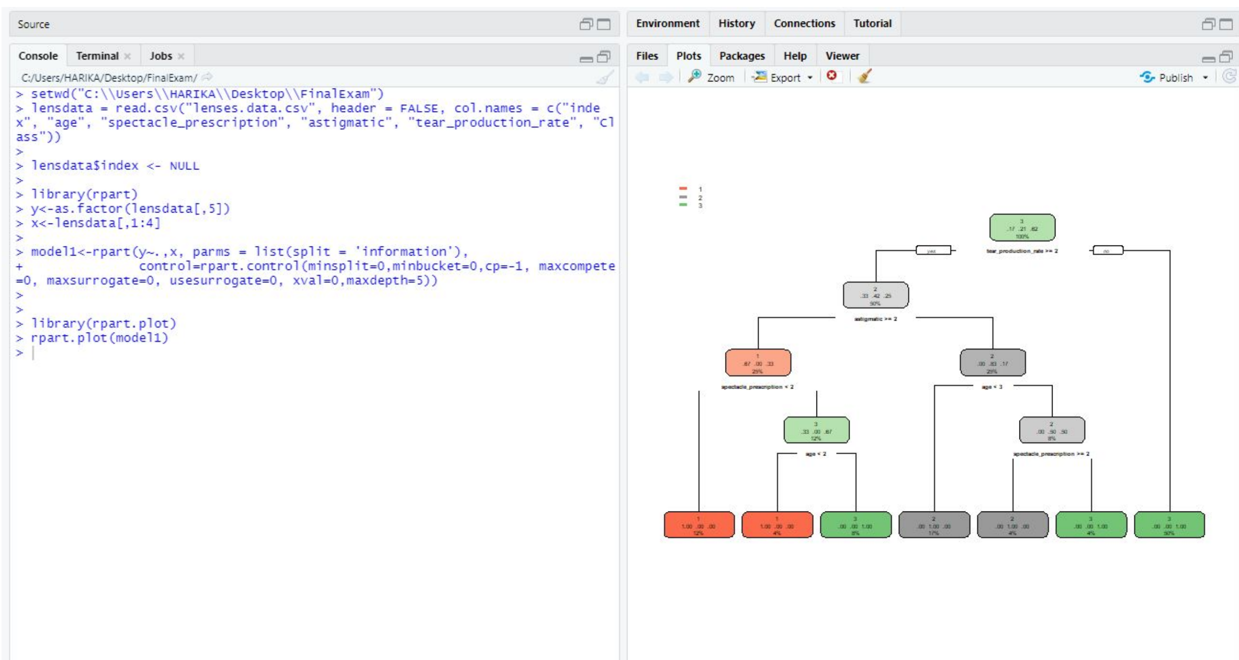



```
Absolute minimum support count: 0

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[17 item(s), 11 transaction(s)] done [0.00s].
sorting and recoding items ... [17 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 done [0.00s].
sorting transactions ... done [0.00s].
writing ... [130 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
>
> itemFrequencyPlot(transactions, topN = 5)
```

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?

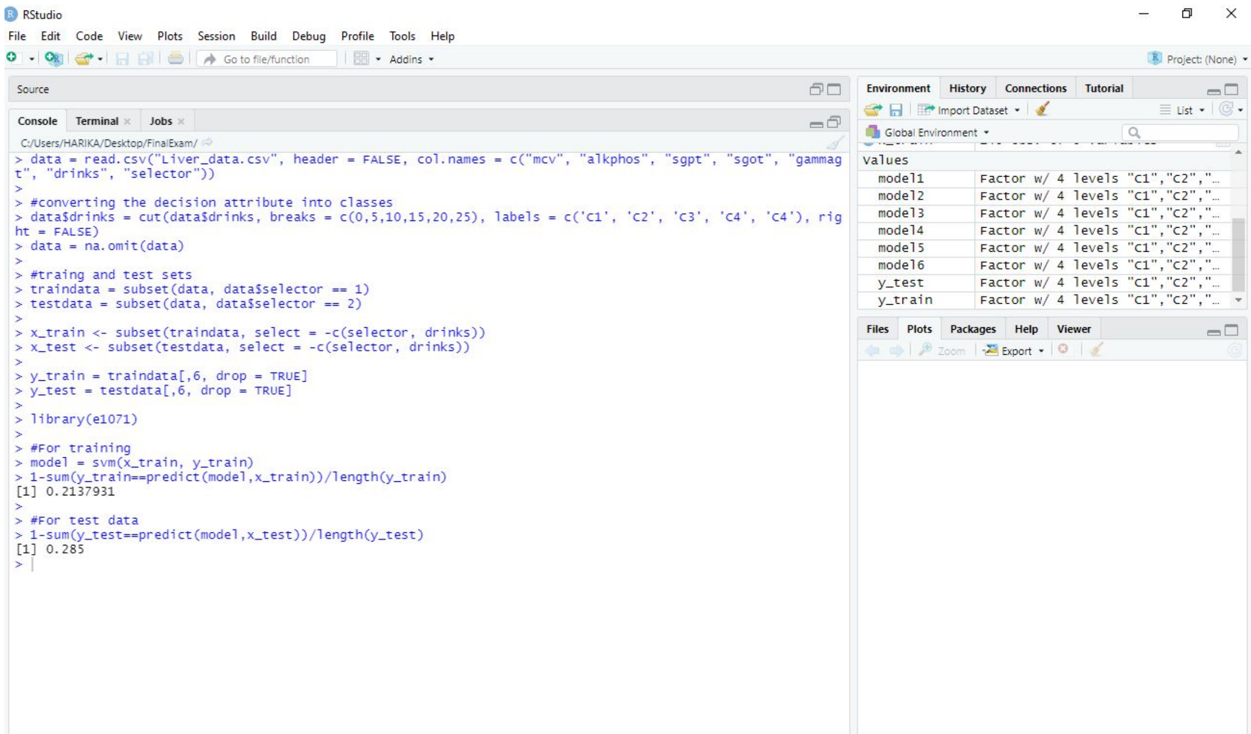
```
> #Information gain
> gain <- sum(y==predict(model1,x,type="class"))/length(y)
> gain
[1] 1
>
> #misclassification error rate
> error_rate <- 1-sum(y==predict(model1,x,type="class"))/length(y)
> error_rate
[1] 0
>
```



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)

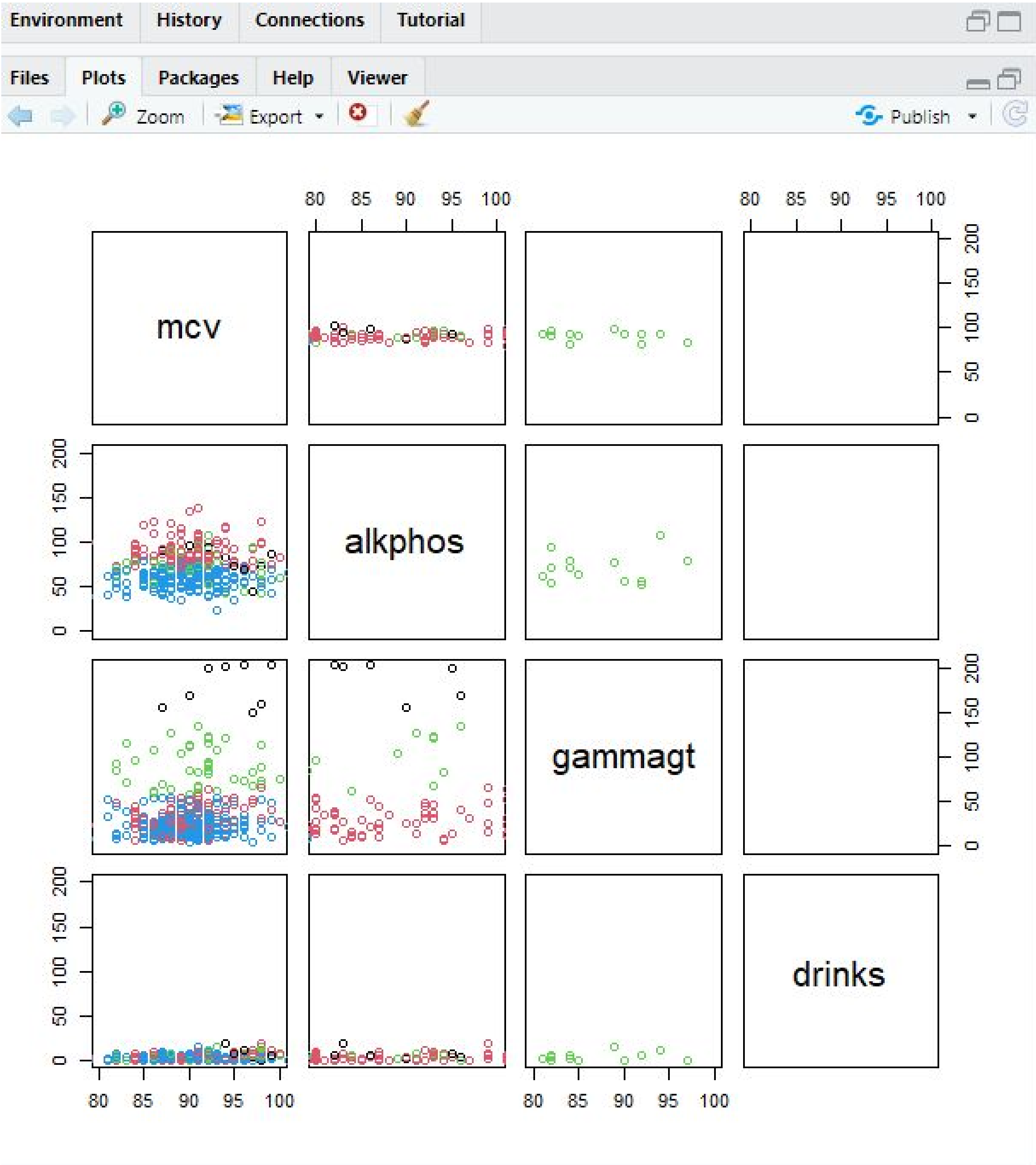
```
>
> #For Training Data
> #knn if k=1
> library(class)
> model1 = knn(x_train, x_test, y_train, k = 1)
> 1-sum(y_train==model1)/length(y_train)
[1] 0.2896552
>
> #knn if k=2
> model2 = knn(x_train, x_train, y_train, k = 2)
> 1-sum(y_train==model2)/length(y_train)
[1] 0.1655172
>
> #knn if k=3
> model3 = knn(x_train, x_train, y_train, k = 3)
> 1-sum(y_train==model3)/length(y_train)
[1] 0.2068966
>
>
> #For Test Data
> #knn if k=1
> model4 = knn(x_train, x_test, y_train, k = 1)
> 1-sum(y_test==model4)/length(y_test)
[1] 0.44
>
> #knn if k=2
> model5 = knn(x_train, x_test, y_train, k = 2)
> 1-sum(y_test==model5)/length(y_test)
[1] 0.44
>
> #knn if k=3
> model6 = knn(x_train, x_test, y_train, k = 3)
> 1-sum(y_test==model6)/length(y_test)
[1] 0.39
>
```

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



---> Now, The misclassification error is high for KNN hence, we can choose SVM over KNN

6.



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)

```
> sort(z)
  [1] -21.200164  -9.377746  -9.268692  -9.141750  -8.595889  -7.911031  -7.134352  -7.067886
  [9]  -7.033645  -6.975723  -6.937815  -6.878679  -6.360330  -6.358610  -6.343812  -6.061018
 [17]  -5.969685  -5.599374  -5.496172  -5.481801  -5.400803  -5.370090  -5.232245  -5.123826
 [25]  -5.114060  -5.006956  -4.909254  -4.908753  -4.855416  -4.745677  -4.538940  -4.499730
 [33]  -4.459298  -4.432670  -4.354265  -4.346507  -4.329650  -4.200860  -4.138610  -4.088660
 [41]  -4.064875  -4.017515  -4.002188  -4.000410  -3.999608  -3.988428  -3.860144  -3.830817
 [49]  -3.817696  -3.782429  -3.768451  -3.708921  -3.684389  -3.677650  -3.634967  -3.625318
 [57]  -3.622589  -3.603167  -3.592849  -3.590947  -3.580911  -3.564583  -3.546365  -3.505900
 [65]  -3.494441  -3.459307  -3.437640  -3.380025  -3.370270  -3.342998  -3.319092  -3.270656
 [73]  -3.254594  -3.246248  -3.241814  -3.232989  -3.228593  -3.222148  -3.219333  -3.213152
 [81]  -3.211454  -3.211050  -3.198724  -3.189995  -3.189810  -3.175534  -3.161884  -3.147789
 [89]  -3.143598  -3.143319  -3.137630  -3.134511  -3.128572  -3.101858  -3.100399  -3.096909
 [97]  -3.092964  -3.092551  -3.088760  -3.071460  -3.070993  -3.065112  -3.059963  -3.057360
[105]  -3.048163  -3.042798  -3.035610  -3.024178  -3.019078  -3.014551  -3.008509  -3.003847
[113]  -2.993174  -2.984002  -2.948282  -2.939099  -2.936503  -2.936488  -2.935785  -2.933105
[121]  -2.896882  -2.891914  -2.873629  -2.856063  -2.855682  -2.854152  -2.835661  -2.828542
[129]  -2.828319  -2.828160  -2.826967  -2.823548  -2.815520  -2.814485  -2.810019  -2.809115
[137]  -2.800195  -2.798689  -2.783580  -2.783244  -2.771490  -2.768274  -2.766765  -2.748036
[145]  -2.721061  -2.717810  -2.713576  -2.708366  -2.707432  -2.706638  -2.705362  -2.703368
[153]  -2.698115  -2.693855  -2.692876  -2.684901  -2.684819  -2.683214  -2.681262  -2.672988
[161]  -2.672246  -2.666344  -2.662731  -2.657205  -2.647594  -2.611798  -2.611069  -2.598381
[169]  -2.597240  -2.595148  -2.592370  -2.590500  -2.589741  -2.586994  -2.585953  -2.578615
[177]  -2.575493  -2.573178  -2.570979  -2.570914  -2.570401  -2.561906  -2.560573  -2.557254
[185]  -2.545845  -2.543108  -2.527819  -2.524139  -2.523433  -2.517064  -2.516159  -2.501099
[193]  -2.499432  -2.497356  -2.495753  -2.492201  -2.492092  -2.484005  -2.477540  -2.477249
[201]  -2.473472  -2.471094  -2.470158  -2.464251  -2.463190  -2.461643  -2.459556  -2.456672
[209]  -2.454286  -2.452216  -2.447735  -2.437809  -2.437410  -2.430328  -2.427655  -2.425600
[217]  -2.424877  -2.421237  -2.409432  -2.401197  -2.399374  -2.397446  -2.397138  -2.390917
[225]  -2.389577  -2.380272  -2.379829  -2.379093  -2.375008  -2.373165  -2.370513  -2.367652
[233]  -2.366689  -2.365369  -2.353898  -2.353653  -2.353079  -2.352901  -2.350541  -2.349672
[241]  -2.348668  -2.344821  -2.342010  -2.337360  -2.337197  -2.335025  -2.334109  -2.331514
[249]  -2.327554  -2.323892  -2.322113  -2.313515  -2.310751  -2.310327  -2.303635  -2.301163
[257]  -2.300648  -2.284633  -2.284572  -2.280690  -2.278578  -2.276458  -2.275715  -2.273848
```

Console Terminal Jobs

C:/Users/HARIKA/Desktop/FinalExam/

```
[265]  -2.273658  -2.269766  -2.264377  -2.263637  -2.258887  -2.254217  -2.253746  -2.245991
[273]  -2.237231  -2.236575  -2.229205  -2.222723  -2.220122  -2.214399  -2.214279  -2.211549
[281]  -2.208445  -2.204393  -2.202724  -2.202538  -2.199290  -2.198450  -2.195617  -2.193336
[289]  -2.188330  -2.186805  -2.183482  -2.179789  -2.178207  -2.172393  -2.164521  -2.160172
[297]  -2.157135  -2.155098  -2.147461  -2.145406  -2.144240  -2.141593  -2.141167  -2.140562
[305]  -2.140216  -2.139240  -2.135110  -2.127599  -2.127568  -2.122789  -2.119759  -2.113375
[313]  -2.112363  -2.112051  -2.110150  -2.109150  -2.109041  -2.101997  -2.100789  -2.090457
[321]  -2.088662  -2.086063  -2.084470  -2.082259  -2.080431  -2.078630  -2.078050  -2.074608
[329]  -2.074393  -2.070827  -2.069752  -2.067025  -2.063743  -2.063599  -2.060653  -2.059792
[337]  -2.055001  -2.050744  -2.050249  -2.050089  -2.049803  -2.048347  -2.041657  -2.041018
[345]  -2.040519  -2.040395  -2.036953  -2.034210  -2.031073  -2.029085  -2.028215  -2.027990
[353]  -2.027671  -2.021070  -2.018450  -2.017346  -2.017248  -2.010556  -2.008954  -2.005927
[361]  -2.002606  -1.999111  -1.998408  -1.997850  -1.997753  -1.996078  -1.996026  -1.993359
[369]  -1.992441  -1.989272  -1.986004  -1.985714  -1.981962  -1.981853  -1.979894  -1.979827
[377]  -1.978391  -1.975982  -1.966884  -1.966734  -1.961525  -1.961065  -1.957340  -1.955891
[385]  -1.954133  -1.953115  -1.952269  -1.951794  -1.951151  -1.950793  -1.949270  -1.948289
[393]  -1.948132  -1.947252  -1.947186  -1.945522  -1.945130  -1.939791  -1.939206  -1.938260
[401]  -1.935022  -1.934854  -1.934798  -1.934331  -1.934159  -1.932068  -1.932019  -1.929389
[409]  -1.929244  -1.926881  -1.926881  -1.925799  -1.925296  -1.924362  -1.922173  -1.921647
[417]  -1.918480  -1.918416  -1.918408  -1.918356  -1.916937  -1.916703  -1.913335  -1.911848
[425]  -1.911835  -1.909280  -1.907751  -1.907410  -1.904674  -1.898521  -1.898075  -1.895925
[433]  -1.895630  -1.894876  -1.894200  -1.892910  -1.890615  -1.888794  -1.888638  -1.887758
[441]  -1.887094  -1.886756  -1.884893  -1.883530  -1.882396  -1.880390  -1.879953  -1.874938
[449]  -1.871919  -1.868939  -1.866530  -1.866400  -1.863759  -1.863122  -1.860843  -1.854617
[457]  -1.852985  -1.851970  -1.851153  -1.846586  -1.844696  -1.844105  -1.841249  -1.839831
[465]  -1.838526  -1.837285  -1.835881  -1.835676  -1.834271  -1.832638  -1.829098  -1.829009
[473]  -1.826599  -1.826165  -1.825414  -1.824602  -1.823778  -1.823689  -1.821642  -1.821596
[481]  -1.817089  -1.817063  -1.816375  -1.815387  -1.813911  -1.812627  -1.809846  -1.809389
[489]  -1.809321  -1.808453  -1.807588  -1.806031  -1.805627  -1.803628  -1.800252  -1.800113
[497]  -1.797638  -1.797328  -1.796657  -1.796045  -1.795155  -1.795026  -1.793689  -1.792634
[505]  -1.790243  -1.788926  -1.787648  -1.786159  -1.785647  -1.785547  -1.782764  -1.782600
[513]  -1.782269  -1.781308  -1.778204  -1.776350  -1.776129  -1.775836  -1.775794  -1.774665
[521]  -1.774332  -1.772895  -1.772890  -1.771901  -1.770915  -1.767164  -1.766146  -1.764316
[529]  -1.763321  -1.761793  -1.761371  -1.759055  -1.757380  -1.755788  -1.753846  -1.752487
[537]  -1.751292  -1.749371  -1.744232  -1.743462  -1.740775  -1.740302  -1.735656  -1.735043
```