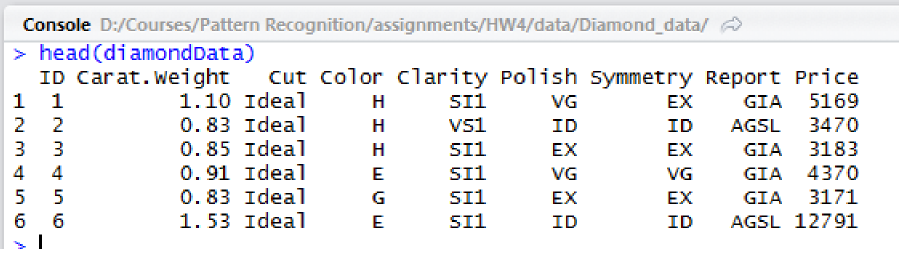
1.

library(gdata)
diamondData<-read.xls("Diamond_Data.xls", perl = "c:/Perl64/bin/perl.exe")

head(diamondData)

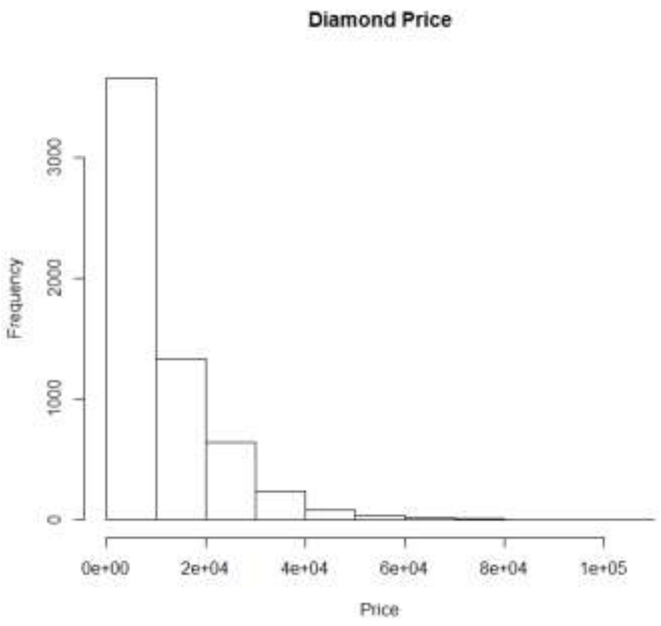

Console D:/Courses/Pattern Recognition/assignments/HW4/data/Diamond_data/

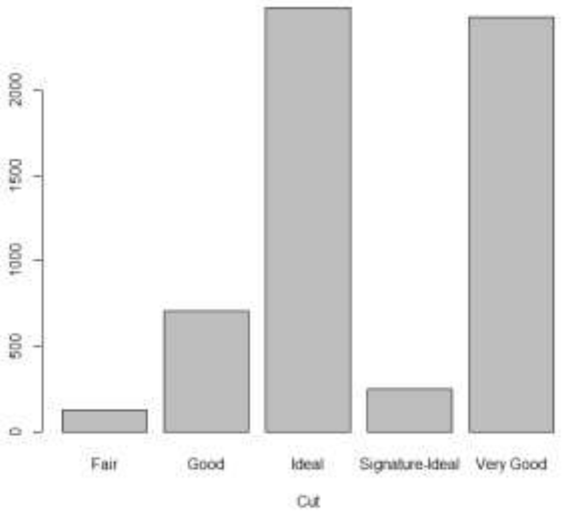
```
> head(diamondData)
  ID Carat.Weight   Cut Color Clarity Polish Symmetry Report Price
1  1         1.10 Ideal     H     SI1     VG       EX    GIA  5169
2  2         0.83 Ideal     H     VS1     ID       ID   AGSL  3470
3  3         0.85 Ideal     H     SI1     EX       EX    GIA  3183
4  4         0.91 Ideal     E     SI1     VG       VG    GIA  4370
5  5         0.83 Ideal     G     SI1     EX       EX    GIA  3171
6  6         1.53 Ideal     E     SI1     ID       ID   AGSL 12791
> |
```

#To install Rule Fit
platform = "windows"
rfhome = "C:/Program Files/R/RFHOME"
source("C:/Program Files/R/RFHOME/rulefit.r")
install.packages("akima", lib=rfhome)
library(akima, lib.loc=rfhome)
#end of rulefit install

1.a Show the distribution of the “cut” and “Price” attributes.

hist(diamondData$Price, main="Diamond Price", xlab="Price")
plot(diamondData$Cut , xlab="Cut")

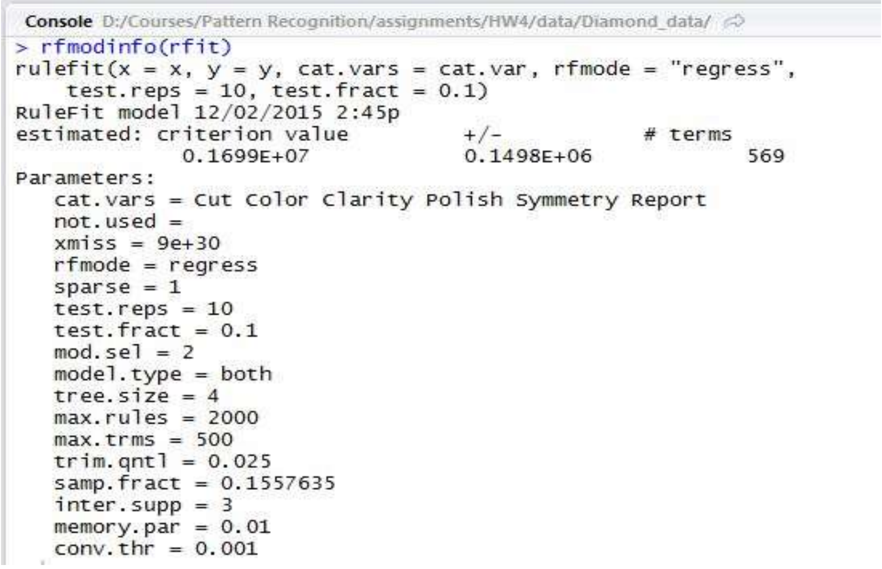




1.b

per <- floor(nrow(diamondData)*5/6)
subs <- sample(nrow(diamondData),per)
train <- diamondData[subs,]
x <- train[,2:8]
y <- train[,9]
cat.var <- c("Cut", "Color", "Clarity","Polish","Symmetry","Report")
rfit <- rulefit(x,y,rfmode="regress",cat.vars=cat.var,test.reps =10,test.fract=0.1)

rfmodinfo(rfit)


Console D:/Courses/Pattern Recognition/assignments/HW4/data/Diamond_data/

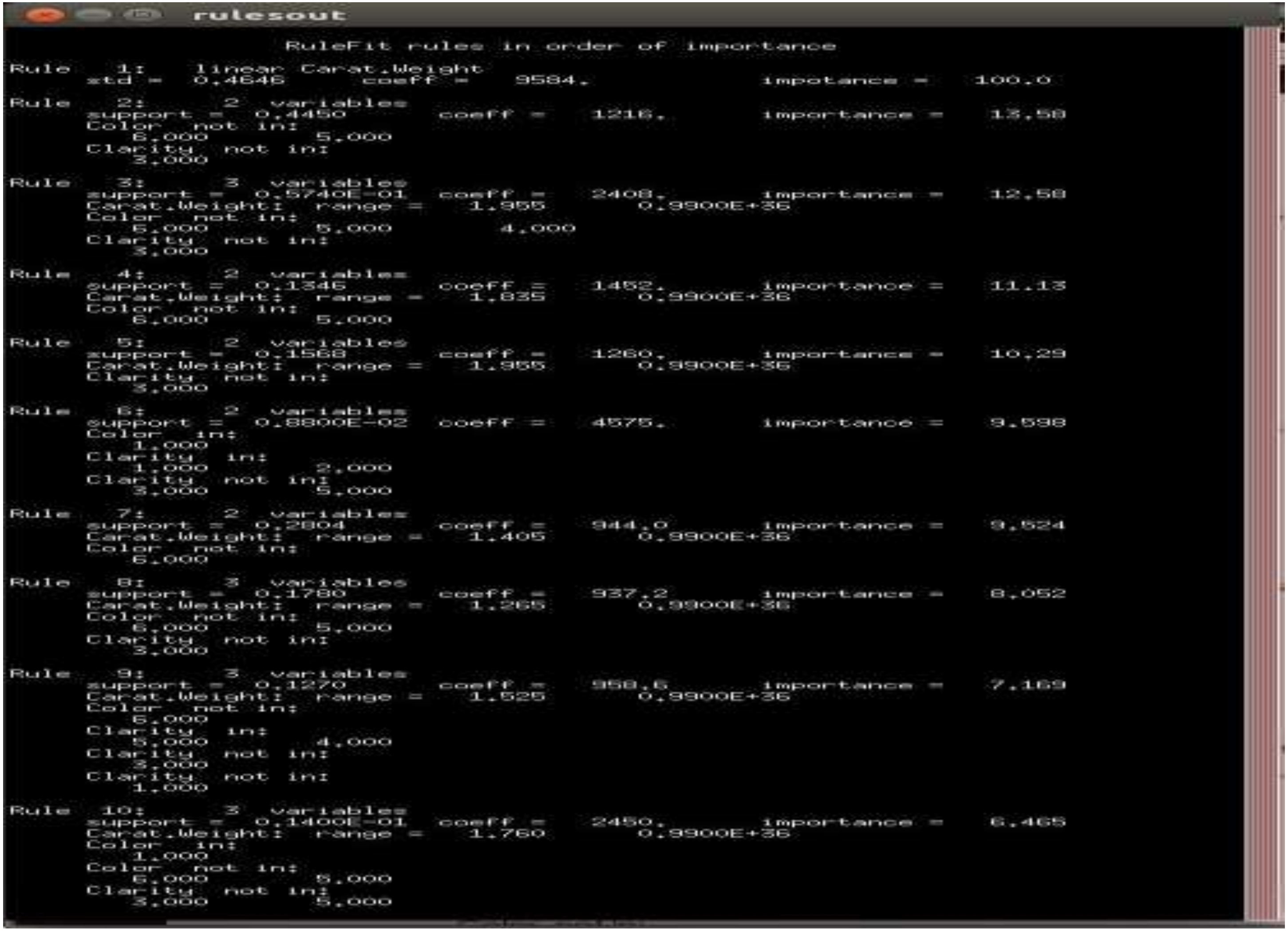
```
> rfmodinfo(rfit)
rulefit(x = x, y = y, cat.vars = cat.var, rfmode = "regress",
    test.reps = 10, test.fract = 0.1)
RuleFit model 12/02/2015 2:45p
estimated: criterion value          +/-           # terms
            0.1699E+07           0.1498E+06              569
Parameters:
   cat.vars = Cut Color Clarity Polish Symmetry Report
   not.used =
   xmiss = 9e+30
   rfmode = regress
   sparse = 1
   test.reps = 10
   test.fract = 0.1
   mod.sel = 2
   model.type = both
   tree.size = 4
   max.rules = 2000
   max.trms = 500
   trim.qntl = 0.025
   samp.fract = 0.1557635
   inter.supp = 3
   memory.par = 0.01
   conv.thr = 0.001
```

Average absolute error: 0.1498E+06

Number of terms: 569

1.c

rls <- rules(beg = 1, end = 10)



```
                         RuleFit rules in order of importance
Rule    1:    linear Carat.Weight
     std =   0.4646         coeff =    9584.              impotance =    100.0
Rule    2:       2  variables
     support =  0.4450          coeff =    1216.          importance =    13.58
     Color  not in:
        6.000        5.000
     Clarity  not in:
        3.000
Rule    3:       3  variables
     support =  0.5740E-01   coeff =    2408.         importance =    12.58
     Carat.Weight:  range =    1.955          0.9900E+36
     Color  not in:
        6.000        5.000          4.000
     Clarity  not in:
        3.000
Rule    4:       2  variables
     support =  0.1346          coeff =    1452.          importance =    11.13
     Carat.Weight:  range =    1.835          0.9900E+36
     Color  not in:
        6.000        5.000
Rule    5:       2  variables
     support =  0.1568          coeff =    1260.          importance =    10.29
     Carat.Weight:  range =    1.955          0.9900E+36
     Clarity  not in:
        3.000
Rule    6:       2  variables
     support =  0.8800E-02   coeff =    4575.          importance =    9.598
     Color   in:
        1.000
     Clarity   in:
        1.000        2.000
     Clarity  not in:
        3.000        5.000
Rule    7:       2  variables
     support =  0.2804          coeff =    944.0         importance =    9.524
     Carat.Weight:  range =    1.405          0.9900E+36
     Color  not in:
        6.000
Rule    8:       3  variables
     support =  0.1780          coeff =    937.2         importance =    8.052
     Carat.Weight:  range =    1.265          0.9900E+36
     Color  not in:
        6.000        5.000
     Clarity  not in:
        3.000
Rule    9:       3  variables
     support =  0.1270          coeff =    958.6         importance =    7.169
     Carat.Weight:  range =    1.525          0.9900E+36
     Color  not in:
        6.000
     Clarity   in:
        5.000        4.000
     Clarity  not in:
        3.000
     Clarity  not in:
        1.000
Rule   10:       3  variables
     support =  0.1400E-01   coeff =    2450.          importance =    6.465
     Carat.Weight:  range =    1.760          0.9900E+36
     Color   in:
        1.000
     Color  not in:
        6.000        5.000
     Clarity  not in:
        3.000        5.000
```

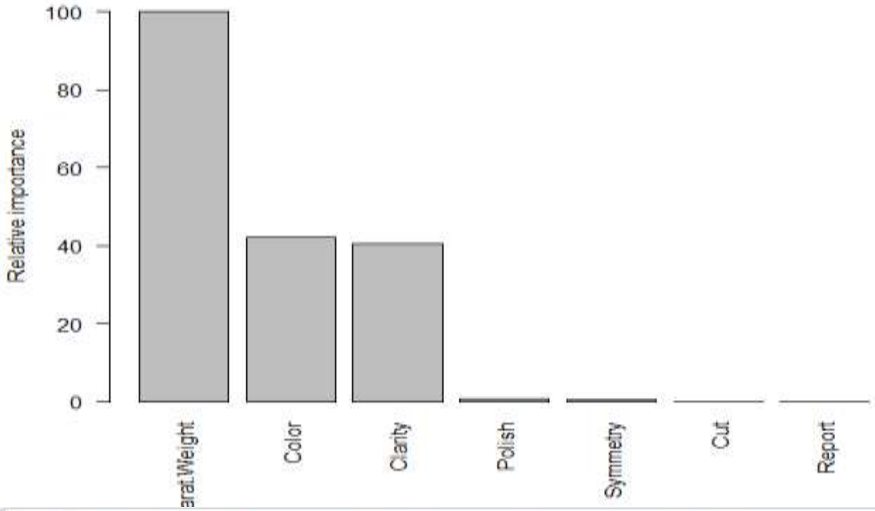
After observing the above output,

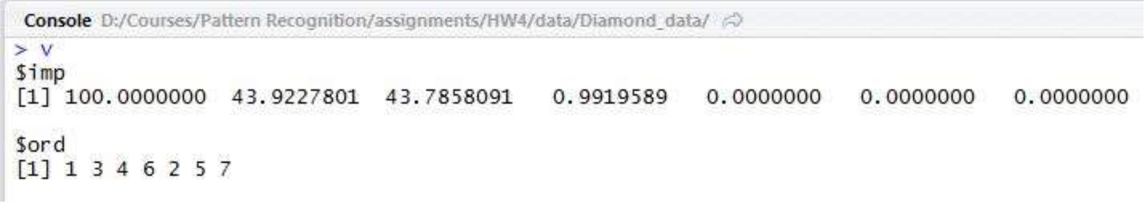
i. There was a liner term in the model ‘carat.weight’. The coefficient is 9584 which suggests increase in the weight. The price will increase by $9584

ii. Rule 3 is based on 3 variables and has importance of 12.58. The rule indicates that if the carat.weight is in the range of 1.955 and color is not in 6, 5, 4 and clarity is not 3 then price will increase by 2408$

1.d.

```
v <- varimp(range = 1:nrow(diamondData[-subs,]),x=diamondData[-subs,][,2:8],wt=rep(1,nrow(diamondData[-subs,])),rth=0,col='grey', donames=T, las=2)
```



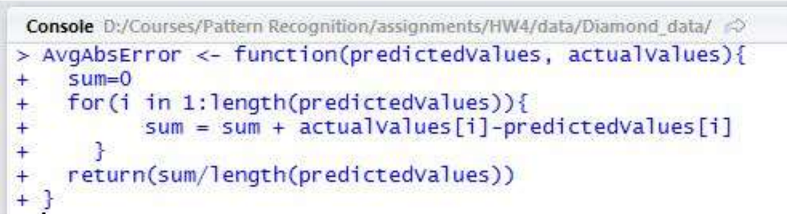

Console D:/Courses/Pattern Recognition/assignments/HW4/data/Diamond_data/

```
> v
$imp
[1] 100.0000000  43.9227801  43.7858091   0.9919589   0.0000000   0.0000000   0.0000000

$ord
[1] 1 3 4 6 2 5 7
```

As seen from the plot the top three most important variables in determining the diamond’s price are: Carat.Weight, Color and Clarity

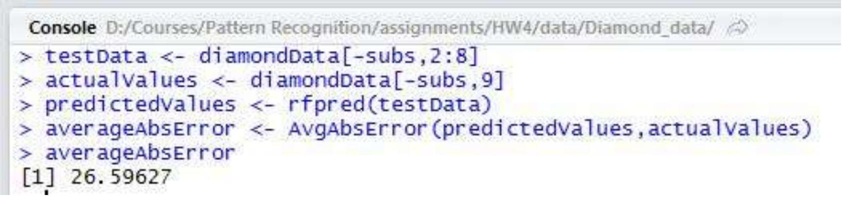
1.e.

```
AvgAbsError <- function(predictedValues, actualValues){
 sum=0
 for(i in 1:length(predictedValues)){
      sum = sum + actualValues[i]-predictedValues[i]
   }
 return(sum/length(predictedValues))
}
```


Console D:/Courses/Pattern Recognition/assignments/HW4/data/Diamond_data/

```
> AvgAbsError <- function(predictedValues, actualValues){
+   sum=0
+   for(i in 1:length(predictedValues)){
+         sum = sum + actualValues[i]-predictedValues[i]
+     }
+   return(sum/length(predictedValues))
+ }
```

```
testData <- diamondData[-subs,2:8]
actualValues <- diamondData[-subs,9]
predictedValues <- rfpred(testData)
averageAbsError <- AvgAbsError(predictedValues,actualValues)
averageAbsError
```



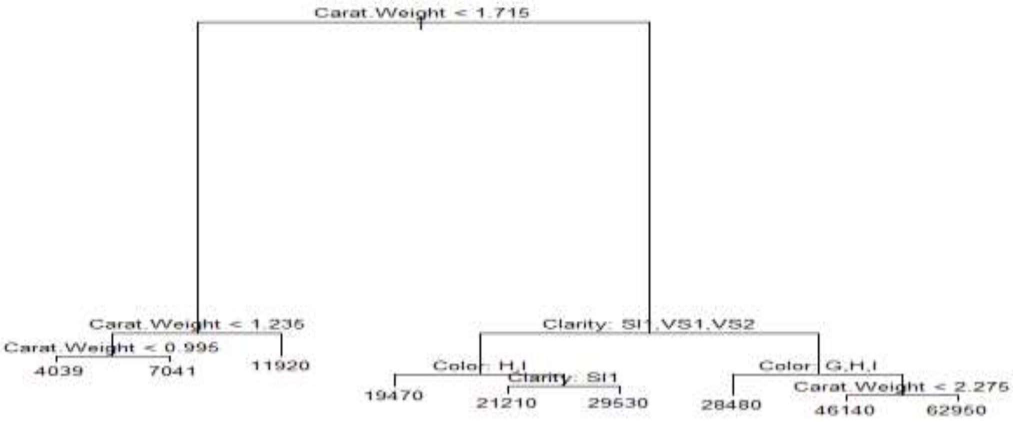
Console D:/Courses/Pattern Recognition/assignments/HW4/data/Diamond_data/

```
> testData <- diamondData[-subs,2:8]
> actualValues <- diamondData[-subs,9]
> predictedValues <- rfpred(testData)
> averageAbsError <- AvgAbsError(predictedValues,actualValues)
> averageAbsError
[1] 26.59627
```

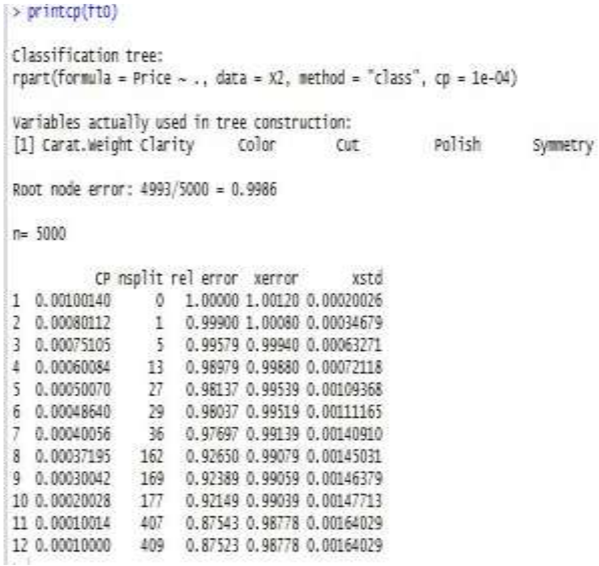
Average Absolute error: 26.59627

1.f.

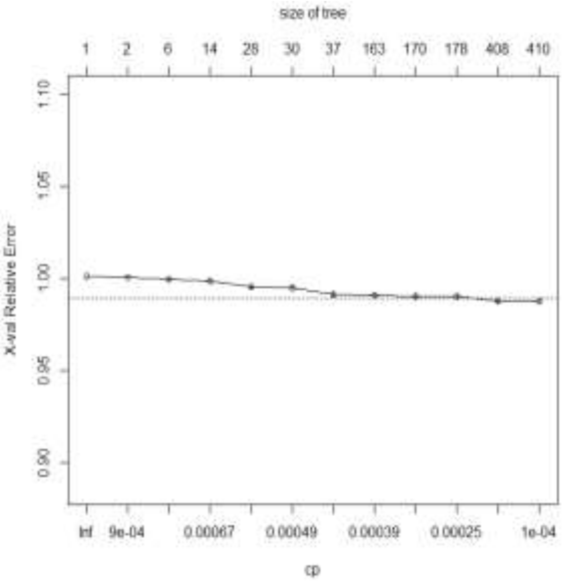
```
x<- train[,2:9]
X1<-as(x,"data.frame")
X2<-as(X1,"data.frame")
install.packages("tree")
library(tree)
dtr<-tree(Price~.,X1)
plot(dtr)
text(dtr, cex=.8,pretty=0)
```



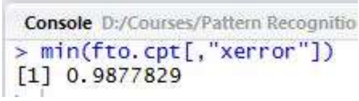
```
install.packages("rpart")
library(rpart)

ftO<-rpart(Price ~ .,data=X2,method="class",cp=0.0001)
printcp(ftO)
plotcp(ftO)
```



```
> printcp(ft0)

Classification tree:
rpart(formula = Price ~ ., data = X2, method = "class", cp = 1e-04)

Variables actually used in tree construction:
[1] Carat.Weight Clarity      Color        Cut          Polish       Symmetry

Root node error: 4993/5000 = 0.9986

n= 5000

          CP nsplit rel error  xerror       xstd
1  0.00100140      0   1.00000 1.00120 0.00020026
2  0.00080112      1   0.99900 1.00080 0.00034679
3  0.00075105      5   0.99579 0.99940 0.00063271
4  0.00060084     13   0.98979 0.99880 0.00072118
5  0.00050070     27   0.98137 0.99539 0.00109368
6  0.00048640     29   0.98037 0.99519 0.00111165
7  0.00040056     36   0.97697 0.99139 0.00140910
8  0.00037195    162   0.92650 0.99079 0.00145031
9  0.00030042    169   0.92389 0.99059 0.00146379
10 0.00020028    177   0.92149 0.99039 0.00147713
11 0.00010014    407   0.87543 0.98778 0.00164029
12 0.00010000    409   0.87523 0.98778 0.00164029
```

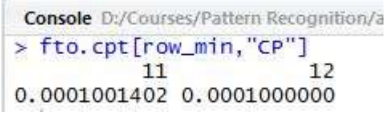


fto.cpt<-ftO$cptable
min(fto.cpt[,"xerror"])



```
> min(fto.cpt[,"xerror"])
[1] 0.9877829
```

row_min<-which(fto.cpt[,"xerror"]==min(fto.cpt[,"xerror"]))
fto.cpt[row_min,"CP"]



```
> fto.cpt[row_min,"CP"]
          11           12 
0.0001001402 0.0001000000 
```

best_cp<-fto.cpt[row_min,"CP"]
dtree<-prune(ftO, best_cp[1])

Error: 0.9877829

2.

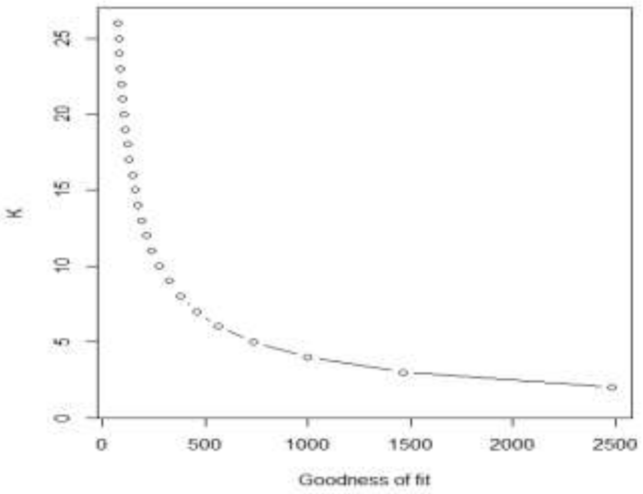
2.a

```
letters<-read.table("az-5000.txt", header=FALSE)
#not selecting first column
letters<-letters[2:5001,2:19]
letters.3means<-kmeans(letters,centers=26)
totwithinss=c()
for(i in 2:26){
  totwithinss[i]=(kmeans(letters,i)$tot.withinss)/i
}
```

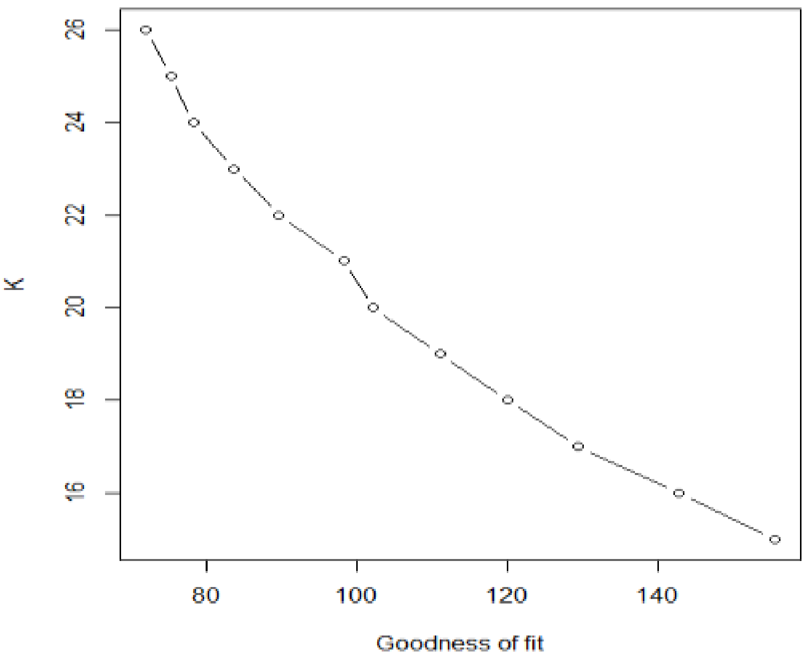
K	J(C)
2	2486.86724
3	1462.92239
4	1001.30631
5	753.04704
6	582.54581
7	456.87470
8	377.80740
9	318.00262
10	280.63136
11	240.86812
12	218.09774
13	192.81951
14	173.09759
15	155.23682
16	142.99360
17	129.09894
18	118.23988
19	111.03241
20	103.47979
21	95.24959
22	88.96936
23	84.98184
24	80.69763
25	75.83029
26	70.09717

2.b

```
plot(totwithinss,1:26, type = "b", xlab = "Goodness of fit", ylab = "K")
```



plot(totwithinss[15:26],15:26, type = "b", xlab = "Goodness of fit", ylab = "K")

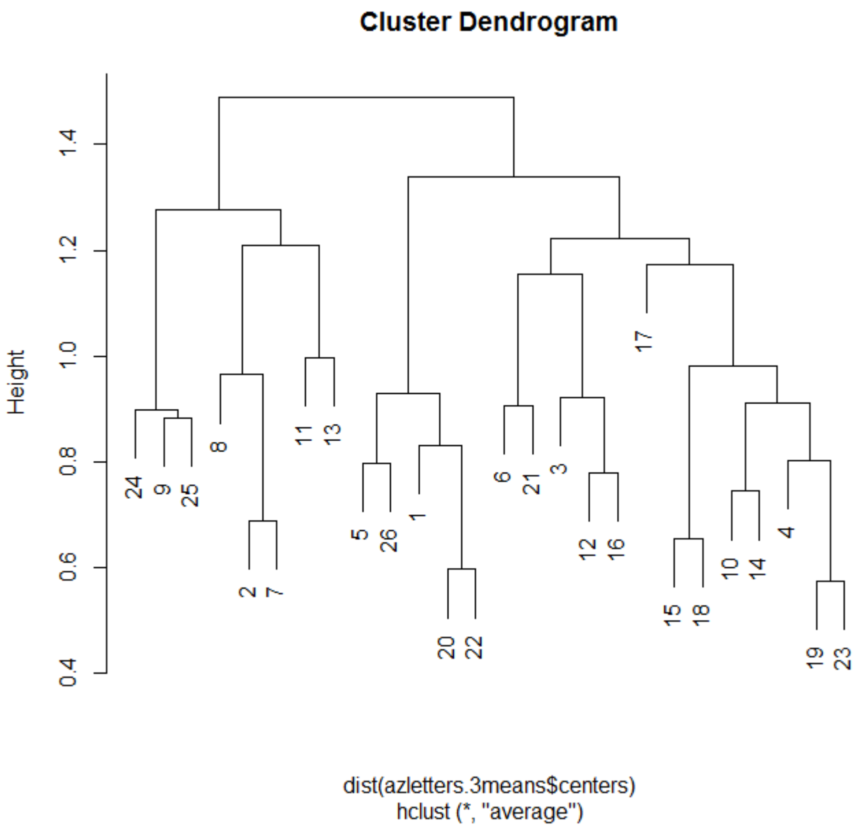


The above plot shows the goodness of fit for K=15 to 26. As seen from the plot above, As the number of clusters increases, the value of withinss decreases. In the above plot, at k = 20, there is a step. This indicates the 20th letter ‘T’ might suggest the number of natural clusters.

3

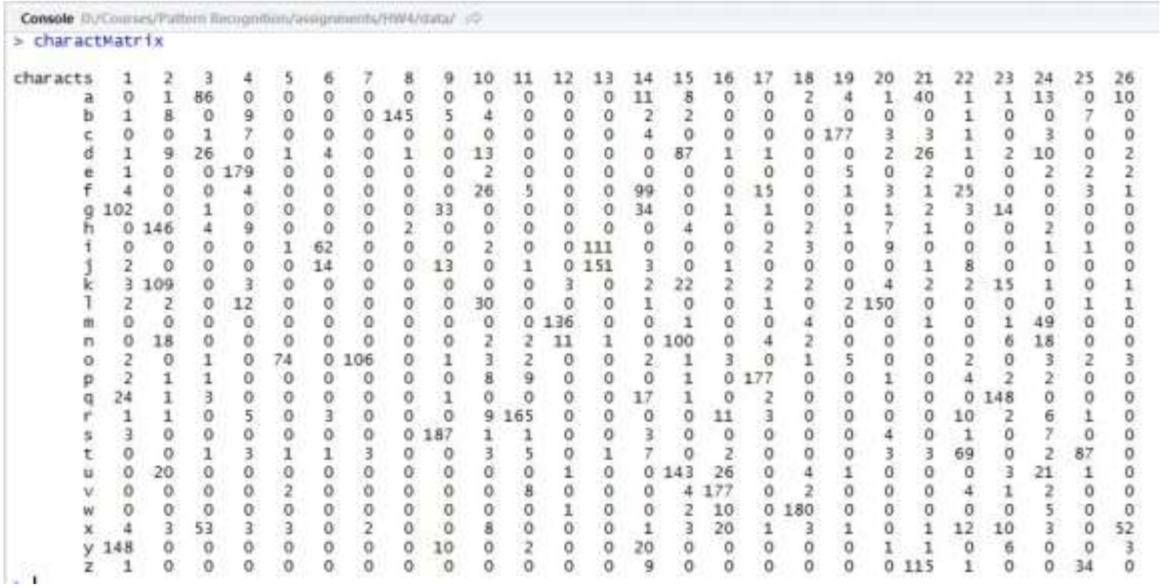
3.a.

```
data <- read.table("az-5000.txt", header=TRUE)
azletters<-read.table("az-5000.txt", header=TRUE)
azletters<-azletters[,-1]
azletters.3means<-kmeans(azletters,centers=26, nstart = 22)
hc <- hclust(dist(azletters.3means$centers), method = "average")
plot(hc)
```



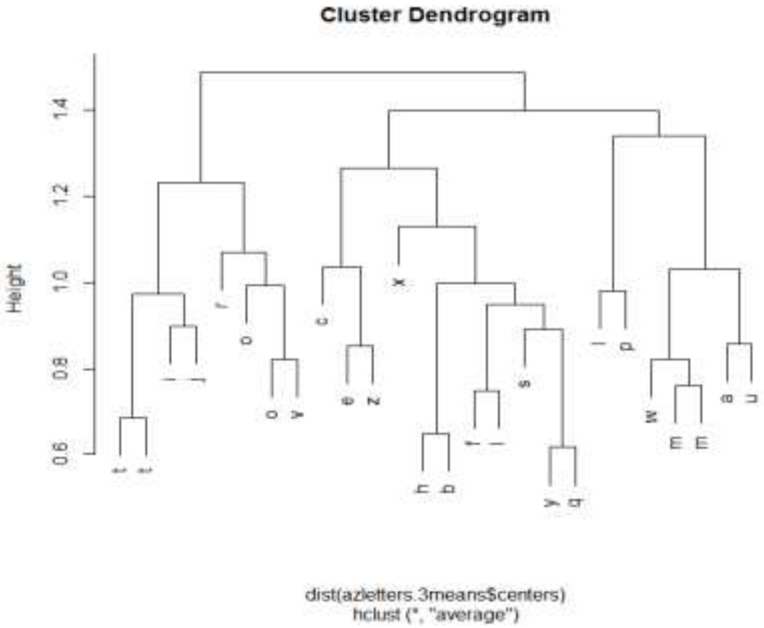
3.b.

```
characts <- data[,1]
charactMatrix <- table(characts,azletters.3means$cluster)
maxCharsArray <- array(26)
for(i in 1:26){
  maxCharsArray[i] <- letters[which.max(charactMatrix[,i])]
}
```



```
Console D:/Courses/Pattern Recognition/assignments/HW4/data/
> charactMatrix

characts   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26
       a   0   1  86   0   0   0   0   0   0   0   0   0   0  11   8   0   0   2   4   1  40   1   1  13   0  10
       b   1   8   0   9   0   0   0 145   5   4   0   0   0   2   2   0   0   0   0   0   0   1   0   0   7   0
       c   0   0   1   7   0   0   0   0   0   0   0   0   0   4   0   0   0   0 177   3   3   1   0   3   0   0
       d   1   9  26   0   1   4   0   1   0  13   0   0   0   0  87   1   1   0   0   2  26   1   2  10   0   2
       e   1   0   0 179   0   0   0   0   0   2   0   0   0   0   0   0   0   0   5   0   2   0   0   2   2   2
       f   4   0   0   4   0   0   0   0   0  26   5   0   0  99   0   0  15   0   1   3   1  25   0   0   3   1
       g 102   0   1   0   0   0   0   0  33   0   0   0   0  34   0   1   1   0   0   1   2   3  14   0   0   0
       h   0 146   4   9   0   0   0   2   0   0   0   0   0   0   4   0   0   2   1   7   1   0   0   2   0   0
       i   0   0   0   0   1  62   0   0   0   2   0   0 111   0   0   0   2   3   0   9   0   0   0   1   1   0
       j   2   0   0   0   0  14   0   0  13   0   1   0 151   3   0   1   0   0   0   0   1   8   0   0   0   0
       k   3 109   0   3   0   0   0   0   0   0   0   3   0   2  22   2   2   2   0   4   2   2  15   1   0   1
       l   2   2   0  12   0   0   0   0   0  30   0   0   0   1   0   0   1   0   2 150   0   0   0   0   1   1
       m   0   0   0   0   0   0   0   0   0   0   0 136   0   0   1   0   0   4   0   0   1   0   1  49   0   0
       n   0  18   0   0   0   0   0   0   0   2   2  11   1   0 100   0   4   2   0   0   0   0   6  18   0   0
       o   2   0   1   0  74   0 106   0   1   3   2   0   0   2   1   3   0   1   5   0   0   2   0   3   2   3
       p   2   1   1   0   0   0   0   0   0   8   9   0   0   0   1   0 177   0   0   1   0   4   2   2   0   0
       q  24   1   3   0   0   0   0   0   1   0   0   0   0  17   1   0   2   0   0   0   0   0 148   0   0   0
       r   1   1   0   5   0   3   0   0   0   9 165   0   0   0   0  11   3   0   0   0   0  10   2   6   1   0
       s   3   0   0   0   0   0   0   0 187   1   1   0   0   3   0   0   0   0   0   4   0   1   0   7   0   0
       t   0   0   1   3   1   1   3   0   0   3   5   0   1   7   0   2   0   0   0   3   3  69   0   2  87   0
       u   0  20   0   0   0   0   0   0   0   0   0   1   0   0 143  26   0   4   1   0   0   0   3  21   1   0
       v   0   0   0   0   2   0   0   0   0   0   8   0   0   0   4 177   0   2   0   0   0   4   1   2   0   0
       w   0   0   0   0   0   0   0   0   0   0   0   1   0   0   2  10   0 180   0   0   0   0   0   5   0   0
       x   4   3  53   3   3   0   2   0   0   8   0   0   0   1   3  20   1   3   1   0   1  12  10   3   0  52
       y 148   0   0   0   0   0   0   0  10   0   2   0   0  20   0   0   0   0   0   1   1   0   6   0   0   3
       z   1   0   0   0   0   0   0   0   0   0   0   0   0   9   0   0   0   0   0   0 115   1   0   0  34   0
```

```
hc$labels <- maxCharsArray
plot(hc)
```



The most common letters in the clusters are ‘n’ and ‘u’.

3.c.

i. From the dendrogram it can be observed that, deeper the location of an item in the dendogram, higher is its frequency of occurrence in the clusters.

ii. Similar letters are present together in the dendogram

3.d.

The missing letters are: d, n, g, k.

The missing letters should be assigned to the cluster in which they have maximum occurrence.

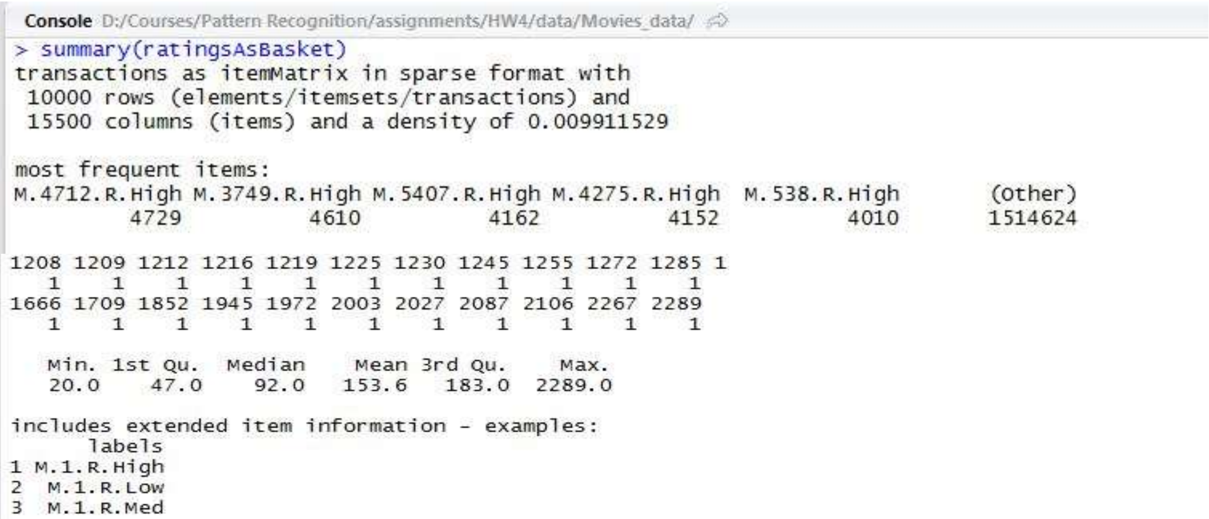
Letter	Cluster
n	15
G	1
d	15
k	2

4.

require(arules)

4.a.

ratingsAsBasket <- read.transactions("ratingsAsBasket.txt", format = "basket")
summary(ratingsAsBasket)



```
Console D:/Courses/Pattern Recognition/assignments/HW4/data/Movies_data/
> summary(ratingsAsBasket)
transactions as itemMatrix in sparse format with
 10000 rows (elements/itemsets/transactions) and
 15500 columns (items) and a density of 0.009911529

most frequent items:
M.4712.R.High M.3749.R.High M.5407.R.High M.4275.R.High  M.538.R.High       (Other)
         4729          4610          4162          4152          4010       1514624

1208 1209 1212 1216 1219 1225 1230 1245 1255 1272 1285 1
   1    1    1    1    1    1    1    1    1    1    1
1666 1709 1852 1945 1972 2003 2027 2087 2106 2267 2289
   1    1    1    1    1    1    1    1    1    1    1

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   20.0    47.0    92.0   153.6   183.0  2289.0

includes extended item information - examples:
       labels
1 M.1.R.High
2   M.1.R.Low
3   M.1.R.Med
```

Number of baskets: 10000

Most frequent item: M.4712.R.High

Title: The Matrix (4712)

Rating: High

Frequency: 4729

Number of movies rated by one rater:

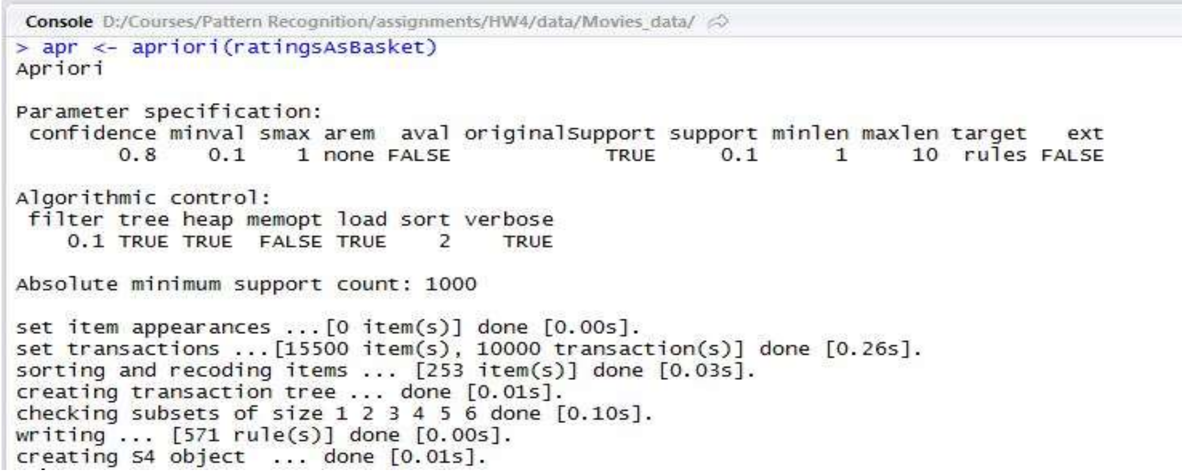
Minimum: 20.0

Maximum: 2289.0

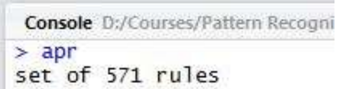
Average: 92.0

4.b.

```
apr <- apriori(ratingsAsBasket)
```

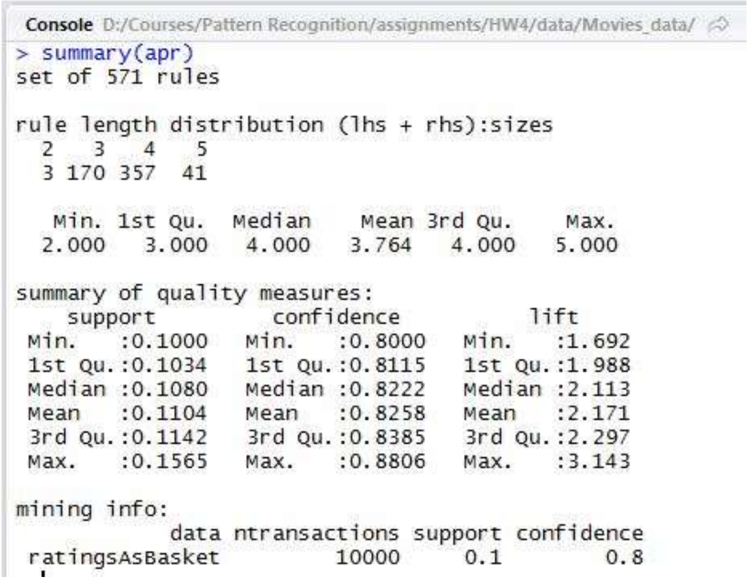


```
Console D:/Courses/Pattern Recognition/assignments/HW4/data/Movies_data/
> apr <- apriori(ratingsAsBasket)
Apriori

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

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

Absolute minimum support count: 1000

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[15500 item(s), 10000 transaction(s)] done [0.26s].
sorting and recoding items ... [253 item(s)] done [0.03s].
creating transaction tree ... done [0.01s].
checking subsets of size 1 2 3 4 5 6 done [0.10s].
writing ... [571 rule(s)] done [0.00s].
creating S4 object  ... done [0.01s].
```



```
Console D:/Courses/Pattern Recogni
> apr
set of 571 rules
```

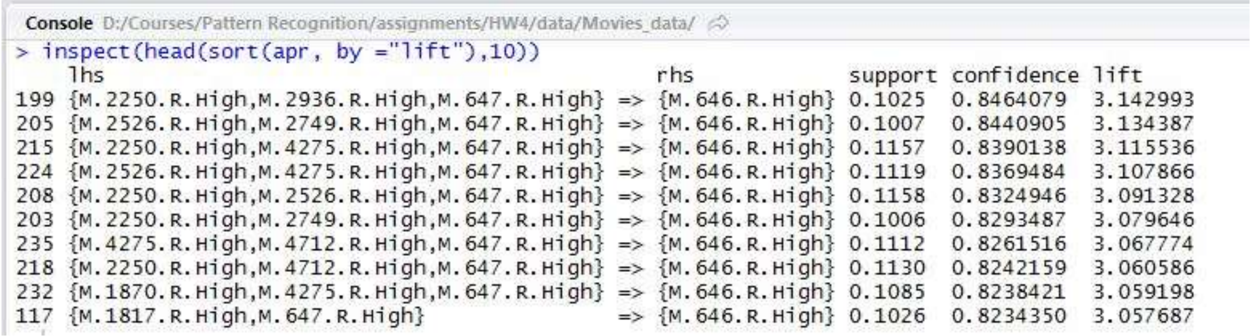
Summary(apr)


Console D:/Courses/Pattern Recognition/assignments/HW4/data/Movies_data/

```
> summary(apr)
set of 571 rules

rule length distribution (lhs + rhs):sizes
  2   3   4   5
  3 170 357  41

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   3.000   4.000   3.764   4.000   5.000

summary of quality measures:
    support         confidence          lift
 Min.   :0.1000   Min.   :0.8000   Min.   :1.692
 1st Qu.:0.1034   1st Qu.:0.8115   1st Qu.:1.988
 Median :0.1080   Median :0.8222   Median :2.113
 Mean   :0.1104   Mean   :0.8258   Mean   :2.171
 3rd Qu.:0.1142   3rd Qu.:0.8385   3rd Qu.:2.297
 Max.   :0.1565   Max.   :0.8806   Max.   :3.143

mining info:
            data ntransactions support confidence
 ratingsAsBasket         10000     0.1        0.8
```

Top 10 rules: With respect to 'lift' measure

inspect(head(sort(apr, by ="lift"),10))


Console D:/Courses/Pattern Recognition/assignments/HW4/data/Movies_data/

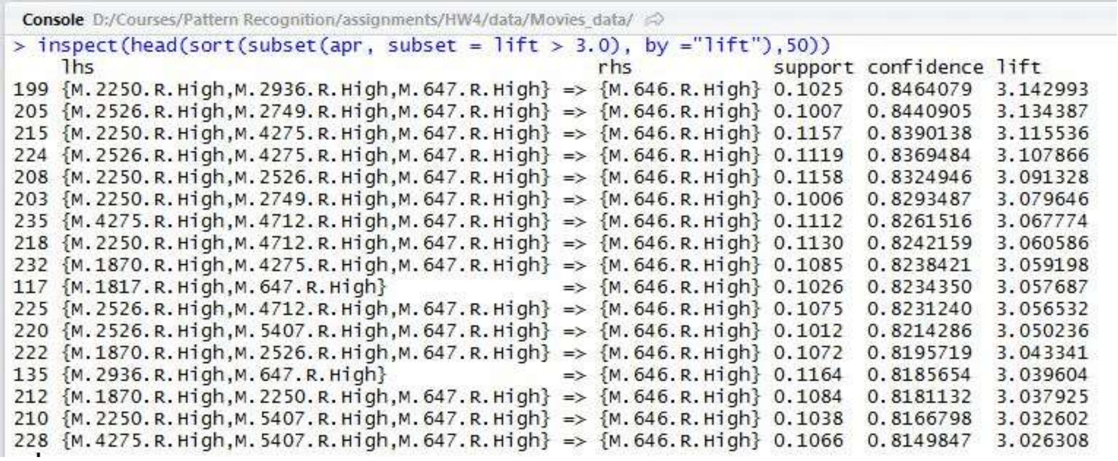
```
> inspect(head(sort(apr, by ="lift"),10))
    lhs                                             rhs              support confidence lift
199 {M.2250.R.High,M.2936.R.High,M.647.R.High} => {M.646.R.High} 0.1025  0.8464079  3.142993
205 {M.2526.R.High,M.2749.R.High,M.647.R.High} => {M.646.R.High} 0.1007  0.8440905  3.134387
215 {M.2250.R.High,M.4275.R.High,M.647.R.High} => {M.646.R.High} 0.1157  0.8390138  3.115536
224 {M.2526.R.High,M.4275.R.High,M.647.R.High} => {M.646.R.High} 0.1119  0.8369484  3.107866
208 {M.2250.R.High,M.2526.R.High,M.647.R.High} => {M.646.R.High} 0.1158  0.8324946  3.091328
203 {M.2250.R.High,M.2749.R.High,M.647.R.High} => {M.646.R.High} 0.1006  0.8293487  3.079646
235 {M.4275.R.High,M.4712.R.High,M.647.R.High} => {M.646.R.High} 0.1112  0.8261516  3.067774
218 {M.2250.R.High,M.4712.R.High,M.647.R.High} => {M.646.R.High} 0.1130  0.8242159  3.060586
232 {M.1870.R.High,M.4275.R.High,M.647.R.High} => {M.646.R.High} 0.1085  0.8238421  3.059198
117 {M.1817.R.High,M.647.R.High}               => {M.646.R.High} 0.1026  0.8234350  3.057687
```

From the above figure we can interpret that, If the user rates Movies 2526 (The Fugitive), 2749 (The hunt for red october) and 647 (Terminator 2: Judgement Day) as high then he will also prefer movie 646 (The Terminator) and rate it 'high'.

4.c.

lift: The strength of a rule is indicated by lift. It is indicated over a co-occurrence of antecedent and consequent. It gives the details of the improvement i.e increase in probability of the consequent for a given antecedent.
(Rule Support) /(Support(Antecedent) * Support(Consequent))

Use subset command to list all the rules with lift > 3.0
inspect(sort(subset(apr, subset = lift > 3), by ="lift"))



```
Console D:/Courses/Pattern Recognition/assignments/HW4/data/Movies_data/
> inspect(head(sort(subset(apr, subset = lift > 3.0), by ="lift"),50))
    lhs                                            rhs              support confidence lift
199 {M.2250.R.High,M.2936.R.High,M.647.R.High} => {M.646.R.High} 0.1025  0.8464079  3.142993
205 {M.2526.R.High,M.2749.R.High,M.647.R.High} => {M.646.R.High} 0.1007  0.8440905  3.134387
215 {M.2250.R.High,M.4275.R.High,M.647.R.High} => {M.646.R.High} 0.1157  0.8390138  3.115536
224 {M.2526.R.High,M.4275.R.High,M.647.R.High} => {M.646.R.High} 0.1119  0.8369484  3.107866
208 {M.2250.R.High,M.2526.R.High,M.647.R.High} => {M.646.R.High} 0.1158  0.8324946  3.091328
203 {M.2250.R.High,M.2749.R.High,M.647.R.High} => {M.646.R.High} 0.1006  0.8293487  3.079646
235 {M.4275.R.High,M.4712.R.High,M.647.R.High} => {M.646.R.High} 0.1112  0.8261516  3.067774
218 {M.2250.R.High,M.4712.R.High,M.647.R.High} => {M.646.R.High} 0.1130  0.8242159  3.060586
232 {M.1870.R.High,M.4275.R.High,M.647.R.High} => {M.646.R.High} 0.1085  0.8238421  3.059198
117 {M.1817.R.High,M.647.R.High}               => {M.646.R.High} 0.1026  0.8234350  3.057687
225 {M.2526.R.High,M.4712.R.High,M.647.R.High} => {M.646.R.High} 0.1075  0.8231240  3.056532
220 {M.2526.R.High,M.5407.R.High,M.647.R.High} => {M.646.R.High} 0.1012  0.8214286  3.050236
222 {M.1870.R.High,M.2526.R.High,M.647.R.High} => {M.646.R.High} 0.1072  0.8195719  3.043341
135 {M.2936.R.High,M.647.R.High}               => {M.646.R.High} 0.1164  0.8185654  3.039604
212 {M.1870.R.High,M.2250.R.High,M.647.R.High} => {M.646.R.High} 0.1084  0.8181132  3.037925
210 {M.2250.R.High,M.5407.R.High,M.647.R.High} => {M.646.R.High} 0.1038  0.8166798  3.032602
228 {M.4275.R.High,M.5407.R.High,M.647.R.High} => {M.646.R.High} 0.1066  0.8149847  3.026308
```

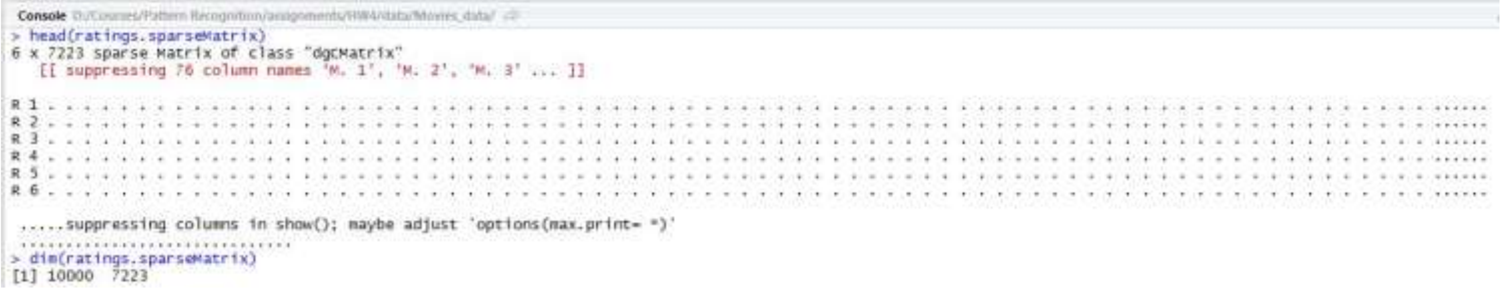
From the above figure we can interpret that, If the user rates Movies 2250 (Die Hard), 2936 (Lethal Weapon) and 647 (Terminator 2: Judgement Day) as high then he will also prefer movie 646 (The Terminator) and rate it 'high'.

5.

```
require(recommenderlab)
```

5.a.

```
ratings <- scan("ratings.txt", what="list", sep = "|")

ratings.matrix <- matrix(ratings, ncol=3,byrow=T)
class(ratings.matrix)<-"numeric"
ratings.sparseMatrix<-sparseMatrix(i=ratings.matrix[,1],j=ratings.matrix[,2],x=ratings.matrix[,3])
dimnames(ratings.sparseMatrix)<-list(user=paste("U", 1:10000),Movie=paste("M.",1:7223))
dim(ratings.sparseMatrix)
```

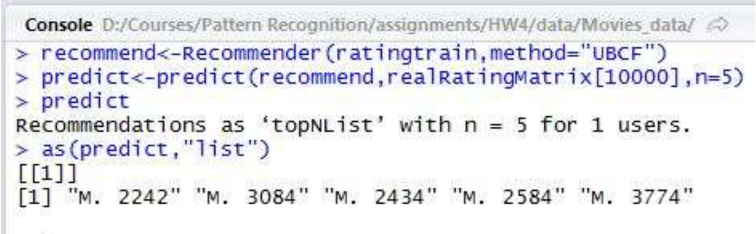


```
Console D:/Courses/Pattern Recognition/assignments/HW4/data/Movies_data/
> head(ratings.sparseMatrix)
6 x 7223 sparse Matrix of class "dgCMatrix"
   [[ suppressing 76 column names 'M. 1', 'M. 2', 'M. 3' ... ]]

R 1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ......
R 2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ......
R 3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ......
R 4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ......
R 5 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ......
R 6 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ......

 .....suppressing columns in show(); maybe adjust 'options(max.print= *)'
 ..............................
> dim(ratings.sparseMatrix)
[1] 10000  7223
```

Dimensions of the sparse matrix are: 10000 7223

5.b

```
realRatingMatrix <- new("realRatingMatrix",data=ratings.sparseMatrix)

realRatingMatrix.split<-floor(nrow(realRatingMatrix)*0.8)
realRatingMatrix.split.sampled<-sample(nrow(realRatingMatrix),realRatingMatrix.split)
ratingtrain<-realRatingMatrix[realRatingMatrix.split.sampled,]

recommend<-Recommender(ratingtrain,method="UBCF")
predict<-predict(recommend,realRatingMatrix[10000],n=10)
predict
as(predict,"list")
```



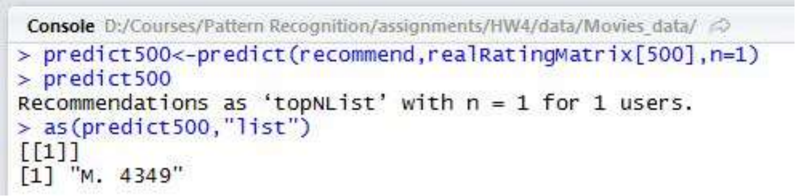
```
Console D:/Courses/Pattern Recognition/assignments/HW4/data/Movies_data/
> recommend<-Recommender(ratingtrain,method="UBCF")
> predict<-predict(recommend,realRatingMatrix[10000],n=5)
> predict
Recommendations as 'topNList' with n = 5 for 1 users.
> as(predict,"list")
[[1]]
[1] "M. 2242" "M. 3084" "M. 2434" "M. 2584" "M. 3774"
```

Top 5 Movie recommendations for user #10000 are:
2242 : Crying Game
3084 : Mission Impossible
2434 : Fargo
2584 : Gone with the wind
3774 : Rain man

5.c.

```
predict500<-predict(recommend,realRatingMatrix[500],n=1)
predict500
```

as(predict500,"list")


Console D:/Courses/Pattern Recognition/assignments/HW4/data/Movies_data/

```
> predict500<-predict(recommend,realRatingMatrix[500],n=1)
> predict500
Recommendations as ‘topNList’ with n = 1 for 1 users.
> as(predict500,"list")
[[1]]
[1] "M. 4349"
```

Highest predicted rating movie for the user #500:
4349 : Good will hunting

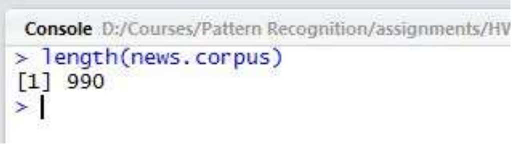
6.

require(tm)

6.a.

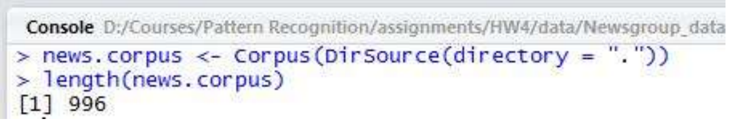
autosData <- DirSource(directory = ".")
news.corpus <- Corpus(DirSource(directory = "."))
length(news.corpus)

folder: ‘rec.autos



```
> length(news.corpus)
[1] 990
> 
```

folder: ‘rec.motorcycles’


Console D:/Courses/Pattern Recognition/assignments/HW4/data/Newsgroup_data

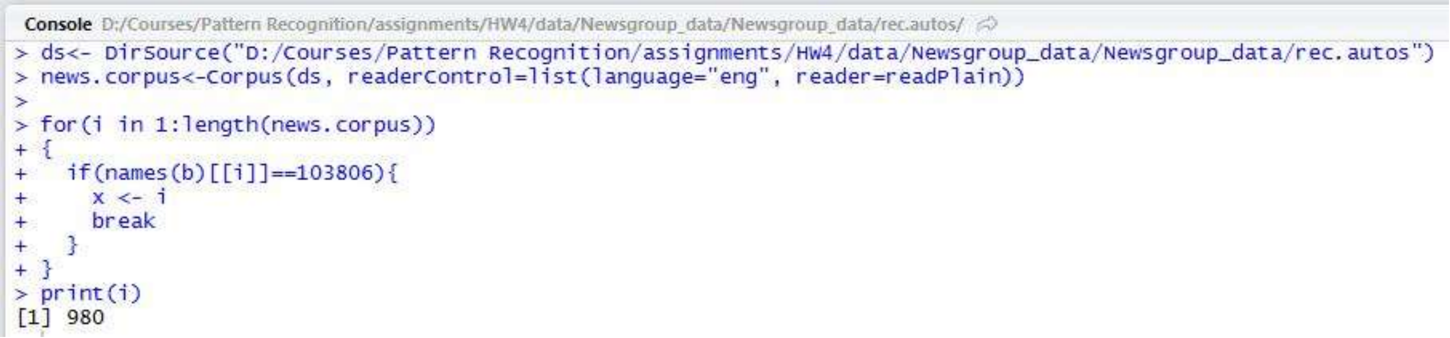
```
> news.corpus <- Corpus(DirSource(directory = "."))
> length(news.corpus)
[1] 996
```

To print the corpus entry corresponding to rec.autos/103806:

ds<- DirSource("D:/Courses/Pattern Recognition/assignments/HW4/data/Newsgroup_data/Newsgroup_data/rec.autos")
news.corpus<-Corpus(ds, readerControl=list(language="eng", reader=readPlain))

```
for(i in 1:length(news.corpus))
{
  if(names(b)[[i]]==103806){
    x <- i
    break
  }
}
print(i)
```

Output: 980

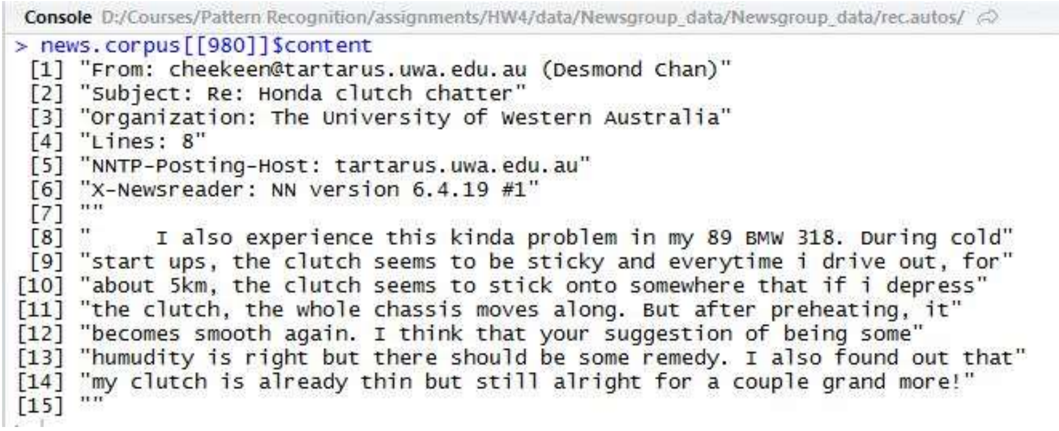

Console D:/Courses/Pattern Recognition/assignments/HW4/data/Newsgroup_data/Newsgroup_data/rec.autos/

```
> ds<- DirSource("D:/Courses/Pattern Recognition/assignments/HW4/data/Newsgroup_data/Newsgroup_data/rec.autos")
> news.corpus<-Corpus(ds, readerControl=list(language="eng", reader=readPlain))
>
> for(i in 1:length(news.corpus))
+ {
+   if(names(b)[[i]]==103806){
+     x <- i
+     break
+   }
+ }
> print(i)
[1] 980
```

6.b.

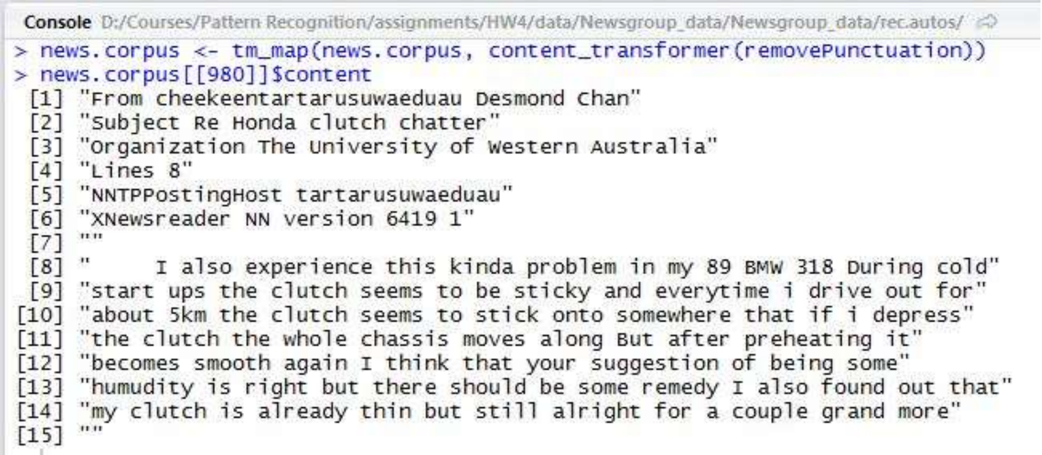
Initial file 103806

news.corpus[[980]]$content # 103806 file is located in the location 980 in the corpus according to the previous solution.


Console D:/Courses/Pattern Recognition/assignments/HW4/data/Newsgroup_data/Newsgroup_data/rec.autos/

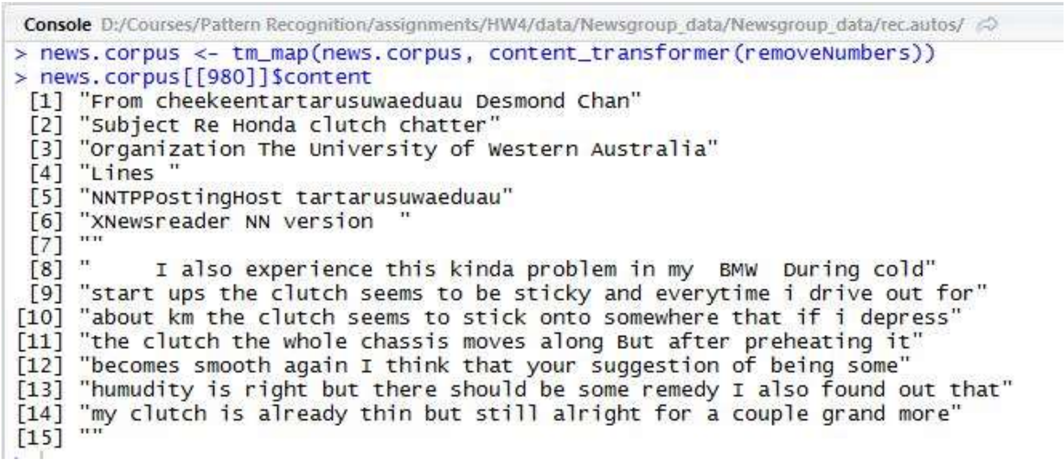
```
> news.corpus[[980]]$content
 [1] "From: cheekeen@tartarus.uwa.edu.au (Desmond Chan)"
 [2] "Subject: Re: Honda clutch chatter"
 [3] "Organization: The University of Western Australia"
 [4] "Lines: 8"
 [5] "NNTP-Posting-Host: tartarus.uwa.edu.au"
 [6] "X-Newsreader: NN version 6.4.19 #1"
 [7] ""
 [8] "      I also experience this kinda problem in my 89 BMW 318. During cold"
 [9] "start ups, the clutch seems to be sticky and everytime i drive out, for"
[10] "about 5km, the clutch seems to stick onto somewhere that if i depress"
[11] "the clutch, the whole chassis moves along. But after preheating, it"
[12] "becomes smooth again. I think that your suggestion of being some"
[13] "humudity is right but there should be some remedy. I also found out that"
[14] "my clutch is already thin but still alright for a couple grand more!"
[15] ""
```

Remove punctuation
news.corpus <- tm_map(news.corpus, content_transformer(removePunctuation))
news.corpus[[980]]$content


Console D:/Courses/Pattern Recognition/assignments/HW4/data/Newsgroup_data/Newsgroup_data/rec.autos/

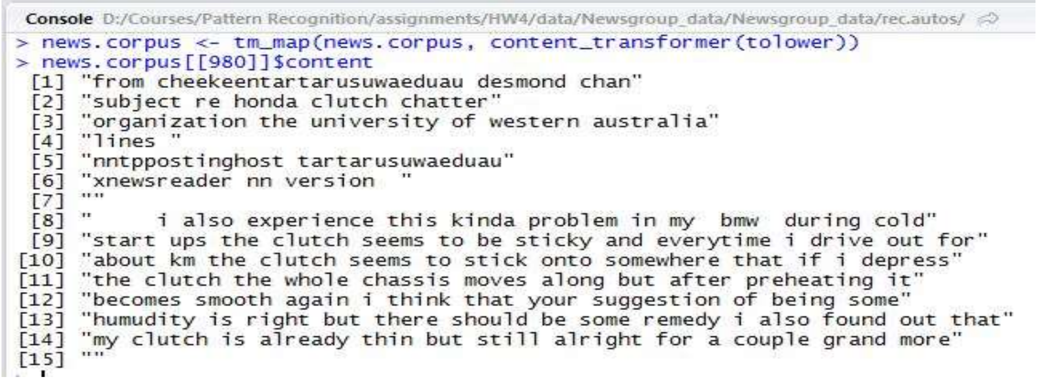
```
> news.corpus <- tm_map(news.corpus, content_transformer(removePunctuation))
> news.corpus[[980]]$content
 [1] "From cheekeentartarusuwaeduau Desmond Chan"
 [2] "Subject Re Honda clutch chatter"
 [3] "Organization The University of Western Australia"
 [4] "Lines 8"
 [5] "NNTPPostingHost tartarusuwaeduau"
 [6] "XNewsreader NN version 6419 1"
 [7] ""
 [8] "      I also experience this kinda problem in my 89 BMW 318 During cold"
 [9] "start ups the clutch seems to be sticky and everytime i drive out for"
[10] "about 5km the clutch seems to stick onto somewhere that if i depress"
[11] "the clutch the whole chassis moves along But after preheating it"
[12] "becomes smooth again I think that your suggestion of being some"
[13] "humudity is right but there should be some remedy I also found out that"
[14] "my clutch is already thin but still alright for a couple grand more"
[15] ""
```

Remove Numbers:
news.corpus <- tm_map(news.corpus, content_transformer(removeNumbers))
news.corpus[[980]]$content


Console D:/Courses/Pattern Recognition/assignments/HW4/data/Newsgroup_data/Newsgroup_data/rec.autos/

```
> news.corpus <- tm_map(news.corpus, content_transformer(removeNumbers))
> news.corpus[[980]]$content
 [1] "From cheekeentartarusuwaeduau Desmond Chan"
 [2] "Subject Re Honda clutch chatter"
 [3] "Organization The University of Western Australia"
 [4] "Lines "
 [5] "NNTPPostingHost tartarusuwaeduau"
 [6] "XNewsreader NN version   "
 [7] ""
 [8] "      I also experience this kinda problem in my  BMW  During cold"
 [9] "start ups the clutch seems to be sticky and everytime i drive out for"
[10] "about km the clutch seems to stick onto somewhere that if i depress"
[11] "the clutch the whole chassis moves along But after preheating it"
[12] "becomes smooth again I think that your suggestion of being some"
[13] "humudity is right but there should be some remedy I also found out that"
[14] "my clutch is already thin but still alright for a couple grand more"
[15] ""
```

Tolower:
news.corpus <- tm_map(news.corpus, content_transformer(tolower))
news.corpus[[980]]$content

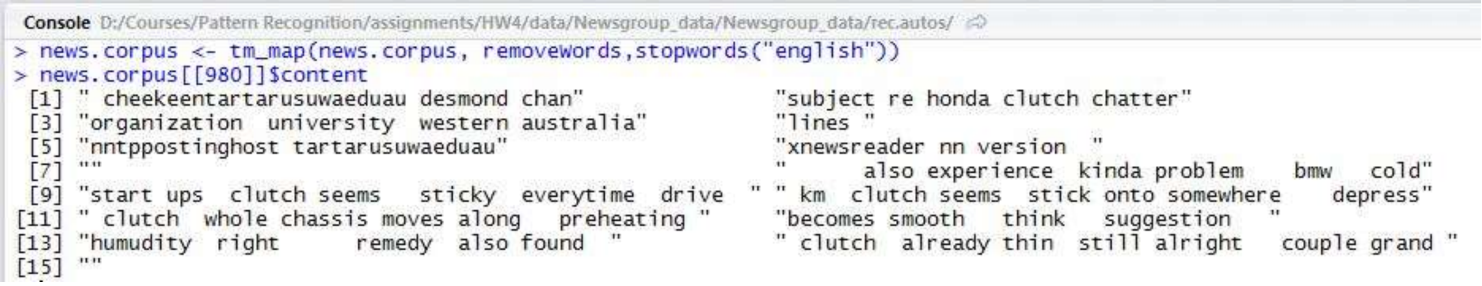

Console D:/Courses/Pattern Recognition/assignments/HW4/data/Newsgroup_data/Newsgroup_data/rec.autos/

```
> news.corpus <- tm_map(news.corpus, content_transformer(tolower))
> news.corpus[[980]]$content
 [1] "from cheekeentartarusuwaeduau desmond chan"
 [2] "subject re honda clutch chatter"
 [3] "organization the university of western australia"
 [4] "lines "
 [5] "nntppostinghost tartarusuwaeduau"
 [6] "xnewsreader nn version  "
 [7] ""
 [8] "     i also experience this kinda problem in my  bmw  during cold"
 [9] "start ups the clutch seems to be sticky and everytime i drive out for"
[10] "about km the clutch seems to stick onto somewhere that if i depress"
[11] "the clutch the whole chassis moves along but after preheating it"
[12] "becomes smooth again i think that your suggestion of being some"
[13] "humudity is right but there should be some remedy i also found out that"
[14] "my clutch is already thin but still alright for a couple grand more"
[15] ""
```

removeWords:

news.corpus <- tm_map(news.corpus, removeWords,stopwords("english"))

news.corpus[[980]]$content

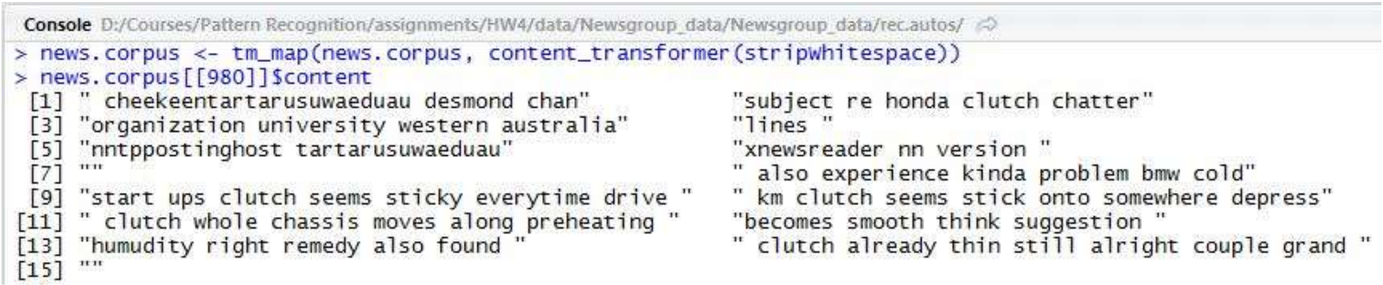

Console D:/Courses/Pattern Recognition/assignments/HW4/data/Newsgroup_data/Newsgroup_data/rec.autos/

```
> news.corpus <- tm_map(news.corpus, removewords,stopwords("english"))
> news.corpus[[980]]$content
 [1] " cheekeentartarusuwaeduau desmond chan"                "subject re honda clutch chatter"
 [3] "organization  university  western australia"           "lines "
 [5] "nntppostinghost tartarusuwaeduau"                      "xnewsreader nn version  "
 [7] ""                                                      "       also experience  kinda problem    bmw   cold"
 [9] "start ups  clutch seems   sticky  everytime  drive   "  " km  clutch seems  stick onto somewhere     depress"
[11] " clutch  whole chassis moves along   preheating "       "becomes smooth    think    suggestion    "
[13] "humudity  right       remedy  also found   "            " clutch  already thin  still alright   couple grand "
[15] ""
```

stripWhitespace:

news.corpus <- tm_map(news.corpus, content_transformer(stripWhitespace))

news.corpus[[980]]$content

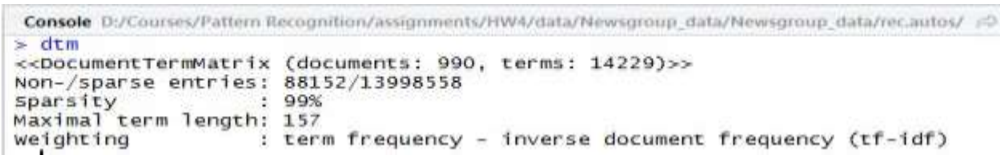

Console D:/Courses/Pattern Recognition/assignments/HW4/data/Newsgroup_data/Newsgroup_data/rec.autos/

```
> news.corpus <- tm_map(news.corpus, content_transformer(stripwhitespace))
> news.corpus[[980]]$content
 [1] " cheekeentartarusuwaeduau desmond chan"           "subject re honda clutch chatter"
 [3] "organization university western australia"        "lines "
 [5] "nntppostinghost tartarusuwaeduau"                 "xnewsreader nn version "
 [7] ""                                                 " also experience kinda problem bmw cold"
 [9] "start ups clutch seems sticky everytime drive "   " km clutch seems stick onto somewhere depress"
[11] " clutch whole chassis moves along preheating "    "becomes smooth think suggestion "
[13] "humudity right remedy also found "                " clutch already thin still alright couple grand "
[15] ""
```

news.corpus <- Corpus(VectorSource((news.corpus)))

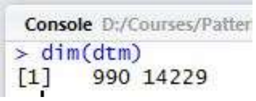
6.c.

dtm <- DocumentTermMatrix(news.corpus, control = list(minWordLength = 1, minDocFreq = 1, weighting = function(x) weightTfIdf(x,normalize = FALSE)))


Console D:/Courses/Pattern Recognition/assignments/HW4/data/Newsgroup_data/Newsgroup_data/rec.autos/

```
> dtm
<<DocumentTermMatrix (documents: 990, terms: 14229)>>
Non-/sparse entries: 88152/13998558
Sparsity           : 99%
Maximal term length: 157
Weighting          : term frequency - inverse document frequency (tf-idf)
```

dim(dtm)

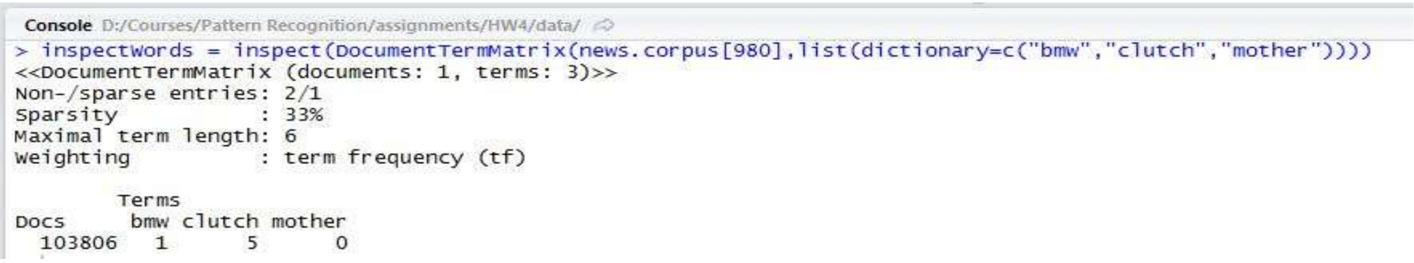


```
> dim(dtm)
[1]   990 14229
```

Dimensions are: 990 14229

6.d.

inspectWords = inspect(DocumentTermMatrix(news.corpus[980],list(dictionary=c("bmw","clutch","mother"))))



Console D:/Courses/Pattern Recognition/assignments/HW4/data/

```
> inspectWords = inspect(DocumentTermMatrix(news.corpus[980],list(dictionary=c("bmw","clutch","mother"))))
<<DocumentTermMatrix (documents: 1, terms: 3)>>
Non-/sparse entries: 2/1
Sparsity           : 33%
Maximal term length: 6
Weighting          : term frequency (tf)

        Terms
Docs     bmw clutch mother
  103806   1      5      0
```

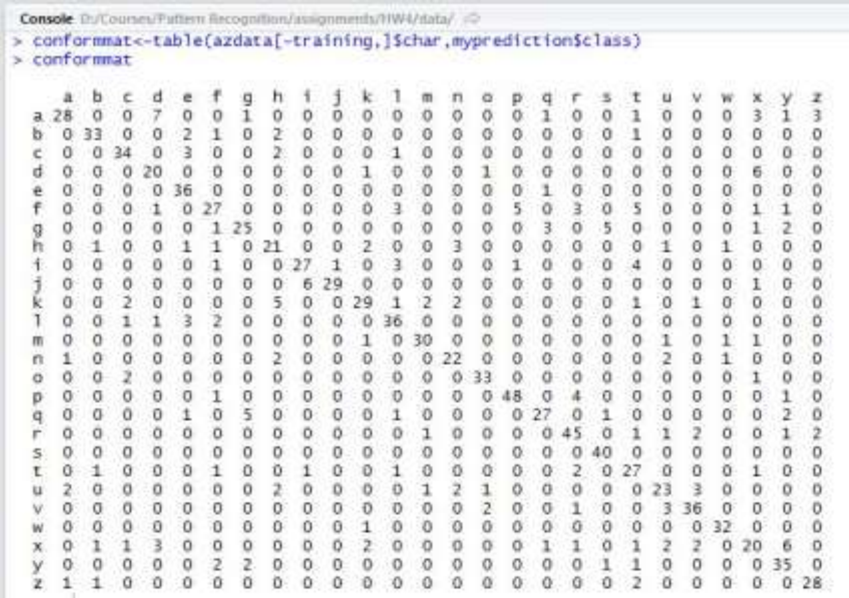
From the above it can be seen that the word ‘bmw’ is present 1 time, ‘clutch’ is present 5 times and the word ‘mother’ is present 0 times in the file number 103806.

The results match with the expected outputs.

7

7.a.

```
azdata=read.table("az-5000.txt",header=T)
training<-sample(1:5000,4000)
trainingData<-azdata[training,]
aztestData<-azdata[-training,]
priors<-c(rep(1/26,26))
library(MASS)
azdatalda<-lda(char~.,azdata,subset=training,prior=priors)
myprediction<-predict(azdatalda,newdata=azdata[-training,],type="response")
conformmat<-table(azdata[-training,]$char,myprediction$class)
conformmat
```

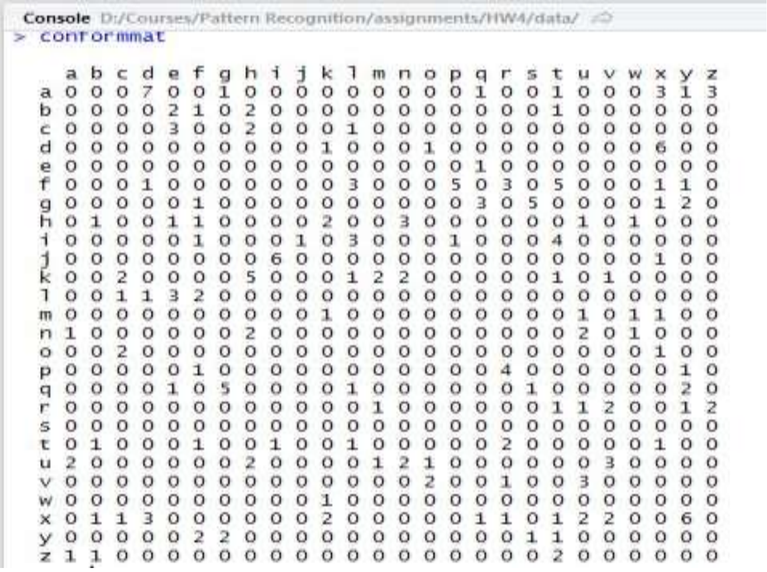


Console D:/Courses/Pattern Recognition/assignments/HW4/data/

```
> conformmat<-table(azdata[-training,]$char,myprediction$class)
> conformmat

     a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u  v  w  x  y  z
  a 28  0  0  7  0  0  1  0  0  0  0  0  0  0  0  0  1  0  0  1  0  0  0  3  1  3
  b  0 33  0  0  2  1  0  2  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0
  c  0  0 34  0  3  0  0  2  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  d  0  0  0 20  0  0  0  0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  6  0  0
  e  0  0  0  0 36  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
  f  0  0  0  1  0 27  0  0  0  0  0  3  0  0  0  5  0  3  0  5  0  0  0  1  1  0
  g  0  0  0  0  0  1 25  0  0  0  0  0  0  0  0  0  3  0  5  0  0  0  0  1  2  0
  h  0  1  0  0  1  1  0 21  0  0  2  0  0  3  0  0  0  0  0  0  1  0  1  0  0  0
  i  0  0  0  0  0  1  0  0 27  1  0  3  0  0  0  1  0  0  0  4  0  0  0  0  0  0
  j  0  0  0  0  0  0  0  0  6 29  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0
  k  0  0  2  0  0  0  0  5  0  0 29  1  2  2  0  0  0  0  0  1  0  1  0  0  0  0
  l  0  0  1  1  3  2  0  0  0  0  0 36  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  m  0  0  0  0  0  0  0  0  0  0  1  0 30  0  0  0  0  0  0  0  1  0  1  1  0  0
  n  1  0  0  0  0  0  0  2  0  0  0  0  0 22  0  0  0  0  0  0  2  0  1  0  0  0
  o  0  0  2  0  0  0  0  0  0  0  0  0  0  0 33  0  0  0  0  0  0  0  0  1  0  0
  p  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0 48  0  4  0  0  0  0  0  0  1  0
  q  0  0  0  0  1  0  5  0  0  0  0  1  0  0  0  0 27  0  1  0  0  0  0  0  2  0
  r  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 45  0  1  1  2  0  0  1  2
  s  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 40  0  0  0  0  0  0  0
  t  0  1  0  0  0  1  0  0  1  0  0  1  0  0  0  0  0  2  0 27  0  0  0  1  0  0
  u  2  0  0  0  0  0  0  2  0  0  0  0  1  2  1  0  0  0  0  0 23  3  0  0  0  0
  v  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  1  0  0  3 36  0  0  0  0
  w  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 32  0  0  0
  x  0  1  1  3  0  0  0  0  0  0  2  0  0  0  0  0  1  1  0  1  2  2  0 20  6  0
  y  0  0  0  0  0  2  2  0  0  0  0  0  0  0  0  0  0  0  1  1  0  0  0  0 35  0
  z  1  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0 28
```

Making the diagonal ‘0’

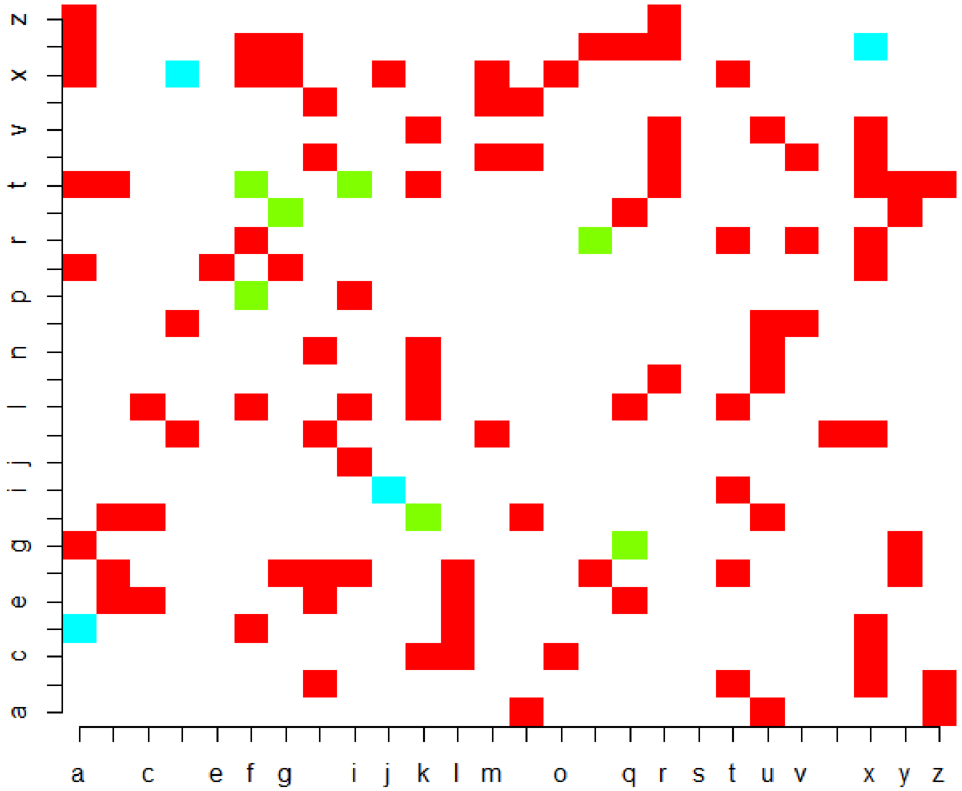
```
for(k in 1:26){
  conformmat[k,k]=0
}
Conformmat
```



Console D:/Courses/Pattern Recognition/assignments/HW4/data/

```
> conformmat

    a b c d e f g h i j k l m n o p q r s t u v w x y z
  a 0 0 0 7 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 3 1 3
  b 0 0 0 0 2 1 0 2 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
  c 0 0 0 0 3 0 0 2 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  d 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 6 0 0
  e 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
  f 0 0 0 1 0 0 0 0 0 0 0 3 0 0 0 5 0 3 0 5 0 0 0 1 1 0
  g 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 3 0 5 0 0 0 0 1 2 0
  h 0 1 0 0 1 1 0 0 0 0 2 0 0 3 0 0 0 0 0 0 1 0 1 0 0 0
  i 0 0 0 0 0 1 0 0 0 1 0 3 0 0 0 1 0 0 0 4 0 0 0 0 0 0
  j 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
  k 0 0 2 0 0 0 0 5 0 0 0 1 2 2 0 0 0 0 0 1 0 1 0 0 0 0
  l 0 0 1 1 3 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  m 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0
  n 1 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 2 0 1 0 0 0
  o 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
  p 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 1 0
  q 0 0 0 0 1 0 5 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 2 0
  r 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 2 0 0 1 2
  s 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  t 0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 0 0 2 0 0 0 0 0 1 0 0
  u 2 0 0 0 0 0 0 2 0 0 0 0 1 2 1 0 0 0 0 0 0 3 0 0 0 0
  v 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 1 0 0 3 0 0 0 0 0
  w 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  x 0 1 1 3 0 0 0 0 0 0 2 0 0 0 0 0 1 1 0 1 2 2 0 0 6 0
  y 0 0 0 0 0 2 2 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0
  z 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0
```

```
image(z=conformmat,zlim=c(1,10),col=rainbow(4), axes=FALSE)
axis(1, at = seq(0, 1, length=length(colnames(conformmat))), labels=colnames(conformmat))
axis(2, at = seq(0, 1, length=length(colnames(conformmat))), labels=colnames(conformmat))
```



7.b.

Color with most confusion is Blue.

Pairs:

{a,d}, {j,i},{d,x},{x,y}