Document Clustering Using K-Means

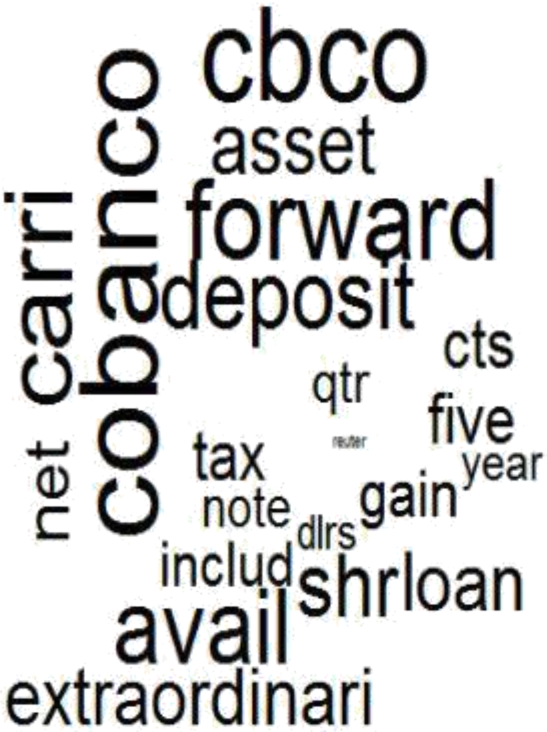
Step 1: Loading the data:

Step 2: Cleaning and Pre Processing the data.

- To lower cases
- Remove spaces
- Remove numbers
- Remove punctuation
- Remove whitespaces
- Remove Stopwords
- Remove my custom stopwords
- Stemming the documents

Step 3:- Creating Document Term Matrix

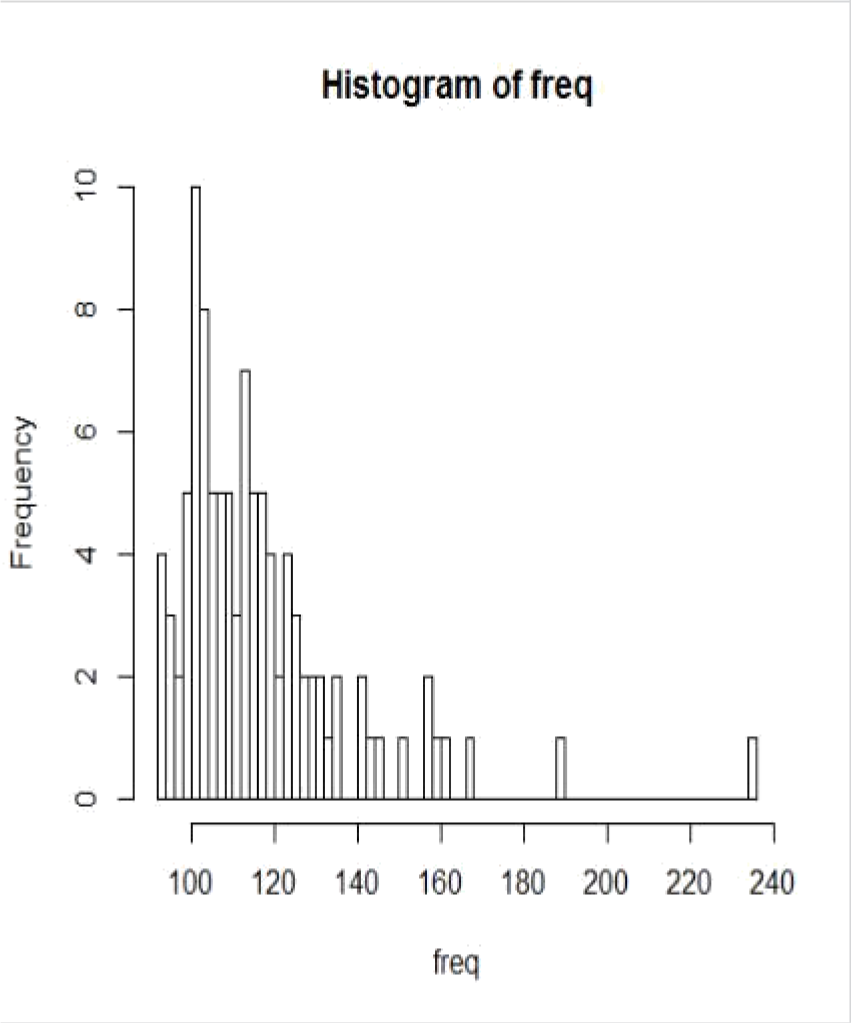
Wordcloud:->



Step 4: Normalization

Normalizing the scores. Normalize the Tf-Idf scores by Euclidean distance using dist function.

Histogram:

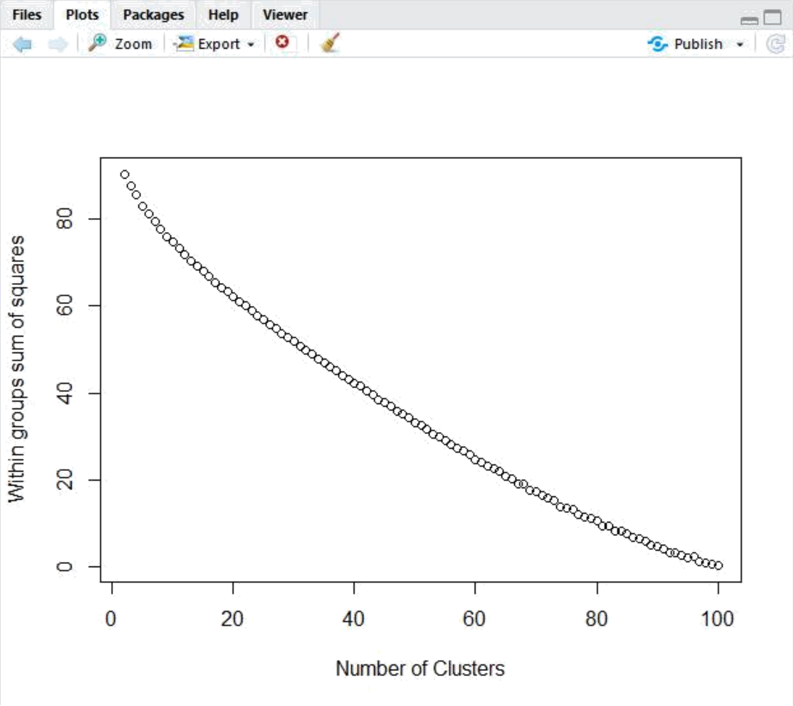


STEP 5: K-MEANS finding the correct number of clusters

kmeans – determine the optimum number of clusters (elbow method)

Observe the below Elbow curve .There is no clear no of Clusters for K-Means.

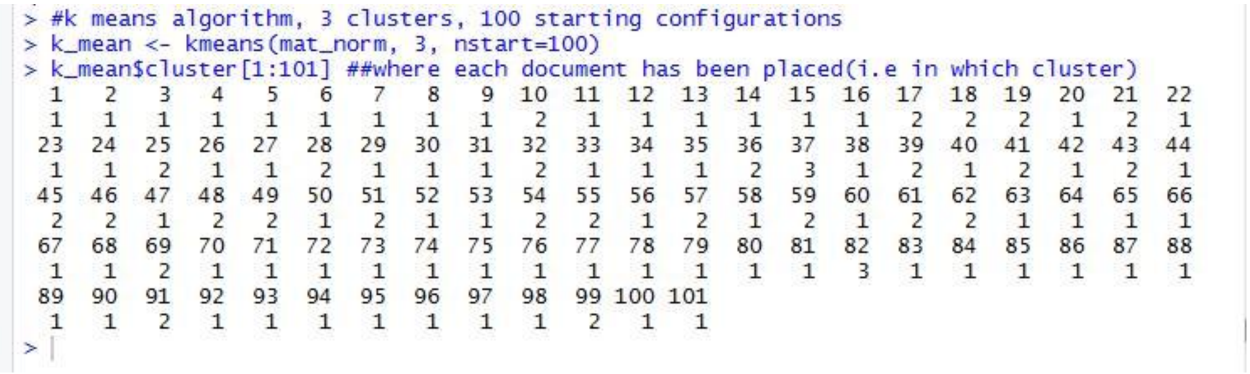
The plot clearly shows that there is no k for which the summed WSS flattens out (no distinct “elbow”).



Observing the above elbow curve we can say that there is not a specific number of clusters for K-Means clustering.

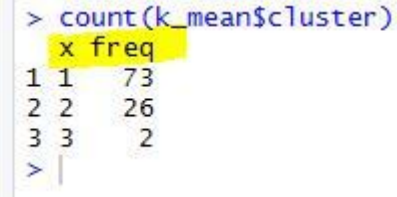
Step 6: K-MEANS Clustering

Below we can see where all the 100 documents have been placed and in which cluster.



```
> #k means algorithm, 3 clusters, 100 starting configurations
> k_mean <- kmeans(mat_norm, 3, nstart=100)
> k_mean$cluster[1:101] ##where each document has been placed(i.e in which cluster)
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22 
  1   1   1   1   1   1   1   1   1   2   1   1   1   1   1   1   2   2   2   1   2   1 
 23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44 
  1   1   2   1   1   2   1   1   1   2   1   1   1   2   3   1   2   1   2   1   2   1 
 45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66 
  2   2   1   2   2   1   2   1   1   2   2   1   2   1   2   1   2   2   1   1   1   1 
 67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88 
  1   1   2   1   1   1   1   1   1   1   1   1   1   1   1   3   1   1   1   1   1   1 
 89  90  91  92  93  94  95  96  97  98  99 100 101 
  1   1   2   1   1   1   1   1   1   1   2   1   1 
> 
```

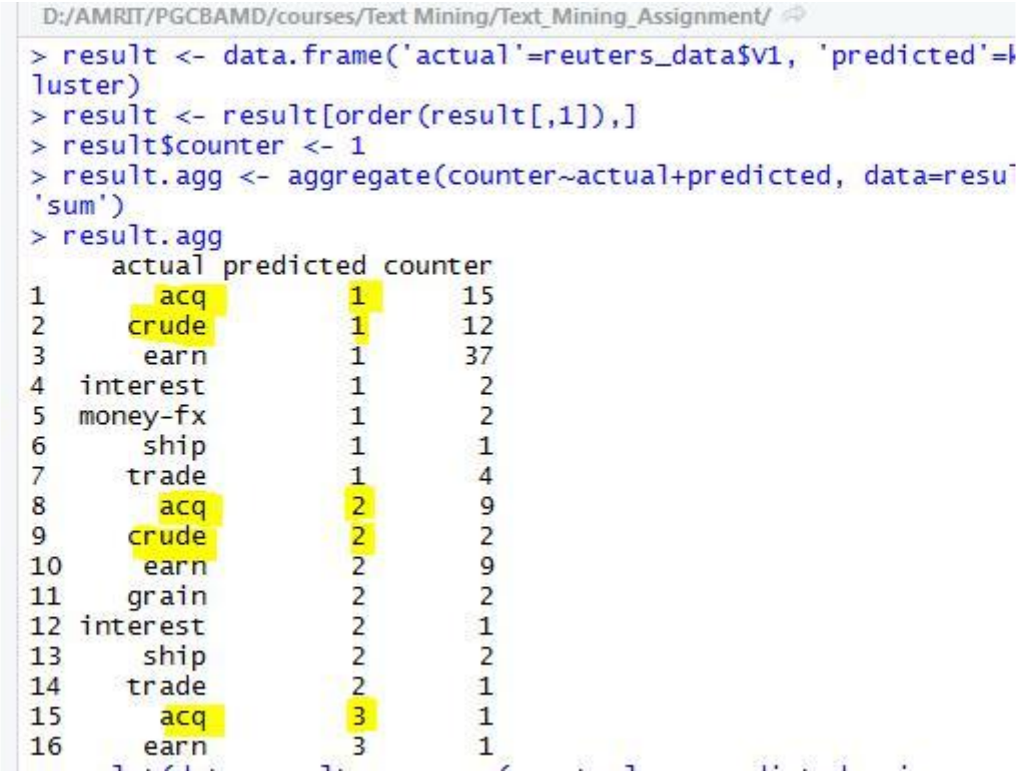
Below we can see the frequency of words in which Cluster.



```
> count(k_mean$cluster)
  x freq
1 1   73
2 2   26
3 3    2
> 
```

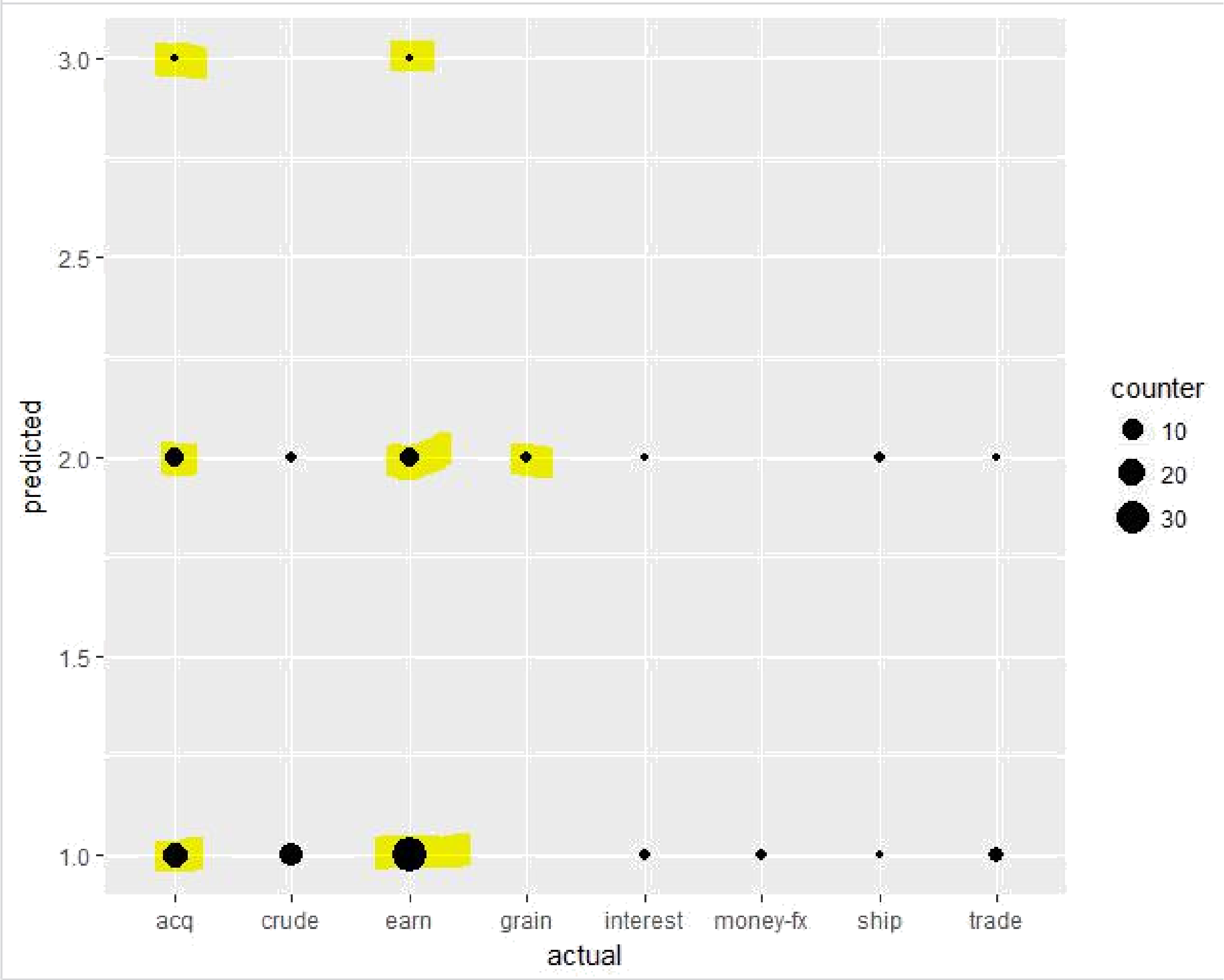
Step 7: Validation of the Model

Model Testing:



D:/AMRIT/PGCBAMD/courses/Text Mining/Text_Mining_Assignment/

```
> result <- data.frame('actual'=reuters_data$V1, 'predicted'=k
luster)
> result <- result[order(result[,1]),]
> result$counter <- 1
> result.agg <- aggregate(counter~actual+predicted, data=resul
'sum')
> result.agg
     actual predicted counter
1       acq         1      15
2     crude         1      12
3      earn         1      37
4  interest         1       2
5  money-fx         1       2
6      ship         1       1
7     trade         1       4
8       acq         2       9
9     crude         2       2
10     earn         2       9
11    grain         2       2
12 interest         2       1
13     ship         2       2
14    trade         2       1
15      acq         3       1
16     earn         3       1
```



- As per clustering 75% of the documents have been classified in the Cluster 1.
- 23% of the documents has been classified to the cluster 2.
- 2% of the documents have been classified to the cluster 3.

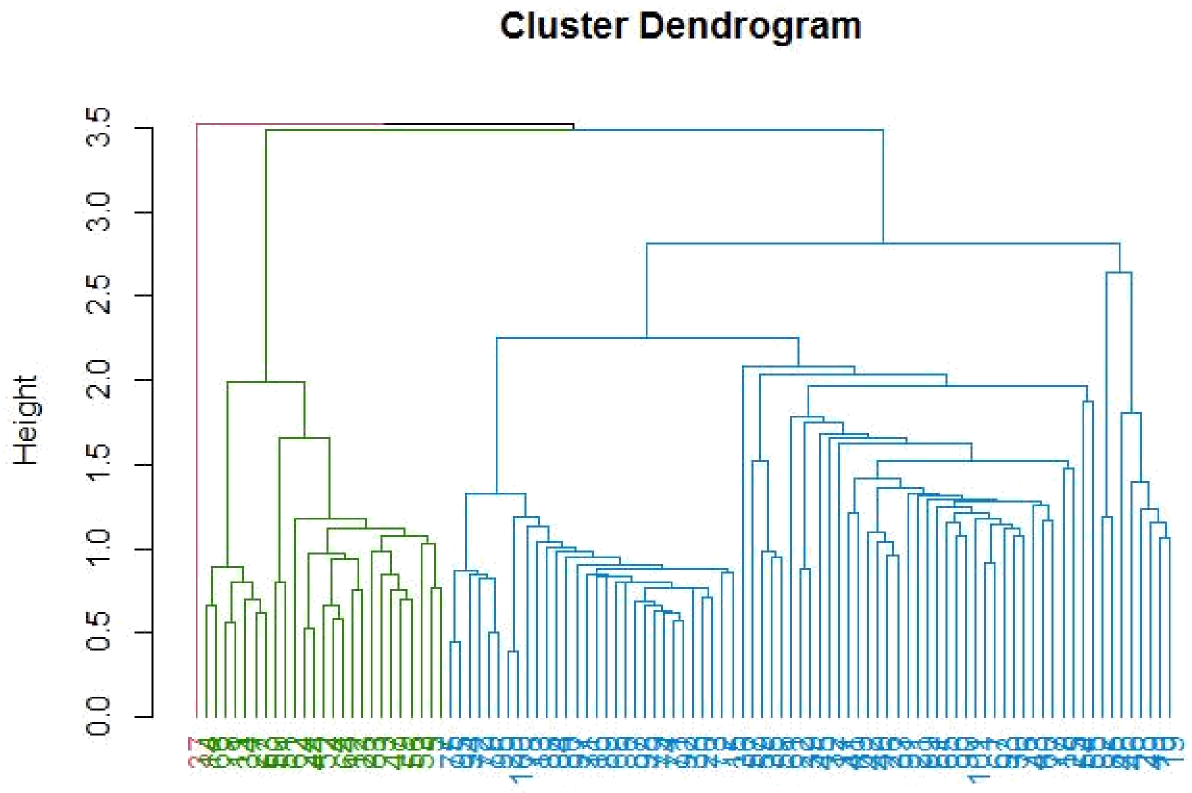
Step 8 : Performing Hierarchical Clustering

Hierarchical Clustering:

Hierarchical methods uses a distance matrix as an input for the clustering algorithm. The clustering output can be displayed in a **dendrogram.**

ward.D: is an Agglomerative clustering

Dendogram:



STEP 9: TOPIC Modelling

Topic Modeling:

It is a type of **unsupervised analysis: topic modeling.**

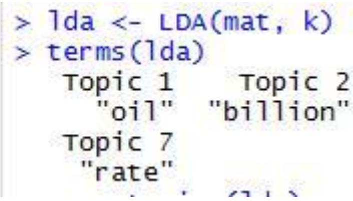
lda() function (Latent Dirichlet Allocation) is being used for it.

Unlike k-means which is a discriminative model (it tries to tell documents apart by conditioning on the contents of the document).

LDA is a generative model (it creates a probabilistic model of how the words in each document were generated/written).

LDA will determine which words are likely generated from a specific topic, then determine the topic of a document by examining these probabilities.

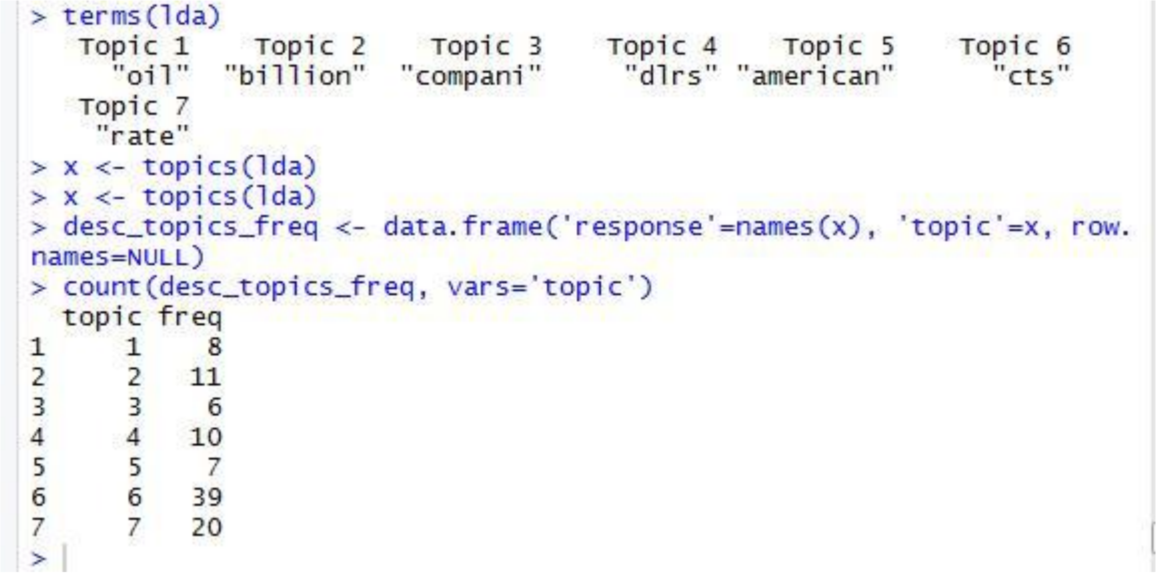
7 topics:



```
> lda <- LDA(mat, k)
> terms(lda)
  Topic 1    Topic 2    Topic 3    Topic 4    Topic 5    Topic 6 
    "oil"  "billion"  "compani"     "dlrs" "american"      "cts" 
  Topic 7 
   "rate" 
```

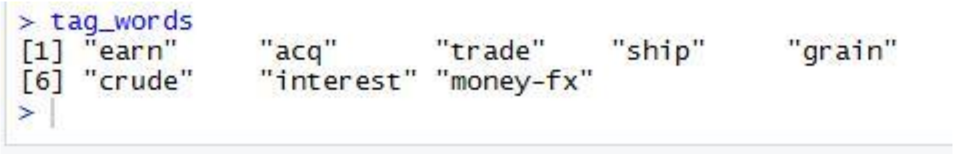
Topics Corresponds well to the tags:

Oil->crude, billion->earn, trade, rate->interest



```
> terms(lda)
  Topic 1    Topic 2    Topic 3    Topic 4    Topic 5    Topic 6 
    "oil"  "billion"  "compani"     "dlrs" "american"      "cts" 
  Topic 7 
   "rate" 
> x <- topics(lda)
> x <- topics(lda)
> desc_topics_freq <- data.frame('response'=names(x), 'topic'=x, row.
names=NULL)
> count(desc_topics_freq, vars='topic')
  topic freq
1     1    8
2     2   11
3     3    6
4     4   10
5     5    7
6     6   39
7     7   20
> 
```

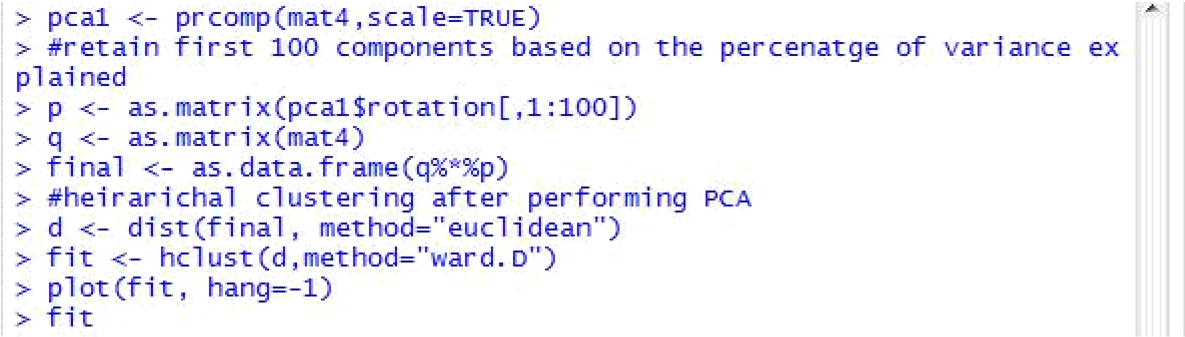
TAG:



```
> tag_words
[1] "earn"     "acq"      "trade"    "ship"     "grain"   
[6] "crude"    "interest" "money-fx"
> 
```

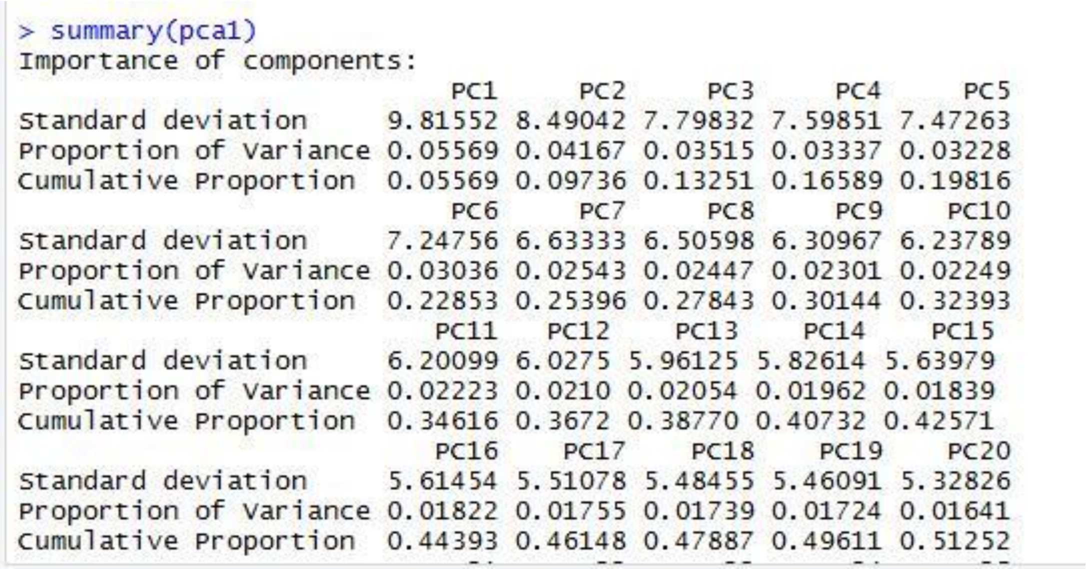
Step 10 :- HIERACHIAL CLUSTERING USING PCA

PCA on the document term matrix and then performed Hierarchical clustering



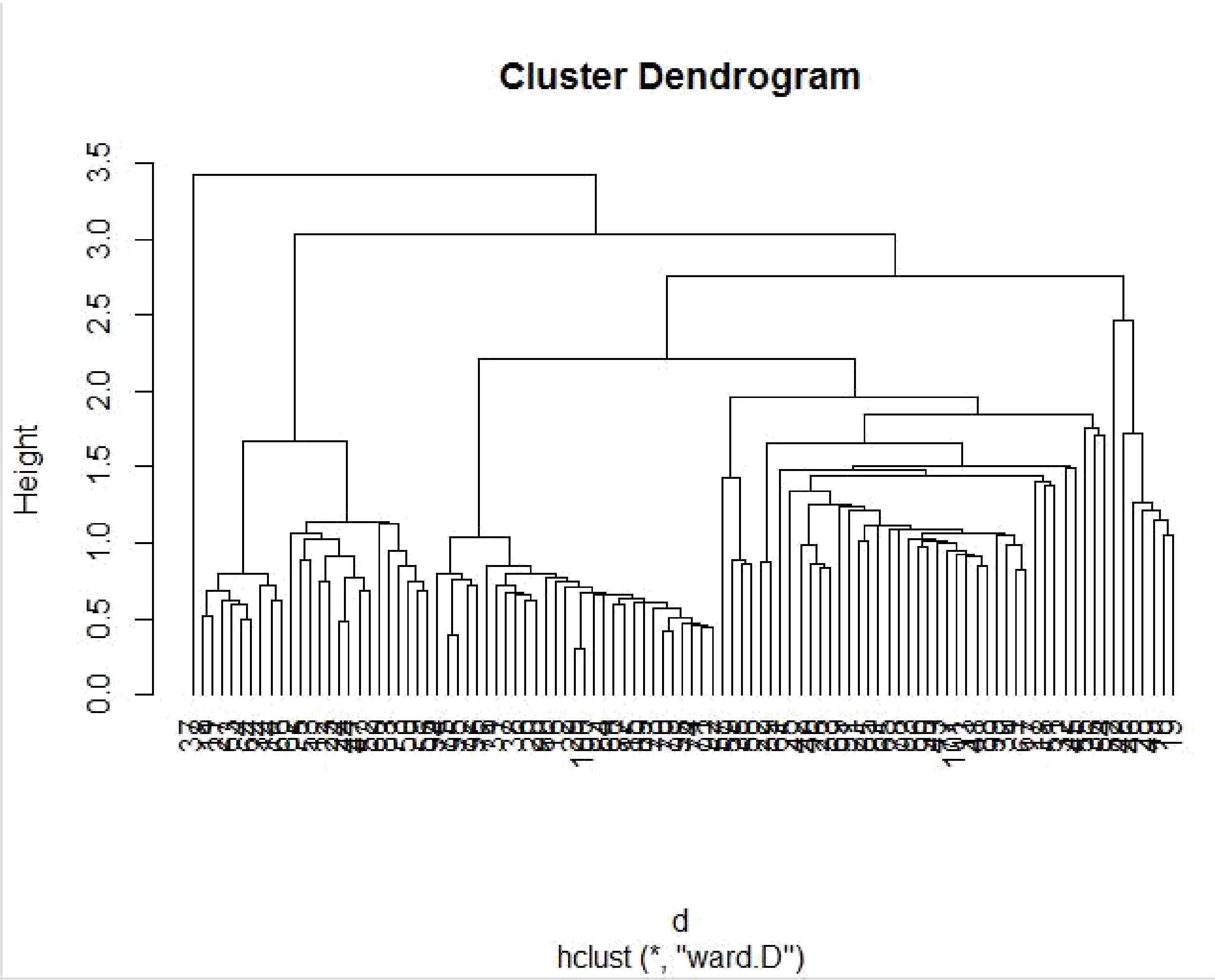
```
> pca1 <- prcomp(mat4,scale=TRUE)
> #retain first 100 components based on the percenatge of variance ex
plained
> p <- as.matrix(pca1$rotation[,1:100])
> q <- as.matrix(mat4)
> final <- as.data.frame(q%*%p)
> #heirarichal clustering after performing PCA
> d <- dist(final, method="euclidean")
> fit <- hclust(d,method="ward.D")
> plot(fit, hang=-1)
> fit
```

PCA Summary:



```
> summary(pca1)
Importance of components:
                           PC1     PC2     PC3     PC4     PC5
Standard deviation     9.81552 8.49042 7.79832 7.59851 7.47263
Proportion of Variance 0.05569 0.04167 0.03515 0.03337 0.03228
Cumulative Proportion  0.05569 0.09736 0.13251 0.16589 0.19816
                           PC6     PC7     PC8     PC9    PC10
Standard deviation     7.24756 6.63333 6.50598 6.30967 6.23789
Proportion of Variance 0.03036 0.02543 0.02447 0.02301 0.02249
Cumulative Proportion  0.22853 0.25396 0.27843 0.30144 0.32393
                          PC11   PC12    PC13    PC14    PC15
Standard deviation     6.20099 6.0275 5.96125 5.82614 5.63979
Proportion of Variance 0.02223 0.0210 0.02054 0.01962 0.01839
Cumulative Proportion  0.34616 0.3672 0.38770 0.40732 0.42571
                          PC16    PC17    PC18    PC19    PC20
Standard deviation     5.61454 5.51078 5.48455 5.46091 5.32826
Proportion of Variance 0.01822 0.01755 0.01739 0.01724 0.01641
Cumulative Proportion  0.44393 0.46148 0.47887 0.49611 0.51252
```

Dendogram:


Cluster Dendrogram
Height
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
d
hclust (*, "ward.D")