Data Visualisation Assignment 4

Sai Krishna Lakshminarayanan

14 March 2019

1.Introduction

The task of text analysis is to be done here on the corpus. The corpus is one that consist of 7612 short text documments in it. Now, exploratory analysis is to be done on this corpus with suitable visualisation techniques.When this is done, we will be able to predict the pattern that is observed in the corpus.

2.Data Preprocessing

2.1 Loading the corpus

Initially, all the required packages are loaded into the code and then the corpus is loaded with help of the VCorpus() function. Encoding and language are given as the standard UTF-8 and english respectively.

```
#Loading all the required packages
library(tm)
```

```
## Warning: package 'tm' was built under R version 3.5.3
```

```
## Loading required package: NLP
```

```
library(SnowballC)
```

```
## Warning: package 'SnowballC' was built under R version 3.5.2
```

```
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 3.5.3
```

```
## Loading required package: RColorBrewer
```

```
## Warning: package 'RColorBrewer' was built under R version 3.5.2
```

```
library(RColorBrewer)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
library(ggdendro)
```

```
## Warning: package 'ggdendro' was built under R version 3.5.3
```

```
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 3.5.3
```

```
library(HSAUR)
```

```
## Warning: package 'HSAUR' was built under R version 3.5.3
```

```
## Loading required package: tools
```

```
library(fpc)
```

```
## Warning: package 'fpc' was built under R version 3.5.3
```

```
library(skmeans)
```

```
## Warning: package 'skmeans' was built under R version 3.5.3
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## ------------------------------------------------------------------------------
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
library(philentropy)
```

```
## Warning: package 'philentropy' was built under R version 3.5.3
```

```
library(treemapify)
```

```
## Warning: package 'treemapify' was built under R version 3.5.3
```

```
library(gplots)
```

```
## Warning: package 'gplots' was built under R version 3.5.3
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:wordcloud':
##
##     textplot
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

```
library(stats)
```

```
corpus1 <- VCorpus(DirSource("corpus", encoding = "UTF-8"), readerControl = list(language = "en
g"))
#Loading the corpus
```

2.2 Cleaning the corpus

Now, the corpus is cleaned by removing the punctations, stop words and annotations. It is changed into lower case to bring about uniformity while considering the words. Numbers are removed as they're literals and are meaningless for text analysis. Additional spaces are removed inorder to avoid them being counted as words in corpus. Common words are manually given and removed inorder to give out meaningul results. Finally, the words are stemmed in order to group together common words into 1 without several separate counts based on tense and so. These are done with the help of tm_map() function.

```
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))#assigning content
 transformer
corpus1 <- tm_map(corpus1, toSpace, "/")#removing / /. @ \\| from the corpus
corpus1 <- tm_map(corpus1, toSpace, "/.")
corpus1 <- tm_map(corpus1, toSpace, "@")
corpus1 <- tm_map(corpus1, toSpace, "\\|")
```

```
corpus1 <- tm_map(corpus1, content_transformer(tolower))#transformation to lower case
corpus1 <- tm_map(corpus1, removeWords, stopwords("english"))#removing stop words,punctuation an
d numbers
corpus1 <- tm_map(corpus1, removePunctuation)
corpus1 <- tm_map(corpus1, removeNumbers)
```

```
corpus1 <- tm_map(corpus1, removeWords, c(letters)) #removing letters and blank spaces
corpus1 <- tm_map(corpus1, stripWhitespace)
```

```
corpus1<-tm_map(corpus1,removeWords, c('subject','will','use','can','organ','line','one','say',
'now','like','get','know','just','think','also','make','may','see','want','nntppostinghost","fro
m", "to","organization", "nntp-posting-host", "article-i.d", "keywords", "originator", "re","rep
ly-to", "last-modified", "distribution","one","write",'need','look'))
```

```
corpus1 <- tm_map(corpus1, stemDocument)
```

2.3 Creating Term Document Matrix

Now, the term document matrix is generated. The sparse terms are removed and then the words along with their frequency are calculated.

```
corpus1.tdm <- TermDocumentMatrix(corpus1)#generating term document matrix
```

```
corpus1.tdm <- removeSparseTerms(corpus1.tdm, 0.9996)#removing terms appearing in very less docs
```

```
corpus1.tdm.matrix <- corpus1.tdm %>% as.matrix()#creating matrix to store the words and its fre
quency

v <- sort(rowSums(corpus1.tdm.matrix),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
head(d, 5)
```

```
##            word freq
## line       line 7452
## write     write 5680
## articl   articl 4949
## univers univers 3906
## peopl     peopl 3511
```

2.4 Creating DocumentTerm Matrix

In the document term matrix, each row is represented by the document and each column by the word. The sparsity threshold is given to remove the sparse values. Then the zero values are removed inorder to reduce the computing time.

```
corpus1.dtm <- DocumentTermMatrix(corpus1, control = list(weighting = function(x) weightTfIdf(x,
normalize = TRUE)))
sparsity_threshold = 0.9995#generating document term matrix
corpus1.dtm<-removeSparseTerms(corpus1.dtm, sparsity_threshold)#removing the sparse values
corpus1.dtm.mat <- corpus1.dtm %>% as.matrix()
corpus1.dtm.mat <- corpus1.dtm.mat[rowSums(corpus1.dtm.mat^2) !=0,]#removing zero values
```

2.5 Generating Random Sample

It will be a tedious task to process on the entire corpus. Therefore, 20% of the data is taken as a sample and worked out.

```
percent = 20
sample_size = nrow(corpus1.dtm.mat) * percent/100
#taking the 20% sample for document term matrix.
corpus1.dtm.mat.sample <- corpus1.dtm.mat[sample(1:nrow(corpus1.dtm.mat), sample_size, replace=F
ALSE),]
```

3. Exploratory Analysis by Speherical K-Means

Now, the exploratory analysis is done initially by the method of spherical k means. For this purpose, k is assumed to be as 8.Then the spherical k mean clustering is done based on it.

```
k=8#k value
corpus1.dtm.mat.sample.skm  <- skmeans(corpus1.dtm.mat.sample,k, method='pclust')
#applying spherical k means to the sample dtm
head(corpus1.dtm.mat.sample.skm$cluster)
```

```
## doc1152  doc855 doc1522 doc3325  doc827 doc5372
##       8       1       8       2       8       4
```

Now, once that is done, then the class of the document is to be considered. Here, it is taken to be 1 as it is present inside the corpus on a whole unlinke the archive example given during tutorial where four distinct journals where explicitly given.

```
corpus1.dtm.mat.sample.skm <- as.data.frame(corpus1.dtm.mat.sample.skm$cluster)#storing the clus
ter values
colnames(corpus1.dtm.mat.sample.skm) = c("cluster")
corpus1.dtm.mat.sample.skm$docs <- rownames(corpus1.dtm.mat.sample.skm)#storing the doc names
corpus1.dtm.mat.sample.skm$docs<-lapply(corpus1.dtm.mat.sample.skm$docs, tm::removeNumbers)#remo
ving the numbers
corpus1.dtm.mat.sample.skm$docs <- unlist(corpus1.dtm.mat.sample.skm$docs)#unlisting them to get
values
head(corpus1.dtm.mat.sample.skm$docs)
```

```
## [1] "doc" "doc" "doc" "doc" "doc" "doc"
```

Then, the sample spherical k means table is generated based on the cluster in rows and doc in columns.

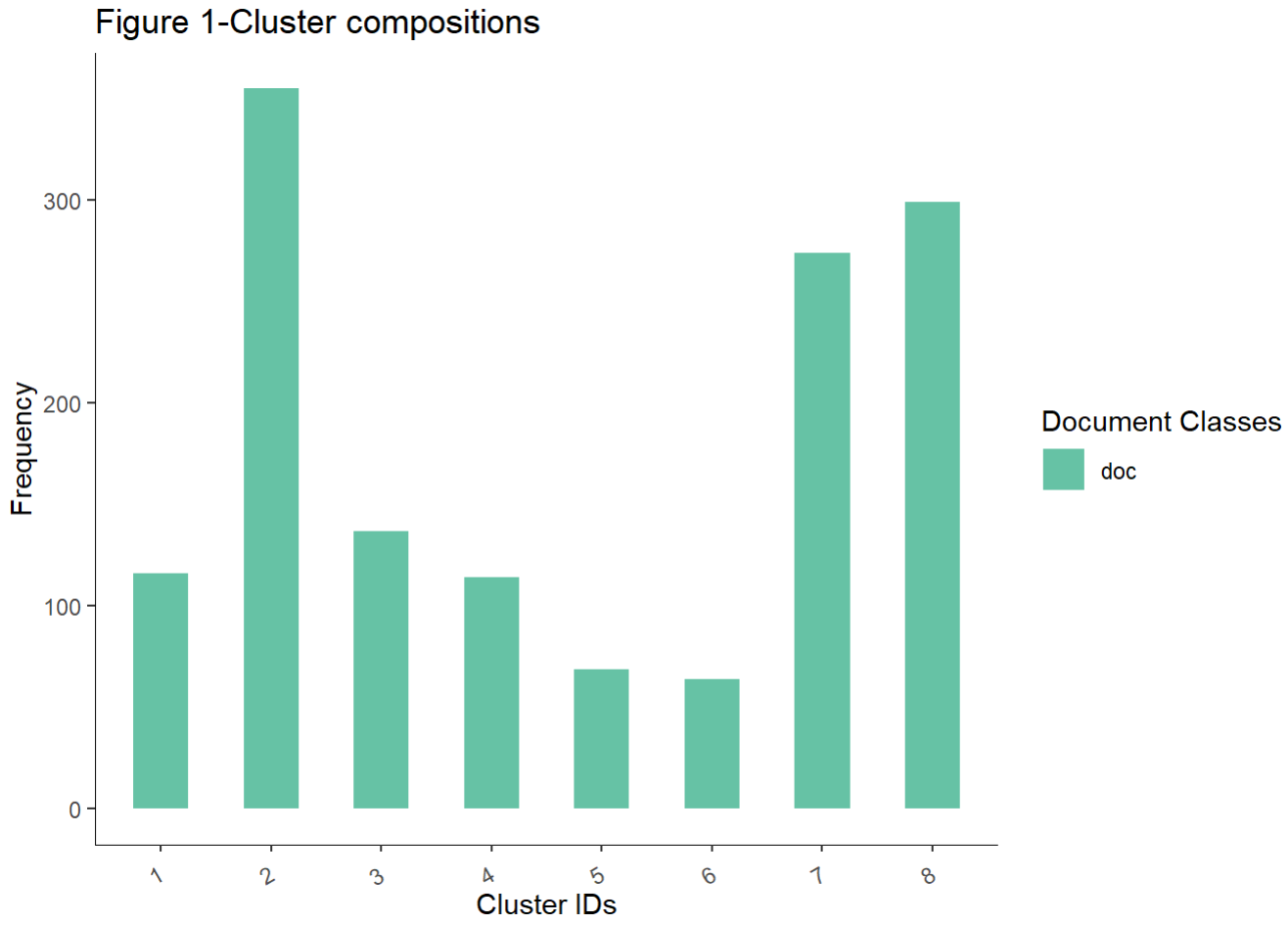
```
corpus1.dtm.mat.sample.skm.table <-table(corpus1.dtm.mat.sample.skm$cluster, corpus1.dtm.mat.sam
ple.skm$docs)
corpus1.dtm.mat.sample.skm.table#generating table for the sample value
```

```
##
##     doc
##   1 116
##   2 355
##   3 137
##   4 114
##   5  69
##   6  64
##   7 274
##   8 299
```

3.1 Cluster Composition

Now, based on the values obtained, the cluster composition is checked for to know the spread of the frequency across the clusters.It is seen that some of the clusters are having high frequency whereas some are having very low frequency in the cluster.

```
corpus1.dtm.mat.sample.skm.table <-as.data.frame.table(corpus1.dtm.mat.sample.skm.table)
# plotting the stacked bar chart to show the cluster compositions
g<- ggplot(corpus1.dtm.mat.sample.skm.table, aes(x=Var1, y=Freq,fill=Var2))
g<- g + geom_bar(width = 0.5, stat="identity") +
  scale_fill_brewer(palette = "Set2", name = "Document Classes") +
  xlab("Cluster IDs") +
  ylab("Frequency") +
  ggtitle("Figure 1-Cluster compositions") +
  theme(panel.grid.major = element_blank(), panel.background = element_blank(), axis.line = elem
ent_line(colour = "black", size = 0.25),  axis.text.x = element_text(angle = 30, hjust=1, vjust
 = .5), legend.key = element_rect(fill = NA, colour = NA, size = 0.25))

g
```



3.2 Explanatory Word Cloud for each cluster

Now, visualising the each cluster topic using word cloud function in order to see the contents of each cluster and understand how meaningful it is in reality.

```
corpus1.tdm.sample <- corpus1.tdm[, rownames(corpus1.dtm.mat.sample)]
#taking only the common documents that are present in document term matrix sample
corpus1.tdm.sample.mat <- corpus1.tdm.sample %>% as.matrix()#changing to matrix


m<- length(unique(corpus1.dtm.mat.sample.skm$cluster))
#cluster count
```

```
set.seed(2474)#seting relevant seed
par(mfrow=c(1,1))#to display one cluster per row


for (i in 1:m) {#for all 8 cluster this loop is put
  #documents present in the cluster for the loop i
  cluster_doc_ids <-which(corpus1.dtm.mat.sample.skm$cluster==i)
  corpus1.tdm.sample.mat.cluster<- corpus1.tdm.sample.mat[, cluster_doc_ids]
#subsets for the documents present
  v <- sort(rowSums(corpus1.tdm.sample.mat.cluster),decreasing=TRUE)#sorting by frequency
  d <- data.frame(word = names(v),freq=v)
  wordcloud(words = d$word, freq = d$freq, scale=c(4,.5), min.freq = 20,
            max.words=30, random.order=FALSE, rot.per=0.3,
            colors=c('#f2f0f7','#cbc9e2','#9e9ac8','#756bb1','#54278f'))
  title(paste("Figure 2.", i,"Cluster",i))
}#applying word cloud function to give out plots
```

Figure 2. 1 Cluster 1

Figure 2. 2 Cluster 2

Figure 2. 3 Cluster 3

Figure 2. 4 Cluster 4

Figure 2. 5 Cluster 5

Figure 2. 6 Cluster 6

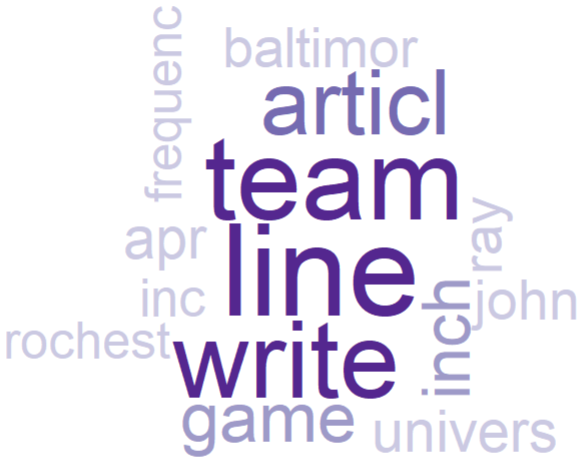
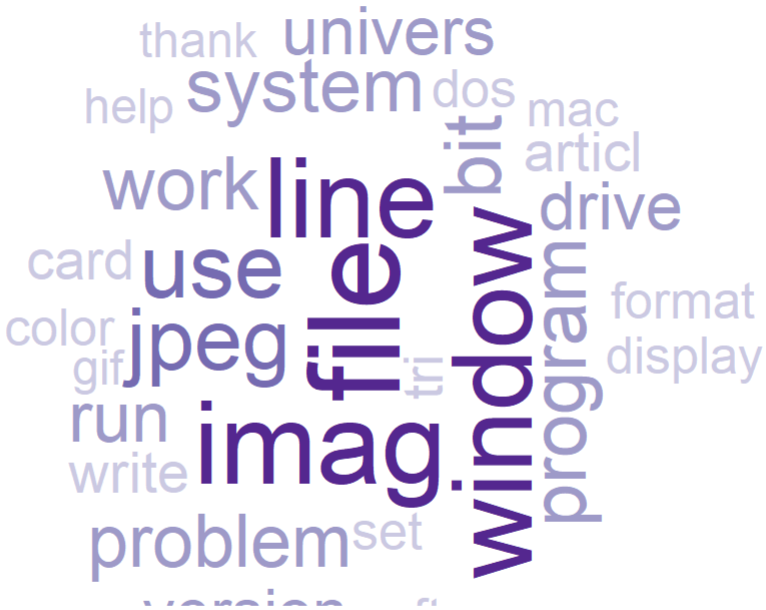


Figure 2. 7 Cluster 7

Figure 2. 8 Cluster 8



4.Exploratory Analysis by Hierarchical Clustering

Hierarchical Clustering helps in getting a visual sense of the clusters in the corpus. It is dependent on generating a distance matrix which is done by help cosine.

```
sim_matrix<-distance(corpus1.dtm.mat.sample, method = "cosine")#calculating the cosine similarit
y
```

```
## Metric: 'cosine'; comparing: 1428 vectors.
```

Upon generating the similarity matrix, the sample hierarchical clustering is created which will then be used for plotting purposes

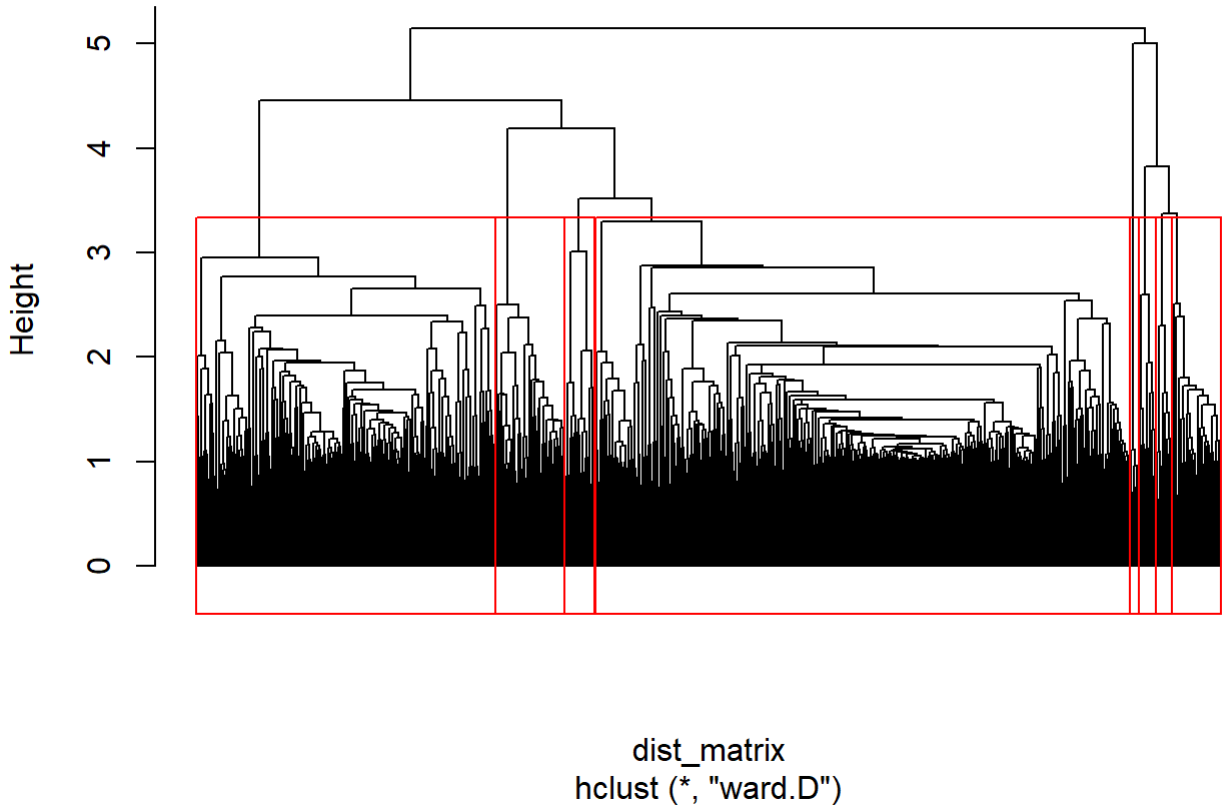
```
colnames(sim_matrix) <- rownames(corpus1.dtm.mat.sample)
rownames(sim_matrix) <- rownames(corpus1.dtm.mat.sample)
#assigning doc names to rows and columns

max_sim <- max(sim_matrix)#max cosine similarity

dist_matrix <- as.dist(max_sim-sim_matrix)
#obtaining distance matrix using the cosine similarity

corpus1.dtm.sample.dend <- hclust(dist_matrix, method = "ward.D")#performing cosine similarity
```

4.1 Dendogram

Now,plotting a dendogram on the sample dendogram values with a cut at pre define value of k as 8.

```
set.seed(2584)
#plotting the dendogram
plot(corpus1.dtm.sample.dend, hang= -1, labels = FALSE,  main = "Figure 3-Cluster dendrogram", s
ub = NULL, xlab = NULL, ylab = "Height")
#creating rectangles on the dendogram for the k number of clusters
rect.hclust(corpus1.dtm.sample.dend, k = 8, border = "red")
```

Figure 3-Cluster dendrogram



4.2 Cutting the Dendogram

Now,checking the clustering in the dendogram after making a horizantal cut that return value of k as 8. Once that is done, then the corpus is examined with its document content in each cluster.

```
k=8#assigning k as 8

corpus1.dtm.sample.dend.cut <- cutree(corpus1.dtm.sample.dend, k=k)
#cutting the dendogram using the cut tree function

m <- length(unique(corpus1.dtm.sample.dend.cut))
#number of clusters in the cut

corpus1.dtm.sample.dend.cut <- as.data.frame(corpus1.dtm.sample.dend.cut)
#generating a data frame and giving column names
colnames(corpus1.dtm.sample.dend.cut) = c("cluster")

#creating a doc column for the cut
corpus1.dtm.sample.dend.cut$docs <- rownames(corpus1.dtm.sample.dend.cut)

#removing the numbers from the documents
corpus1.dtm.sample.dend.cut$docs<-lapply(corpus1.dtm.sample.dend.cut$docs, tm::removeNumbers)

#unlisting the list for docs.
corpus1.dtm.sample.dend.cut$docs <- unlist(corpus1.dtm.sample.dend.cut$docs)
```

```
# generating a frequency table for the dendogram cut values for the clusters
corpus1.dtm.sample.dend.cut.table <-table(corpus1.dtm.sample.dend.cut$cluster, corpus1.dtm.sampl
e.dend.cut$docs)

#displays the documents frequency per cluster
corpus1.dtm.sample.dend.cut.table
```

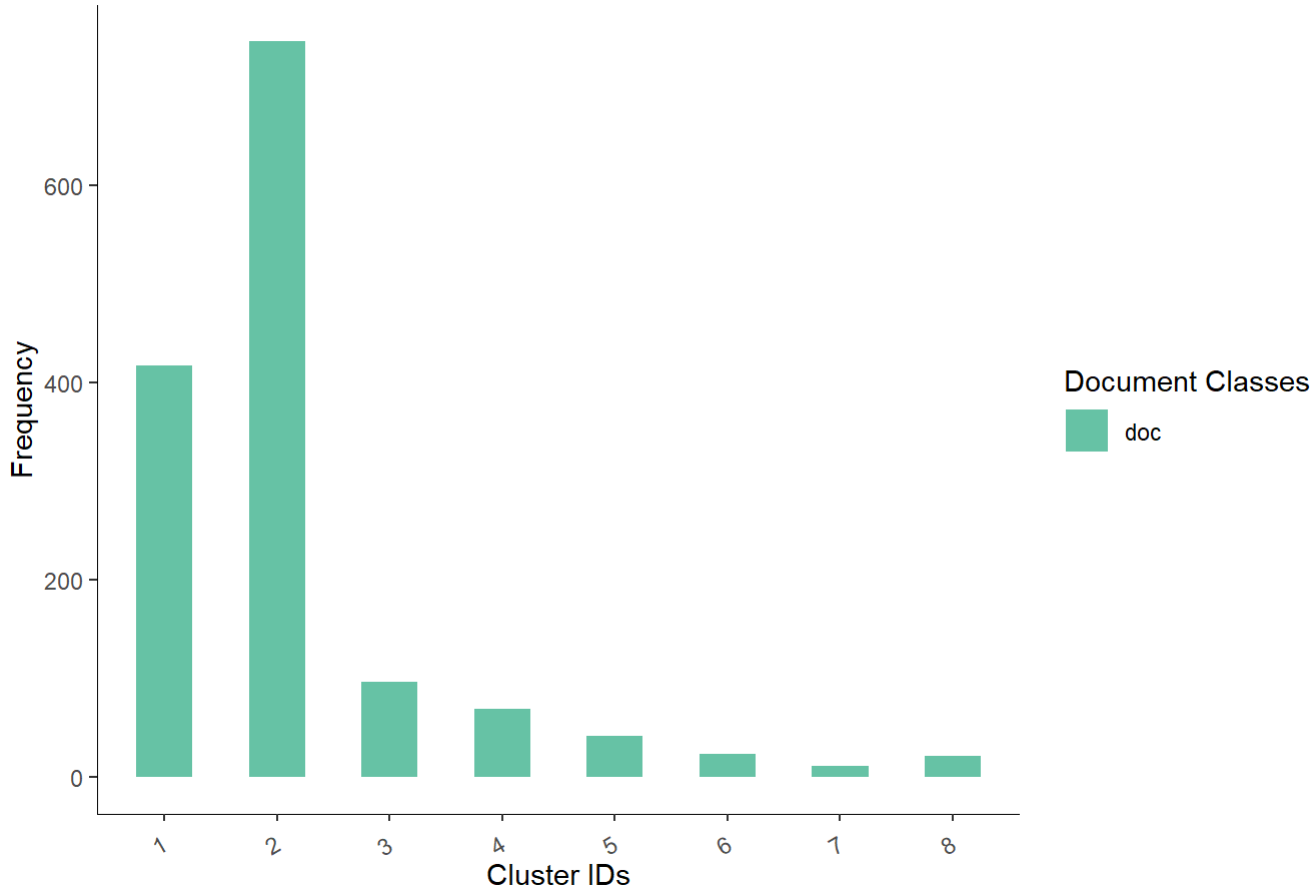
```
##
##     doc
##   1 417
##   2 745
##   3  97
##   4  69
##   5  42
##   6  24
##   7  12
##   8  22
```

4.3 Cluster Composition

Now, the cluster composition is to be done in the similar way like the methodolgy followed during spherical k means.

```
corpus1.dtm.sample.dend.cut.table <-as.data.frame.table(corpus1.dtm.sample.dend.cut.table)
#generating a data frame of the dendogram cut table and then creating stacked bar graph to show
 cluster compositions
g<- ggplot(corpus1.dtm.sample.dend.cut.table, aes(x=Var1, y=Freq,fill=Var2))
g<- g + geom_bar(width = 0.5, stat="identity") +
  scale_fill_brewer(palette = "Set2", name = "Document Classes") +
  xlab("Cluster IDs") +
  ylab("Frequency") +
  ggtitle("Figure 4-Cluster compositions") +
  theme(panel.grid.major = element_blank(), panel.background = element_blank(), axis.line = elem
ent_line(colour = "black", size = 0.25),  axis.text.x = element_text(angle = 30, hjust=1, vjust
 = .5), legend.key = element_rect(fill = NA, colour = NA, size = 0.25))

g
```

Figure 4-Cluster compositions



4.4 Word Cloud for each cluster

Class labels are not present but the document terms present can be used to summarise each cluster. The same approach is followed like in the case of spherical k means clustering. For the cluster present in the cut tree function, word frequencies plot is generated. Therefore, 8 in total is generated.

```
m <- length(unique(corpus1.dtm.sample.dend.cut$cluster))#number of clusters present


set.seed(1478)
par(mfrow=c(1,1))


for (i in 1:m) {#for each of the 8 cluster present loop
  cut_doc_ids <-which(corpus1.dtm.sample.dend.cut$cluster==i)# documents present in the correspo
nding cluster i

  corpus1.tdm.sample.mat.cluster<- corpus1.tdm.sample.mat[, cut_doc_ids]
  #subset documents present to this cluster

  v <- sort(rowSums(corpus1.tdm.sample.mat.cluster),decreasing=TRUE)#sorting based on the freque
ncy
  d <- data.frame(word = names(v),freq=v)
  wordcloud(words = d$word, freq = d$freq, scale=c(3,.2), min.freq = 3,
          max.words=40, random.order=FALSE, rot.per=0.35,
          colors=c('#feedde','#fdbe85','#fd8d3c','#e6550d','#a63603'))
 title(paste("Figure 5.",i,"num clusters  = ", k, "; cluster", i))
}#using word cloud function to generate the plots
```

Figure 5. 1 num clusters = 8 ; cluster 1

Figure 5. 2 num clusters = 8 ; cluster 2

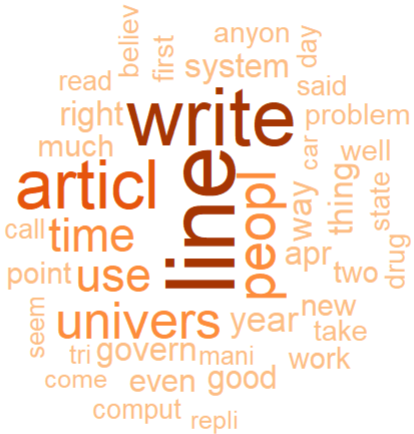


Figure 5. 3 num clusters = 8 ; cluster 3

Figure 5. 4 num clusters = 8 ; cluster 4

Figure 5. 5 num clusters = 8 ; cluster 5

Figure 5. 6 num clusters = 8 ; cluster 6

Figure 5. 7 num clusters = 8 ; cluster 7

Figure 5. 8 num clusters = 8 ; cluster 8

4.5 Tree Map

The tree map can be used to plot the top occuring terms in each cluster. This will help us to conduct hypothesis on the documents in each cluster and to understand and explain about what it is talking about. It resembles the world cloud plots which were used above.Though 2 variables can be considered for colour and size in tree map, here only the frequency is considered.

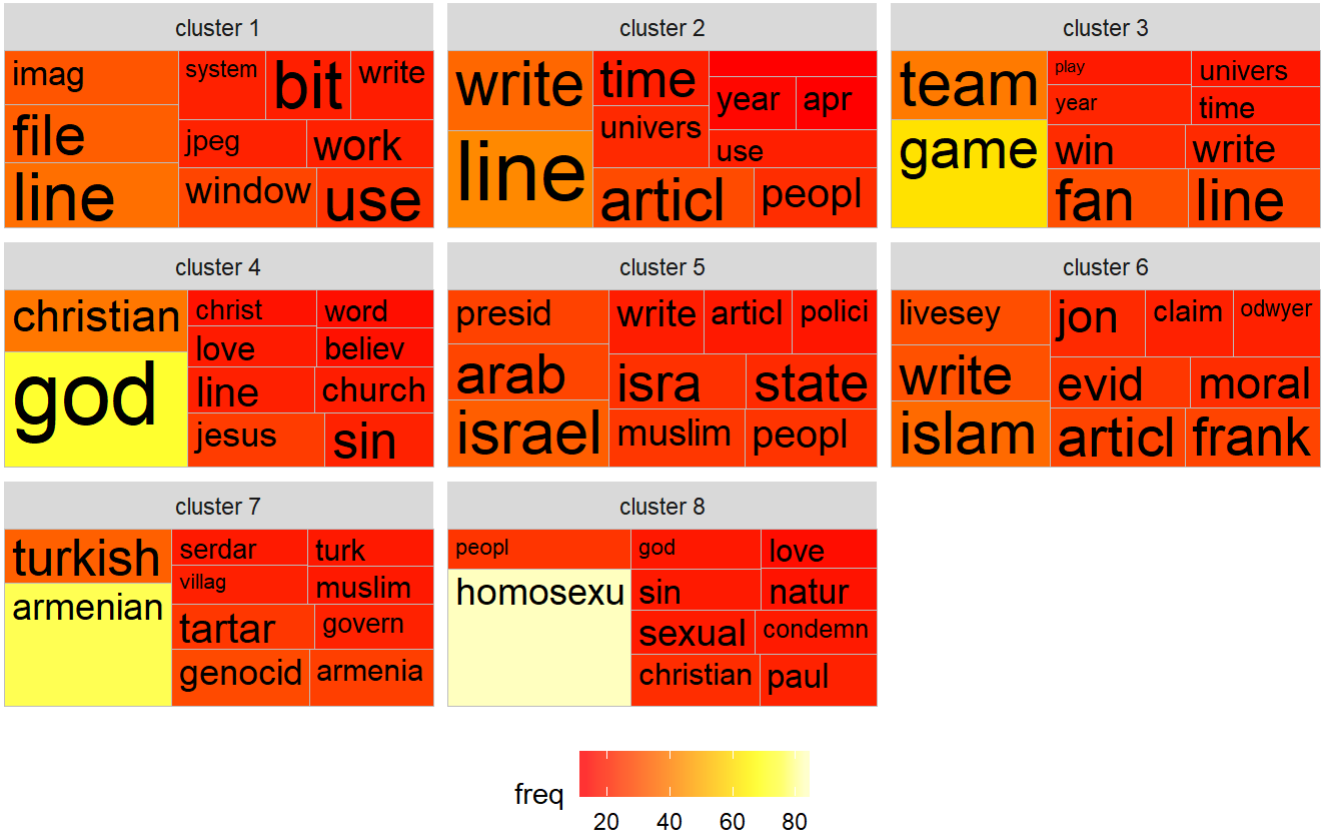
```
m <- length(unique(corpus1.dtm.sample.dend.cut$cluster))#number of clusters present at the cut

n <-10#total number of terms to show per cluster

df <- data.frame(word=character(), freq = double(),cluster = integer())#generating empty dataset

for (i in 1:m) {#for all 8 clusters in the loop
  cut_doc_ids <-which(corpus1.dtm.sample.dend.cut$cluster==i)#documents corresponding to the clu
ster i

  corpus1.tdm.sample.mat.cluster<- corpus1.tdm.sample.mat[, cut_doc_ids]
  #subset of the documents matrix present

  v <- sort(rowSums(corpus1.tdm.sample.mat.cluster),decreasing=TRUE)#sorting based on the freque
ncy
  d <- data.frame(word = names(v),freq=v, cluster=i)


  d[,2] <- scale(d[,2],center=FALSE, scale=TRUE)
  #scaling to reduce the domination of large values

  d <-d[1:n,]#considering the initial n

  df<- rbind(df,d)#binding this with df
}

df$freq <- as.vector(df$freq)#converting to vector

clust_name<-function(x){
   paste("cluster", x)
}#renaming as cluster 1 to 8


 df$cluster<- as.character(apply(df["cluster"], MARGIN = 2,FUN =clust_name ))#apply function use
d on cluster column

#plotting the tree map using geom_treemap()
gg<- ggplot(df, aes(area = freq, fill = freq, subgroup=cluster, label = word)) +
geom_treemap() +#assigning tree map function
geom_treemap_subgroup_border() +
geom_treemap_subgroup_text(place = "centre", grow = T, alpha = 0.5, colour ="#99d8c9", fontface
 = "italic", min.size = 10) +
geom_treemap_text(fontface = "italic", colour = "white", place = "centre", grow = TRUE)+
ggtitle("Figure 6-Tree Map")

gg
```

Figure 6-Tree Map

Now the tree map can be facted in order to get better visually separable out inorder to understand the cluster summary efficiently. Based on this, the most frequent terms in each cluster is seen and hypothesis is oberserved to explore the meaning of the documents clustered together.

```
gg<- ggplot(df, aes(area = freq, fill = freq, subgroup=cluster, label = word)) +
geom_treemap() +
  geom_treemap_text(grow = T, reflow = T, colour = "black") +
  facet_wrap( ~ cluster) +#faceting the tree msp
#giving heat map like colours
  scale_fill_gradientn(colours = heat.colors(n, alpha = 0.8)) +
  theme(legend.position = "bottom") +
  labs(title = "Figure 7-Tree Map facted by Cluster ", caption = "The area of each term is propo
rtional to its relative frequency within cluster")

gg#desired output
```

Figure 7-Tree Map facted by Cluster



The area of each term is proportional to its relative frequency within cluster

5.Conclusion

From the frequently occuring words found using the tree maps,patterns can be hypothesised based on the clusters. It is to be noted that for each execution the cluster changes and the terms occuring in them changes therefore only a broad consensus can be given from the readings without naming the clustering. This is because, due to random sampling and so it may be doing on other set which will give some other set of words and so.

From the observed most frequent words in the 8 clusters, each cluster can be said to be based on the topics as follows,

1. One cluster is containting documents that can be said to be about God as itcontent related to God,chirstians and Chruch.
2. One cluster is hypothesised to be about games as it contains frequently occuring terms as game, play , teams and so on.
3. One other cluster is considered to be about sex education as that contains terms like homo sexuality , gay and so.

4)One cluster is about computers as it contains terms like window,disk,drive and so.

5)One cluster is considered to be about law and order as it contains terms like law,system,people,govern and so on

6)One cluster is about religion particularly as it contains terms like muslim,arab and so.

7. One cluster can be said to be about calling internationally as it contains terms like swedon,germany and phone and tone.

8. Finally last cluster can be said to be about work as it talks about people, work , time,system and so on.

It is to be noted that k means is un realiable and tedious and it can’t be said with complete assertion about them and can be only considered as an estimate for the execution that is done at that corresponding time.

In this way, exploratory analysis is done on the corpus successfully and the hypothesis are done based on the clusters considered to give out meaningful insights into the documents. ##6.Reference 1. Tutorial and Lecture from week 8