```
getwd()
#set working directory
setwd("C:/Users/Kailash/Documents/TextMining")
#load tm library
library(tm)
#Create Corpus
docs <- Corpus(DirSource("C:/Users/Kailash/Documents/TextMining"))
#Check details
summary(docs)
#Inspect all documents in Corpus
inspect(docs)
#inspect a particular document
writeLines(as.character(docs[[30]]))

#Transform to lower case
docs <- tm_map(docs,content_transformer(tolower))
#remove potentiallyy problematic symbols
toSpace <- content_transformer(function(x, pattern) { return (gsub(pattern, " ",
x))})
docs <- tm_map(docs, toSpace, "-")
docs <- tm_map(docs, toSpace, ":")
docs <- tm_map(docs, toSpace, "'")
docs <- tm_map(docs, toSpace, ".")
docs <- tm_map(docs, toSpace, ".        ")
docs <- tm_map(docs, toSpace, " -")
docs <- tm_map(docs, toSpace, """)
docs <- tm_map(docs, toSpace, """)
#remove punctuation
docs <- tm_map(docs, removePunctuation)
#Strip digits
docs <- tm_map(docs, removeNumbers)

#remove stopwords
docs <- tm_map(docs, removeWords, stopwords("english"))
#remove whitespace
docs <- tm_map(docs, stripWhitespace)
#Good practice to check after each step.
writeLines(as.character(docs[[30]]))

#Explain stemming
#Need SnowballC library for stemming
#library(Snowball)
#Stem document
docs <- tm_map(docs,stemDocument)


docs <- tm_map(docs, content_transformer(gsub),
               pattern = "organiz", replacement = "organ")

docs <- tm_map(docs, content_transformer(gsub),
               pattern = "organis", replacement = "organ")

docs <- tm_map(docs, content_transformer(gsub),
               pattern = "andgovern", replacement = "govern")

docs <- tm_map(docs, content_transformer(gsub),
               pattern = "inenterpris", replacement = "enterpris")

docs <- tm_map(docs, content_transformer(gsub),
               pattern = "team-", replacement = "team")

#remove recalcitrant stopwords
```

```
myStopwords <- c("can", "say","one","way","use",
                                        "also","howev","tell","will",
                                        "much","need","take","tend","even",
                                        "like","particular","rather","said",
                                        "get","well","make","ask","come","end",
                                        "first","two","help","often","may",
                                        "might","see","someth","thing","point",
                                        "post","look","right","now","think","'ve ",
                                        "'re ")

docs <- tm_map(docs, removeWords, myStopwords)


writeLines(as.character(docs[[30]]))
#Create term-document matrix
tdm <- TermDocumentMatrix(docs)
#view it
tdm
#Create document-term matrix
dtm <- DocumentTermMatrix(docs)
#print a summary
dtm

#convert dtm to matrix
m<-as.matrix(dtm)
#write as csv file
write.csv(m,file="dtmEight2Late.csv")
#shorten rownames for display purposes
rownames(m) <- paste(substring(rownames(m),1,3),rep("..",nrow(m)),
                     substring(rownames(m),
nchar(rownames(m))-12,nchar(rownames(m))-4))
#compute distance between document vectors
d <- dist(m)
#run hierarchical clustering using Ward's method
groups <- hclust(d,method="ward.D")
#plot, use hang to ensure that labels fall below tree
plot(groups, hang=-1)
#cut into 2 subtrees (experiment) - try 2,3,4,5,6
rect.hclust(groups,2)

#kmeans clustering

#kmeans - run with nstart=100 and k=2,3,5 to compare results with hclust
kfit <- kmeans(d, 2, nstart=100)
#plot - need library cluster
library(cluster)
clusplot(as.matrix(d), kfit$cluster, color=T, shade=T, labels=2, lines=0)
#print contents of kfit
print(kfit)
#print cluster sizes
kfit$size
#print clusters (members)
kfit$cluster
#write clusters to csv file
write.csv(kfit$cluster,file="KMClustGroups2.csv")
#ss between cluster centers
kfit$betweenss
#what about a simple scatter plot??

#kmeans - how to determine optimal number of clusters?
#look for "elbow" in plot of summed intra-cluster distances (withinss) as fn of k
wss <- 2:29
for (i in 2:29) wss[i] <- sum(kmeans(d,
```

```
                                centers=i,nstart=25)$withinss)
plot(2:29, wss[2:29], type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")

####
```