DA_Assignment_TEXT_ANALYTICS
17BCS028 SOMYADEEP SHRIVASTAVA

1. Load and text-analyze the data using any platform/package such as R, Python-NLTK, etc. You may want to do some pre-processing and clean-up if needed, use a tokenizer, a stemmer, count some word frequencies, recognize phrases, etc.

```
library(ggplot2)
library(readr)
library(tm)
library(wordcloud)
library(plyr)
library(lubridate)
library(syuzhet)
```

```
#Import the twitter data set
tweetsdata=read.csv('/home/samroadie/Desktop/DA_Lab/LAB5',stringsAsFactors = FALSE)
```

```
options(warn=-1)
summary(tweetsdata)
```

OUTPUT

```
      X.1             X          text          favorited
 Min.   :    1   Min.   :    1   Length:14940       Mode :logical
 1st Qu.: 3736   1st Qu.: 3736   Class :character   FALSE:14940
 Median : 7470   Median : 7470   Mode  :character
 Mean   : 7470   Mean   : 7470
 3rd Qu.:11205   3rd Qu.:11205
 Max.   :14940   Max.   :14940

 favoriteCount       replyToSN         created         truncated
```

```
Min.   :   0.000   Length:14940       Length:14940       Mode :logical
1st Qu.:   0.000   Class :character   Class :character   FALSE:14243
Median :   0.000   Mode  :character   Mode  :character   TRUE :697
Mean   :   1.071
3rd Qu.:   0.000
Max.   :3166.000

  replyToSID             id             replyToUID       statusSource
Min.   :2.210e+10   Min.   :8.010e+17   Min.   :1.918e+06   Length:14940
1st Qu.:8.015e+17   1st Qu.:8.013e+17   1st Qu.:3.915e+07   Class :character
Median :8.529e+17   Median :8.015e+17   Median :1.458e+08   Mode  :character
Mean   :8.384e+17   Mean   :8.256e+17   Mean   :4.303e+16
3rd Qu.:8.540e+17   3rd Qu.:8.535e+17   3rd Qu.:1.480e+09
Max.   :8.555e+17   Max.   :8.555e+17   Max.   :8.543e+17
NA's   :14054                           NA's   :13838
 screenName        retweetCount    isRetweet       retweeted
Length:14940       Min.   :   0.0   Mode :logical   Mode :logical
Class :character   1st Qu.:   1.0   FALSE:3948      FALSE:14940
Mode  :character   Median :  40.0   TRUE :10992
                   Mean   : 223.8
                   3rd Qu.: 197.0
                   Max.   :5170.0

 created_date           hour           isRetweetNum     retweetedNum
Min.   :2016-11-22   Length:14940       Min.   :0.0000   Min.   :0
1st Qu.:2016-11-23   Class :character   1st Qu.:0.0000   1st Qu.:0
Median :2016-11-23   Mode  :character   Median :1.0000   Median :0
Mean   :2017-01-28                      Mean   :0.7357   Mean   :0
3rd Qu.:2017-04-16                      3rd Qu.:1.0000   3rd Qu.:0
Max.   :2017-04-21                      Max.   :1.0000   Max.   :0

    tweet
Min.   :1
1st Qu.:1
Median :1
Mean   :1
3rd Qu.:1
Max.   :1
```

Code

```
tweetsdata$created_date=as.Date(tweetsdata$created,format='%Y-%m-%d
%H:%M:%S')#convert created to date format
tweetsdata$hour = format(as.POSIXct(tweetsdata$created,format="%Y-%m-%d
%H:%M:%S"),"%H")#Extract Hour from the date
tweetsdata$isRetweetNum=ifelse(tweetsdata$isRetweet==FALSE,0,1)#Numerical
variable to indicate whether a tweet was retweet
tweetsdata$retweetedNum=ifelse(tweetsdata$retweeted==FALSE,0,1)#Total number of
times a tweet was tetweeted
tweetsdata$tweet=c(1)#Additional column that will help us in summing up total tweets
```

Preprocessing Data

```
some_txt<-gsub("(RT|via)((?:\\b\\w*@\\w+)+)","",tweetsdata$text)
some_txt<-gsub("http[^[:blank:]]+","",some_txt)
some_txt<-gsub("@\\w+","",some_txt)
some_txt<-gsub("[[:punct:]]"," ",some_txt)
some_txt<-gsub("[^[:alnum:]]"," ",some_txt)


text_corpus <- Corpus(VectorSource(some_txt))
text_corpus <- tm_map(text_corpus, removePunctuation)
text_corpus <- tm_map(text_corpus, content_transformer(tolower))
text_corpus <- tm_map(text_corpus, tm::removeWords, tm::stopwords('english'))
text_corpus <- tm_map(text_corpus, removeWords,
c("00bd","will","00a0","amp","00b8","looking","for?"))
corpus <- TermDocumentMatrix(text_corpus)
corpus <- as.matrix(corpus)
corpus <- sort(rowSums(corpus),decreasing=TRUE)

df <- data.frame(word = names(corpus),freq=corpus)

head(df, 20)
```

Output top 20 words with their frequency

	word	freq
	<fct>	<dbl>
demonetization	demonetization	14547
modi	modi	3105
india	india	3040
narendra	narendra	1566
rich	rich	1511
find	find	1422
dear	dear	1411
implement	implement	1400
actually	actually	1374
people	people	1218
bank	bank	1046
cash	cash	725
impact	impact	702
lakh	lakh	686
support	support	685
terrorists	terrorists	659
nation	nation	600
since	since	596
move	move	558
third	third	552

Most Popular Users

```
y=ddply(tweetsdata, .(screenName), numcolwise(sum))
popularUsers=y[,c("screenName","retweetCount","tweet")]
popularUsers=popularUsers[order(-popularUsers$retweetCount),]
popularUsers=head(popularUsers,n=10)
popularUsers
```

	screenName	retweetCount	tweet
	<chr>	<int>	<dbl>
1141	apoliceshanigm2	7677	2
5166	Krishna20977027	7677	2
7012	ParthPa07241800	5916	13
135	1SunnyElias	5170	1
8020	rayyat9tfoi	5170	1
9692	subhashjsr	5170	1
9942	sxP6DbxfufguCc0	5170	1
8509	sainath_kits	4280	11
8487	SahilBalu456	3772	2
9850	SurenderBalu1	3772	2

Most Replies

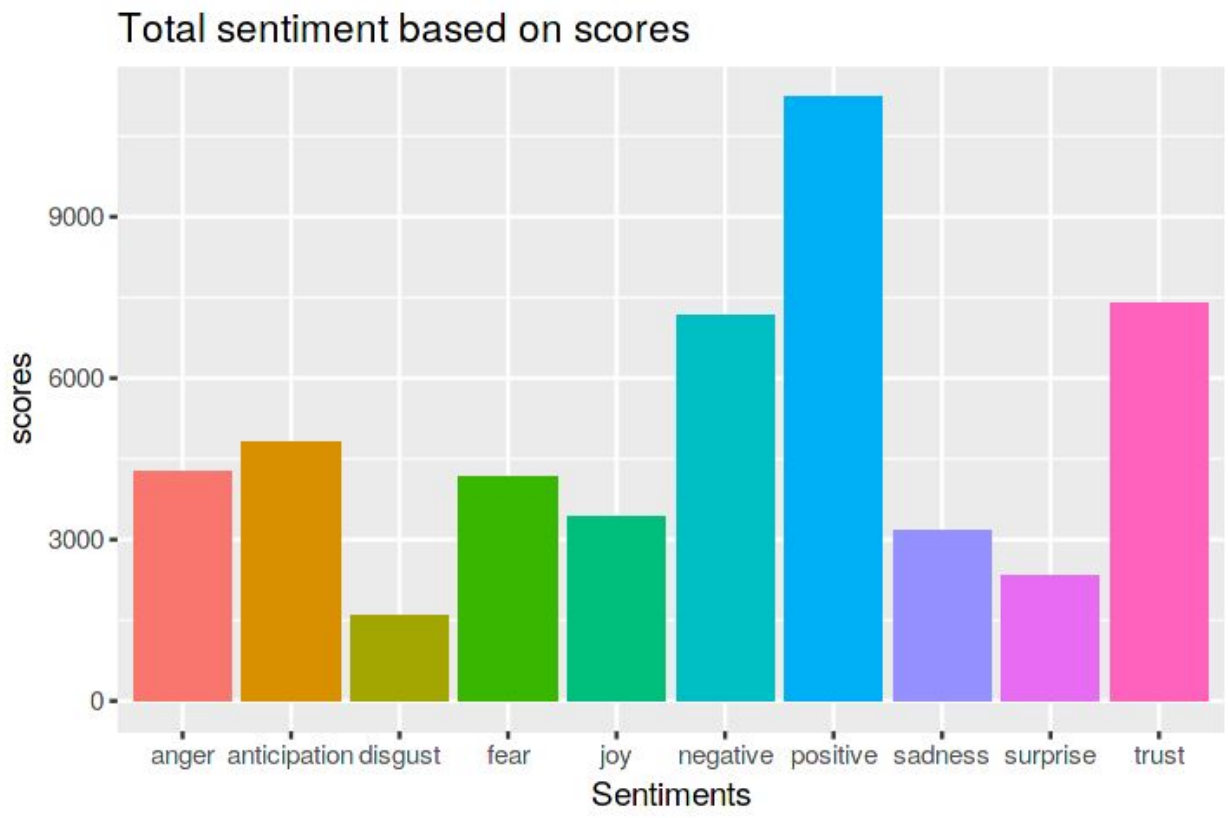
```
Replies=tweetsdata[is.na(tweetsdata$replyToSN)==FALSE,]
y=ddply(Replies, .(replyToSN), numcolwise(sum))
Replies=y[,c("replyToSN","tweet")]
Replies=Replies[order(-Replies$tweet),]
Replies=head(Replies,n=20)
colnames(Replies)=c("User","RepliesReceived")
Replies
```

	<chr>	<dbl>
369	narendramodi	77
421	PMOIndia	21
518	sardesairajdeep	17
66	ArvindKejriwal	16
114	centerofright	13
377	ndtv	12
49	ANI_news	10
608	timesofindia	10
125	CNNnews18	9
180	evanspiegel	9
558	Stupidosaur	9
264	jamewils	7
41	AmmU_MaanU	6
99	BJP4India	6
276	John_Miller_GLR	6
313	madmanweb	6
370	NarendraModi98	6
63	arunjaitley	5
153	digvijaya_28	5
161	dna	5

Perform sentiment analysis on the messages (use any suitable algorithm/library/package to do this)

#library already imported above

```
mysentiment<-get_nrc_sentiment((some_txt))
####used to classify sentiment scores
Sentimentscores<-data.frame(colSums(mysentiment[,]))
names(Sentimentscores)<-"Score"
SentimentScores<-cbind("sentiment"=rownames(Sentimentscores),Sentimentscores)
rownames(SentimentScores)<-NULL
ggplot(data=SentimentScores,aes(x=sentiment,y=Score))+geom_bar(aes(fill=sentiment),stat = "identity")+
  theme(legend.position="none")+
  xlab("Sentiments")+ylab("scores")+ggtitle("Total sentiment based on scores")
```



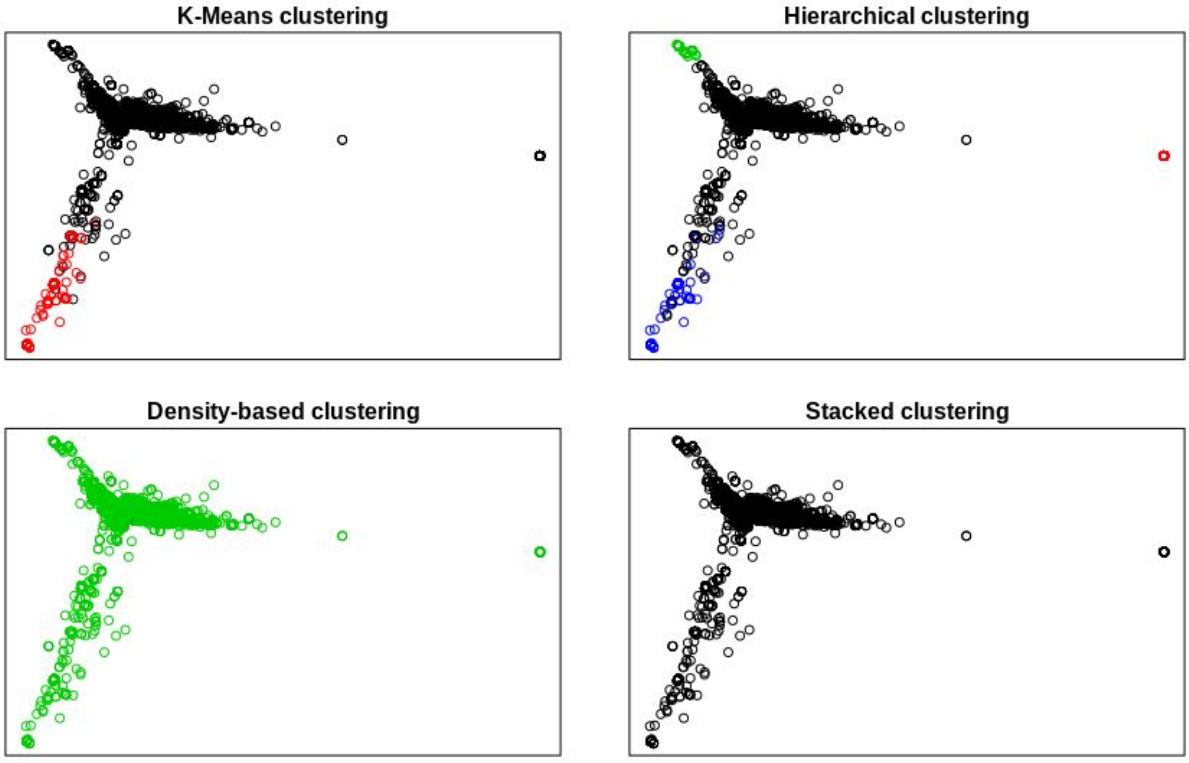
Discover several major classes of messages, i.e., cluster and/or classify them into a few groups of related messages (based on their textual content in addition to other attributes). Classification based only on timestamp, sender, etc. is not valid.

CLUSTERING

```
corpus = tm::Corpus(tm::VectorSource(some_txt))
corpus.cleaned <- tm::tm_map(corpus, function(x) iconv(x, to='UTF-8', sub='byte'))
corpus.cleaned <- tm::tm_map(corpus.cleaned, tm::removeWords,
tm::stopwords('english'))
corpus.cleaned <- tm::tm_map(corpus, tm::stemDocument, language = "english")
corpus.cleaned <- tm::tm_map(corpus.cleaned, tm::stripWhitespace)
```

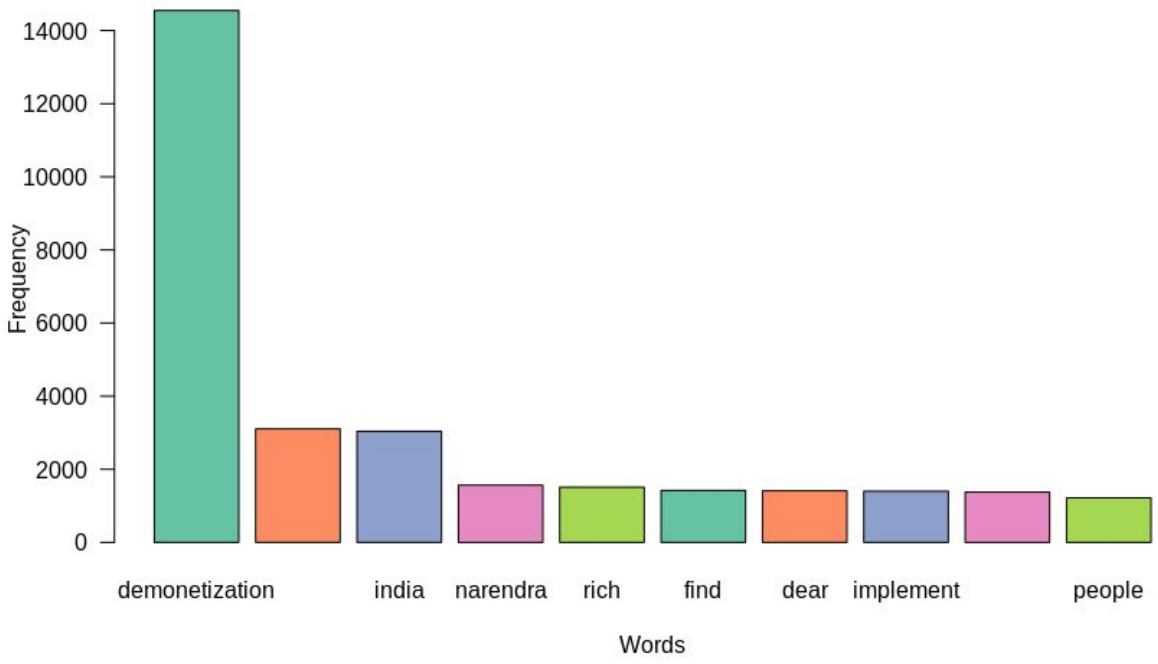
```
#DOING CLUSTER ANALYSIS ON ONLY 9000 CORPUSES
tdm <- tm::DocumentTermMatrix(corpus.cleaned[1:9000])
tdm.tfidf <- tm::weightTfIdf(tdm)
tdm.tfidf <- tm::removeSparseTerms(tdm.tfidf, 0.999)
tfidf.matrix <- as.matrix(tdm.tfidf)

dist.matrix = proxy::dist(tfidf.matrix, method = "cosine")

clustering.kmeans <- kmeans(tfidf.matrix,2,nstart=100)
clustering.hierarchical <- hclust(dist.matrix, method = "ward.D2")
clustering.dbscan <- dbscan::hdbscan(dist.matrix, minPts = 10)

master.cluster <- clustering.kmeans$cluster
slave.hierarchical <- cutree(clustering.hierarchical, k = 4)
slave.dbscan <- clustering.dbscan$cluster
stacked.clustering <- rep(NA, length(master.cluster))
names(stacked.clustering) <- 1:length(master.cluster)
for (cluster in unique(master.cluster)) {
  indexes = which(master.cluster == cluster, arr.ind = TRUE)
  slave1.votes <- table(slave.hierarchical[indexes])
  slave1.maxcount <- names(slave1.votes)[which.max(slave1.votes)]
  slave1.indexes = which(slave.hierarchical == slave1.maxcount, arr.ind =
TRUE)
  slave2.votes <- table(slave.dbscan[indexes])
  slave2.maxcount <- names(slave2.votes)[which.max(slave2.votes)]
  stacked.clustering[indexes] <- slave2.maxcount
}



points <- cmdscale(dist.matrix, k = 2)

palette <- colorspace::diverge_hcl(2) # Creating a color palette
previous.par <- par(mfrow=c(2,2), mar = rep(1.5, 4))
```

```
plot(points, main = 'K-Means clustering', col = as.factor(master.cluster),
    mai = c(0, 0, 0, 0), mar = c(0, 0, 0, 0),
    xaxt = 'n', yaxt = 'n', xlab = '', ylab = '')
plot(points, main = 'Hierarchical clustering', col =
as.factor(slave.hierarchical),
    mai = c(0, 0, 0, 0), mar = c(0, 0, 0, 0),
    xaxt = 'n', yaxt = 'n', xlab = '', ylab = '')
plot(points, main = 'Density-based clustering', col = as.factor(slave.dbscan),
    mai = c(0, 0, 0, 0), mar = c(0, 0, 0, 0),
    xaxt = 'n', yaxt = 'n', xlab = '', ylab = '')
plot(points, main = 'Stacked clustering', col = as.factor(stacked.clustering),
    mai = c(0, 0, 0, 0), mar = c(0, 0, 0, 0),
    xaxt = 'n', yaxt = 'n', xlab = '', ylab = '')
par(previous.par) # recovering the original plot space parameters
```

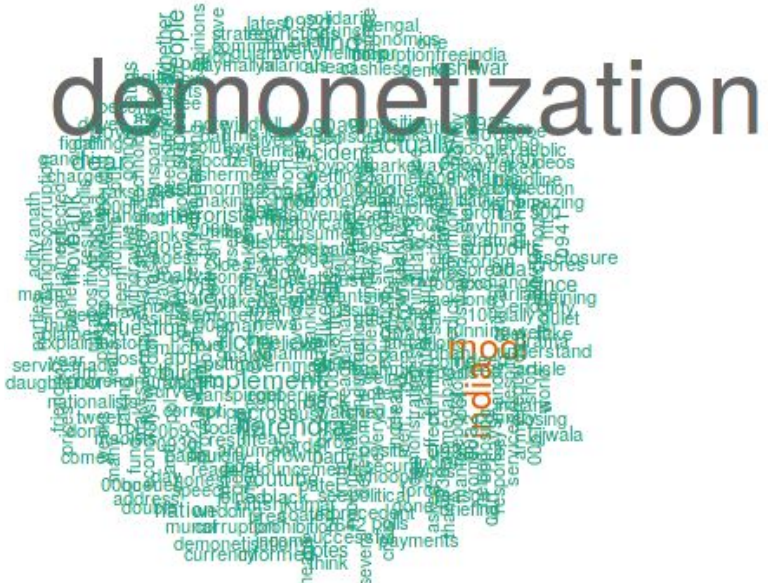


Generate some form of visualization of the messages: from the simplest of plots to the most sophisticated visualizations
PLOTS

```
text_corpus <- Corpus(VectorSource(some_txt))
text_corpus <- tm_map(text_corpus, removePunctuation)
text_corpus <- tm_map(text_corpus, content_transformer(tolower))
text_corpus <- tm_map(text_corpus, tm::removeWords, tm::stopwords('english'))
text_corpus <- tm_map(text_corpus, removeWords,
c("00bd","will","00a0","amp","00b8","looking","for?"))
corpus <- TermDocumentMatrix(text_corpus)
corpus <- as.matrix(corpus)
corpus <- sort(rowSums(corpus),decreasing=TRUE)

df <- data.frame(word = names(corpus),freq=corpus)

library(RColorBrewer)
coul <- brewer.pal(5, "Set2")
barplot(height=df[1:10,]$freq, names=df[1:10,]$word, col=coul ,horiz=F, las=1,xlab =
"Words",ylab ="Frequency")
```




```
wordcloud(text_corpus, min.freq = 50,
        max.words=1500, rot.per=0.35,
        colors=brewer.pal(8, "Dark2"), scale = c(3,0.5))
```



6. Test a hypothesis of your own, i.e., state a hypothesis (e.g., demonetization had majority popular support), based on the results of your text analytics, classification, sentiment analysis, etc. show that the hypothesis is either true or false. Can you show that your conclusion is statistically valid?

HYPOTHESIS TESTING

```
install.packages("ggpubr")
library(BSDA)

positive_mean = mean(mysentiment$positive)
sigmap = sd(mysentiment$positive)
negative_mean = mean(mysentiment$negative)
sigman = sd(mysentiment$negative)

countp = length(mysentiment$positive)
countn = length(mysentiment$negative)


z_calcs1 = zsum.test(mean.x = positive_mean,sigma.x = sigmap,n.x =
countp,mean.y = negative_mean,sigma.y =sigman,n.y = countn )
```

#Null hypothesis: H0: There is no significant difference between the means

```
print(z_calcs1)
```

```
        Two-sample z-Test

data:  Summarized x and y
z = 29.365, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.2526787 0.2888206
sample estimates:
mean of x mean of y
0.7512718 0.4805221
```

henceforth our null hypothesis rejected as their is significant difference between the means. Hence positive and negative belongs to different sample