Uttar Pradesh(India) Election Tweets

Background

Uttar Pradesh is the largest state in India. During 2017 elections we tried to predict election outcomes based on Twitter data.

There are 4 major parties in UP for election

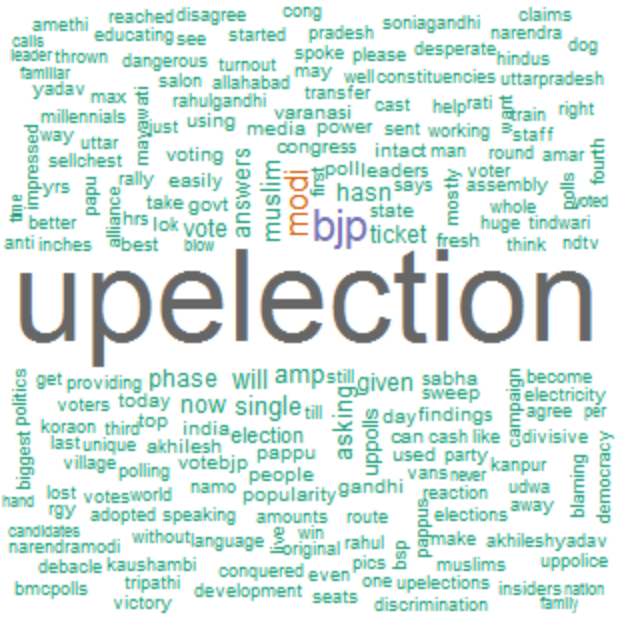
Party	Leader
BJP	Narendra Modi, Amit shah
SP	Akhilesh Yadav, Mulayam singh Yadav
BSP	Mayawati
INC	Rahul Gandhi, Sonia Gandhi

Scenerio- BSP and BJP decided to run election on their own. However, SP and INC formed a coalition. So for the purpose of analysis of sentiments we will consider tweets for both the parties (SP and INC) together. Till feburary 2017 only SP was having its government in UP.

Code

```
# libraries

library(dplyr)

library(plyr)

library(syuzhet)

library(SnowballC)

library(RCurl)

library(twitteR)

library(tm)

library(wordcloud)

# Downloading tweets

up_ele_tweets <- searchTwitter("#UPElection2017",n = 10000, lang = "en",resultType = "recent")

# writing tweets as dataframe

dpy <- ldply(up_ele_tweets,function(t) t$toDataFrame())

write.csv(dpy,"D:\\Documents\\Desktop\\Data Science\\Twitter\\dpy.csv")
```

```
# Getting text from dataframe

up_ele_tweets_text <- sapply(up_ele_tweets, function(x) x$getText())

write.csv(up_ele_tweets_text,"D:\\Documents\\Desktop\\Data
Science\\Twitter\\up_ele_tweets_text.csv")

# cleaning tweets using gsub to remove punctuation.

sum_txt1 <- gsub("(RT|via)((?:\\b\\w*@\\w+)+)","",up_ele_tweets_text)

sum_txt2 <- gsub("http[^[:blank:]]+","",sum_txt1)

sum_tx3 <- gsub("@\\w+","",sum_txt2)

sum_tx4 <- gsub("[[:punct:]]"," ", sum_tx3)

sum_tex5 <- gsub("[^[:alnum:]]", " ", sum_tx4)

sum_tx6 <- gsub("RT  ","", sum_tex5)

# data frame is not good for text so convert it corpus and cleaning

up_ele_corpus <- Corpus(VectorSource(sum_tx6))

up_ele_clean <- tm_map(up_ele_corpus, content_transformer(tolower)) #converting everything to
lower cases

up_ele_clean <- tm_map(up_ele_clean,removeWords, stopwords("english")) #stopword are words like
of, the, a, as..

up_ele_clean <- tm_map(up_ele_clean, removeNumbers)

up_ele_clean <- tm_map(up_ele_clean, stripWhitespace)

up_ele_clean <- tm_map(up_ele_clean, removeWords, "UPElection2017") # Removing #UPElection2017
as it is obviously will be there in

# Wordcloud

pal <- brewer.pal(8,"Dark2")

wordcloud(up_ele_clean,min.freq = 125,max.words = Inf, width=1000,height=1000,random.order  =
TRUE,colors = pal)
```

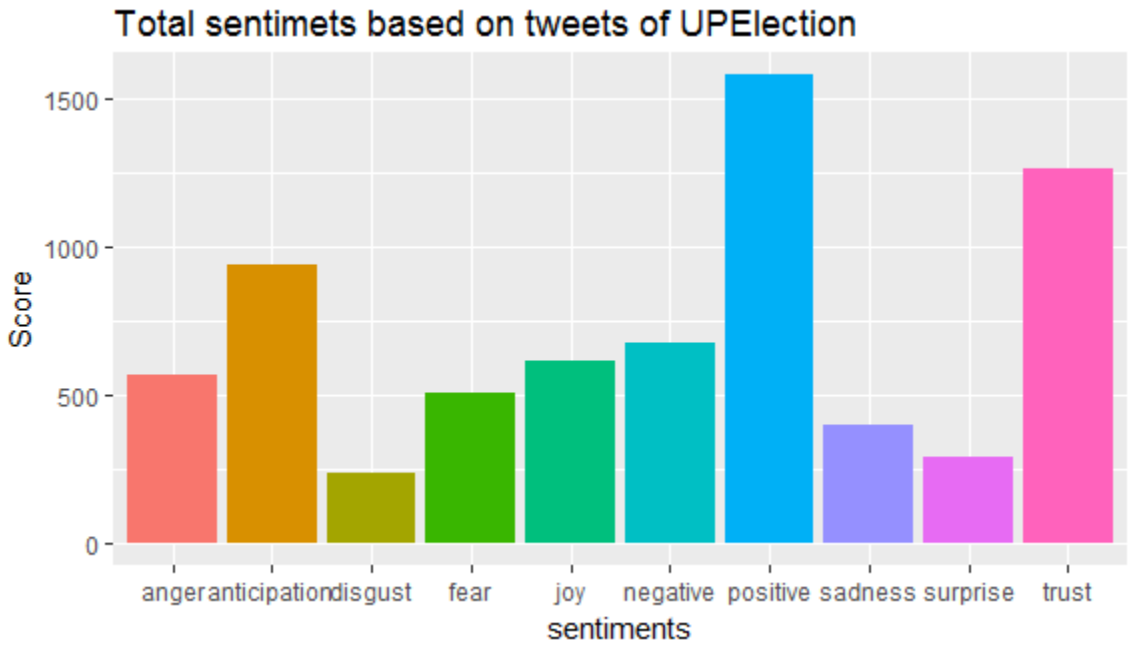


Removing word upelection and checking wordcloud

up_ele_clean_up_ele <- tm_map(up_ele_clean, removeWords, "upelection")

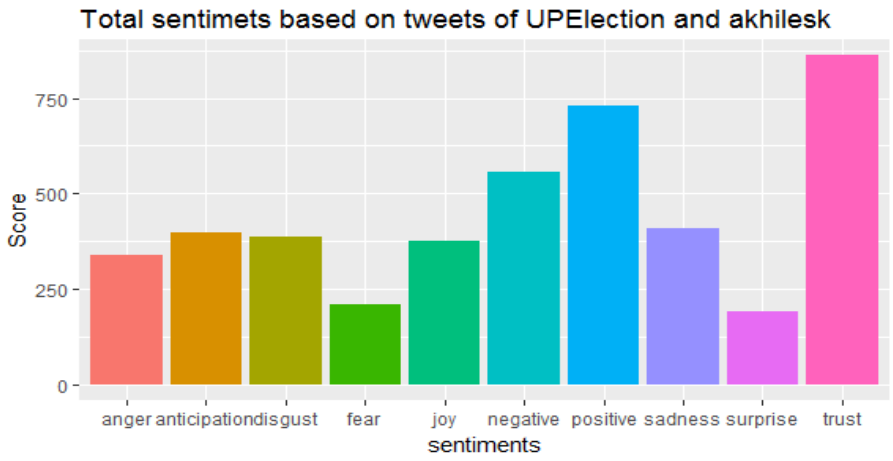
wordcloud(up_ele_clean_up_ele,min.freq = 75,max.words = Inf, width=1000,height=1000,random.order = FALSE,colors = pal)

```
# checking sentiments score

mysentiment <- get_nrc_sentiment(sum_tx6)

sentimentscore <- data.frame(colSums(mysentiment[,]))

names(sentimentscore) <- "Score"

sentimentscore <- cbind("sentiment" = rownames(sentimentscore), sentimentscore)

rownames(sentimentscore)<- NULL

# Plot sentiment

ggplot(data = sentimentscore,aes(x = sentiment,y = Score)) +

 geom_bar(aes(fill=sentiment),stat = "identity") +

 theme(legend.position = "none") +

 xlab("sentiments") +

 ylab("Score") +

 ggtitle("Total sentimets based on tweets of UPElection")
```

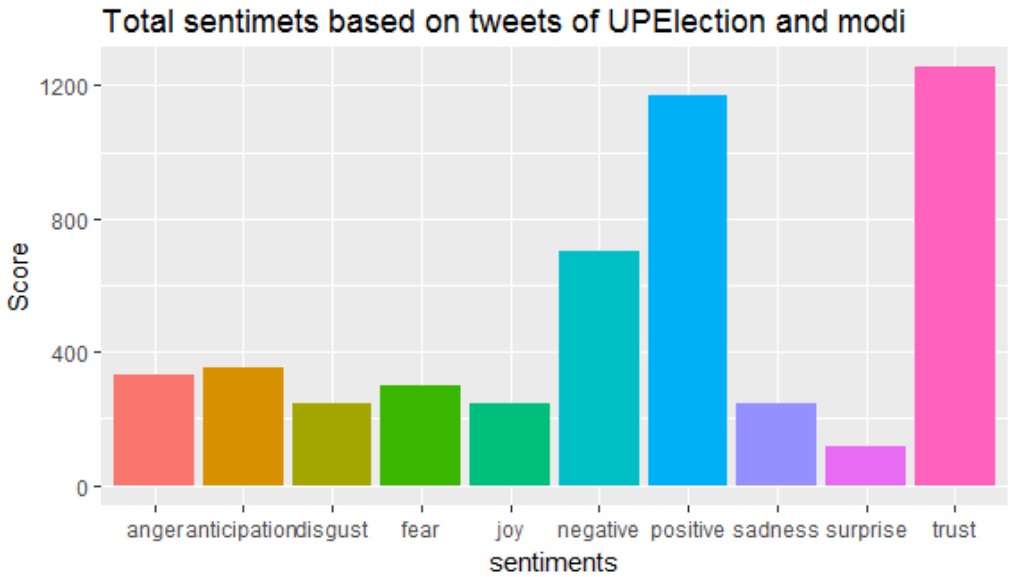


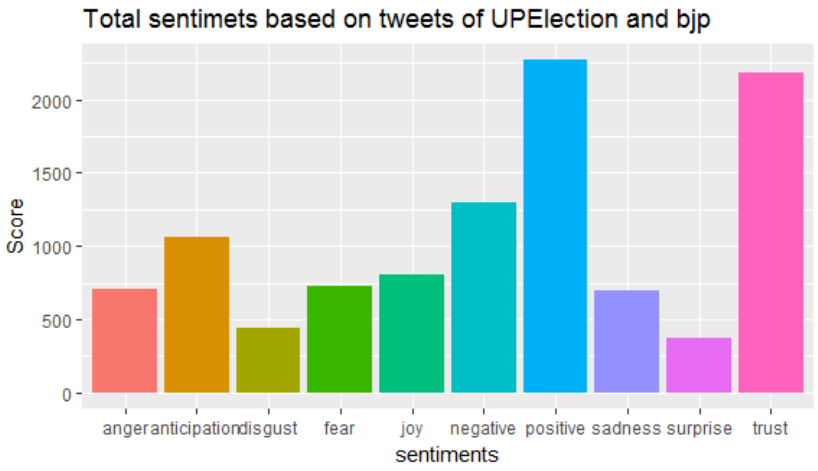
```
# checking word frequency

tdm <- TermDocumentMatrix(up_ele_clean)

mydata.df <- as.data.frame(inspect(tdm))
count<- as.data.frame(rowSums(mydata.df))
count$word = rownames(count)
colnames(count) <- c("count","word" )
count<-count[order(count$count, decreasing=TRUE), ]
```

Note: We fetch tweets of high frequency word like BJP, akhilesh, modi, congress, pappu, mayawati, bsp, rahul which contains **UP Election** as well, so we get sentiments regarding both.

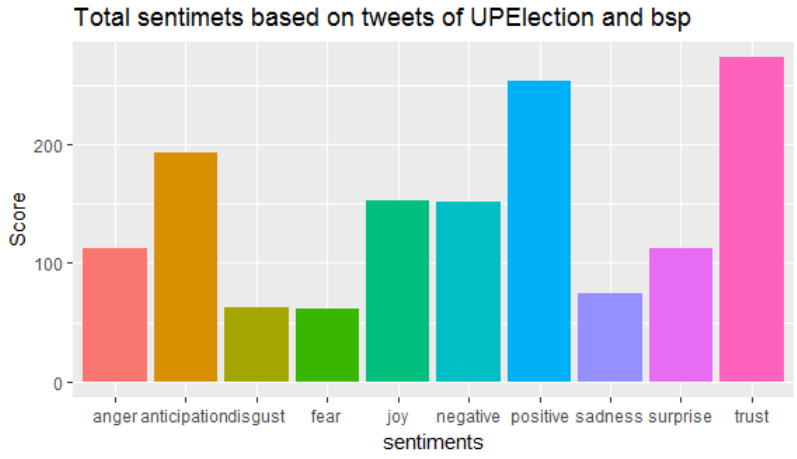


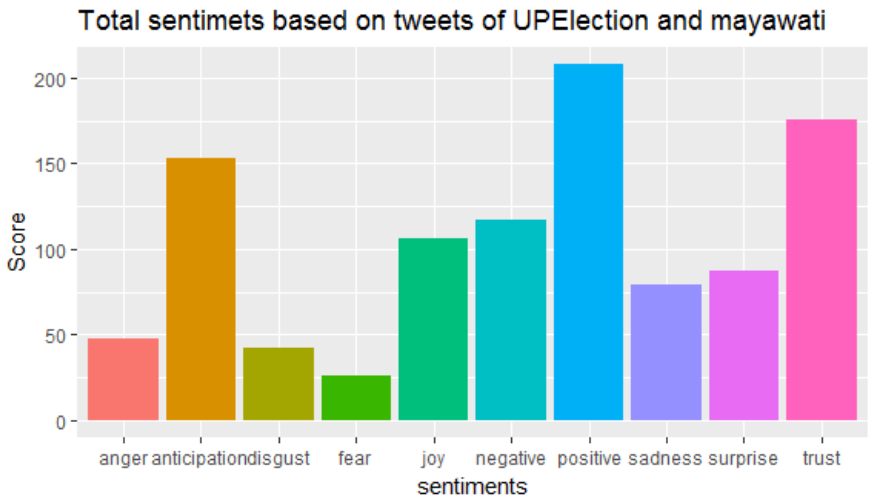
For SP majority of the sentiment is positive; however there is substantial negative sentiment as well.



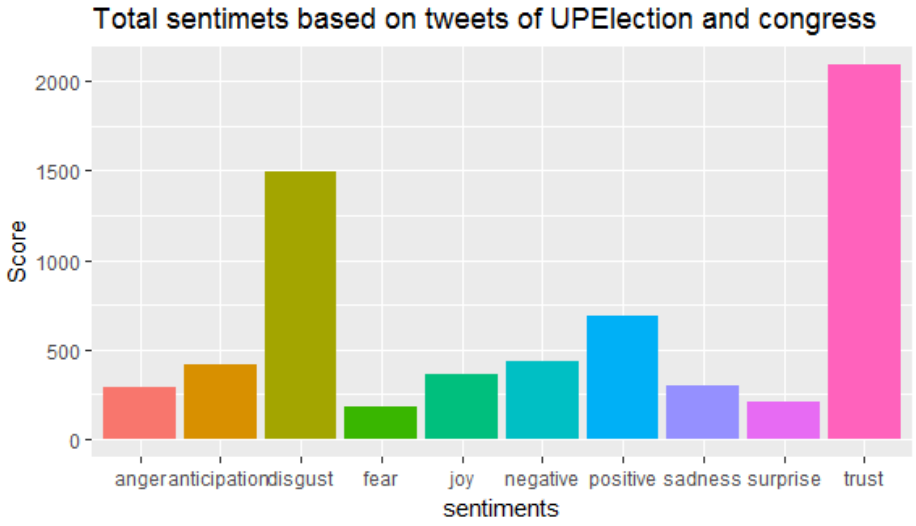


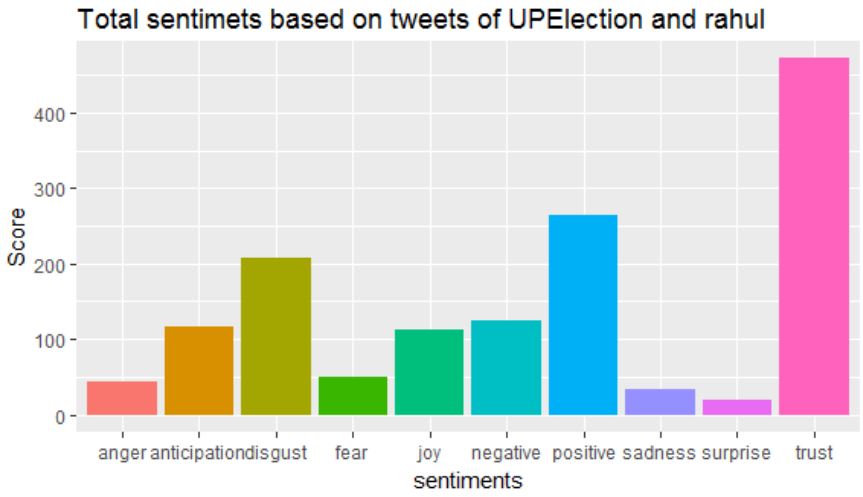
For MODI and BJP majority of the sentiment is positive and negative sentiment is relatively less.





For BSP and its leader people are tweeting mixed sentiments.





People trust congress and its leader but at the same time they are also showing disgust towards them. So far the best graph is shown by BJP. However to quantify sentiments I wrote the addition code.

Sntiment analysis Cont.. (code 2)

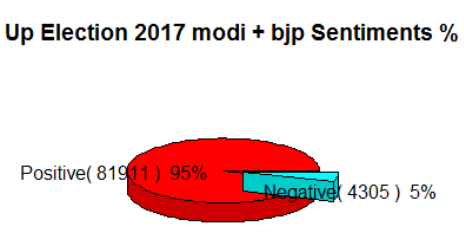
In last script we calculated overall sentiment, but failed to provide any numerical value to it.

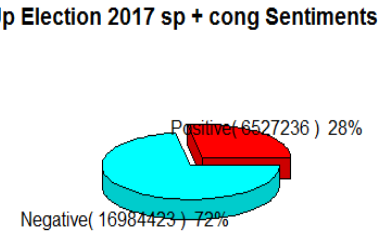
In this script we aim at calculating just positive and negative sentiment and assign percentage to each of them.

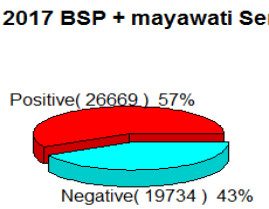
For this we used already available list of positive and negative words. By matching positive/negative words in the tweets with those words in list we calculated score and percentage.

To calculate winning chances of any party we clubbed all tweets related to that party.

For ex –
BJP + Modi + Amit Shah
Akhilesh + Sp + Rahul + Congress
Mayawati + BSP
Results ->

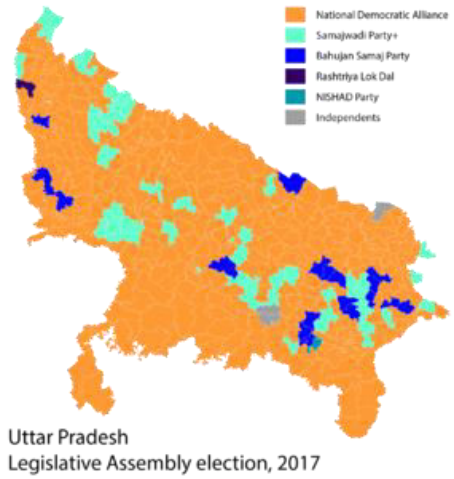






From the above graphs we can say that chances of winning for BJP is substantially high. While sp + congress coalation is expected to perform worst.

Results-



Party	Positive sentiment	Seats won out of 403
BJP (NDA)	95	312
SP + INC	28	47
BSP	57	19

Sentiment Analysis was successful to find out inclination of voters towards BJP and Modi. This was confirmed with election results.

NOTE -

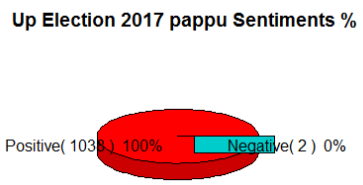
For sentiment Analysis we have selected tweets containing only English language. However, if we consider tweets from other regional languages we will get more accurate sentiment of people.

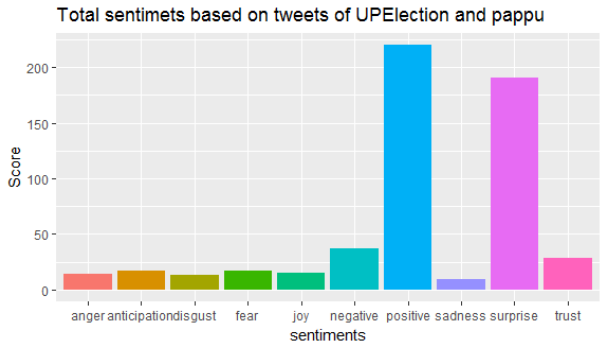
Sarcasm –

Sentiment Analysis Fails to detect Sarcasm.

"Pappu" is the name used by various leaders to mock Rahul Gandhi (leader of INC). However, we got almost 100 % positive sentiment for the word Pappu from both the codes. This positive sentiment is nothing but sarcasm.

While we removed punctuations from the tweets, it also removed emojis. Further work is required to use emojis to extract sentiments from tweets.





Research is still going on in this area of sentiment analysis.

Further work –

1- Geographical data can be collected from twitter and Sentiment analysis can be done for each seat in Election.

2- Model can be trained on the given data available and used during 2022 elections.