Technique that computers use
to extract worthwhile information
to extract Text worthwhile
information from the human
language Analysis in a smart
and efficient manner.

Text analysis in R and Python

Name : Modem Praveen
Branch : Computer Science and Engineering
Reg No : 17BCS035

Data Set : Tweets on demonetisation in India

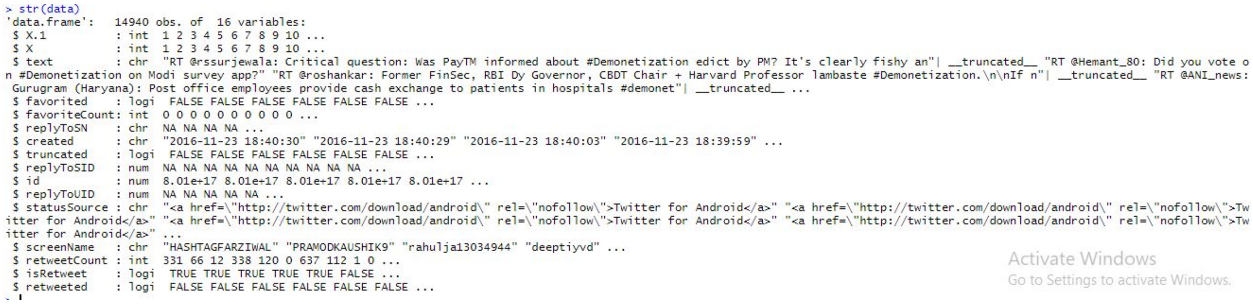
Path of dataset : Kaggle

Load dataset :

data <- read.csv(file.choose(), sep = ",", stringsAsFactors = FALSE) #Demonitization_tweets.csv

Note : we should load data by sep "," and stringsAsFactors = F

Structure of the data set:



```
> str(data)
'data.frame':   14940 obs. of  16 variables:
 $ X.1          : int  1 2 3 4 5 6 7 8 9 10 ...
 $ X            : int  1 2 3 4 5 6 7 8 9 10 ...
 $ text         : chr  "RT @rssurjewala: Critical question: Was PayTM informed about #Demonetization edict by PM? It's clearly fishy an"| __truncated__ "RT @Hemant_80: Did you vote o
n #Demonetization on Modi survey app?" "RT @roshankar: Former FinSec, RBI Dy Governor, CBDT Chair + Harvard Professor lambaste #Demonetization.\n\nIf n"| __truncated__ "RT @ANI_news:
 Gurugram (Haryana): Post office employees provide cash exchange to patients in hospitals #demonet"| __truncated__ ...
 $ favorited    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ favoriteCount: int  0 0 0 0 0 0 0 0 0 0 ...
 $ replyToSN    : chr  NA NA NA NA ...
 $ created      : chr  "2016-11-23 18:40:30" "2016-11-23 18:40:29" "2016-11-23 18:40:03" "2016-11-23 18:39:59" ...
 $ truncated    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
 $ replyToSID   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ id           : num  8.01e+17 8.01e+17 8.01e+17 8.01e+17 8.01e+17 ...
 $ replyToUID   : num  NA NA NA NA NA ...
 $ statusSource : chr  "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>" "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Tw
itter for Android</a>" "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>" "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Tw
itter for Android</a>" ...
 $ screenName   : chr  "HASHTAGFARZIWAL" "PRAMODKAUSHIK9" "rahulja13034944" "deeptiyvd" ...
 $ retweetCount : int  331 66 12 338 120 0 637 112 1 0 ...
 $ isRetweet    : logi  TRUE TRUE TRUE TRUE TRUE FALSE ...
 $ retweeted    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
>
```

str(data)

Data cleaning :

- Before going ahead we should install ggplot2 , dplyr , tidytext , igraph, ggraph ,widyr , tidyr , wordcloud ,SnowballC ,tidyverse , topicmodels , RTextTools , tm ,syuzhet

```
clean_tweets <- function(x) {
  x %>%
  str_remove_all("@[[:alnum:]]+") %>%
  str_remove_all("\\<U[^\\>]*\\>") %>%
  str_remove_all(" ?(f|ht)(tp)(s?)(://)(.*)[.|/](.*)") %>%

str_remove_all("(^|[^&\\p{L}\\p{M}\\p{Nd}_\u200c\u200d\ua67e\u05be\u05f3\u05f4\u309b\u309c\u30a0\u30fb\u3003\u0f0b\u0f0c\u00b7])(#|\uFF03)(?!\uFE0F|\u20E3)([\\p{L}\\p{M}\\p{Nd}_\u200c\u200d\ua67e\u05be\u05f3\u05f4\u309b\u309c\u30a0\u30fb\u3003\u0f0b\u0f0c\u00b7]*[\\p{L}\\p{M}][\\p{L}\\p{M}\\p{Nd}_\u200c\u200d\ua67e\u05be\u05f3\u05f4\u309b\u309c\u30a0\u30fb\u3003\u0f0b\u0f0c\u00b7]*)") %>%
  str_replace_all("&amp;", "and") %>%
  str_remove_all("[[:punct:]]") %>%
  str_remove_all("^RT:? ") %>%
  str_remove_all("#[[:alnum:]]+") %>%
  str_remove_all("[[:digit:]]+") %>%
  str_replace_all("\\\n", " ") %>%
  str_to_lower() %>%
  str_trim("both")
}
data$cleaned <- clean_tweets(data$text)
```

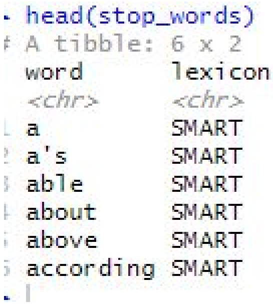
- We are replacing it with a cleaned column.

Tokenization :

```
data_clean <- data %>%

 dplyr::select(ctext) %>%

 unnest_tokens(word, ctext) #Tokenization
```

Stop words:

```
data("stop_words")

head(stop_words)
```

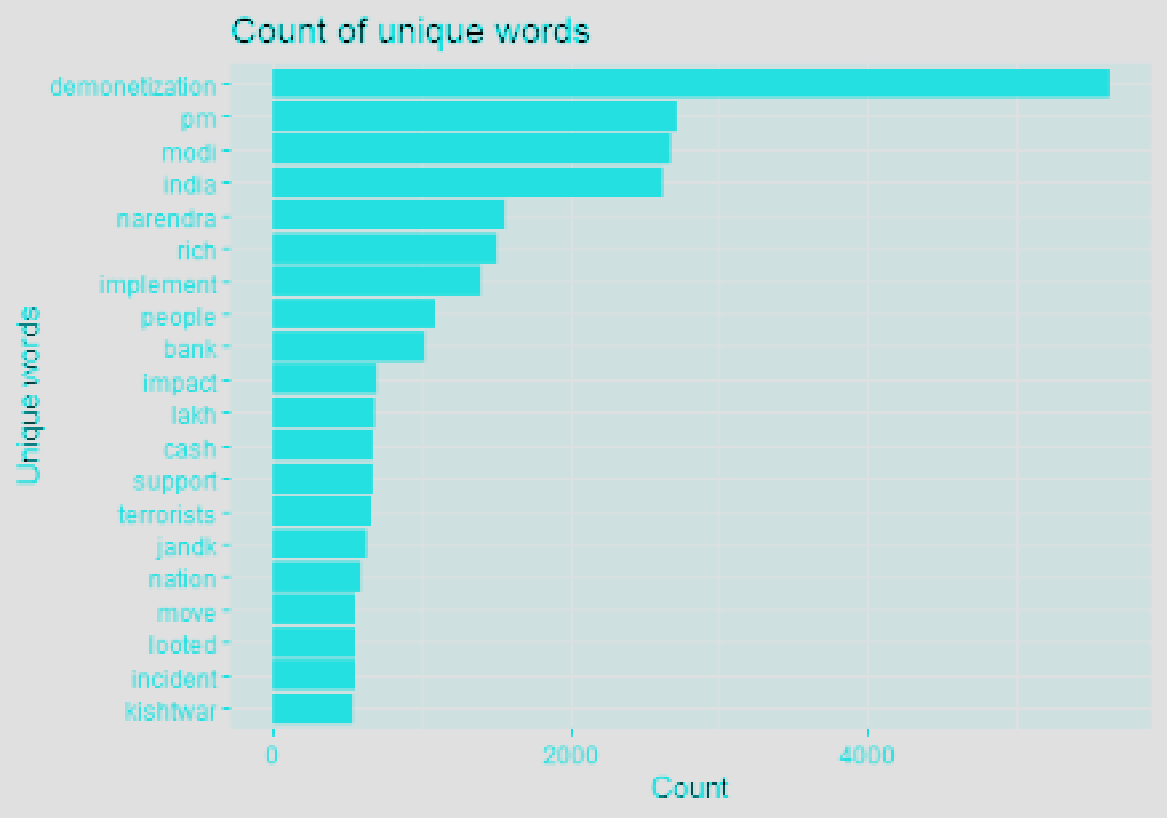


```
> head(stop_words)
# A tibble: 6 x 2
  word      lexicon
  <chr>     <chr>
1 a         SMART
2 a's       SMART
3 able      SMART
4 about     SMART
5 above     SMART
6 according SMART
> |
```

Cleaning using Stop words :

```
cleaned_tweets <- data_clean %>%
  anti_join(stop_words) %>%
  filter(!word %in% tolower(data$screenName)) %>%
  filter(!word == "ed") %>%
  filter(!word == "dear") %>%
  filter(!word == "httpst") %>%
  filter(!word == "https") %>%
  filter(!word == "dont") %>%
  filter(!word == "put") %>%
  filter(!word == "urautelaforever") %>%
  filter(!word == "rs")
```

Unique words plot :

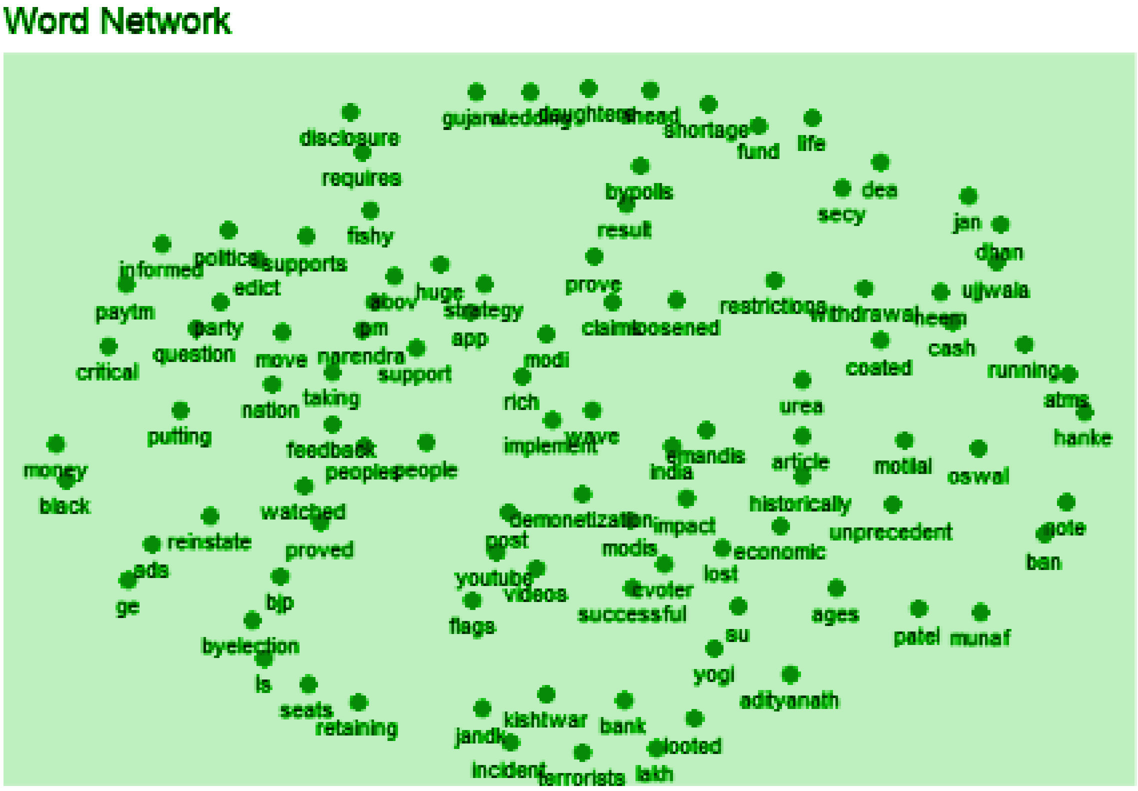


Little more preprocessing :

```
data_paired <- cleaned_tweets %>%

  dplyr::select(word) %>%

  unnest_tokens(paired_words, word, token = "ngrams", n = 2)

data_paired %>%

  dplyr::count(paired_words, sort = TRUE)
```

```
data_separated <- data_paired %>%
  separate(paired_words, c("word1", "word2"), sep = " ")
data_filtered <- data_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)
data_words_count <- data_filtered %>%
  dplyr::count(word1, word2, sort = TRUE)
```

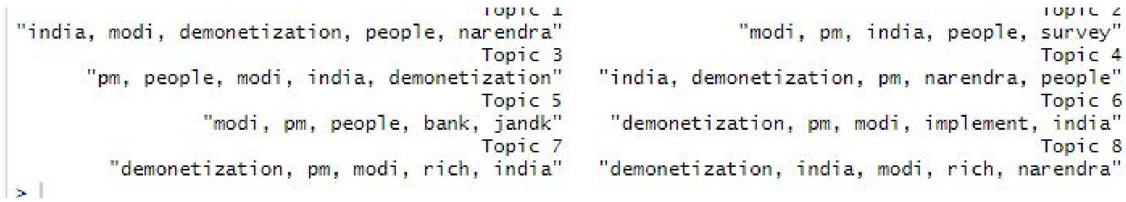
Word Network Plot



Tweets classification:

```
cl <- VCorpus(VectorSource(cleaned_tweets))

td <- TermDocumentMatrix(cl, control = list(wordLengths = c(1, Inf)))

dt <- as.DocumentTermMatrix(td)

lda <- LDA(dt, k = 8)

term <- terms(lda, 5)

(term <- apply(term, MARGIN = 2, paste, collapse = ", "))
```

Output:



```
                                         Topic 1                                              Topic 2
"india, modi, demonetization, people, narendra"                "modi, pm, india, people, survey"
                                         Topic 3                                              Topic 4
     "pm, people, modi, india, demonetization"   "india, demonetization, pm, narendra, people"
                                         Topic 5                                              Topic 6
               "modi, pm, people, bank, jandk"    "demonetization, pm, modi, implement, india"
                                         Topic 7                                              Topic 8
      "demonetization, pm, modi, rich, india"   "demonetization, india, modi, rich, narendra"
> 
```

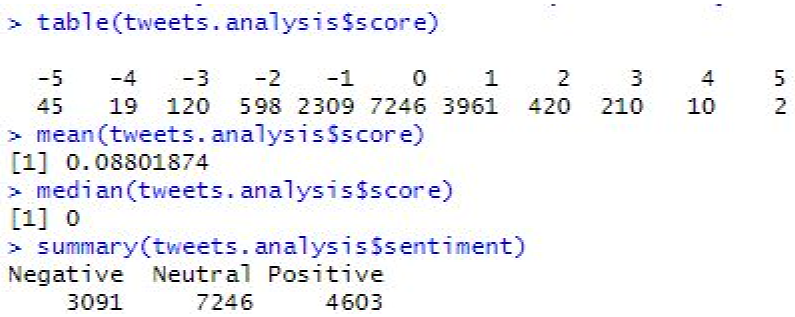
Sentimental analysis:

Load data:

```
tweets <-as.character(data$ctext)

pos <- scan(file.choose(), what= "character", comment.char= ";")

neg <- scan(file.choose(), what= "character", comment.char= ";")
```

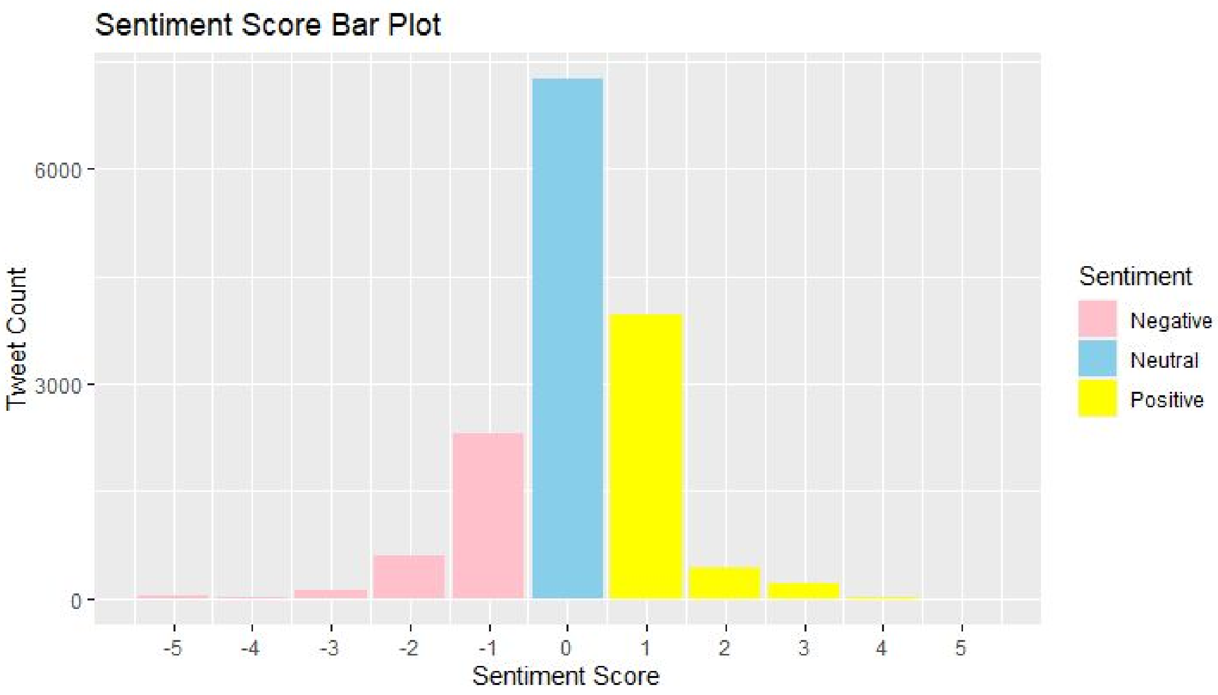
```
sent.score <- function(sentences, pos.words, neg.words, .progress='none')
{
 require(plyr)
 require(stringr)
  scores <- laply(sentences, function(sentence, pos.words, neg.words)
 {
  sentence <- gsub('[[:cntrl:]]', '', sentence)
  sentence <- gsub('(RT|via)((?:\\b\\W*@\\W+)+)', '', sentence)
  sentence <- gsub('http.*','',  sentence)
  sentence <- gsub('https.*','',  sentence)
  sentence <- gsub('@\\w+', '', sentence)
  sentence <- gsub('[[:punct:]]', '', sentence)
  sentence <- gsub('[[:digit:]]', '', sentence)
  sentence <- gsub('http[s]?\\w+', '', sentence)
  sentence <- gsub('[ \t]{2,}', '', sentence)
  sentence <- gsub('^\\s+|\\s+$', '', sentence)
  sentence <- sentence[!is.na(sentence)]
  sentence <- tolower(sentence)
  word.list <- str_split(sentence, '\\s+')
  words <- unlist(word.list)
  neg.matches <- match(words, neg.words)
  pos.matches <- match(words, pos.words)
```

```
    pos.matches <- !is.na(pos.matches)

    neg.matches <- !is.na(neg.matches)



    score <- sum(pos.matches) - sum(neg.matches)



    return(score)

  }, pos.words, neg.words, .progress=.progress )



  scr.df <- data.frame(score=scores, text=sentences)

  return(scr.df)

}
```



```
> table(tweets.analysis$score)

  -5   -4   -3   -2   -1    0    1    2    3    4    5 
  45   19  120  598 2309 7246 3961  420  210   10    2 
> mean(tweets.analysis$score)
[1] 0.08801874
> median(tweets.analysis$score)
[1] 0
> summary(tweets.analysis$sentiment)
Negative  Neutral Positive 
    3091     7246     4603 
```

Plot



Emotion classification :

```
wrd.df <- as.vector(data$ctext)

emotion.df <- get_nrc_sentiment(wrd.df)

emotion.df2 <- cbind(data$ctext, emotion.df)
```

Hypothesis Testing

Most of the positive tweets support towards demonetisation was from those who anticipated it.

Assuming the distribution of the populations to be normal

Null hypothesis H0: No significant difference between both means # Test statistic Z,

```
z_two = function(mu1, mu2, sigma1, sigma2, n1, n2){
  zt = (mu1-mu2)/sqrt(sigma1^2/n1+sigma2^2/n2)
  return(zt)
}
#sample means

mu1 <- mean(emotion_sample$anticipation)
mu2 <- mean(emotion_sample$positive)

#sample sizes
n1 <- length(emotion_sample$anticipation)
n2 <- length(emotion_sample$positive)
#sample variances
```

```
sigma1_m <- var(emotion_sample$anticipation)
sigma2_m <- var(emotion_sample$positive)

# Calculating value of Z
z2_calu <- z_two(mu1, mu2, sigma1_m, sigma2_m, n1, n2)
print(z2_calu)
z2_calum <- abs(z2_calu)

 Critical value of z for 5% LOS
z_cri_5 = 1.96
print(z_cri_5)

Decision on null hypothesis
if (z2_calum > z_cri_5){
  print ("Reject H0")
  print("Statistically validated")
} else {
  print ("Accept H0")
  print("Statistically validated")
}
```

Reject H0 : Both means are significantly different