Quiz_1_2_KeerthiRBollam

```
## Question 1: Tweets analysis for Donald Trump @ POTUS and creating the word
cloud
library(twitteR)

## Warning: package 'twitteR' was built under R version 3.3.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.3.3

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:twitteR':
##
##     id, location

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

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

library(ggplot2)

setup_twitter_oauth(api_key, api_secret, token, token_secret)

## [1] "Using direct authentication"

trump = userTimeline('POTUS', n=50)

## converting to data frame ##
df.trump = twListToDF(trump)

## String operations
library(stringr)

## Warning: package 'stringr' was built under R version 3.3.3

library(stringi)

## Warning: package 'stringi' was built under R version 3.3.3

words_list = str_split(df.trump$text, ' ')
## Each tweet is consdiering as a list
words_list[1]
```

```
## [[1]]
## [1] "Weekly"                    "Address"
## [3] "-"                         "tune"
## [5] "in!"                       "#MAGA"
## [7] "<U+27A1><U+FE0F>https://t.co/hJNQcZQrbx" "https://t.co/x3Kw88f0De"

words_list[[1]][1]

## [1] "Weekly"

## Collate all the lists at one place
words_list = str_split(df.trump$text, ' ')
allwords=unlist(words_list)
length(allwords)

## [1] 120

df.allwords = as.data.frame(table(allwords))
head(df.allwords)

##                    allwords Freq
## 1                         -    1
## 2                    \n@POTUS    1
## 3 \nhttps://t.co/qtaHFoGXkf    2
## 4                     "attack    1
## 5                        "in    1
## 6                    "renewed    1

df.allwords = df.allwords %>% arrange(-Freq)

library(tm)

## Warning: package 'tm' was built under R version 3.3.3

## Loading required package: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':
##
##     annotate

common_stopwords = stopwords()
custom_stopwords = c('&amp;')
all_stopwords = c(common_stopwords, custom_stopwords)
df.allwords = df.allwords[! df.allwords$allwords %in% all_stopwords,]

print(df.allwords)

##                                  allwords Freq
## 3                                  .@POTUS    3
## 7                \nhttps://t.co/qtaHFoGXkf    2
```

```
## 8                                    Address    2
## 10                                     defeat    2
## 12                                  Manchester    2
## 14                                   terrorism    2
## 15                                         The    2
## 16                                          UK    2
## 17                                      Weekly    2
## 19                                           -    1
## 20                                    \n@POTUS    1
## 21                                     "attack    1
## 22                                         "in    1
## 23                                    "renewed    1
## 24                                        "The    1
## 25                                       #MAGA    1
## 26                           @realDonaldTrump    1
## 27 <U+27A1><U+FE0F>https://t.co/hJNQcZQrbx    1
## 28 <U+27A1><U+FE0F>https://t.co/lShUEgXOPD    1
## 31                                     achieve    1
## 33                                     attacks    1
## 34                                      brings    1
## 35                                      Budget    1
## 36                                         can    1
## 37                                     deliver    1
## 38                                demonstrates    1
## 39                                      depths    1
## 40                                determination    1
## 41                                        evil    1
## 42                                        face    1
## 43                                      faiths    1
## 44                                       First    1
## 45                                       focus    1
## 46                                      Friday    1
## 47                                      future    1
## 48                                       great    1
## 49                                        hate    1
## 50                                        hope    1
## 51                     https://t.co/6tcy4SmQFh    1
## 52                     https://t.co/9Xqsa3czFr    1
## 53                     https://t.co/qtaHFoGXkf    1
## 54                     https://t.co/x3Kw88f0De    1
## 55                                           I    1
## 56                                immigration"    1
## 57                                         in!    1
## 58                                     include    1
## 59                                intolerance.    1
## 61                                        Join    1
## 62                                     lasting    1
## 63                                        many    1
## 66                                        must    1
## 67                                     nations    1
```

```
## 68                                      NATO    1
## 69                                       Our    1
## 70                                  Portland    1
## 71                                   prayers    1
## 72                                      read    1
## 73                                    right:    1
## 74                                 security"    1
## 75                                  standing    1
## 76                                    stands    1
## 77                                  Taxpayer    1
## 78                                terrorism"    1
## 79                               terrorism."    1
## 81                                     them.    1
## 82                                      trip    1
## 83                                      tune    1
## 84                             unacceptable.    1
## 85                                     unite    1
## 87                                   victims    1
## 88                                   violent    1
## 89                                        w/    1

#### creating the WordCloud ####

library(wordcloud)

## Warning: package 'wordcloud' was built under R version 3.3.3

## Loading required package: RColorBrewer

wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq = 1)

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## https://t.co/x3Kw88f0De could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## https://t.co/qtaHFoGXkf could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## "renewed could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## #MAGA could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## Taxpayer could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## Budget could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## https://t.co/6tcy4SmQFh could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## https://t.co/qtaHFoGXkf could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## standing could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## <U+27A1><U+FE0F>https://t.co/lShUEgXOPD could not be fit on page. It will
## not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## unite could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## unacceptable. could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## right: could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## achieve could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## great could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## immigration" could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## https://t.co/9Xqsa3czFr could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## UK could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## violent could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## NATO could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## @realDonaldTrump could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## terrorism" could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## Manchester could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## The could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## lasting could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## intolerance. could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## tune could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## Weekly could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
## 1): .@POTUS could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## Join could not be fit on page. It will not be plotted.

## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## hope could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(df.allwords$allwords, df.allwords$Freq, min.freq =
1):
## terrorism." could not be fit on page. It will not be plotted.
```


```
## END ##

# Question Number 2 : Use date column to check day wise no of tweets. Plot
line chart

Tweets <- read.csv("C:/Users/Kikku/Google Drive/1 Data Science/11
unstructured data analytics/Lab/potus_tweets.csv")
head(Tweets)

##   X
## 1 1
## 2 2
## 3 3
## 4 4
## 5 5
## 6 6
##
text
## 1                                                    Weekly Address -
tune in! #MAGA <U+27A1><U+FE0F>https://t.co/hJNQcZQrbx
https://t.co/x3Kw88f0De
## 2           The violent attacks in Portland on Friday are unacceptable.
The victims were standing up to hate and intolerance. Our prayers are w/
```

```
them.
## 3 Join me as I deliver the Weekly Address and read more about our Taxpayer
First Budget <U+27A1><U+FE0F>https://t.co/lShUEgXOPD https://t.co/9Xqsa3czFr
## 4                  "The NATO of the future must include a great focus on
terrorism and immigration" \n@POTUS @realDonaldTrump
\nhttps://t.co/qtaHFoGXkf
## 5                .@POTUS is right: "attack on Manchester in the UK
demonstrates the depths of the evil we face with terrorism."
https://t.co/qtaHFoGXkf
## 6                                  .@POTUS trip brings "renewed hope that
nations of many faiths can unite to defeat terrorism"
\nhttps://t.co/qtaHFoGXkf
##   favorited favoriteCount replyToSN             created truncated
## 1     FALSE          9230        NA 2017-06-02 23:18:46     FALSE
## 2     FALSE         55101        NA 2017-05-29 14:51:00     FALSE
## 3     FALSE         10182        NA 2017-05-26 16:48:49     FALSE
## 4     FALSE         19716        NA 2017-05-25 20:25:15     FALSE
## 5     FALSE         15326        NA 2017-05-25 20:23:54     FALSE
## 6     FALSE         15294        NA 2017-05-25 20:23:21     FALSE
##   replyToSID           id replyToUID
## 1         NA 8.707818e+17         NA
## 2         NA 8.692044e+17         NA
## 3         NA 8.681469e+17         NA
## 4         NA 8.678390e+17         NA
## 5         NA 8.678387e+17         NA
## 6         NA 8.678385e+17         NA
##
statusSource
## 1                 <a href="http://twitter.com" rel="nofollow">Twitter Web
Client</a>
## 2 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for
iPhone</a>
## 3                 <a href="http://twitter.com" rel="nofollow">Twitter Web
Client</a>
## 4 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for
iPhone</a>
## 5 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for
iPhone</a>
## 6 <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for
iPhone</a>
##   screenName retweetCount isRetweet retweeted longitude latitude
## 1      POTUS         2058     FALSE     FALSE        NA       NA
## 2      POTUS        12457     FALSE     FALSE        NA       NA
## 3      POTUS         2244     FALSE     FALSE        NA       NA
## 4      POTUS         5023     FALSE     FALSE        NA       NA
## 5      POTUS         3651     FALSE     FALSE        NA       NA
## 6      POTUS         3380     FALSE     FALSE        NA       NA
```

```r
#convert the timestamp column in to date time object
Tweets$timestamp = as.Date(Tweets$created)
head(Tweets$timestamp)
```

```
## [1] "2017-06-02" "2017-05-29" "2017-05-26" "2017-05-25" "2017-05-25"
## [6] "2017-05-25"
```

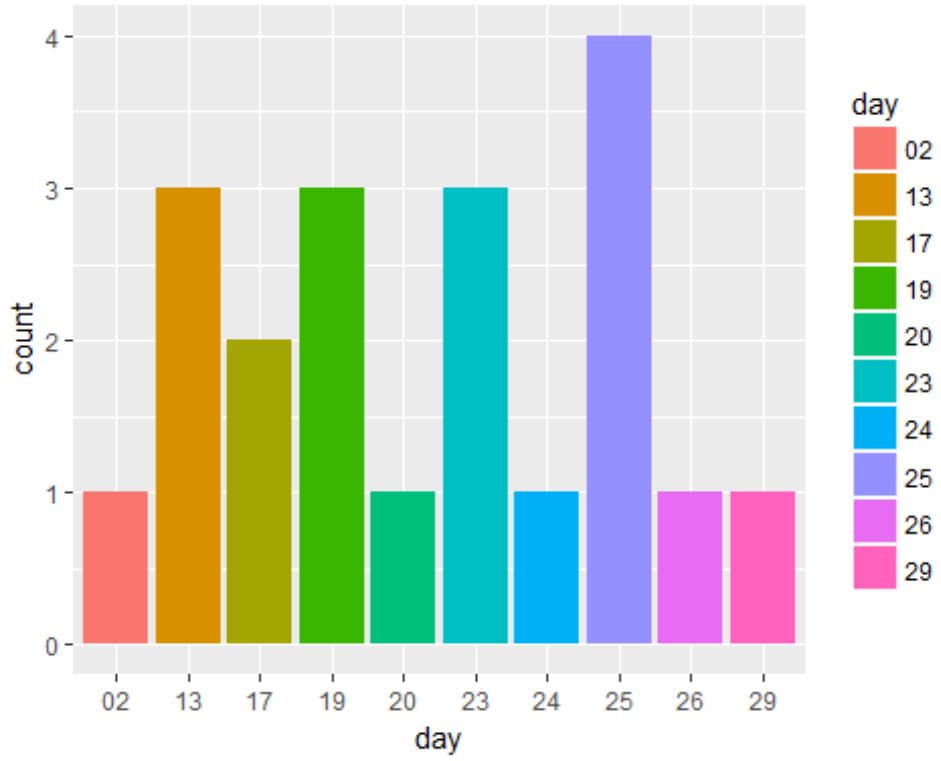
```r
## Extract date as per the requirement
Tweets$day = format(Tweets$timestamp, "%d")
Tweets$day
```

```
##  [1] "02" "29" "26" "25" "25" "25" "25" "24" "23" "23" "23" "20" "19" "19"
## [15] "19" "17" "17" "13" "13" "13"
```

```r
## Day wise number of Tweets
tweets_day = Tweets %>% group_by(day) %>% summarise(count=n())
tweets_day
```

```
## # A tibble: 10 × 2
##      day count
##    <chr> <int>
## 1     02     1
## 2     13     3
## 3     17     2
## 4     19     3
## 5     20     1
## 6     23     3
## 7     24     1
## 8     25     4
## 9     26     1
## 10    29     1
```

```r
## Plotting a bar plot as per the requirement

bar_day <- ggplot(tweets_day, aes(x=day, y=count)) + geom_bar(stat =
"identity", aes(fill = day))
bar_day
```

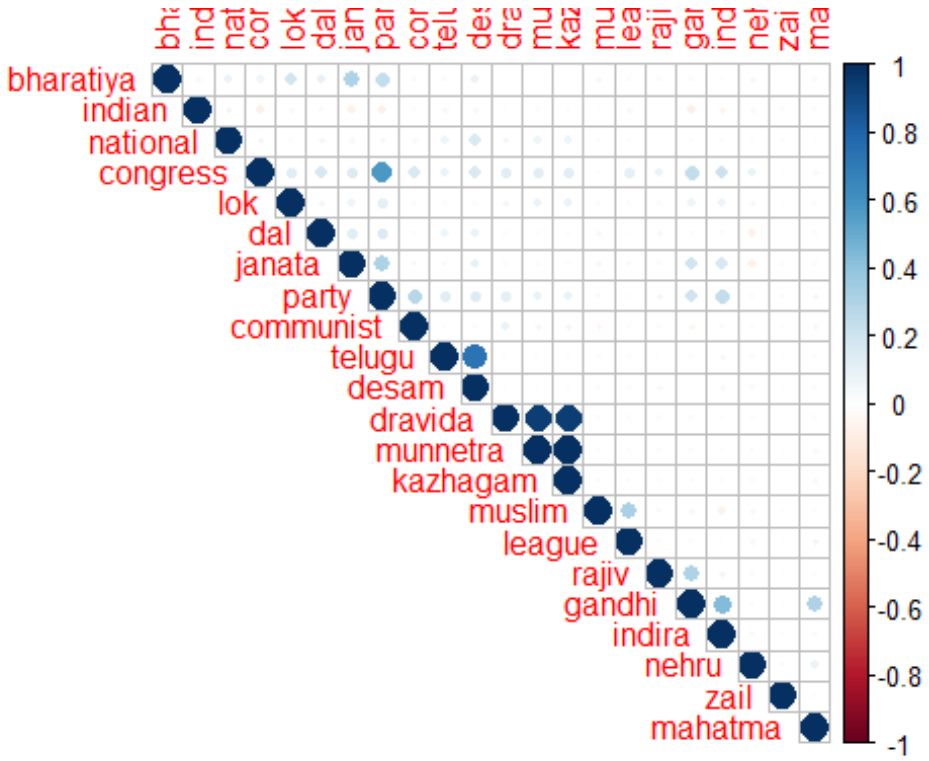



```
## END of Q2 ##

################## Question 3 : Read the book (India after Gandhi)
####################

library(pdftools)

## Warning: package 'pdftools' was built under R version 3.3.3

setwd('C:/Users/Kikku/Google Drive/1 Data Science/11 unstructured data
analytics/Lab/Quizes')
book_text = pdf_text("india-after-gandhi.pdf")

#Removing all the special characters
book_text_transformed = gsub("[^A-Za-z///' ]", "", book_text)

################## Question 3a: Printing total number of documents
##################
docs = Corpus(VectorSource(book_text_transformed))

print(docs)

## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 770
```

```
################ Question 3c : Compute TDM and DTM ####################

#cleaning corpus before calculating TDM and DTM
docs <- tm_map(docs, content_transformer(tolower))
docs <- tm_map(docs, removeWords, stopwords())

#compute TDM
tdm <- TermDocumentMatrix(docs)
words <- as.matrix(tdm)

# compute DTM
dtm = DocumentTermMatrix(docs)
dtm = as.matrix(dtm)

##############Question 3b : Calculate tot number of words identified by TDM
#######################

words_freq <- as.data.frame(rowSums(words))
nrow(words_freq)

## [1] 31906

############# Question 3d: Identify bottom 15 words that has appeared
################

words_freq <- as.data.frame(rowSums(words))
names(words_freq) <- 'count'
words_freq$words <- rownames(words_freq)
words_freq <- words_freq %>% arrange(count)

bottom15 = words_freq[1:15, "words"]
head(bottom15, 15)

##  [1] "indiaafter"             "forira"
##  [3] "sasha"                  "suja"
##  [5] "andindividual"          "blackwill"
##  [7] "pathof"                 "pitfalls"
##  [9] "placeto"                "rediscoverindia"
## [11] "ndia"                   "shaking"
## [13] "entertainmentsepilogue" "risespart"
## [15] "armiesbrother"

############# Question 3e: Identify bottom 15 words that has appeared
################
words_freq_1 <- words_freq %>% arrange(-count)

top10 = words_freq_1[1:10, "words"]
head(top10, 10)

##  [1] "india"      "indian"     "one"        "state"      "government"
##  [6] "minister"   "nehru"      "congress"   "also"       "now"
```

```r
# END of Q3#

########### Question 4: Important parties and characters & association
between them#############


# Converting the matrix to a data frame

df_dtm = as.data.frame(dtm)

# subsetting the document term matrix with important characters and parties.
df_dtm_imp = subset(df_dtm,
select=c("bharatiya","indian","national","congress","lok","dal","janata","par
ty",

"communist","telugu","desam","dravida","munnetra","kazhagam","muslim","league
",

"rajiv","gandhi","indira","nehru","zail","mahatma"))
head(df_dtm_imp)

##   bharatiya indian national congress lok dal janata party communist telugu
## 1         0      0        0        0   0   0      0     0          0      0
## 2         0      0        0        0   0   0      0     0          0      0
## 3         0      0        0        0   0   0      0     0          0      0
## 4         0      1        0        0   0   0      0     0          0      0
## 5         0      0        0        0   0   0      0     0          0      0
## 6         0      0        0        1   0   0      0     0          0      0
##   desam dravida munnetra kazhagam muslim league rajiv gandhi indira nehru
## 1     0       0        0        0      0      0     0      0      0     0
## 2     0       0        0        0      0      0     0      1      0     0
## 3     0       0        0        0      0      0     0      0      0     0
## 4     0       0        0        0      0      0     0      0      0     0
## 5     0       0        0        0      0      0     0      0      0     0
## 6     0       0        0        0      0      0     0      0      0     0
##   zail mahatma
## 1    0       0
## 2    0       0
## 3    0       0
## 4    0       0
## 5    0       0
## 6    0       0

#creating a correlation matrix
df_dtm_imp_cor = cor(df_dtm_imp)

head(df_dtm_imp_cor)

##             bharatiya      indian   national    congress         lok
## bharatiya  1.00000000 -0.03343015 0.08904228  0.06651665  0.19911986
## indian    -0.03343015  1.00000000 0.06107030 -0.08470039 -0.03766972
```

```
## national   0.08904228  0.06107030 1.00000000  0.05715096  0.02483187
## congress   0.06651665 -0.08470039 0.05715096  1.00000000  0.13304682
## lok        0.19911986 -0.03766972 0.02483187  0.13304682  1.00000000
## dal        0.10597115 -0.03245652 0.05661711  0.17283507  0.05640606
##                  dal      janata       party   communist      telugu
## bharatiya  0.10597115  0.30908688  0.24002466 -0.03001967  0.02127977
## indian    -0.03245652 -0.07865843 -0.08321402  0.01755054 -0.04842383
## national   0.05661711  0.04023598  0.04166810 -0.03501068  0.07151490
## congress   0.17283507  0.15171670  0.57449693  0.16530978  0.08644251
## lok        0.05640606  0.07721356  0.11077399 -0.02318530 -0.01574149
## dal        1.00000000  0.13427312  0.15910137 -0.01704209  0.06820333
##                 desam      dravida     munnetra     kazhagam       muslim
## bharatiya  0.08359926 -0.008588971 -0.009092255 -0.009092255 -0.041667927
## indian    -0.04747369  0.004347709 -0.019199856 -0.019199856  0.006768382
## national   0.16199232  0.053525774  0.078753284  0.078753284 -0.021036649
## congress   0.16652551  0.128550674  0.128355855  0.128355855  0.013030860
## lok       -0.01067611  0.060564513  0.082955867  0.082955867 -0.020185996
## dal        0.08065848 -0.011035853 -0.011682516 -0.011682516  0.049220931
##                 league        rajiv      gandhi      indira        nehru
## bharatiya -0.019972621  0.028825024 -0.02653895  0.01813751 -0.046007324
## indian    -0.041530110  0.001796669 -0.08558112 -0.06304916 -0.038232622
## national   0.016630712  0.001023239 -0.01913212 -0.02539255 -0.001689672
## congress   0.112837342  0.085266410  0.24172974  0.21856971  0.093772404
## lok       -0.024726843 -0.026764833  0.07309523  0.07615141 -0.046055385
## dal        0.002667434  0.047490843  0.01336742  0.02112848 -0.065644203
##                  zail      mahatma
## bharatiya -0.006420819 -0.032502057
## indian    -0.022896913  0.023125854
## national  -0.022387592 -0.003001302
## congress  -0.009402119  0.036359735
## lok       -0.007949211 -0.020428111
## dal       -0.008250023 -0.026507033

## Use corrplot to visualize the associations
library(corrplot)

## Warning: package 'corrplot' was built under R version 3.3.3

corrplot(df_dtm_imp_cor, type = "upper")
```


bharatiya
indian
national
congress
lok
dal
janata
party
communist
telugu
desam
dravida
munnetra
kazhagam
muslim
league
rajiv
gandhi
indira
nehru
zail
mahatma
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1