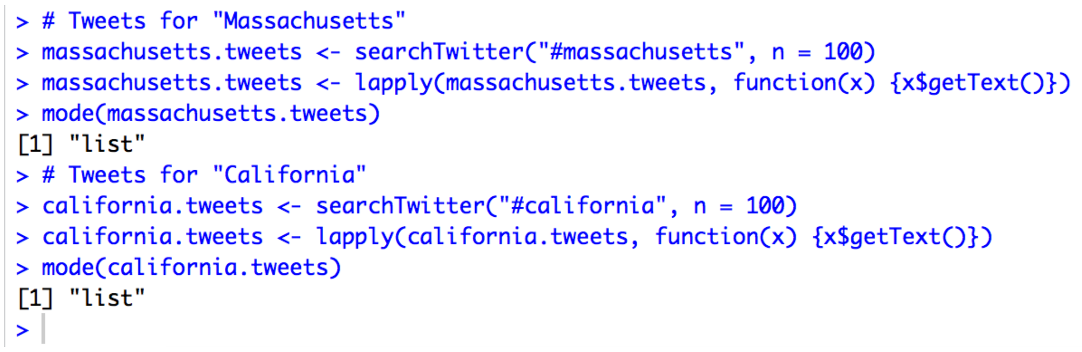
CS688 Web Analytics and Mining
Homework#4

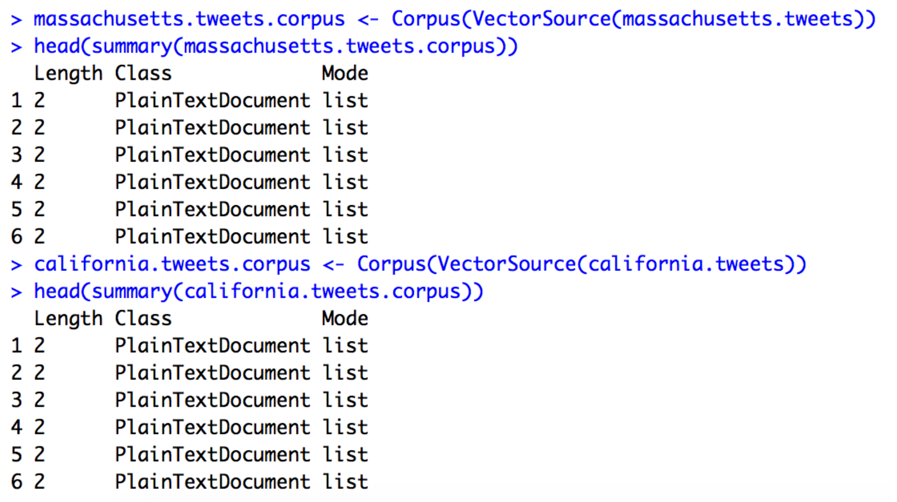
Twitter Mining:

a) States chosen: Massachusetts and California
    Creating a list of tweets for both states.



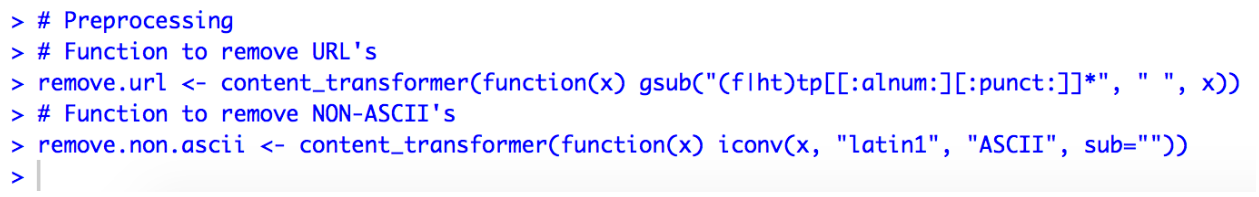
```
> # Tweets for "Massachusetts"
> massachusetts.tweets <- searchTwitter("#massachusetts", n = 100)
> massachusetts.tweets <- lapply(massachusetts.tweets, function(x) {x$getText()})
> mode(massachusetts.tweets)
[1] "list"
> # Tweets for "California"
> california.tweets <- searchTwitter("#california", n = 100)
> california.tweets <- lapply(california.tweets, function(x) {x$getText()})
> mode(california.tweets)
[1] "list"
> 
```

b) Creating a corpus for both states.



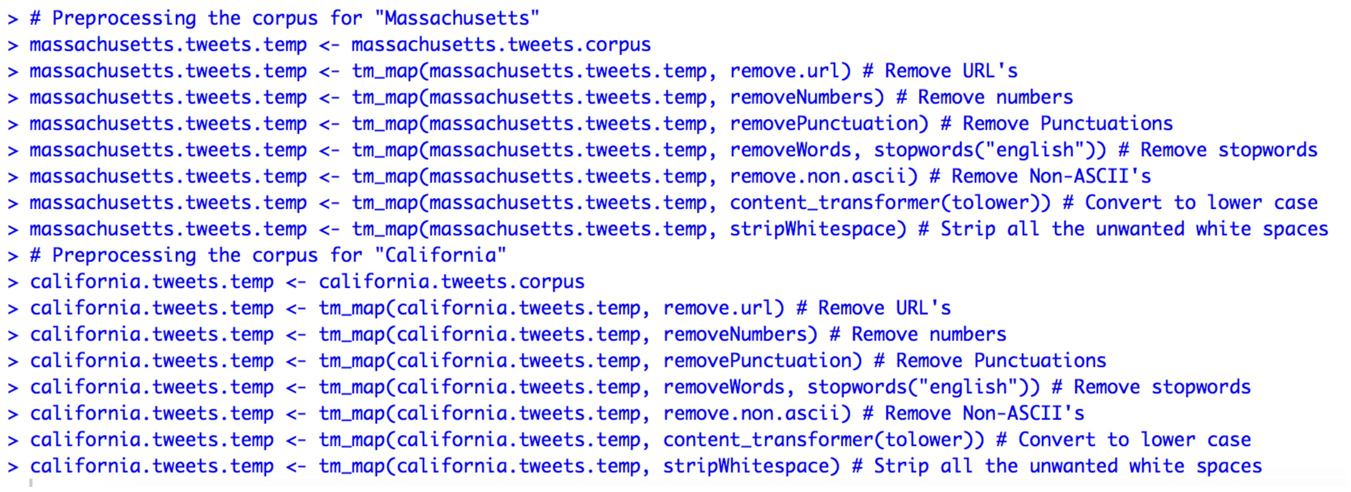
```
> massachusetts.tweets.corpus <- Corpus(VectorSource(massachusetts.tweets))
> head(summary(massachusetts.tweets.corpus))
  Length Class            Mode
1 2      PlainTextDocument list
2 2      PlainTextDocument list
3 2      PlainTextDocument list
4 2      PlainTextDocument list
5 2      PlainTextDocument list
6 2      PlainTextDocument list
> california.tweets.corpus <- Corpus(VectorSource(california.tweets))
> head(summary(california.tweets.corpus))
  Length Class            Mode
1 2      PlainTextDocument list
2 2      PlainTextDocument list
3 2      PlainTextDocument list
4 2      PlainTextDocument list
5 2      PlainTextDocument list
6 2      PlainTextDocument list
```

c) Preprocessing a corpus:
    User defined function to remove URL's and Non-ASCII characters.



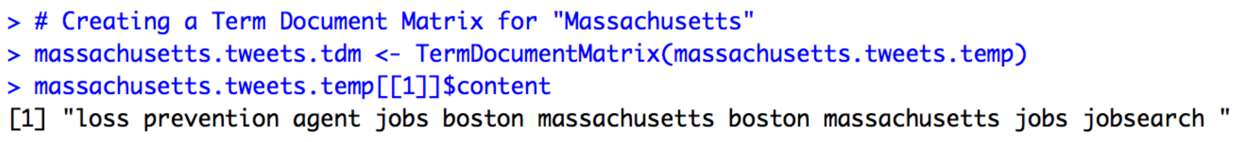
```
> # Preprocessing
> # Function to remove URL's
> remove.url <- content_transformer(function(x) gsub("(f|ht)tp[[:alnum:][:punct:]]*", " ", x))
> # Function to remove NON-ASCII's
> remove.non.ascii <- content_transformer(function(x) iconv(x, "latin1", "ASCII", sub=""))
> 
```

Built-in function from tm package for preprocessing:

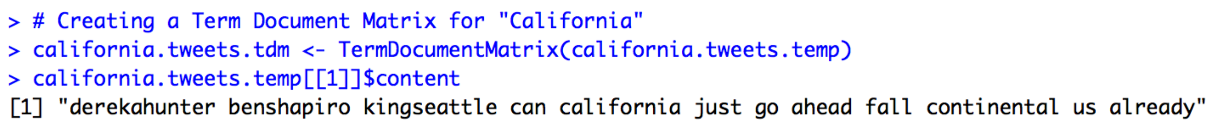


```
> # Preprocessing the corpus for "Massachusetts"
> massachusetts.tweets.temp <- massachusetts.tweets.corpus
> massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, remove.url) # Remove URL's
> massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, removeNumbers) # Remove numbers
> massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, removePunctuation) # Remove Punctuations
> massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, removeWords, stopwords("english")) # Remove stopwords
> massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, remove.non.ascii) # Remove Non-ASCII's
> massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, content_transformer(tolower)) # Convert to lower case
> massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, stripWhitespace) # Strip all the unwanted white spaces
> # Preprocessing the corpus for "California"
> california.tweets.temp <- california.tweets.corpus
> california.tweets.temp <- tm_map(california.tweets.temp, remove.url) # Remove URL's
> california.tweets.temp <- tm_map(california.tweets.temp, removeNumbers) # Remove numbers
> california.tweets.temp <- tm_map(california.tweets.temp, removePunctuation) # Remove Punctuations
> california.tweets.temp <- tm_map(california.tweets.temp, removeWords, stopwords("english")) # Remove stopwords
> california.tweets.temp <- tm_map(california.tweets.temp, remove.non.ascii) # Remove Non-ASCII's
> california.tweets.temp <- tm_map(california.tweets.temp, content_transformer(tolower)) # Convert to lower case
> california.tweets.temp <- tm_map(california.tweets.temp, stripWhitespace) # Strip all the unwanted white spaces
```

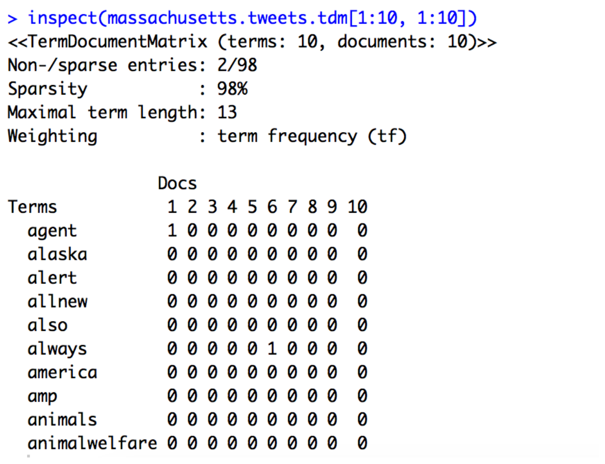
(d) Creating a term document matrix:



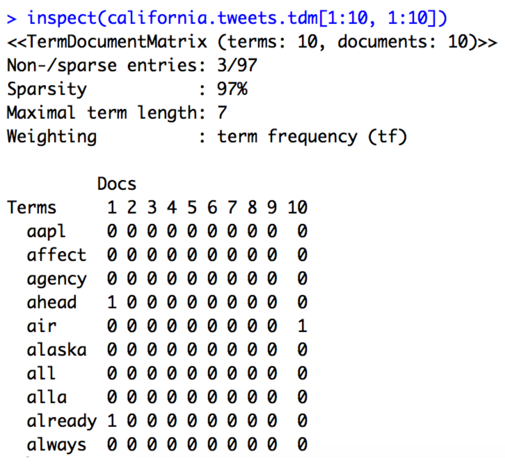
```
> # Creating a Term Document Matrix for "Massachusetts"
> massachusetts.tweets.tdm <- TermDocumentMatrix(massachusetts.tweets.temp)
> massachusetts.tweets.temp[[1]]$content
[1] "loss prevention agent jobs boston massachusetts boston massachusetts jobs jobsearch "
```



```
> # Creating a Term Document Matrix for "California"
> california.tweets.tdm <- TermDocumentMatrix(california.tweets.temp)
> california.tweets.temp[[1]]$content
[1] "derekahunter benshapiro kingseattle can california just go ahead fall continental us already"
```

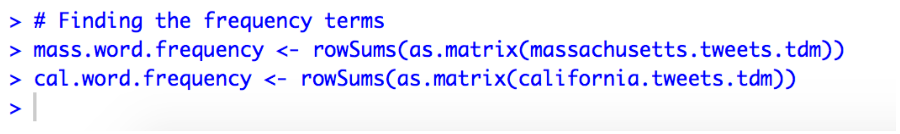


```
> inspect(massachusetts.tweets.tdm[1:10, 1:10])
<<TermDocumentMatrix (terms: 10, documents: 10)>>
Non-/sparse entries: 2/98
Sparsity           : 98%
Maximal term length: 13
Weighting          : term frequency (tf)

              Docs
Terms          1 2 3 4 5 6 7 8 9 10
  agent        1 0 0 0 0 0 0 0 0  0
  alaska       0 0 0 0 0 0 0 0 0  0
  alert        0 0 0 0 0 0 0 0 0  0
  allnew       0 0 0 0 0 0 0 0 0  0
  also         0 0 0 0 0 0 0 0 0  0
  always       0 0 0 0 0 1 0 0 0  0
  america      0 0 0 0 0 0 0 0 0  0
  amp          0 0 0 0 0 0 0 0 0  0
  animals      0 0 0 0 0 0 0 0 0  0
  animalwelfare 0 0 0 0 0 0 0 0 0  0
```



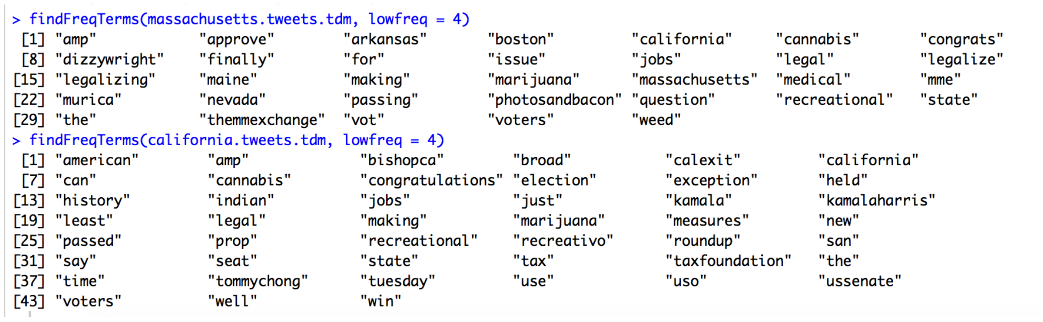
```
> inspect(california.tweets.tdm[1:10, 1:10])
<<TermDocumentMatrix (terms: 10, documents: 10)>>
Non-/sparse entries: 3/97
Sparsity           : 97%
Maximal term length: 7
Weighting          : term frequency (tf)

         Docs
Terms     1 2 3 4 5 6 7 8 9 10
  aapl    0 0 0 0 0 0 0 0 0  0
  affect  0 0 0 0 0 0 0 0 0  0
  agency  0 0 0 0 0 0 0 0 0  0
  ahead   1 0 0 0 0 0 0 0 0  0
  air     0 0 0 0 0 0 0 0 0  1
  alaska  0 0 0 0 0 0 0 0 0  0
  all     0 0 0 0 0 0 0 0 0  0
  alla    0 0 0 0 0 0 0 0 0  0
  already 1 0 0 0 0 0 0 0 0  0
  always  0 0 0 0 0 0 0 0 0  0
```

(e) Creating a Frequent terms in both states:

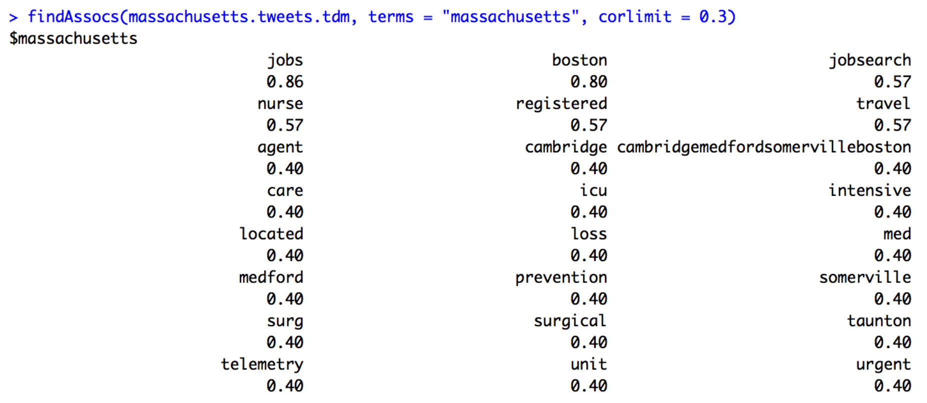


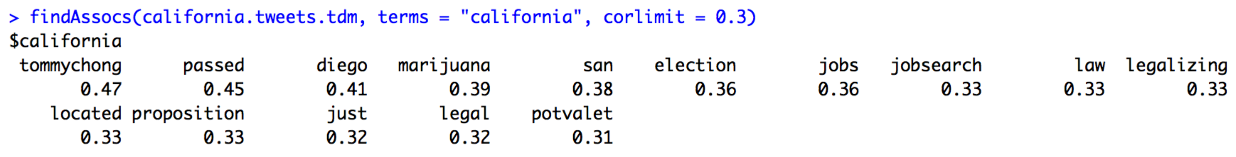
```
> # Finding the frequency terms
> mass.word.frequency <- rowSums(as.matrix(massachusetts.tweets.tdm))
> cal.word.frequency <- rowSums(as.matrix(california.tweets.tdm))
>
```

Most frequent terms:


```
> findFreqTerms(massachusetts.tweets.tdm, lowfreq = 4)
 [1] "amp"           "approve"       "arkansas"      "boston"        "california"    "cannabis"      "congrats"
 [8] "dizzywright"   "finally"       "for"           "issue"         "jobs"          "legal"         "legalize"
[15] "legalizing"    "maine"         "making"        "marijuana"     "massachusetts" "medical"       "mme"
[22] "murica"        "nevada"        "passing"       "photosandbacon" "question"      "recreational"  "state"
[29] "the"           "themmexchange" "vot"           "voters"        "weed"
> findFreqTerms(california.tweets.tdm, lowfreq = 4)
 [1] "american"      "amp"           "bishopca"      "broad"         "calexit"       "california"
 [7] "can"           "cannabis"      "congratulations" "election"    "exception"     "held"
[13] "history"       "indian"        "jobs"          "just"          "kamala"        "kamalaharris"
[19] "least"         "legal"         "making"        "marijuana"     "measures"      "new"
[25] "passed"        "prop"          "recreational"  "recreativo"    "roundup"       "san"
[31] "say"           "seat"          "state"         "tax"           "taxfoundation" "the"
[37] "time"          "tommychong"    "tuesday"       "use"           "uso"           "ussenate"
[43] "voters"        "well"          "win"
```

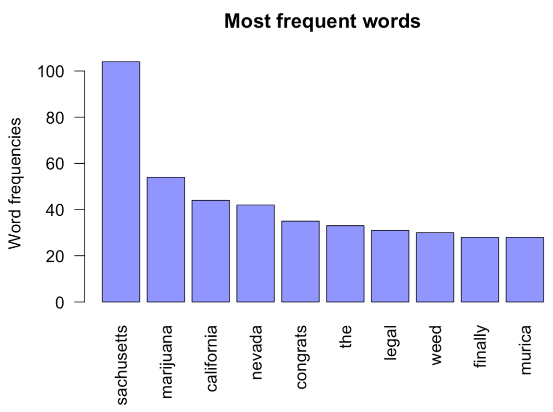
Association of frequent terms with the words “Massachusetts” and “California”


```
> findAssocs(massachusetts.tweets.tdm, terms = "massachusetts", corlimit = 0.3)
$massachusetts
                                 jobs                           boston                        jobsearch
                                 0.86                             0.80                             0.57
                                nurse                       registered                           travel
                                 0.57                             0.57                             0.57
                                agent                        cambridge cambridgemedfordsomervilleboston
                                 0.40                             0.40                             0.40
                                 care                              icu                        intensive
                                 0.40                             0.40                             0.40
                              located                             loss                              med
                                 0.40                             0.40                             0.40
                              medford                       prevention                       somerville
                                 0.40                             0.40                             0.40
                                 surg                         surgical                          taunton
                                 0.40                             0.40                             0.40
                            telemetry                             unit                           urgent
                                 0.40                             0.40                             0.40
```

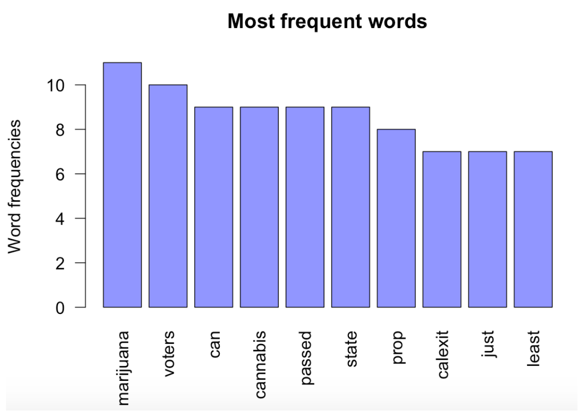

```
> findAssocs(california.tweets.tdm, terms = "california", corlimit = 0.3)
$california
 tommychong      passed       diego   marijuana         san    election        jobs   jobsearch         law  legalizing
       0.47        0.45        0.41        0.39        0.38        0.36        0.36        0.33        0.33        0.33
    located proposition        just       legal    potvalet
       0.33        0.33        0.32        0.32        0.31
```

Graphic plot of MOST frequent terms:

MASSACHUSETTS

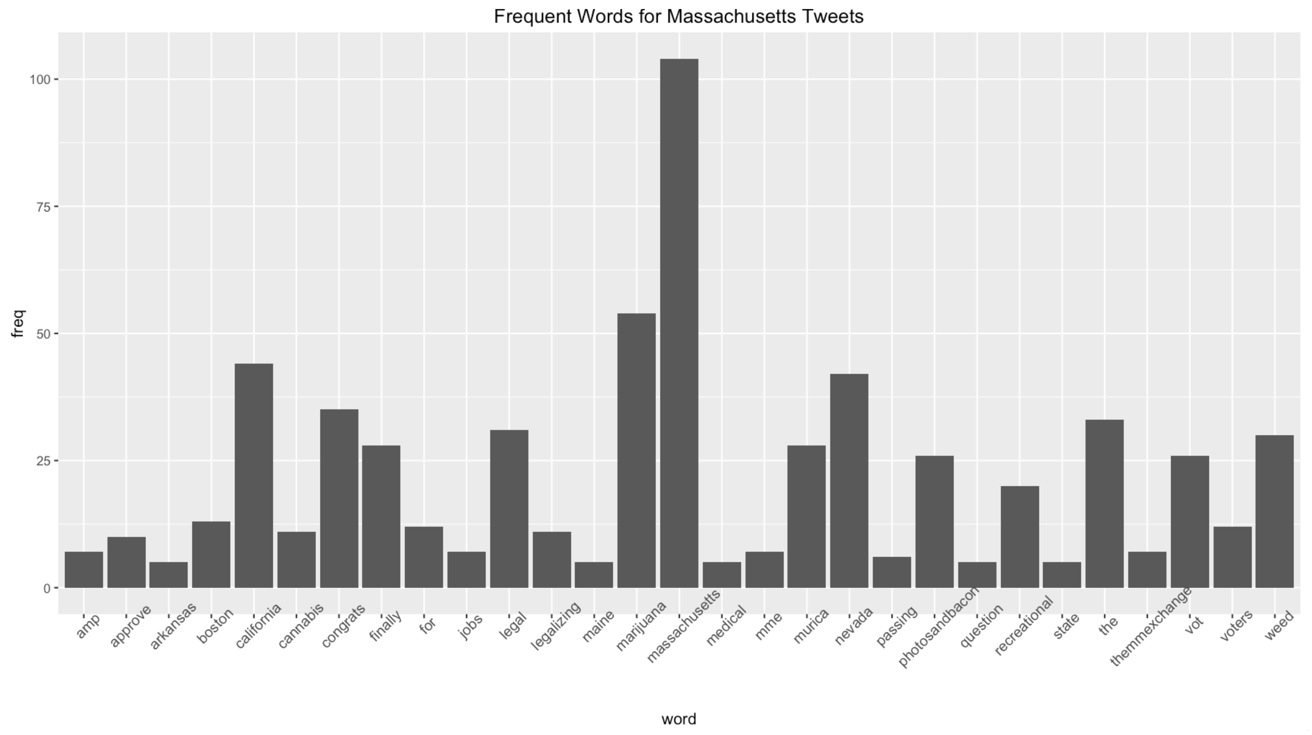


CALIFORNIA

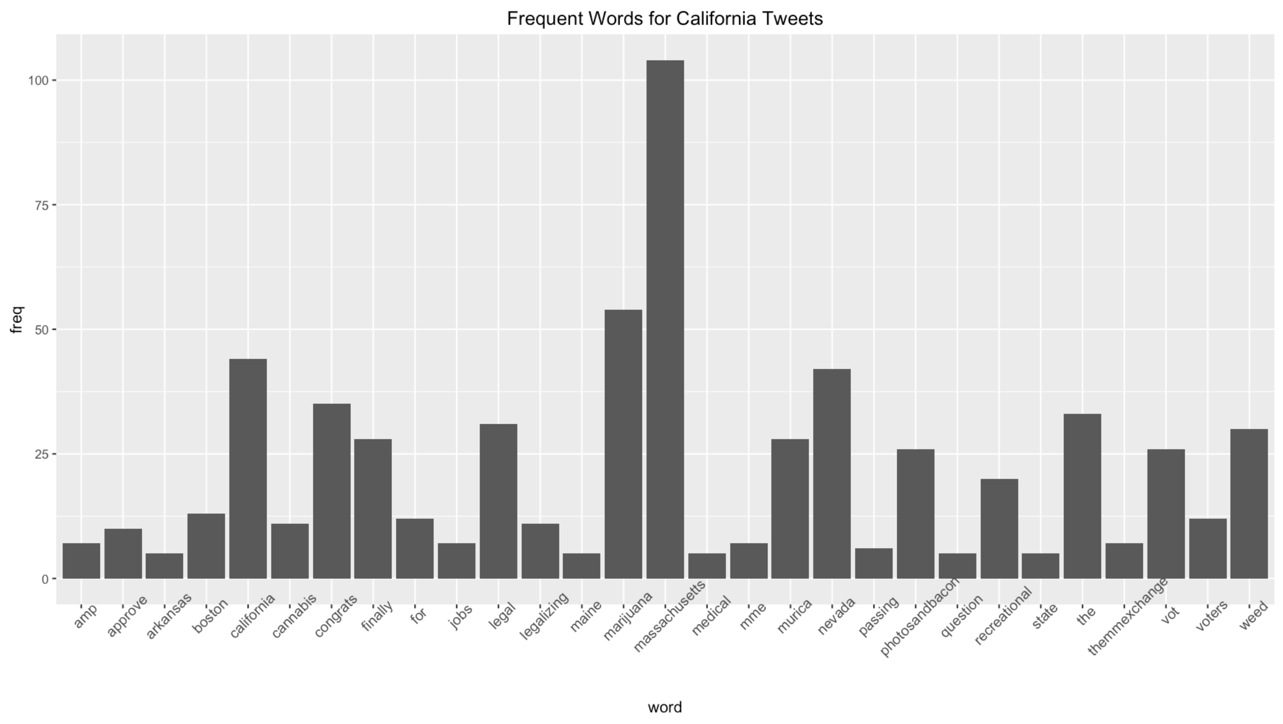


Visual plot of Frequent terms:

MASSACHUSETTS:

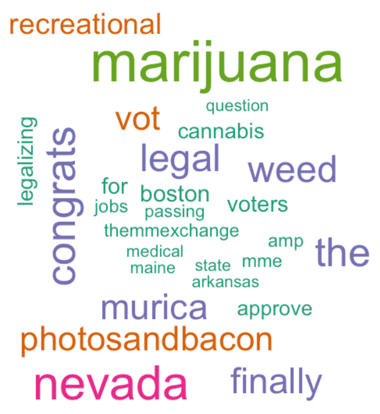


CALIFORNIA:

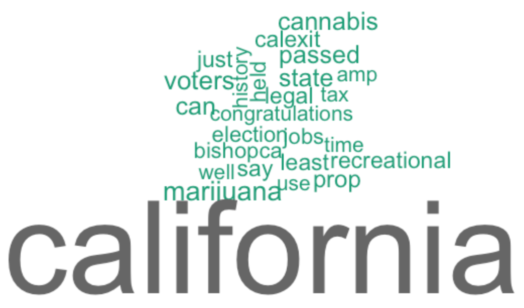


(f) Word Cloud:

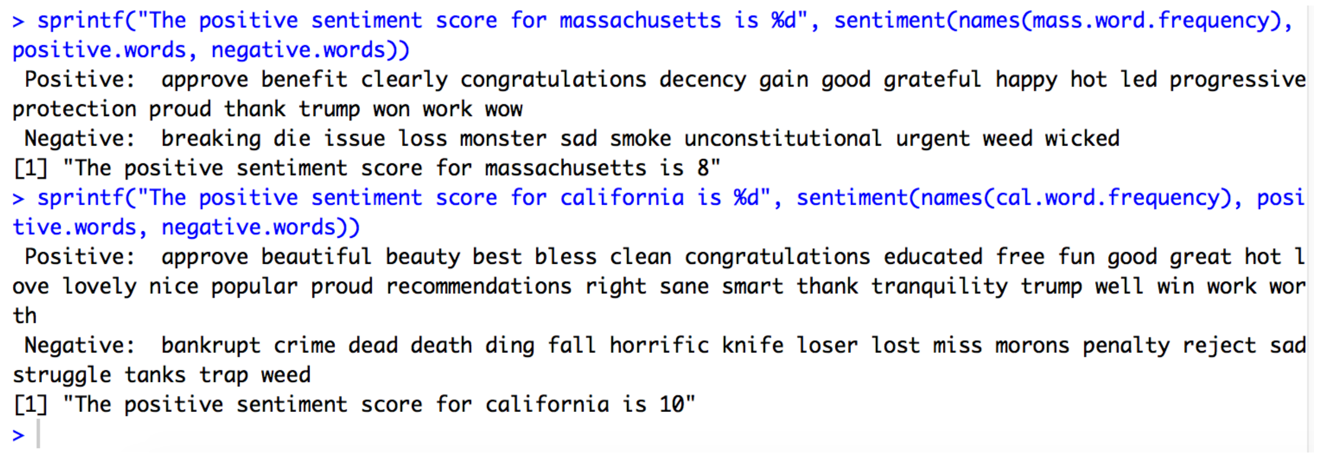
MASSACHUSETTS



CALIFORNIA



(g) Sentiment analysis:



```
> sprintf("The positive sentiment score for massachusetts is %d", sentiment(names(mass.word.frequency),
positive.words, negative.words))
 Positive:  approve benefit clearly congratulations decency gain good grateful happy hot led progressive
protection proud thank trump won work wow
 Negative:  breaking die issue loss monster sad smoke unconstitutional urgent weed wicked
[1] "The positive sentiment score for massachusetts is 8"
> sprintf("The positive sentiment score for california is %d", sentiment(names(cal.word.frequency), posi
tive.words, negative.words))
 Positive:  approve beautiful beauty best bless clean congratulations educated free fun good great hot l
ove lovely nice popular proud recommendations right sane smart thank tranquility trump well win work wor
th
 Negative:  bankrupt crime dead death ding fall horrific knife loser lost miss morons penalty reject sad
struggle tanks trap weed
[1] "The positive sentiment score for california is 10"
> 
```

The California has more positive sentiment than Massachusetts.

R- Code:

```
rm(list = ls())
setwd("/Users/Ravi/Documents/Fall-2016/CS688/Homeworks/HW#4/")
library("twitteR")
library("ROAuth")
library("bitops")
library("RCurl")
library("rjson")
library("tm")
library("SnowballC")
library("wordcloud")
library("tm.plugin.webmining")
library("ggplot2")
cat("\014")

# Problem-a
# Tweets for "Massachusetts"
massachusetts.tweets <- searchTwitter("#massachusetts", n = 100)
massachusetts.tweets <- lapply(massachusetts.tweets, function(x) {x$getText()})
mode(massachusetts.tweets)

# Tweets for "California"
california.tweets <- searchTwitter("#california", n = 100)
california.tweets <- lapply(california.tweets, function(x) {x$getText()})
mode(california.tweets)

# Problem-b
# Creating a corpus for "Massachusetts"
massachusetts.tweets.corpus <- Corpus(VectorSource(massachusetts.tweets))
california.tweets.corpus <- Corpus(VectorSource(california.tweets))

# Problem-c
# Preprocessing
# Function to remove URL's
remove.url <- content_transformer(function(x) gsub("(f|ht)tp[[:alnum:][:punct:]]*", " ", x))
# Function to remove NON-ASCII's
remove.non.ascii <- content_transformer(function(x) iconv(x, "latin1", "ASCII", sub=""))

# Preprocessing the corpus for "Massachusetts"
massachusetts.tweets.temp <- massachusetts.tweets.corpus
massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, remove.url) # Remove URL's
massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, removeNumbers) # Remove numbers
massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, removePunctuation) # Remove Punctuations
massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, removeWords, stopwords("english")) # Remove stopwords
# massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, stemDocument)
massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, remove.non.ascii) # Remove Non-ASCII's
massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, content_transformer(tolower)) # Convert to lower case
massachusetts.tweets.temp <- tm_map(massachusetts.tweets.temp, stripWhitespace) # Strip all the unwanted white spaces

# Preprocessing the corpus for "California"
california.tweets.temp <- california.tweets.corpus
california.tweets.temp <- tm_map(california.tweets.temp, remove.url) # Remove URL's
california.tweets.temp <- tm_map(california.tweets.temp, removeNumbers) # Remove numbers
california.tweets.temp <- tm_map(california.tweets.temp, removePunctuation) # Remove Punctuations
california.tweets.temp <- tm_map(california.tweets.temp, removeWords, stopwords("english")) # Remove stopwords
california.tweets.temp <- tm_map(california.tweets.temp, remove.non.ascii) # Remove Non-ASCII's
california.tweets.temp <- tm_map(california.tweets.temp, content_transformer(tolower)) # Convert to lower case
california.tweets.temp <- tm_map(california.tweets.temp, stripWhitespace) # Strip all the unwanted white spaces
```

```
# Problem-d
# Creating a Term Document Matrix for "Massachusetts"
massachusetts.tweets.tdm <- TermDocumentMatrix(massachusetts.tweets.temp)
massachusetts.tweets.temp[[1]]$content
inspect(massachusetts.tweets.tdm[1:10, 1:10])

# Creating a Term Document Matrix for "California"
california.tweets.tdm <- TermDocumentMatrix(california.tweets.temp)
california.tweets.temp[[1]]$content
inspect(california.tweets.tdm[1:10, 1:10])

# Problem-e
# Finding the frequency terms
mass.word.frequency <- rowSums(as.matrix(massachusetts.tweets.tdm))
cal.word.frequency <- rowSums(as.matrix(california.tweets.tdm))

mass.ordered <- order(mass.word.frequency)
cal.ordered <- order(cal.word.frequency)
mass.word.frequency[tail(mass.ordered)]
cal.word.frequency[tail(cal.ordered)]

findFreqTerms(massachusetts.tweets.tdm, lowfreq = 4)
findAssocs(massachusetts.tweets.tdm, terms = "massachusetts", corlimit = 0.3)
findFreqTerms(california.tweets.tdm, lowfreq = 4)
findAssocs(california.tweets.tdm, terms = "california", corlimit = 0.3)

mass.freq.frame <- data.frame(word = names(sort(mass.word.frequency, decreasing = TRUE)), freq = sort(mass.word.frequency,
decreasing = TRUE))
cal.freq.frame <- data.frame(word = names(sort(cal.word.frequency, decreasing = TRUE)), freq = sort(cal.word.frequency,
decreasing = TRUE))

# Most frequent terms
barplot(mass.freq.frame[1:10,]$freq, las = 2, names.arg = mass.freq.frame[1:10,]$word, col = rgb(0,0,1,0.5), main = "Most frequent
words", ylab = "Word frequencies")
barplot(cal.freq.frame[2:11,]$freq, las = 2, names.arg = cal.freq.frame[2:11,]$word, col = rgb(0,0,1,0.5), main = "Most frequent
words", ylab = "Word frequencies")

# Frequent terms
ggplot(subset(mass.freq.frame, freq>4), aes(word, freq)) + geom_bar(stat = "identity") +
  ggtitle("Frequent Words for Massachusetts Tweets") + theme(axis.text.x=element_text(angle=45, size =10))

ggplot(subset(mass.freq.frame, freq>4), aes(word, freq)) + geom_bar(stat = "identity") +
  ggtitle("Frequent Words for California Tweets") + theme(axis.text.x=element_text(angle=45, size =10))

# Problem-f
# Creating a Word cloud for both states
wordcloud(names(mass.word.frequency), mass.word.frequency, min.freq = 5, colors=brewer.pal(8, "Dark2"))
wordcloud(names(cal.word.frequency), cal.word.frequency, min.freq = 5, colors=brewer.pal(8, "Dark2"))


# Problem-g
# Sentiment analysis
sentiment <- function(text, pos.words, neg.words) {
  text <- gsub('[[:punct:]]', '', text)
  text <- gsub('[[:cntrl:]]', '', text)
  text <- gsub('\\d+', '', text)
  # text <- tolower(text)
  # split the text into a vector of words
  words <- strsplit(text, '\\s+')
  words <- unlist(words)
  # find which words are positive
  pos.matches <- match(words, pos.words)
  pos.matches <- !is.na(pos.matches)
  # find which words are negative
  neg.matches <- match(words, neg.words)
  neg.matches <- !is.na(neg.matches)
  # calculate the sentiment score
  score <- sum(pos.matches) - sum(neg.matches)
  cat("sum of pos and neg:", sum(pos.matches), sum(neg.matches), "\n")
  cat (" Positive: ", words[pos.matches], "\n")
  cat (" Negative: ", words[neg.matches], "\n")
  return (score)
}

# Read positve and negative text files
positive.words <- scan('positive.words.txt', what = "character", comment.char = ';')
negative.words <- scan('negative-words.txt', what = "character", comment.char = ';')

sprintf("The positive sentiment score for massachusetts is %d", sentiment(names(mass.word.frequency), positive.words,
negative.words))
sprintf("The positive sentiment score for california is %d", sentiment(names(cal.word.frequency), positive.words, negative.words))
```