HW2 Num2

Aaron Coates

5/4/2019

2.1 (a.)

```
setwd("~/Documents/GitHub/MMSS_311_2")

TwitterData <- read.csv('/Users/aaroncoates/Downloads/trumptweets.csv')

library(tidytext)
library(tm)
```

```
## Loading required package: NLP
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

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

```
library(broom)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(stringr)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
library(tidyr)
```

2.1 (b.)

```
tweetcorpus <- Corpus(VectorSource(as.vector(TwitterData$text)))

processedcorpus <- tweetcorpus %>%
  tm_map(removeWords, stopwords("english")) %>%
  tm_map(content_transformer(tolower)) %>%
```

```
  tm_map(content_transformer(stemDocument), language = "english") %>%
  tm_map(content_transformer(removePunctuation))
```

```
## Warning in tm_map.SimpleCorpus(., removeWords, stopwords("english")):
## transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(tolower)):
## transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(stemDocument),
## language = "english"): transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(removePunctuation)):
## transformation drops documents
```

2.1 (c.)

```
DTMatrix <- DocumentTermMatrix(processedcorpus)
SparseDTMatrix <- removeSparseTerms(DTMatrix, .99)
inspect(SparseDTMatrix[1:10,1:10])
```

```
## <<DocumentTermMatrix (documents: 10, terms: 10)>>
## Non-/sparse entries: 10/90
## Sparsity           : 90%
## Maximal term length: 7
## Weighting          : term frequency (tf)
## Sample             :
##     Terms
## Docs america back better busi come join like live mani new
##   1        0    0      0    0    0    1    0    1    0   0
##   10       0    0      0    0    0    0    0    0    0   0
##   2        0    0      0    0    0    0    0    0    0   1
##   3        1    2      1    1    2    0    1    0    1   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
##   7        0    0      0    0    0    0    0    0    0   0
##   8        0    0      0    0    0    0    0    0    0   0
##   9        0    0      0    0    0    0    0    0    0   0
```

2.1 (d.)

```
tidymatrix <- tidy(DTMatrix)
```

2.1 (e.)

```
TfIdfMat <- DocumentTermMatrix(processedcorpus, control
                          = list(weighting = weightTfIdf))
SparseTfIdfMat <- removeSparseTerms(TfIdfMat, .99)
inspect(SparseTfIdfMat[1:10,1:10])
```

```
## <<DocumentTermMatrix (documents: 10, terms: 10)>>
## Non-/sparse entries: 10/90
## Sparsity           : 90%
## Maximal term length: 7
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
## Sample             :
##     Terms
```

```
## Docs  america      back    better      busi      come     join      like
##   1  0.000000 0.0000000 0.0000000 0.0000000 0.0000000 2.149356 0.0000000
##   10 0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
##   2  0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
##   3  0.172845 0.3998882 0.2271556 0.2360408 0.4410037 0.000000 0.1725976
##   4  0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
##   5  0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
##   6  0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
##   7  0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
##   8  0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
##   9  0.000000 0.0000000 0.0000000 0.0000000 0.0000000 0.000000 0.0000000
##     Terms
## Docs     live      mani       new
##   1  1.911577 0.0000000 0.0000000
##   10 0.000000 0.0000000 0.0000000
##   2  0.000000 0.0000000 0.5292588
##   3  0.000000 0.2048043 0.0000000
##   4  0.000000 0.0000000 0.0000000
##   5  0.000000 0.0000000 0.0000000
##   6  0.000000 0.0000000 0.0000000
##   7  0.000000 0.0000000 0.0000000
##   8  0.000000 0.0000000 0.0000000
##   9  0.000000 0.0000000 0.0000000
```

2.2 (a.)

```
popterms <- tidymatrix %>%
  group_by(term) %>%
  summarize(frequency = sum(count)) %>%
  arrange(desc(frequency))
popterms[1:20,1:2]
```

```
## # A tibble: 20 x 2
##    term            frequency
##    <chr>               <dbl>
##  1 twitter             15283
##  2 realdonaldtrump      8384
##  3 web                  5869
##  4 trump                4215
##  5 will                 4156
##  6 great                4005
##  7 amp                  2730
##  8 thank                2577
##  9 the                  2485
## 10 get                  1672
## 11 just                 1663
## 12 make                 1515
## 13 donald               1459
## 14 presid               1364
## 15 like                 1318
## 16 america              1317
## 17 run                  1293
## 18 obama                1285
## 19 need                 1268
## 20 new                  1253
```

2.2 (b.) Post-Election

```
TwitterDataDate <- TwitterData
TwitterDataDate$date <- as.Date(TwitterData$created_at, '%m-%d-%Y')

PostTwitterData <- subset(TwitterDataDate, date >= as.Date('2016-11-08'))
PreTwitterData <- subset(TwitterDataDate, date <= as.Date('2016-11-08'))

posttweetcorpus <- Corpus(VectorSource(as.vector(PostTwitterData$text)))

postprocessedcorpus <- posttweetcorpus %>%
  tm_map(removeWords, stopwords("english")) %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(content_transformer(stemDocument), language = "english") %>%
  tm_map(content_transformer(removePunctuation))
```

```
## Warning in tm_map.SimpleCorpus(., removeWords, stopwords("english")):
## transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(tolower)):
## transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(stemDocument),
## language = "english"): transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(removePunctuation)):
## transformation drops documents
```

```
postDTMatrix <- DocumentTermMatrix(postprocessedcorpus)
postSparseDTMatrix <- removeSparseTerms(postDTMatrix, .99)
posttidymatrix <- tidy(postDTMatrix)

postpopterms <- posttidymatrix %>%
  group_by(term) %>%
  summarize(frequency = sum(count)) %>%
  arrange(desc(frequency))
postpopterms[1:20,1:2]
```

```
## # A tibble: 20 x 2
##    term     frequency
##    <chr>        <dbl>
##  1 twitter       1204
##  2 will           581
##  3 great          553
##  4 amp            500
##  5 the            289
##  6 news           219
##  7 fake           191
##  8 tax            191
##  9 thank          189
## 10 peopl          185
## 11 media          181
## 12 america        177
## 13 just           170
## 14 now            170
## 15 iphonert       168
## 16 big            165
```

```
## 17 get            165
## 18 job            165
## 19 today          165
## 20 trump          162
```

2.2 (b.) Pre-Election

```
pretweetcorpus <- Corpus(VectorSource(as.vector(PreTwitterData$text)))

preprocessedcorpus <- pretweetcorpus %>%
  tm_map(removeWords, stopwords("english")) %>%
  tm_map(content_transformer(tolower)) %>%
  tm_map(content_transformer(stemDocument), language = "english") %>%
  tm_map(content_transformer(removePunctuation))
```

```
## Warning in tm_map.SimpleCorpus(., removeWords, stopwords("english")):
## transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(tolower)):
## transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(stemDocument),
## language = "english"): transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(removePunctuation)):
## transformation drops documents
```

```
preDTMatrix <- DocumentTermMatrix(preprocessedcorpus)
preSparseDTMatrix <- removeSparseTerms(preDTMatrix, .99)
inspect(preSparseDTMatrix[1:10,1:10])
```

```
## <<DocumentTermMatrix (documents: 10, terms: 10)>>
## Non-/sparse entries: 14/86
## Sparsity           : 86%
## Maximal term length: 8
## Weighting          : term frequency (tf)
## Sample             :
##     Terms
## Docs again all also alway amaz america american amp and announc
##   1      4   1    1     3    2      14        3  17   4       2
##   10     0   0    0     0    0       0        1   0   0       0
##   2      0   0    0     0    0       0        0   0   0       0
##   3      0   0    0     0    0       0        0   1   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
##   7      0   0    0     0    0       0        0   0   0       1
##   8      0   0    0     0    0       0        0   1   0       0
##   9      0   0    0     0    0       0        0   0   0       0
```

```
pretidymatrix <- tidy(preDTMatrix)

prepopterms <- pretidymatrix %>%
  group_by(term) %>%
  summarize(frequency = sum(count)) %>%
  arrange(desc(frequency))
prepopterms[1:20,1:2]
```

```
## # A tibble: 20 x 2
##    term            frequency
##    <chr>               <dbl>
##  1 twitter             14079
##  2 realdonaldtrump      8289
##  3 web                  5830
##  4 trump                4053
##  5 will                 3575
##  6 great                3452
##  7 thank                2388
##  8 amp                  2230
##  9 the                  2196
## 10 get                  1507
## 11 just                 1493
## 12 donald               1429
## 13 make                 1364
## 14 run                  1263
## 15 like                 1248
## 16 obama                1223
## 17 presid               1204
## 18 need                 1191
## 19 america              1140
## 20 can                  1138
```

For Pre-Election, we see many campaign related words, such as "Obama" and "Make America great", which was part of Trump's slogan, "Make America great again." For Post-Election, we see many of Trump's focuses while in the presidency, such as "fake news", "taxes", and "jobs."

2.2 (c.)

```
hashtagcorpus <- Corpus(VectorSource(as.vector(TwitterData$text)))

hashtag <- function(x) gsub("[^#[:alnum:][:space:]]", "", x)
htcorpus2 <- tm_map(hashtagcorpus, content_transformer(hashtag)) %>%
  tm_map(removeWords, stopwords("english")) %>%
  tm_map(content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(hashtagcorpus,
## content_transformer(hashtag)): transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., removeWords, stopwords("english")):
## transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., content_transformer(tolower)):
## transformation drops documents
```

```
htDTMatrix <- DocumentTermMatrix(htcorpus2)
tidyhash <- tidy(htDTMatrix)
```

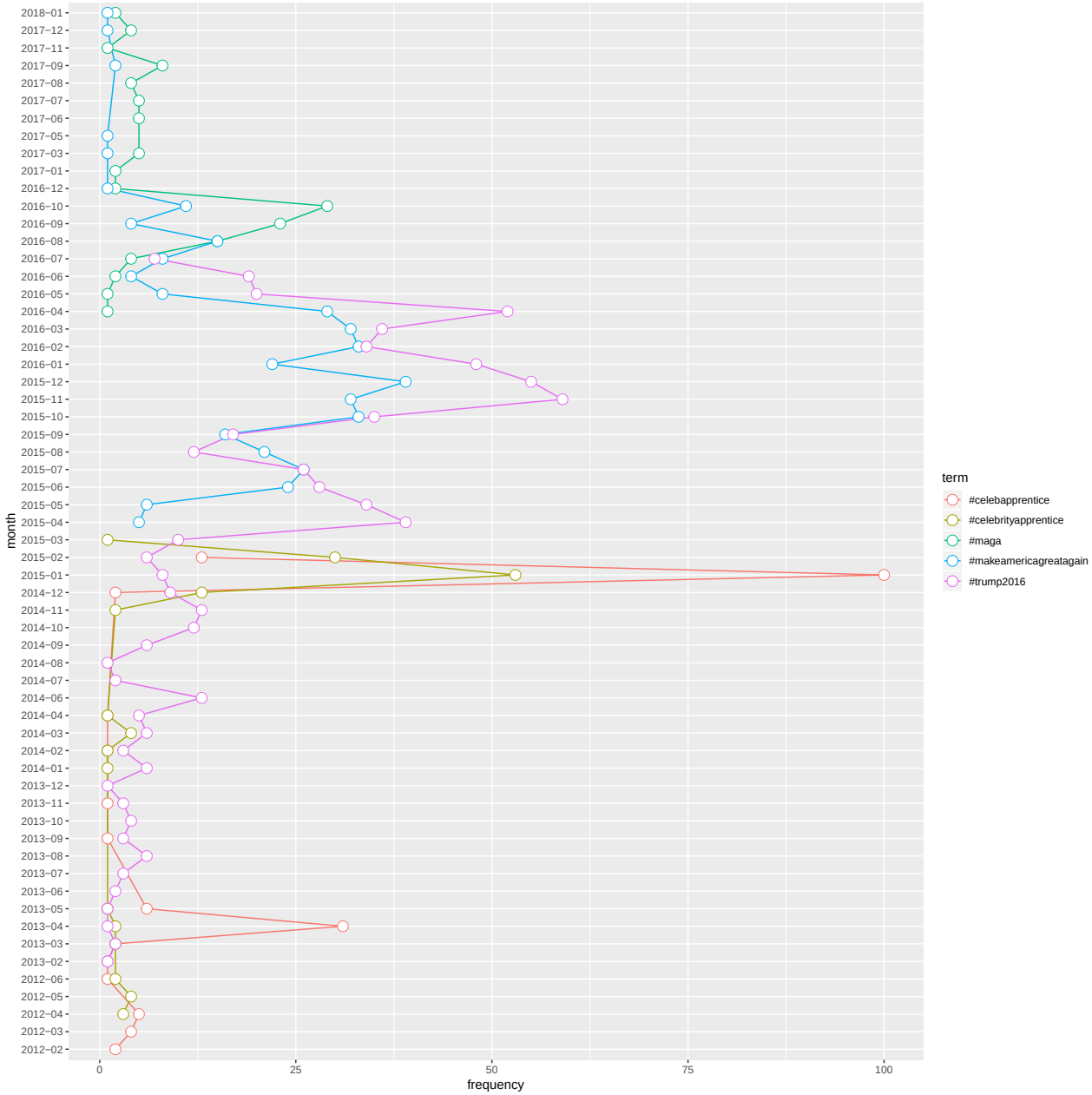
2.2 (d.)

```
popht <- tidyhash %>%
  group_by(term) %>%
  summarize(frequency = sum(count)) %>%
  arrange(desc(frequency))

hashtagdaddy <- subset(popht, grepl("#", term))
hashtagdaddy[1:5, 1:2]
```

```
## # A tibble: 5 x 2
##   term                    frequency
##   <chr>                       <dbl>
## 1 #trump2016                    648
## 2 #makeamericagreatagain        375
## 3 #celebapprentice              171
## 4 #celebrityapprentice          120
## 5 #maga                         113
```

2.2 (e.)

```
DateHashtag <- tidyhash %>%
  subset(term == '#maga' | term == '#trump2016'
         | term == '#celebapprentice' | term == '#celebrityapprentice'
         | term == '#makeamericagreatagain')

x <- 1:17200
TwitterData$document <- x
TwitterData$document <- as.character(TwitterData$document)

final <- inner_join(DateHashtag, TwitterData, by = 'document')
final <- final[, c('document', 'term', 'created_at', 'count')]

final$date <- as.Date(final$created_at, '%m-%d-%Y')
final$month <- format(final$date, '%Y-%m')

maga <- final %>%
  group_by(month, term) %>%
  summarise(frequency = sum(count))

maga <- arrange(maga, month)
```



2.2 (f.)

```
Crooked <- TwitterData
Crooked <- select(Crooked, c(text, created_at)) %>%
  unnest_tokens(bigram, text, token='ngrams', n=2)
Crooked$date <- as.Date(Crooked$created_at, '%m-%d-%Y')
Crooked$month <- format(Crooked$date, '%Y-%m')

Crooked <- subset(Crooked, bigram=='crooked hillary')

for (i in 1:nrow(Crooked)) {
  Crooked$count[i] =1
}
```

```r
finalcrooked <- Crooked %>%
  group_by(month) %>%
  summarise(frequency = sum(count))
```