COSC5610 CMV Analysis

Noah Asaria, David Reddy, Eddie Chapman

```
library(jsonlite)
library(lubridate)
library(SnowballC)
library(tidyverse)
library(topicmodels)
library(tm)
library(tidytext)
```

Data import

Data is separated into two spreadsheets:

- `threads.csv`
- `comments.csv`

Thread data

After importing, any threads that do not start with `CMV:` are dropped. These are moderator notes or other irrelevant threads.

We remove `CMV:` from the remaining thread titles. We also remove the default moderator note that is included at the bottom of most thread texts.

```
threads <- read_csv("threads.csv", col_names = TRUE) %>%
  filter(str_starts(title, coll("CMV:"))) %>%
  mutate(title = str_replace(title, "CMV:", ""),
         text = str_replace(text, "\\*Hello, users of CMV\\! .* \\*Happy CMVing\\!\\*", ""),
         id = as.factor(id),
         timestamp = as.Date.POSIXct(timestamp),
         week = week(timestamp),
         year = year(timestamp),
         ups = as.integer(ups),
         downs = as.integer(downs))
```

Comment data

Comment data contains the author, OP, thread, and timestamp of all threads.

We drop any comments that are posted by the thread's OP, by the DeltaBot, or by a deleted account.

We also drop any comments correponding to irrelevant threads identified in the `threads` dataframe.

```
comments <- read_csv('comments.csv', col_names = TRUE) %>%
  filter(author != op,
         author != 'DeltaBot',
         author != '[DELETED]') %>%
  mutate(id = as.factor(id),
         thread = as.factor(thread),
```

```
        timestamp = as.Date.POSIXct(timestamp)) %>%
  semi_join(threads, by = c("thread" = "id"))
```

Summary statistics

Here we identify the total number of comments and unique commentors for each thread.

```
users_per_thread <- comments %>%
  select(thread, author) %>%
  distinct() %>%
  group_by(thread) %>%
  count(name = 'n_users')

comments_per_thread <- comments %>%
  select(thread, id) %>%
  distinct() %>%
  group_by(thread) %>%
  count(name = 'n_comments')

threads <- threads %>%
  left_join(users_per_thread, by = c("id" = "thread")) %>%
  left_join(comments_per_thread, by = c("id" = "thread")) %>%
  mutate(comments_per_user = n_users / n_comments)
```

Sampling

(Temporary) we reduce the dataset to improve computation time.

Threads are selected which fall between the median and 3rd quartile measure of number of unique users, number of total comments, and upvotes.

This leaves us with 505 threads.

```
popular_threads <- threads %>%
  filter(n_users >= 17, n_users <= 29) %>%
  filter(n_comments >= 32, n_comments <= 60) %>%
  filter(ups >= 10, ups <= 30)
```

Threads are grouped into weeks and the average number of threads per week is calculated.

Any weeks featuring a thread count between the mean thread count and 3rd quartile thread count are retained.

This leaves us with 303 threads.

```
threads_per_week <- popular_threads %>%
  group_by(year, week) %>%
  count(name = "threads_per_week") %>%
  arrange(year, week)

popular_threads <- popular_threads %>%
  left_join(threads_per_week)

sample <- popular_threads %>%
  filter(threads_per_week >= 5, threads_per_week <= 10)
```

```
summary(sample)
```

```
##       id               title              author              text
##  Length:303         Length:303         Length:303         Length:303
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    timestamp               ups            downs        week
##  Min.   :2014-04-02   Min.   :10.00   Min.   :0   Min.   : 1.00
##  1st Qu.:2014-06-28   1st Qu.:13.00   1st Qu.:0   1st Qu.:13.00
##  Median :2014-10-20   Median :17.00   Median :0   Median :20.00
##  Mean   :2014-10-15   Mean   :17.74   Mean   :0   Mean   :23.46
##  3rd Qu.:2015-02-02   3rd Qu.:21.00   3rd Qu.:0   3rd Qu.:35.00
##  Max.   :2015-04-29   Max.   :30.00   Max.   :0   Max.   :52.00
##       year         n_users        n_comments   comments_per_user
##  Min.   :2014   Min.   :17.00   Min.   :32.0   Min.   :0.2833
##  1st Qu.:2014   1st Qu.:19.00   1st Qu.:37.0   1st Qu.:0.4500
##  Median :2014   Median :22.00   Median :43.0   Median :0.5135
##  Mean   :2014   Mean   :22.21   Mean   :43.7   Mean   :0.5198
##  3rd Qu.:2015   3rd Qu.:25.00   3rd Qu.:49.0   3rd Qu.:0.5802
##  Max.   :2015   Max.   :29.00   Max.   :60.0   Max.   :0.8438
##  threads_per_week
##  Min.   : 5.000
##  1st Qu.: 6.000
##  Median : 9.000
##  Mean   : 8.188
##  3rd Qu.:10.000
##  Max.   :10.000
```

Tokenizing

Our text cleaning function removes URLs, symbols and formatting, and a few common reddit terms.

```
clean_text <- function(text) {
  str_replace_all(
    text,
    c("https?:\\/\\/.*[\r\n]*" = " ",                          # URLs
      "_____|\\\n|&amp;|#x200B;|nbsp|&gt;" = " ",              # Symbols and formatting
      "'s" = " ")
  )
}
```

The following terms are removed due to their common usage in the r/ChangeMyView corpus.

```
my_stop_words <- c(
  "edit", "reddit", "cmv", "change", "view", "people", "person", "post",
  "vote", "delta", "score", "comment", "debate", "life", "feel", "time"
)
```

First, the text component of the threads is created by joining the thread title with the thread body text. This is what will be analyzed during topic modelling.

The thread text is tokenized into individual words. Non-alphabetic words are removed, as are any matches between the thread words and the two stop word lists. Words shorter than 3 characters are also dropped.

```
tokens <- sample %>%
  mutate(text = paste(title, text)) %>%
  mutate(text = clean_text(text)) %>%
  select(id, text) %>%
  unnest_tokens(word, text) %>%
  filter(str_detect(word, "[a-z]$")) %>%
  filter(!word %in% my_stop_words) %>%
  filter(!word %in% stop_words$word) %>%
  filter(length(word) > 2)
```

The remaining words are saved in a vector to be used as a dictionary during stem completion.

```
dictionary <- tokens %>%
  select(word) %>%
  unique() %>%
  arrange(word)

dictionary <- as.vector(dictionary$word)
```

Stemming is performed on the thread words. The stems are completed using the dictionary vector so that they are standardized and readable. Lost terms are dropped. The term frequencies are tallied for conversion to a document term matrix.

```
tokens <- tokens %>%
  mutate(word = wordStem(word, language = "english")) %>%
  mutate(word = stemCompletion(word, dictionary = dictionary)) %>%
  select(id, word) %>%
  filter(word != "") %>%
  count(id, word, sort = TRUE) %>%
  ungroup()

tokens
```

```
## # A tibble: 18,807 x 3
##    id        word          n
##    <chr>     <chr>     <int>
##  1 t3_25l3gr word         21
##  2 t3_29koa9 football     21
##  3 t3_29koa9 soccer       19
##  4 t3_2yrgi3 women        18
##  5 t3_25ikk0 car          17
##  6 t3_25ikk0 insurance    17
##  7 t3_26d20x language     17
##  8 t3_2a6nk5 animal       17
##  9 t3_2p6y53 dog          17
## 10 t3_2pkcvf immigrant    17
## # ... with 18,797 more rows
```

Modelling

The thread words are coverted to a document term matrix, and passed to the LDA function.

You can play with the `k =` argument to `LDA()` to pick different numbers of topics. 16 seems to be reasonable.

```
cmv_dtm <- tokens %>%
  cast_dtm(id, word, n)

cmv_lda <- cmv_dtm %>%
  LDA(k = 25)
```

Topics are visualized by their most frequently occuring words.

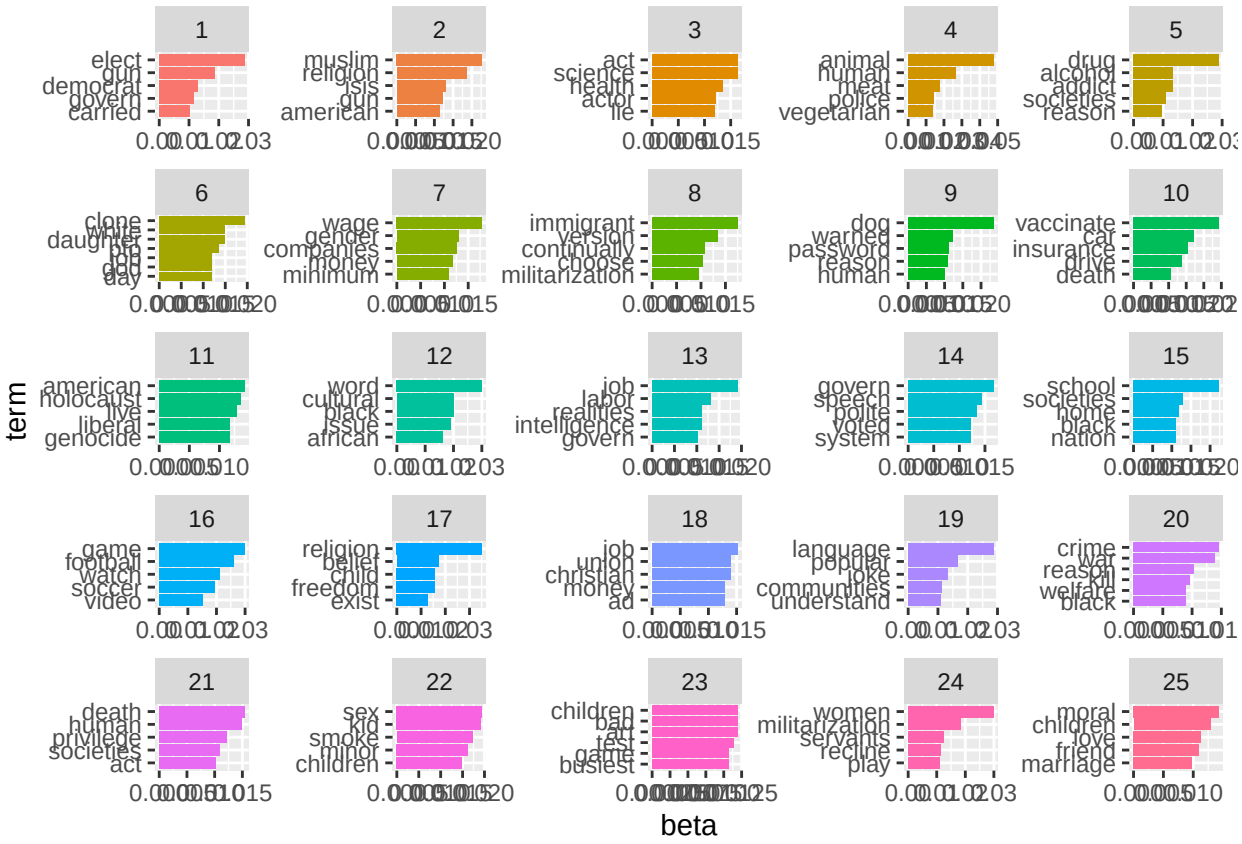
```
cmv_topics <- tidy(cmv_lda, matrix = "beta")

top_terms <- cmv_topics %>%
  group_by(topic) %>%
  top_n(5, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms
```

```
## # A tibble: 129 x 3
##    topic term       beta
##    <int> <chr>     <dbl>
##  1     1 elect    0.0287
##  2     1 gun      0.0185
##  3     1 democrat 0.0129
##  4     1 govern   0.0114
##  5     1 carried  0.0102
##  6     2 muslim   0.0227
##  7     2 religion 0.0186
##  8     2 isis     0.0130
##  9     2 gun      0.0122
## 10     2 american 0.0114
## # ... with 119 more rows
```

```
top_terms %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip() +
  scale_x_reordered()
```



The titles of the most representative threads for each topic are displayed for inspection.

```
thread_topics <- tidy(cmv_lda, matrix = "gamma")

topic_threads <- thread_topics %>%
  group_by(topic) %>%
  arrange(topic, desc(gamma)) %>%
  top_n(3, gamma) %>%
  ungroup() %>%
  left_join(sample, by = c("document" = "id")) %>%
  select(topic, title)

topic_threads
```

```
## # A tibble: 75 x 2
##    topic title
##    <int> <chr>
##  1     1 " Democracy is neither desirable nor fundamentally good."
##  2     1 " Democracy cannot be sustained, and will soon fail."
##  3     1 " There is no good reason for a store to ban the concealed carry of fi~
##  4     2 " Muslims are the most discriminated against in general (on Reddit and~
##  5     2 " Gun violence, education reform, concussions in American football, th~
##  6     2 " Expecting Muslims to protest against ISIS, is a double standard stee~
##  7     3 " I think moral laws can be established by logic &amp; science and wou~
##  8     3 " I don't think acting's hard. Just hear me out..."
##  9     3 " The Hobbit movies are over indulgent commercial diarrhea in the same~
## 10     4 " The strongest ethical arguments for veganism are stronger than those~
## # ... with 65 more rows
```