Homework 5

Allison Collins

11/25/2019

General NLP/Pre-processing

1. Load the data

```
dem_text <- paste(readLines("~/Documents/Problem-Set-5/Party Platforms Data/d16.txt"), collapse=" ")
rep_text <- paste(readLines("~/Documents/Problem-Set-5/Party Platforms Data/r16.txt"), collapse=" ")

d_doc = VCorpus(VectorSource(dem_text))
r_doc = VCorpus(VectorSource(rep_text))
```

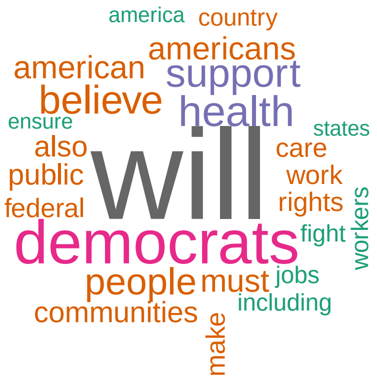
Defining a cleaning function to leverage since we will use on several different corpuses.

```
clean_text <- function(doc) {
  doc <- tm_map(doc, removePunctuation)
  doc <- tm_map(doc, tolower)
  doc <- tm_map(doc, removeNumbers)
  doc <- tm_map(doc, removeWords, stopwords("english"))
  doc <- tm_map(doc, stripWhitespace)
  doc <- tm_map(doc, PlainTextDocument)
}
```

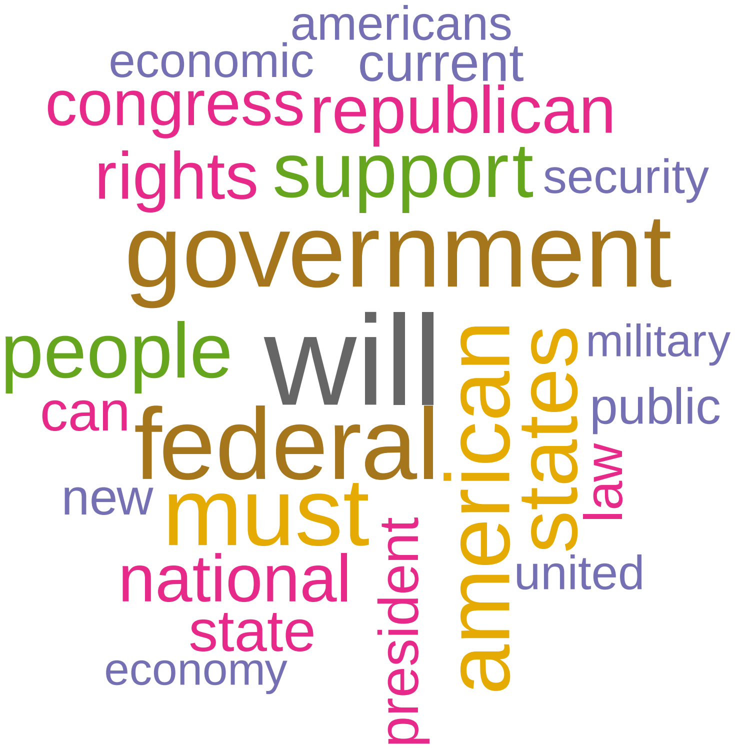
2. Create a document-term matrix and preprocess the platforms

Create frequencies - doing separately to avoid the cutoff

```
d_frequency <- sort(colSums(as.matrix(d_dtm)),
                    decreasing=TRUE)

set.seed(2345) #for word clouds

d_wordcloud <- wordcloud(names(d_frequency), d_frequency, max.words = 25,
                         random.order = FALSE, main = "Republican Party",
                         colors = brewer.pal(8, "Dark2"), scale = c(4, 0.3),
                         fixed.asp=TRUE)
```



```
r_frequency <- sort(colSums(as.matrix(r_dtm)),
                    decreasing=TRUE)

set.seed(2345) #for word clouds

r_wordcloud <- wordcloud(names(r_frequency), r_frequency, max.words = 25,
                          random.order = FALSE, main = "Republican Party",
                          colors = brewer.pal(8, "Dark2"), scale = c(4, 0.3),
                          fixed.asp=TRUE)
```



We can see some similarities and differences across the platforms. Both parties use “will” a lot, which makes sense, as we are looking toward the future in an election, and reference their own parties. In both, there are also clear references to America, Americans etc. and terms relating to government. We then see some differences which relate to common party platforms - e.g. “health” appears for democrats and “military” and “security” appears for Republicans. It’s also interesting to note that with democrats, “fight” and “workers” appear which show a base they could be targeting. Finally, “new” appears with republicans, which makes sense as democrats were the incumbent party.

Sentiment Analysis

4. Use the “Bing” and “AFINN” dictionaries to calculate the sentiment of each cleaned party platform. Present the results however you’d like (e.g., visually and/or numerically).

```
#Democrats

#Bing
d_text <- as.character(d_docs[[1]])
d_tbl <- tibble(txt=d_text)
```

```
d_bing <- d_tbl %>%
  unnest_tokens(word, txt) %>%
  inner_join(get_sentiments("bing")) %>%
  count(sentiment, sort = TRUE)
```

```
## Joining, by = "word"
```

```
#AFINN
d_afinn <- d_tbl %>%
  unnest_tokens(word, txt) %>%
  inner_join(get_sentiments("afinn"))
```

```
## Joining, by = "word"
```

```
#Make a table for words with negative and positive scores
d_afinn_table <- matrix(c(sum(d_afinn$value <0),sum(d_afinn$value >0)),ncol=2,byrow=TRUE)

#Display ouuputs for democrats
d_bing
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 positive   1372
## 2 negative    811
```

```
d_afinn_table
```

```
##      [,1] [,2]
## [1,]  737 1578
```

```
sum(d_afinn$value)
```

```
## [1] 1303
```

```
#Republicans
r_text <- as.character(r_doc[[1]])
r_tbl <- tibble(txt=r_text)

r_bing <- r_tbl %>%
  unnest_tokens(word, txt) %>%
  inner_join(get_sentiments("bing")) %>%
  count(sentiment, sort = TRUE)
```

```
## Joining, by = "word"
```

```
#AFINN
r_afinn <- r_tbl %>%
  unnest_tokens(word, txt) %>%
  inner_join(get_sentiments("afinn"))
```

```
## Joining, by = "word"
```

```
#Make a table for words with negative and positive scores
r_afinn_table <- matrix(c(sum(r_afinn$value <0),sum(r_afinn$value >0)),ncol=2,byrow=TRUE)

#Display ouuputs for republicans
r_bing
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 positive   1577
## 2 negative   1245
```

```
r_afinn_table
```

```
##      [,1] [,2]
## [1,]  973 1679
```

```
sum(r_afinn$value)
```

```
## [1] 939
```

5. Compare and discuss the sentiments of these platforms: which party tends to be more optimistic about the future? Does this comport with your perceptions of the parties?

Looking at the outputs, the ratio of positive to negative sentiment words is higher for democrats than republicans (in both bing and if we consider <0 negative and >0 positive for AFINN). Additionally, summing all the scores in AFINN further gives a higher value to Democrats. This is aligned with my expectations that democrats would have a more optimistic message - republicans were not in power at the time, thus it makes sense that they would be more negative to try to induce people to think they needed a change from the status quo.

Topic Models

6. With a general sense of sentiments of the party platforms (i.e., the tones related to how parties talk about their roles in the political future), now explore the topics they are highlighting in their platforms. This will give a sense of the key policy areas they're most interested in. Fit a topic model for each of the major parties (i.e. two topic models) using the latent Dirichlet allocation algorithm, initialized at k = 5 topics as a start. Present the results however you'd like (e.g., visually and/or numerically).

First, I am creating training + testing datasets, since we want to test perplexity using a holdout per class conversation

```
dem_words <- strsplit(dem_text," ")
dem_words <-unlist(dem_words)

sample_d = sample.split(dem_words,SplitRatio = 0.8)
train_dem =subset(dem_words, sample_d ==TRUE)
test_dem =subset(dem_words, sample_d == FALSE)

train_dem_c <- VCorpus(VectorSource(paste(train_dem, collapse = " ")))
test_dem_c <- VCorpus(VectorSource(paste(test_dem, collapse = " ")))

d_doc_train <- clean_text(train_dem_c)
d_dtm_train <- DocumentTermMatrix(d_doc_train)

d_doc_test <- clean_text(test_dem_c)
d_dtm_test <- DocumentTermMatrix(d_doc_test)
```

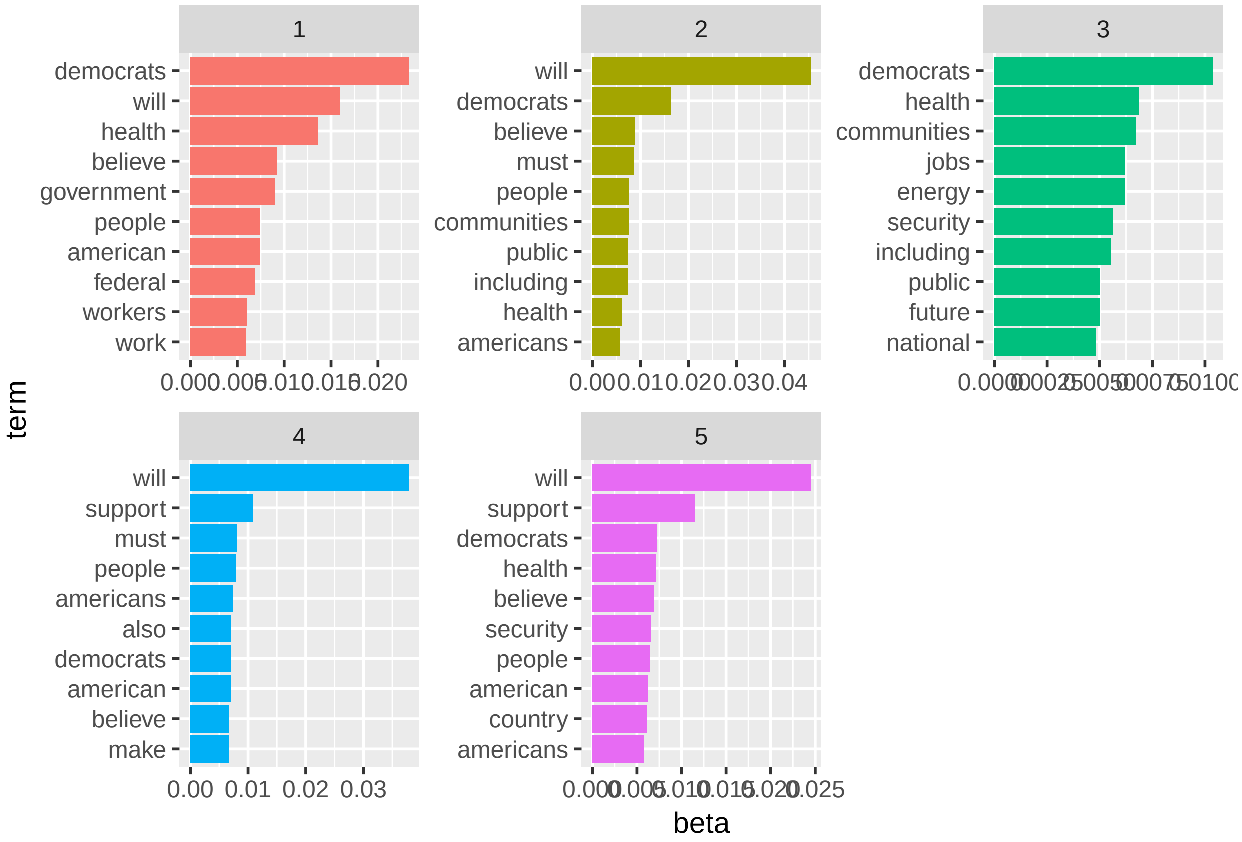
```r
rep_words <- strsplit(rep_text," ")
rep_words <-unlist(rep_words)

sample_r = sample.split(rep_words,SplitRatio = 0.8)
train_rep =subset(rep_words,sample_r ==TRUE)
test_rep =subset(rep_words,sample_r ==FALSE)

train_rep_c <- VCorpus(VectorSource(paste(train_rep, collapse = " ")))
test_rep_c <- VCorpus(VectorSource(paste(test_rep, collapse = " ")))

r_doc_train <- clean_text(train_rep_c)
r_dtm_train <- DocumentTermMatrix(r_doc_train)

r_doc_test <- clean_text(test_dem_c)
r_dtm_test <- DocumentTermMatrix(r_doc_test)
```

Now creating a function I will use several times to display the top terms per topic

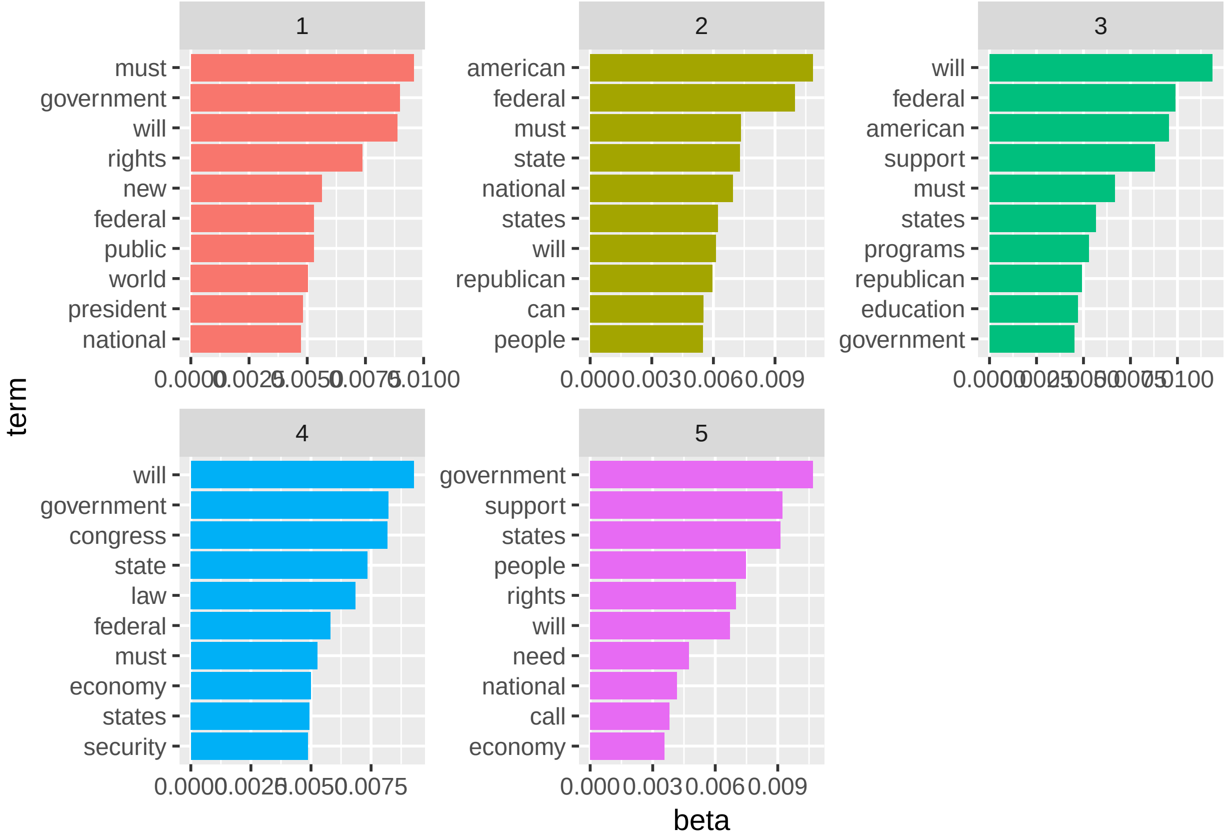
```r
#citation for some of the displays: https://www.tidytextmining.com/topicmodeling.html

plot_terms <- function(topics) {
  d_top_terms <- topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

  d_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()
  }
```

Now, creating topic models using the training data sets, first for k = 5 per instructions

```r
d_lda <- LDA(d_dtm_train, k = 5, control = list(seed = 1234))
d_topics <-tidy(d_lda, matrix = "beta")
plot_terms(d_topics)
```



```r
r_lda <- LDA(r_dtm, k = 5, control = list(seed = 1234))
r_topics <-tidy(r_lda, matrix = "beta")
plot_terms(r_topics)
```



7. Describe the general trends in topics that emerge from this stage. Are the parties focusing on similar or different topics, generally?

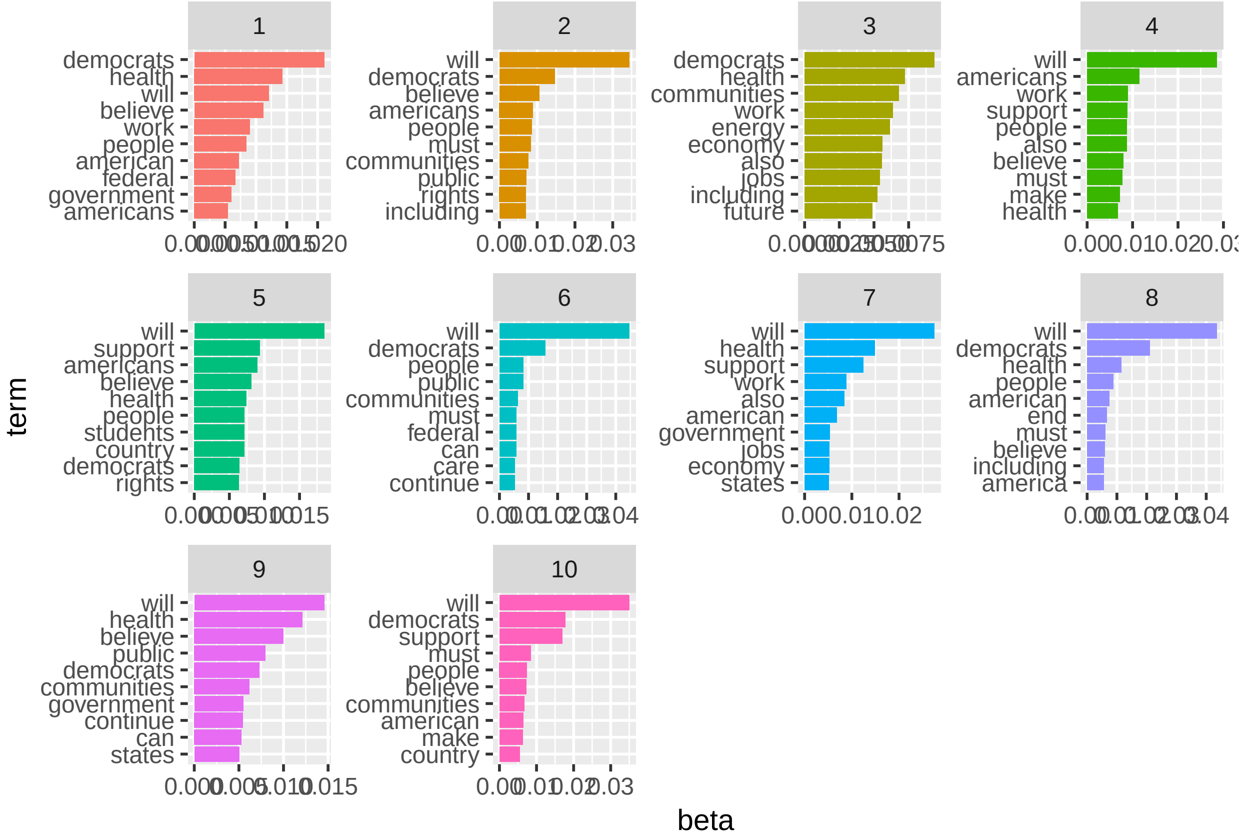
There is a lot of overlap in the parties talking about themselves and terms generally related to government. Similar to above when we looked at the word clouds, we can see certain topics emerging for the two parties that are more commonly associated with each one's platform, e.g. health, workers rights etc. with democrats and security etc. for republicans.

8. Fit 6 more topic models at the follow levels of k for each party: 5, 10, 25. Present the results however you'd like (e.g., visually and/or numerically).

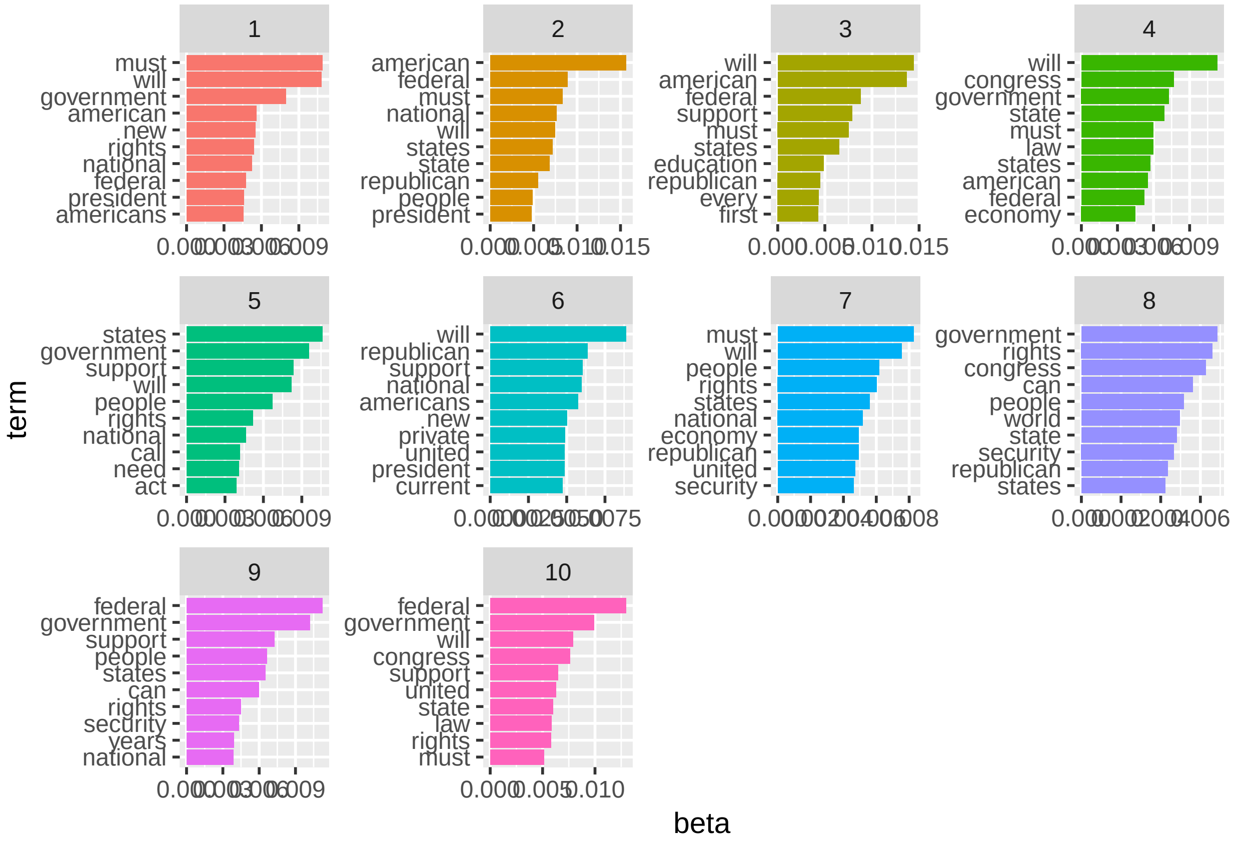
Per piazza, doing for 10, 25 as we already did 5

For 10 topic models

```
#Per piazza, doing for 10, 25 as we already did 5

d_lda_10 <- LDA(d_dtm_train, k = 10, control = list(seed = 1234))
d_topics_10 <-tidy(d_lda_10, matrix = "beta")

r_lda_10 <- LDA(r_dtm_train, k = 10, control = list(seed = 1234))
r_topics_10 <-tidy(r_lda_10, matrix = "beta")

plot_terms(d_topics_10)
```

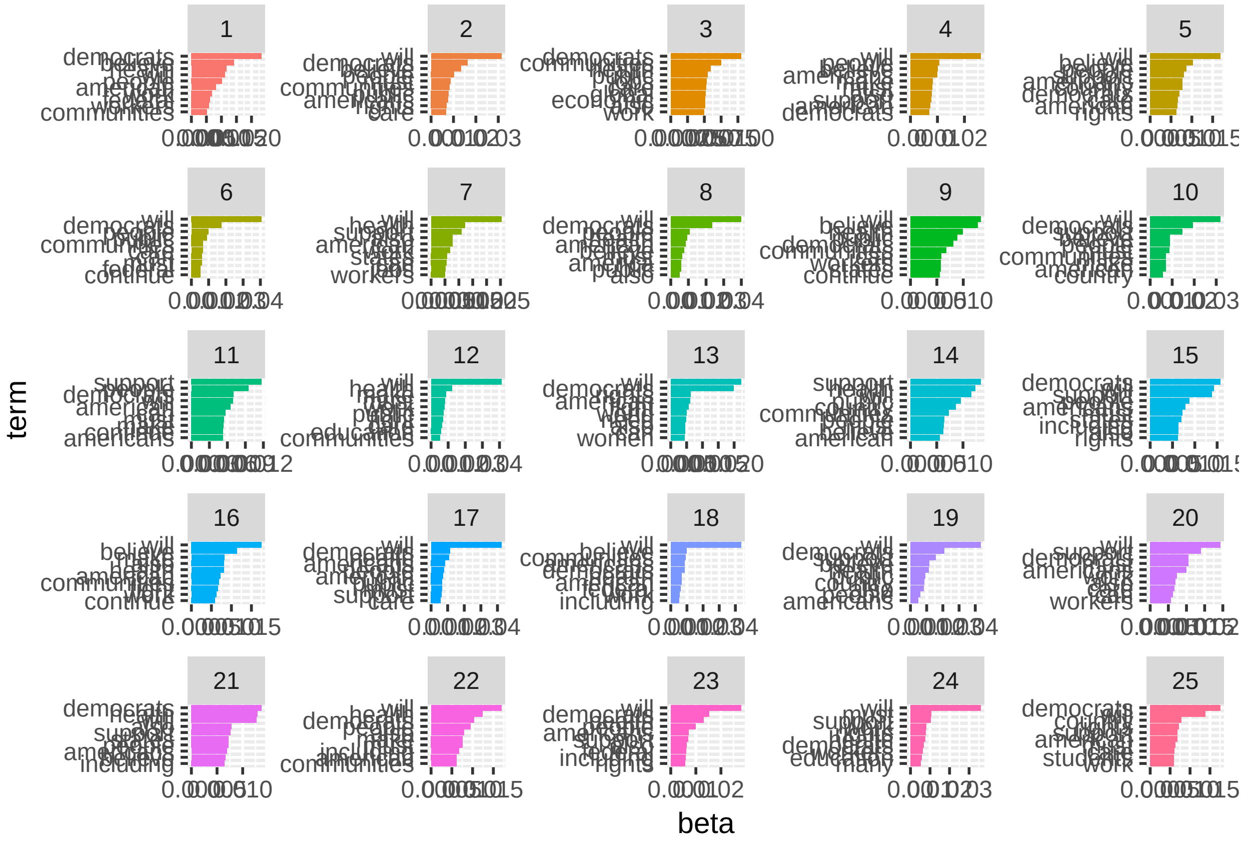


```
plot_terms(r_topics_10)
```

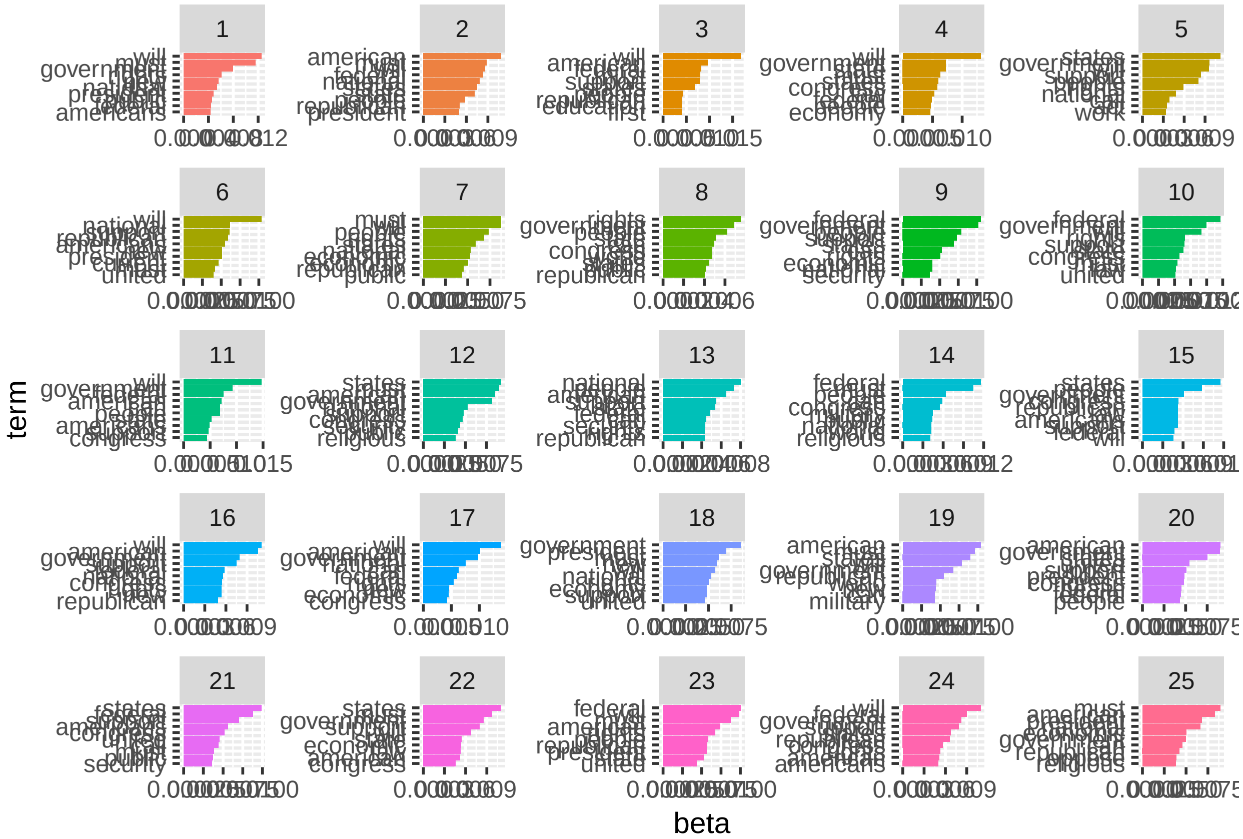


For 25 topic models

```
#Per piazza, doing for 10, 25 as we already did 5

d_lda_25 <- LDA(d_dtm_train, k = 25, control = list(seed = 1234))
d_topics_25 <-tidy(d_lda_25, matrix = "beta")

r_lda_25 <- LDA(r_dtm_train, k = 25, control = list(seed = 1234))
r_topics_25 <-tidy(r_lda_25, matrix = "beta")

plot_terms(d_topics_25)
```



```r
plot_terms(r_topics_25)
```



##9. Calculate the perplexity of each model iteration and describe which technically fits best.

```
#For democrats
perplexity(d_lda,d_dtm_test)
```

```
## [1] 897.5014
```

```
perplexity(d_lda_10,d_dtm_test)
```

```
## [1] 901.0312
```

```
perplexity(d_lda_25,d_dtm_test)
```

```
## [1] 911.8072
```

```
#For republicans
perplexity(r_lda,r_dtm_test)
```

```
## [1] 1378.378
```

```
perplexity(r_lda_10,r_dtm_test)
```

```
## [1] 1424.683
```

```
perplexity(r_lda_25,r_dtm_test)
```

```
## [1] 1438.862
```

Above, you can see the perplexity for each of the model runs. For both, the 5 topic model fits best (by a small margin compared to 10-topic) and it continues to get worse for the 25-topic. We know they fit best because they have the lowest perplexity.

10. Building on the previous question, display a barplot of the k = 10 model for each party, and offer some general inferences as to the main trends that emerge. Are there similar themes between the parties? Do you think k = 10 likely picks up differences more efficiently? Why or why not?

I have displayed the barplots of top terms by topic for k = 10 above. Looking back to them to answer this question, I would echo that we see similar themes emerge for each party as discussed from the wordclouds - however, in some instances, this does not appear in the k = 5 top terms, and 10 may better capture it. For instance, when we go to a topic of 10 for republicans, we can see that "private" (and consequently a topic built around it) appears, a stance typically held by republicans that contrasts democrats (e.g. privatization of economy, etc) which does not get a topic in the model of 5. However, I would say that we can still see some distinctions at the level of 5 - as we can see topics specific to each party emerging for ex. for Democrats there is one that appears to be around public health. At 25, there is much repetition of words among topics. It is worth noting as well that perhaps additional removal of more custom stopwords particular to such political speeches could reduce the overlap a bit across the speeches.

Conclusion

11. Per the opening question, based on your analyses (including exploring party brands, general tones/sentiments, political outlook, and policy priorities), which party would you support in the 2020 election (again, this is hypothetical)?

These analyses would reinforce to me that I should support the democratic party. Firstly, as a young individual, I identify with messages of optimism. Secondly, for the topics, key words, etc. we surfaced for each party, the democratic party's messaging was in line with issues that are important to me such as healthcare (I am in health program and that is a space I previously worked in), workers' rights etc. I am less inclined to support military spend etc. and military, security etc. popped up for republican topics.