Homework 3

Sanittawan Tan

11/22/2019

Load libraries

```
library(tm)
```

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

```
library(grid)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(wordcloud2)
library(SnowballC)
library(ggplot2)
```

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

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

```
library(gridExtra)
library(topicmodels)
library(tidyverse)
```

```
## Registered S3 method overwritten by 'rvest':
##   method            from
##   read_xml.response xml2
```

```
## -- Attaching packages ---------------------------------- tidyverse 1.2.1 --
```

```
## v tibble  2.1.3     v purrr   0.3.3
## v tidyr   1.0.0     v dplyr   0.8.3
## v readr   1.3.1     v stringr 1.4.0
## v tibble  2.1.3     v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------- tidyverse_conflicts() --
## x ggplot2::annotate() masks NLP::annotate()
## x dplyr::combine()    masks gridExtra::combine()
## x dplyr::filter()     masks stats::filter()
## x dplyr::lag()        masks stats::lag()
```

```
library(tidytext)
```

General NLP/Preprocessing

Question 1 — Load text

```
# I set working directory here
rep_dir <- file.path(".", "textrep")
dem_dir <- file.path(".", "textdem")

# create 2 different corpus, each with 1 document
rep_doc <- VCorpus(DirSource("textrep"))
dem_doc <- VCorpus(DirSource("textdem"))
rep_doc
```

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

```
dem_doc
```

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

After having created the corpus, I checked the documents here. Since the output is long, I am omitting it from the writeup.

```
writeLines(as.character(rep_doc))
writeLines(as.character(dem_doc))
```

Question 2 — Preprocess and create document-term matrix

```
process_text <- function(doc) {
    # convert to lower case
    doc <- tm_map(doc, tolower)
    # remove stopwords
    doc <- tm_map(doc, removeWords, stopwords("english"))
    # remove numbers
    doc <- tm_map(doc, removeNumbers)
    # remove punctuation
    doc <- tm_map(doc, removePunctuation,
                  preserve_intra_word_dashes = TRUE)
    # remove whitespace
    doc <- tm_map(doc, stripWhitespace)
    # convert to plaintext
    doc <- tm_map(doc, PlainTextDocument)
}
```

```
p_rep <- process_text(rep_doc)
p_dem <- process_text(dem_doc)

# I checked the document here
# The outputs are too long; not printing out
# writeLines(as.character(p_rep))
# writeLines(as.character(p_dem))
```

This is a custom function for removing specific characters and words.

```
remove_words <- function(doc, word) {
    for (i in seq(doc)) {
        # substitute word with empty string
        doc[[i]]$content <- gsub(word, "", doc[[i]]$content)

        # remove leading and trailing whitespace
        doc[[i]]$content <- gsub("^\\s+|\\s+$", "", doc[[i]]$content)
    }
    return(doc)
}
```

I think that there is ground for removing the words Republican/Republicans and Democrat/Democrats from the Republican and Democratic platforms respectively since the documents are going to refer to these terms a lot, but they do not contribute to new information since we already know which platform is from which party. The word Republicans appears 119 times while the word Democrats appear 207 times.

```
things_to_remove <- c("/", "\\", "@", "-", "’")

# Republicans
for (char in c("republicans", "republican", "will", "must", "-")) {
    p_rep <- remove_words(p_rep, char)
}

# Democrats
for (char in c("democrats", "democrat", "will", "must", "-")) {
    p_dem <- remove_words(p_dem, char)
}
```

There are no words to be put back together since I chose to preserved words that were connected by hyphens when I eliminated the punctuation.

I also created the stemmed versions.

```
p_rep_st <- tm_map(p_rep, stemDocument)
p_dem_st <- tm_map(p_dem, stemDocument)
```

Create Document-Term matrix

```
rep_dtm <- DocumentTermMatrix(p_rep)
dem_dtm <- DocumentTermMatrix(p_dem)
```

Question 3 — Inspection via wordcloud

Regarding the commonly used words, they are displayed in the wordclouds. For less commonly used words, I will pull from the frequency lists instead. (Discussion at the end)

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

```
## government    federal   american     states    support     people
##        137        134        121        110        100         98
```

```
set.seed(986)
wordcloud(names(r_frequency), r_frequency,
          scale=c(2,0.5), max.words = 150,
```

```
          random.order = FALSE,
          rot.per = 0.30,
          main = "Title",
          colors = brewer.pal(6, "Set2")
          )
```


```
tail(r_frequency, 30)
```

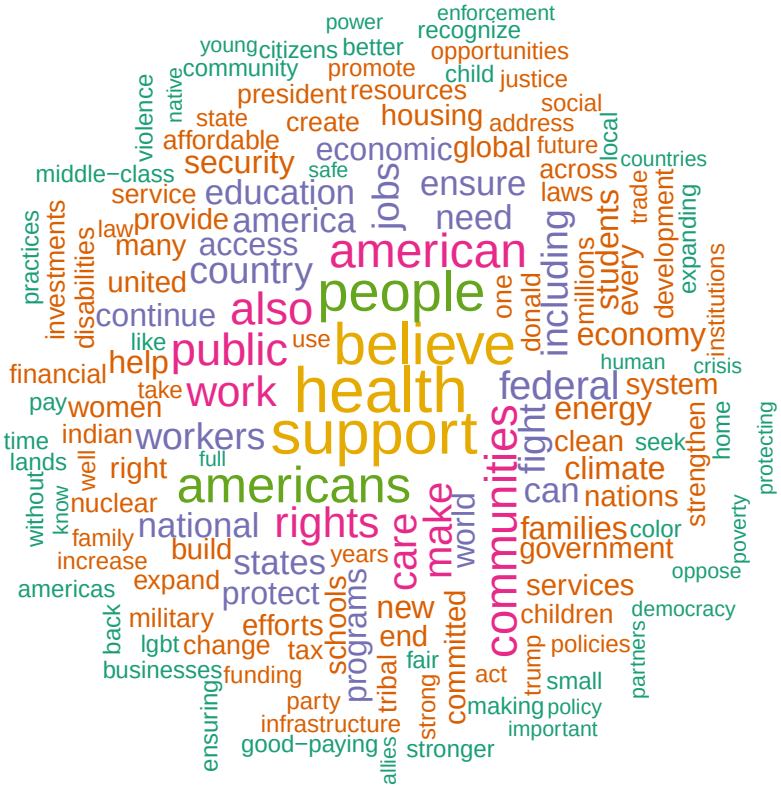
```
##      wilson         win        wind     windsor      winter       wired
##           1           1           1           1           1           1
##      wished    withdraw  withdrawal   withdrawn    withhold   withstood
##           1           1           1           1           1           1
##      womans     wonders        word  work-based      worked world-class
##           1           1           1           1           1           1
##       worse  worthwhile      wreaks       wreck    wreckage       wrote
##           1           1           1           1           1           1
##    xinjiang      yazidi  yesterdays       youth        zika         zip
##           1           1           1           1           1           1
```

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

```
##    health   support   believe    people americans  american
##       130       123       117       107        92        86
```

```
set.seed(986)
wordcloud(names(d_frequency), d_frequency,
          scale=c(2,0.5), max.words = 150,
          random.order = FALSE,
```

```
          rot.per = 0.30,
          main = "Title",
          colors = brewer.pal(6, "Dark2")
          )
```




```
tail(d_frequency, 30)
```

```
##     western    whenever     whereas    wherever      whites wholesalers
##           1           1           1           1           1           1
##      widely       widen       wider  widespread     widowed        wifi
##           1           1           1           1           1           1
##    wildfire    wireless      wishes       woman       words   workplace
##           1           1           1           1           1           1
##       worth     wounded      wounds  wrongdoing     wyoming  year-round
##           1           1           1           1           1           1
##     yezidis     zealand        zero        zika    zikajust       zones
##           1           1           1           1           1           1
```

Top 5 commonly used words in the platforms for the Republican Party are government, federal, american, states, support, and people. The top 5 commonly used words for the Democratic Party are different: health, support, believe, people, americans, american. However, the two words in common are "people", "support" and "american(s)," which makes perfect sense since both parties are speaking to American citizens and they are to "support" the people and various policies. The less commonly used words (based on the wordclouds) are also different but not by much. Both parties mentioned issues such as healthcare and trade although words related to health and healthcare appear more often in the Democratic platform. In both platforms, we also see issues such as foreign policy and defense, specifically the word nuclear that appears in the Democrat's wordcloud.

I can sense several differences between the two parties based on the wordclouds. Firstly, while the Republican Party appears to focus on issues such as trade, health, education, and military, they tend to place equal

weights across a range of issues. What I do not see from the Democrats' wordcloud is religious issue which is present with relatively high frequency in the Republican platform. In contrast, one thing that jumps out of the Democratic Party's platform to me is diversity and class. There are words like working class, good-paying (jobs), Indian (which I assume to be native Americans), students, disabilities, and women. What is missing from the Republican platform is climate change which appears with high frequency in the Democratic platform. These differences suggest that each party has different policy priorities though some of them overlapped.

Sentiment Analysis

Question 4 — Dictionary Approach

Instead of using the CSV file, I directly casted the document-term matrix to tidy text data. I noticed that the platforms of both parties in the CSV file were cut off at the end (not by much). Next, I wrote a customized function to perform sentiment analysis using the dictionary approach.

```
analyze_sentiments <- function(tidy_data, sent_dict) {
    rv <- tidy_data %>%
        inner_join(get_sentiments(sent_dict), by = c(term = "word"))
    return(rv)
}
```

Bing

```
rep_bing <- analyze_sentiments(rep_tidy, "bing")

dem_bing <- analyze_sentiments(dem_tidy, "bing")
```

```
rep_bing <- rep_bing %>%
    mutate(score = ifelse(sentiment == "positive", 1, -1))
# not sure why but there are repeated terms (possibly from inner join)
rep_bing <- distinct(rep_bing, term, .keep_all = TRUE)

dem_bing <- dem_bing %>%
    mutate(score = ifelse(sentiment == "positive", 1, -1))
# not sure why but there are repeated terms (possibly from inner join)
dem_bing <- distinct(dem_bing, term, .keep_all = TRUE)
```

```
sum(rep_bing$score * rep_bing$count)
```

```
## [1] 351
```

```
sum(dem_bing$score * dem_bing$count)
```

```
## [1] 588
```

AFINN

```
rep_afinn <- analyze_sentiments(rep_tidy, "afinn")
rep_afinn <- distinct(rep_afinn, term, .keep_all = TRUE)
dem_afinn <- analyze_sentiments(dem_tidy, "afinn")
dem_afinn <- distinct(dem_afinn, term, .keep_all = TRUE)
sum(rep_afinn$value * rep_afinn$count)
```

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

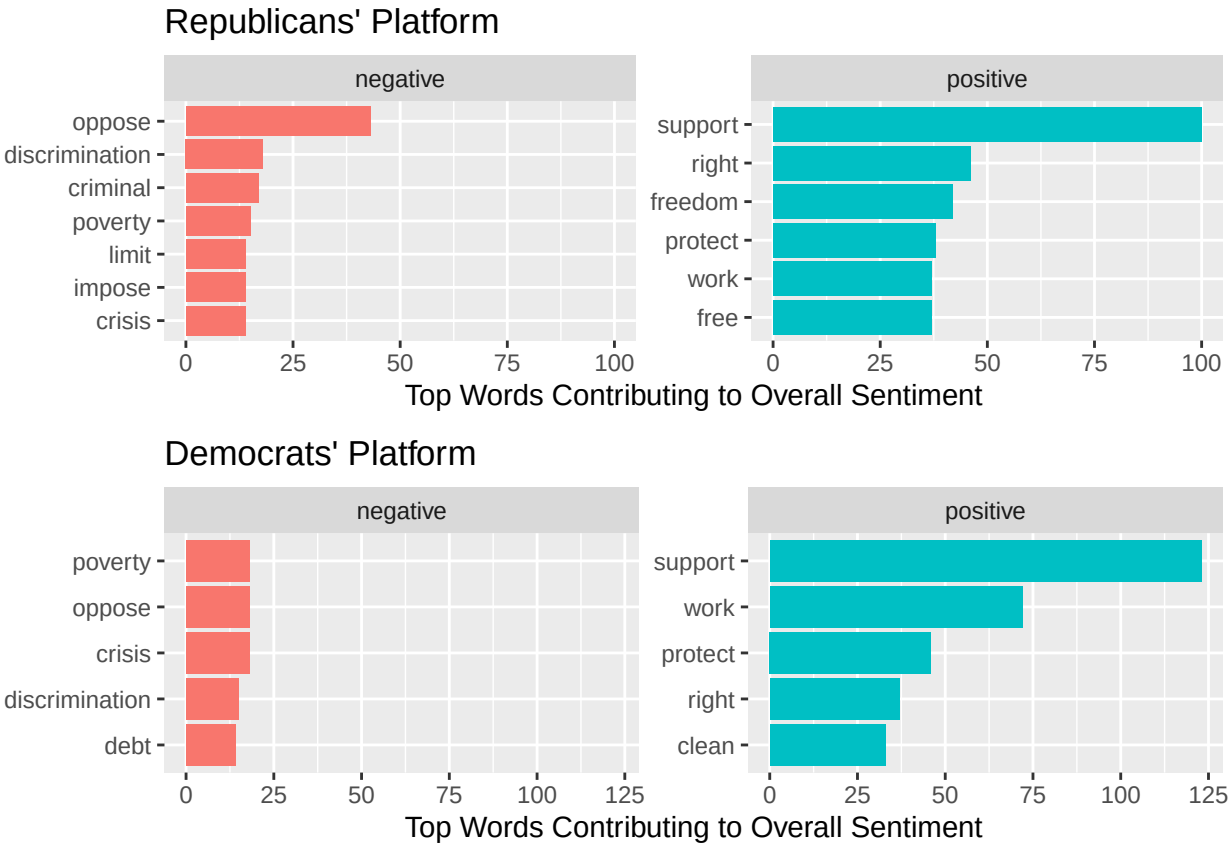
```
sum(dem_afinn$value * dem_afinn$count)
```

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

Numerics By naively summing up the product of the number of times a word appear and the translated sentiment score (in 1 and -1 forms in the case of Bing), we found that overall the Democrats' platform is more positive than that of the Republicans via the Bing dictionary. This is similar to the result we have from AFINN dictionary. We also found that both platforms are positive which may suggest optimism. Notice that the scores from AFINN is higher due to the scale differences. If we really want to compare the 2 approaches, I think that standardization to make 2 scales comparable will be needed. However, for this exercise, we are only interested in understanding the direction of sentiments and raw scores should be sufficient.

Visuals I am going to plot the top 15 terms that contribute to the overall sentiment for each party's platform.

```
rep_bing_sorted <- rep_bing %>%
    arrange(desc(count))

dem_bing_sorted <- dem_bing %>%
    arrange(desc(count))

p1 <- rep_bing_sorted %>%
  group_by(sentiment) %>%
  top_n(5, count) %>%
  ungroup() %>%
  mutate(term = reorder(term, count)) %>%
  ggplot(aes(term, count, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Top Words Contributing to Overall Sentiment",
       x = NULL) +
  coord_flip() +
  ggtitle("Republicans' Platform")

p2 <- dem_bing_sorted %>%
  group_by(sentiment) %>%
  top_n(5, count) %>%
  ungroup() %>%
  mutate(term = reorder(term, count)) %>%
  ggplot(aes(term, count, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Top Words Contributing to Overall Sentiment",
       x = NULL) +
  coord_flip() +
  ggtitle("Democrats' Platform")

grid.arrange(p1, p2, nrow = 2)
```



The top words that contribute to overall sentiments are not starkly different.

Question 5 — Discussion

Which party tends to be more optimistic about the future? Does this comport with your perceptions of the parties?

Based on the sentiment analysis of each party's platform via the dictionary approach, we found that the Democratic platform is more positive, or optimistic about the future, than the Republican platform given that positivism implies optimism. Based on conventional wisdom, platforms are, on average, optimistic, positive, and forward-looking since they were put out by the party and not necessarily candidates per se. Those who followed the 2016 race closely would know that Donald Trump is a more pessimistic candidate than Hilary Clinton. This is reflected in his speeches, especially when he spoke about the fact that Americans jobs were stolen and immigrants. Nevertheless, we still found that both platforms are positive though the Democratic Party's platform is more positive. Therefore, our findings comport with our understanding about party's platforms.

Regarding the parties, although I am inclined to generalize that the Republican party is, on average, less optimistic about the future than the Democratic Party in presidential campaigns, it appears to depend largely on the candidates. Obviously, one of the characteristics of conservatism is that they favor gradual change over abrupt revolution that progressives prefer. Therefore, it may be deductable that conservatives are not so optimistic about the future. Some of them might even yearn the "golden eara" of the past. Nevertheless, the Republican party had a candidate like Ronald Reagan who is famous for his "Morning in American" advertisement which is very positive and optimistic. This is in stark contrast with Donald Trump's campaign in 2016. Therefore, the parties' optimism about the future tends to evolve over the years and so do my perceptions. In short, my answer is it depends on the presidential candidates. And, yes, if we were to compare Donald Trump vs. Ronald Reagan and Hilary Clinton vs. Barack Obama, both parties have become less

optimistic/positive in their campaigns in 2016.

Topic Models

Question 6 — Fit 2 topic models using LDA $k = 5$ topics

```
# need to clean up rows with sum == 0
remove_all_zeros <- function(dtm) {
    row_total <- apply(dtm, 1, sum)
    rv <- dtm[row_total > 0,]
    return(rv)
}
new_rep_dtm <- remove_all_zeros(rep_dtm)
new_dem_dtm <- remove_all_zeros(dem_dtm)
# check
rows_with_zero_sum <- apply(new_rep_dtm, 1, sum)
print(new_rep_dtm[rows_with_zero_sum == 0,]$dimnames[1][[1]])
```

```
## NULL
```

```
rows_with_zero_sum2 <- apply(new_dem_dtm, 1, sum)
print(new_dem_dtm[rows_with_zero_sum2 == 0,]$dimnames[1][[1]])
```

```
## NULL
```

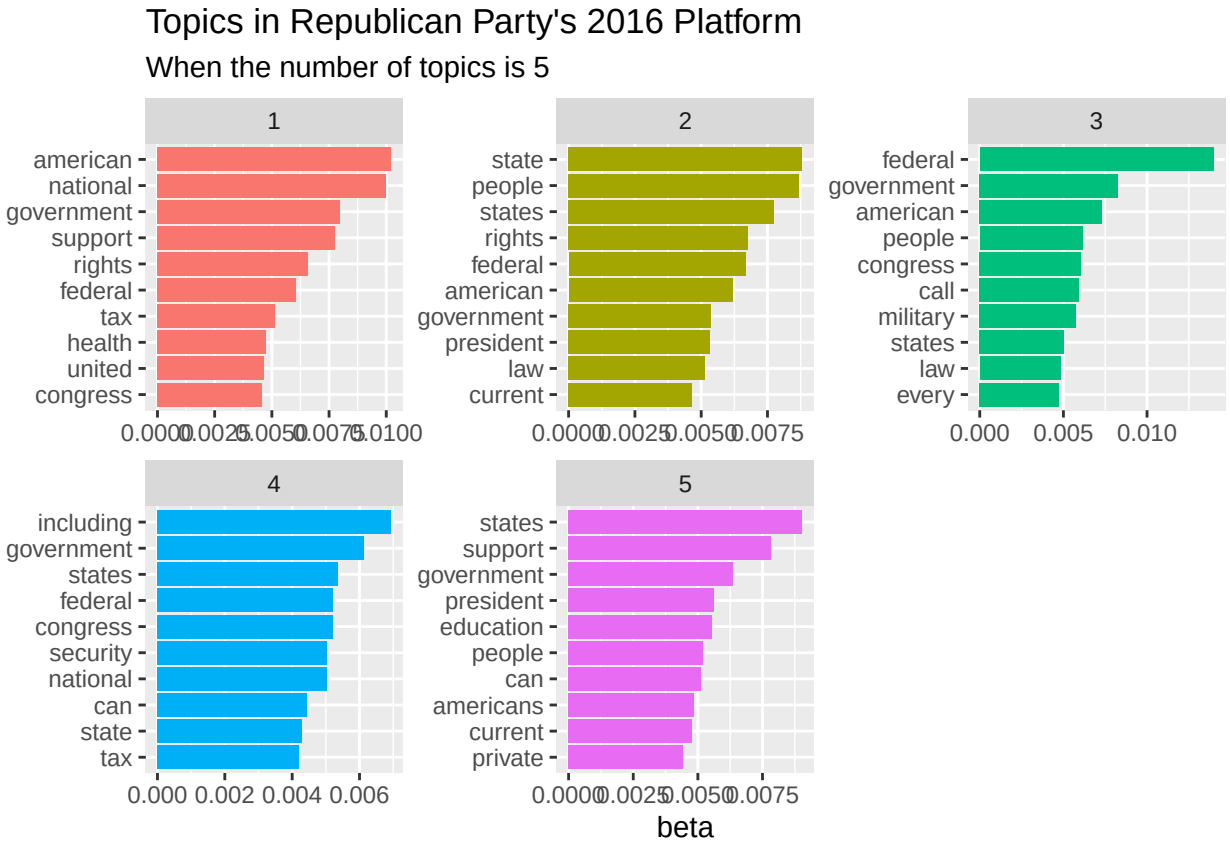
```
get_top_terms <- function(topics) {
    top_terms <- topics %>%
        group_by(topic) %>%
        top_n(10, beta) %>%
        ungroup() %>%
        arrange(topic, -beta)
    return(top_terms)
}

plot_by_topics <- function(top_terms, party, num_topics) {
    Title <- paste("Topics in", party, "2016 Platform", sep=" ")
    Subtitle <- paste("When the number of topics is", num_topics, sep=" ")

    top_terms %>%
        mutate(topic = as.factor(topic),
                term = reorder_within(term, beta, topic)) %>%
        ggplot(aes(term, beta, fill = topic)) +
        geom_col(show.legend = FALSE) +
        facet_wrap(~ topic, scales = "free") +
        coord_flip() +
        scale_x_reordered() +
        labs(y = "beta",
            x = NULL,
            title = Title,
            subtitle = Subtitle)
}
```

```
rep_lda <- LDA(new_rep_dtm, k = 5, control = list(seed = 54))
rep_5topics <- tidy(rep_lda, matrix = "beta")
```

```
r_5_perpl <- c(perplexity(rep_lda, newdata = new_rep_dtm), 5)
print(r_5_perpl)
```

```
## [1] 2363.274    5.000
```

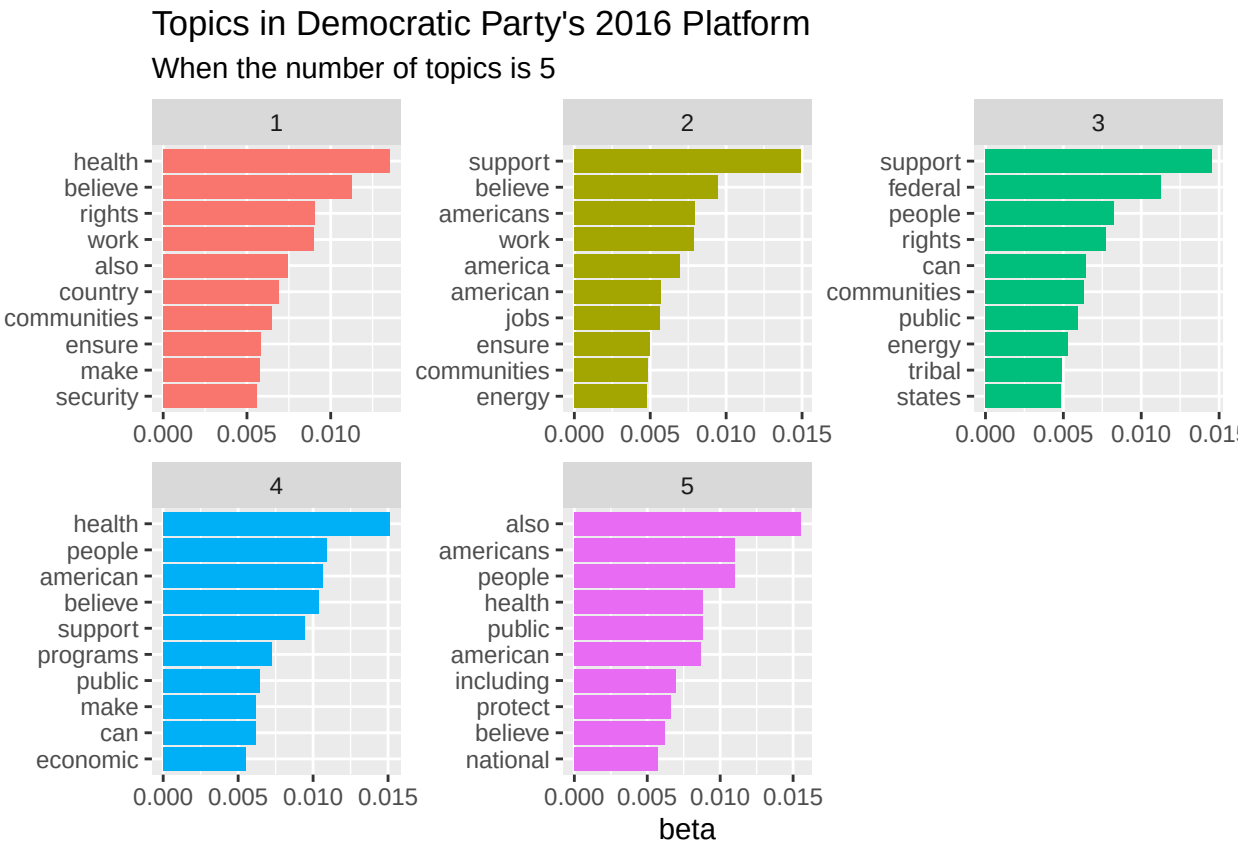
```
rep_top_terms <- get_top_terms(rep_5topics)
plot_by_topics(rep_top_terms, "Republican Party's", "5")
```




```
dem_lda <- LDA(dem_dtm, k = 5, control = list(seed = 12345))
dem_5topics <- tidy(dem_lda, matrix = "beta")
d_5_perpl <- c(perplexity(dem_lda, newdata = new_dem_dtm), 5)
print(d_5_perpl)
```

```
## [1] 1674.802    5.000
```

```
dem_top_terms <- get_top_terms(dem_5topics)
plot_by_topics(dem_top_terms, "Democratic Party's", "5")
```



Question 7 — General trends in topics of each party

Are parties focusing on similar or different topics, generally?

Based on 5 topic-model, the general trend for both parties are not very clear since many topics are redundant. For instance, in the results of the Republican Party, we see economic, energy, federal and congressional issues. In the results for the Democratic Party, we see a lot of issues related to health in 4 out of 5 topics. This suggests that grouping the platforms into 5 issues may not be sufficient to capture all of the underlying topics in the data since we know that there are more than 5 policy issues that were included in the platforms. The main contrast that we take away from this model is that the health issue is missing in the Republican Party's topics, while the Democrats seem to focus much more on the issue. Next, we move on to topic models with 10 and 25 topics to see if we will obtain better results.

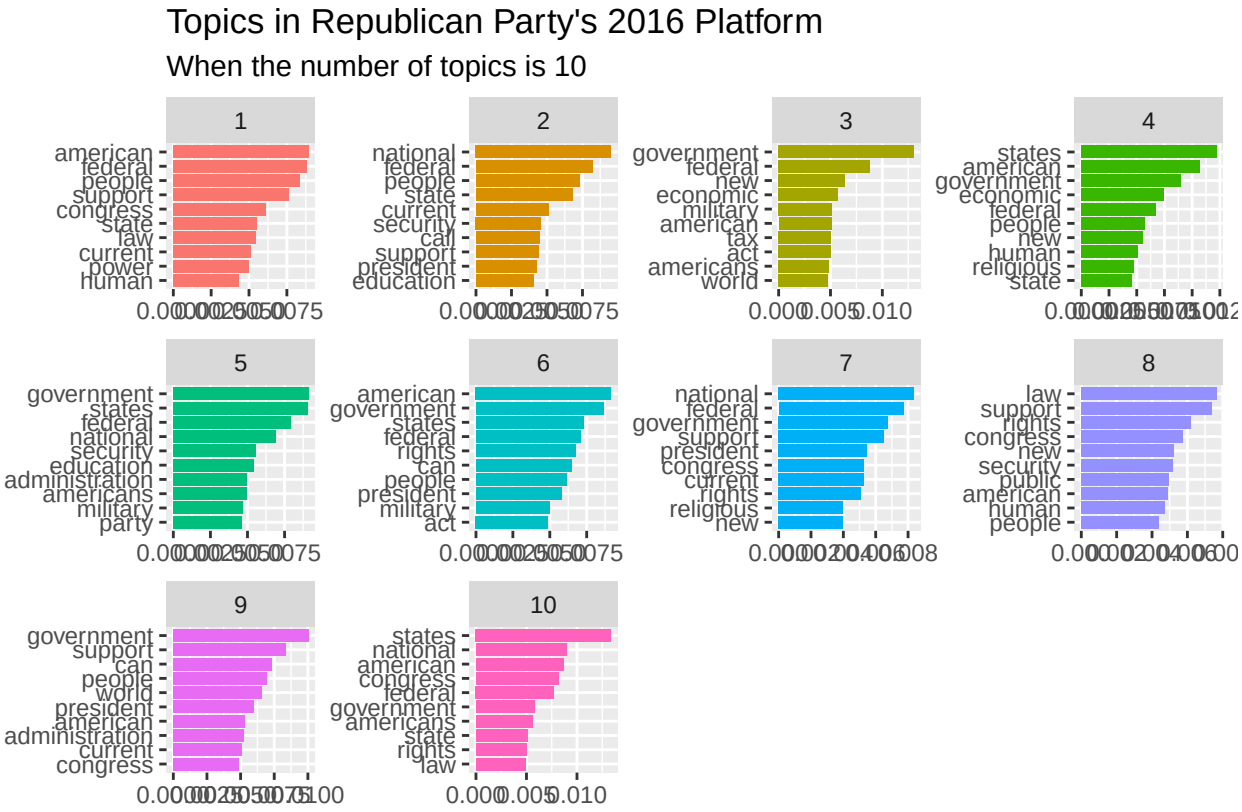
Question 8 — Fit 6 more topic models using LDA $k = 5, 10, 25$ (3 for each party)

k = 10

```
rep_10lda <- LDA(new_rep_dtm, k = 10, control = list(seed = 56723))
rep_10topics <- tidy(rep_10lda, matrix = "beta")
r_10_perpl <- c(perplexity(rep_10lda, newdata = new_rep_dtm), 10)
print(r_10_perpl)
```

```
## [1] 2364.254   10.000
```

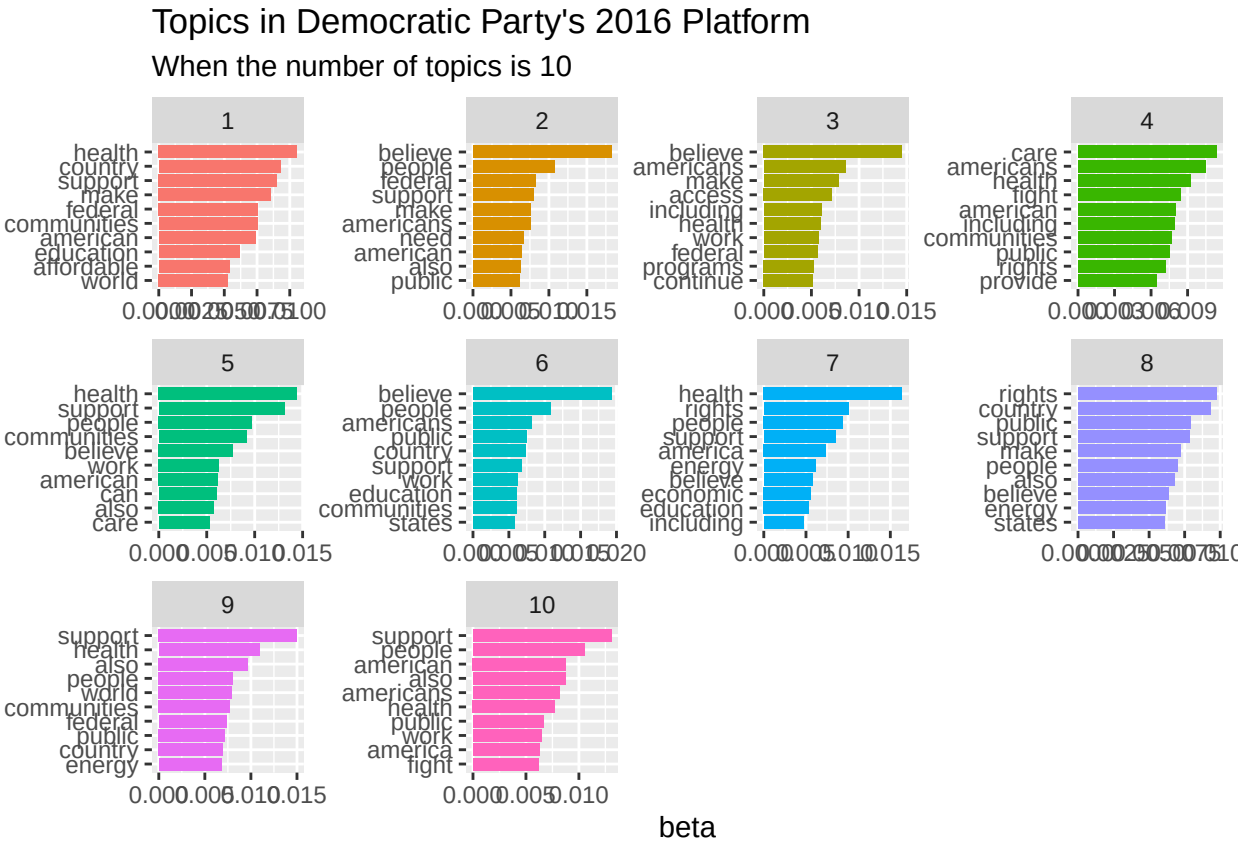
```
rep_10top_terms <- get_top_terms(rep_10topics)
plot_by_topics(rep_10top_terms, "Republican Party's", "10")
```




```
dem_10lda <- LDA(dem_dtm, k = 10, control = list(seed = 1234))
dem_10topics <- tidy(dem_10lda, matrix = "beta")
d_10_perpl <- c(perplexity(dem_10lda), 10)
print(d_10_perpl)
```

```
## [1] 1675.83   10.00
```

```
dem_10top_terms <- get_top_terms(dem_10topics)
plot_by_topics(dem_10top_terms, "Democratic Party's", "10")
```



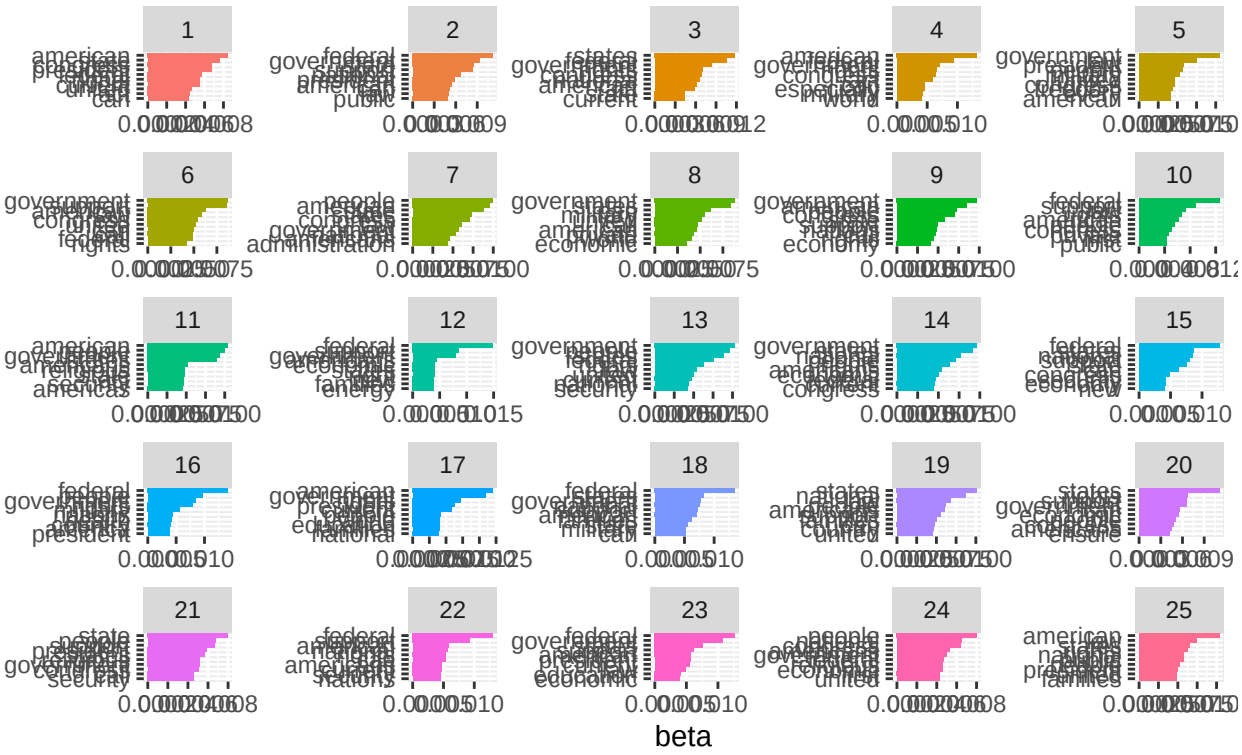
I think the results slightly improved when the number of topics is equal to 10 because we see clearer partitions between topics than in the 5-topic model. Nevertheless, there are some overlapping topics in the results for both parties. We start to see bigger contrast between the two parties' platforms. For example, the Republican party seems to be focusing on education, state and congressional affairs, economic, military and national security issues while the Democratic party put more emphasis on jobs, healthcare, students, and climate change. Based on this result, one would think that there are not a lot of overlapping issues between the two parties. Next, we fit the 25-topic model.

k = 25

```
rep_25lda <- LDA(new_rep_dtm, k = 25, control = list(seed = 1234))
rep_25topics <- tidy(rep_25lda, matrix = "beta")
r_25_perpl <- c(perplexity(rep_25lda), 25)
print(r_25_perpl)
```

```
## [1] 2366.983   25.000
```

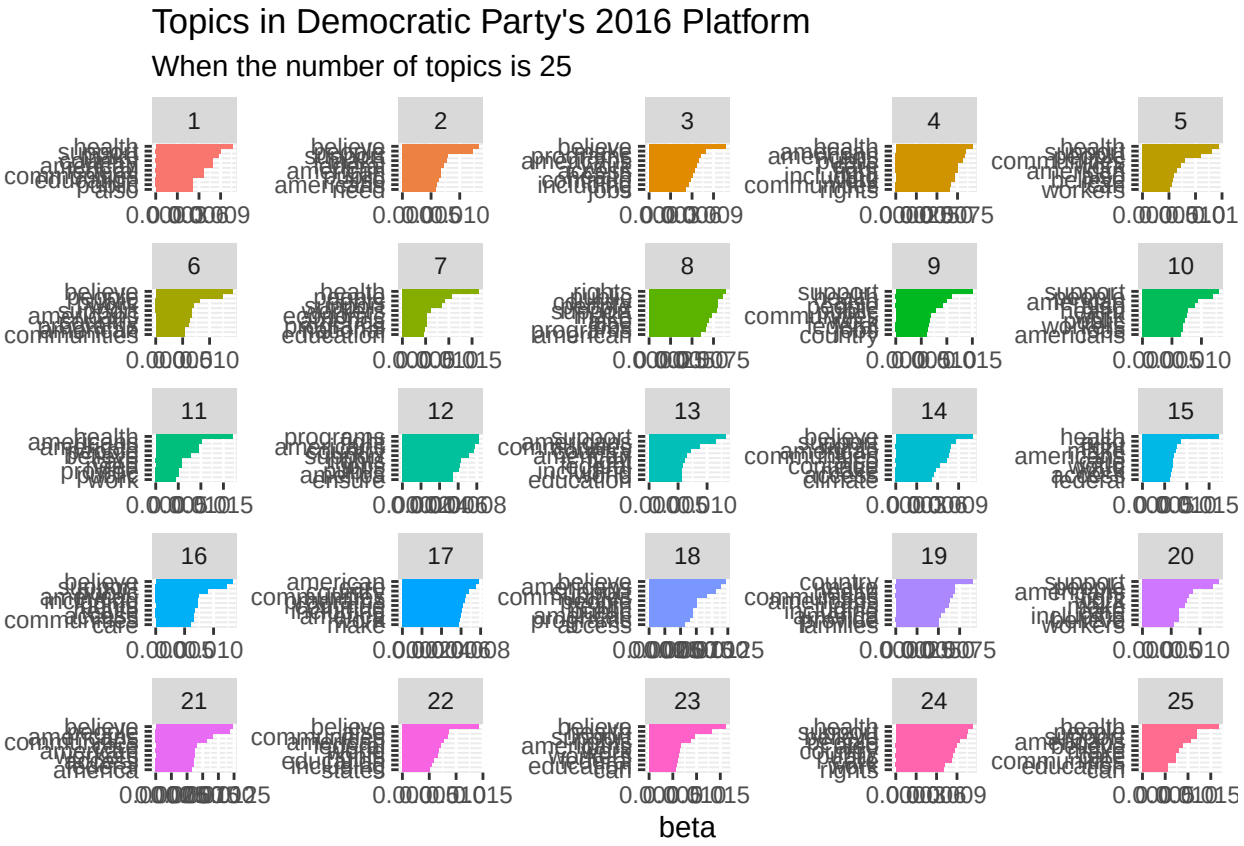
```
rep_25top_terms <- get_top_terms(rep_25topics)
plot_by_topics(rep_25top_terms, "Republican Party's", "25")
```




```
dem_25lda <- LDA(dem_dtm, k = 25, control = list(seed = 1234))
dem_25topics <- tidy(dem_25lda, matrix = "beta")
d_25_perpl <- c(perplexity(dem_25lda), 25)
print(d_25_perpl)
```

```
## [1] 1680.436   25.000
```

```
dem_25top_terms <- get_top_terms(dem_25topics)
plot_by_topics(dem_25top_terms, "Democratic Party's", "25")
```



Based on eyeball inspection of the topic distribution, I do not think that when $k = 25$ topics add more valuable information as we can see multiple redundant topics.

Question 9 — Perplexity

This is where the behavior of my data and analysis turns strange and (possibly) problematic. In class and elsewhere, it seems that the perplexity scores are expected to decrease rather than increase as the number of topics (k) increases. In my first implementation where I built a corpus with each row corresponding to each line in the platform, I obtained the expected behavior of the perplexity score.

However, after I changed my implementation to building 1 corpus for each party with 1 document which is the platform, my document-term matrix is in the form of 1 row by the number of terms, hence it has exactly 1 row with 0% sparsity. This is the same for both parties. As a result, I obtained such strange results since my perplexity score is increasing rather than decreasing. After revisiting the LDA model which looks at the co-occurrences of words between documents, my second implementation does not make much sense since there is technically 1 document in each document-term matrix. This thread also points out the same issue. When I reread the question which asks to fit a topic model for each of the major parties (totaling 2 models), I am unclear if this means that the document-term matrix should include exactly 2 documents, corresponding to each row. However, I would have 1 document-term matrix with different parties (which will not give me 2 topic models for each party). So, this might be a potential way to rectify and considered a more correct way to implement the analysis.

Another reason I can come up with is that $k = 5$ is really the best for the model. However, this conflicts with the eyeball inspection that I discussed in the previous section.

Below, I present my perplexity score findings.

Perplexity score over a range of number of topics for the Republican Party's platform

```
perplx_rep <- t(data.frame(r_5_perpl, r_10_perpl, r_25_perpl))
colnames(perplx_rep) <-  c("score", "k")
perplx_rep <- as.data.frame(perplx_rep)
```

Perplexity score over a range of number of topics for the Democratic Party's platform

```
perplx_dem <- t(data.frame(d_5_perpl, d_10_perpl, d_25_perpl))
colnames(perplx_dem) <-  c("score", "k")
perplx_dem <- as.data.frame(perplx_dem)
```

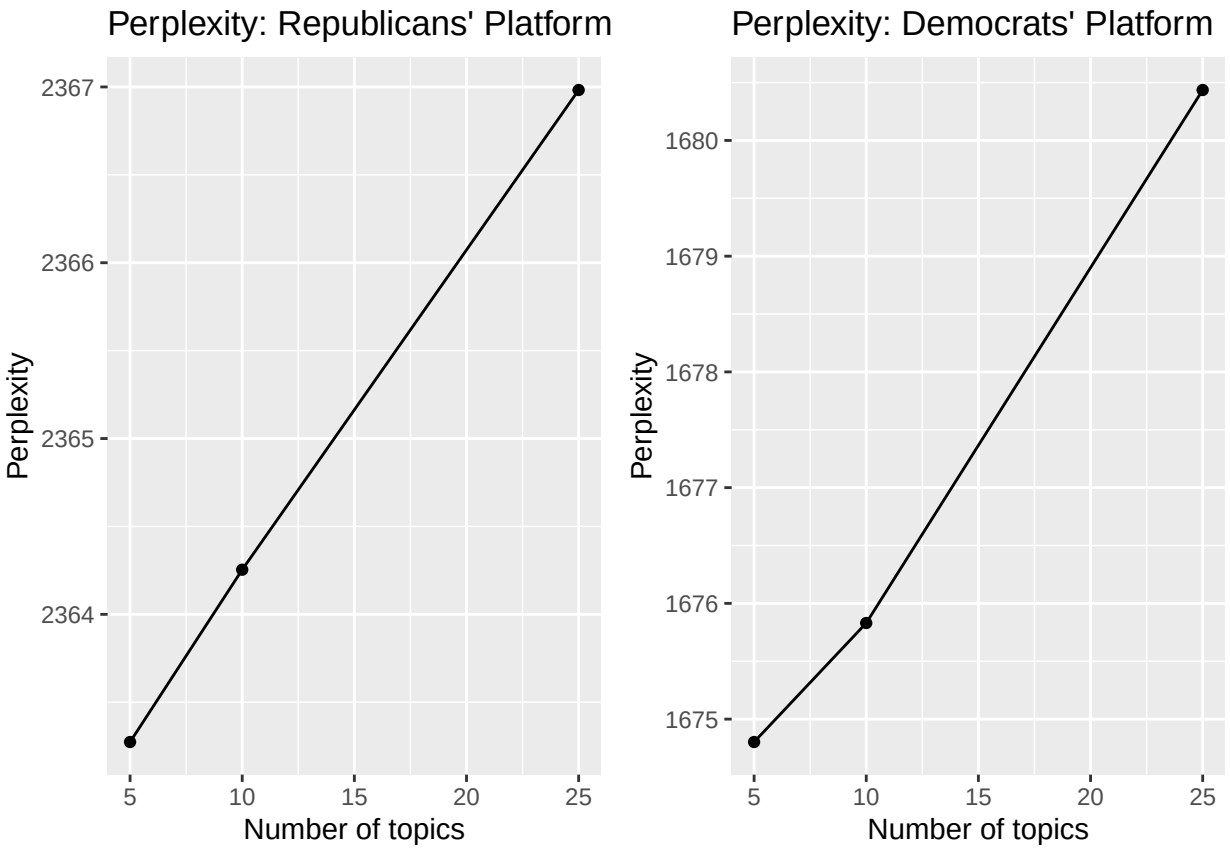
```
print(perplx_dem)
```

```
##               score  k
## d_5_perpl  1674.802  5
## d_10_perpl 1675.830 10
## d_25_perpl 1680.436 25
```

```
print(perplx_rep)
```

```
##               score  k
## r_5_perpl  2363.274  5
## r_10_perpl 2364.254 10
## r_25_perpl 2366.983 25
```

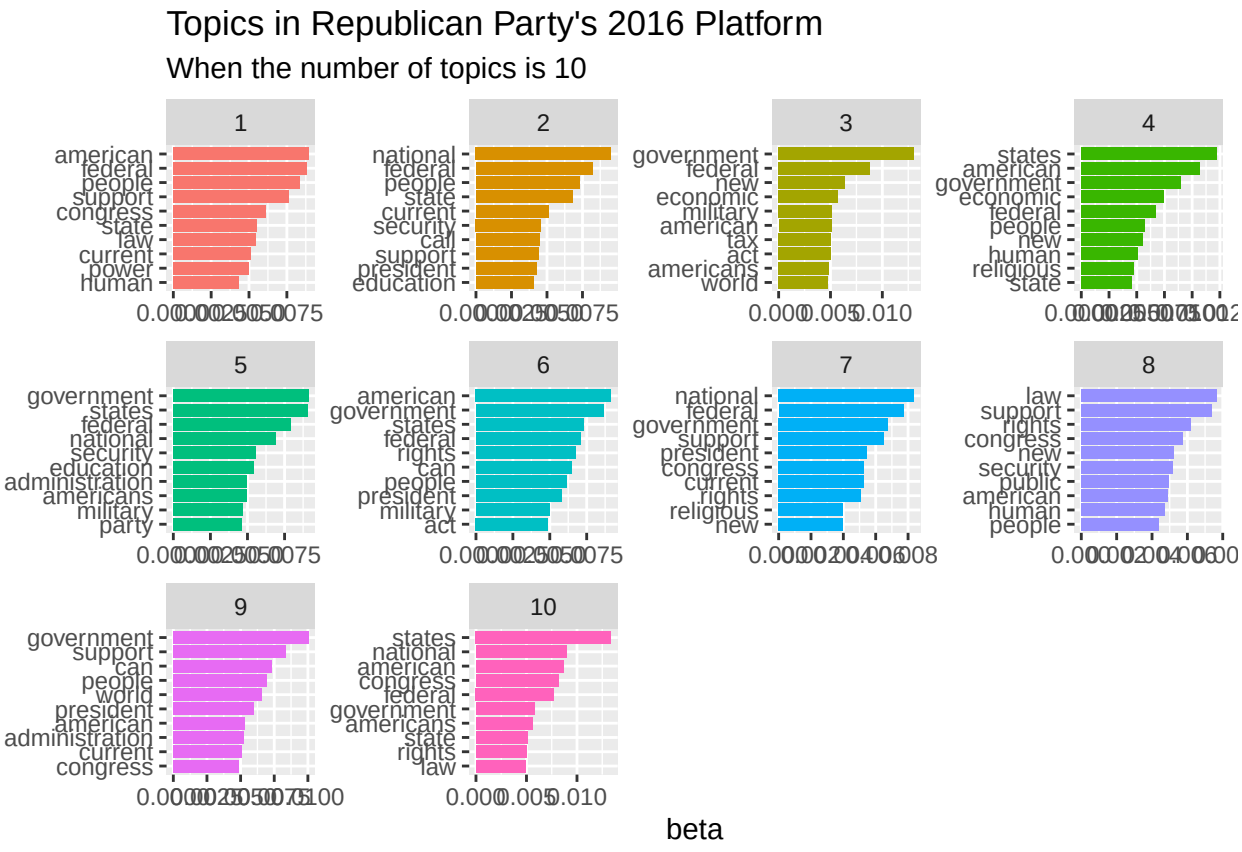
```
p3 <- ggplot(perplx_rep, aes(x = k, y = score)) +
        geom_point() +
        geom_line() +
        labs(title = "Perplexity: Republicans' Platform",
            x = "Number of topics",
            y = "Perplexity")

p4 <- ggplot(perplx_dem, aes(x = k, y = score)) +
        geom_point() +
        geom_line() +
        labs(title = "Perplexity: Democrats' Platform",
            x = "Number of topics",
            y = "Perplexity")

grid.arrange(p3, p4, nrow = 1)
```



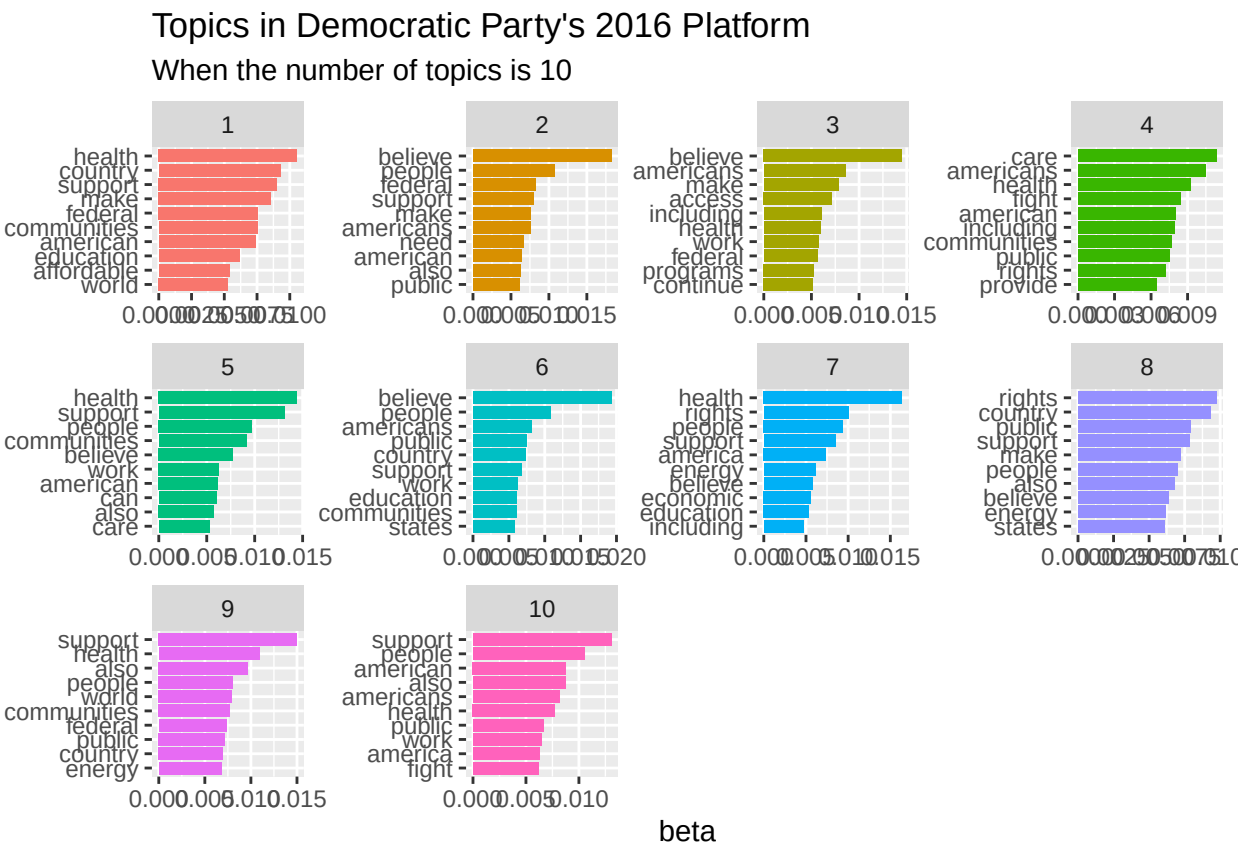
Question 10 — Comparison when $k = 10$ topics

I plotted the 10 topics emerging from each party's platform again below.

```
plot_by_topics(rep_10top_terms, "Republican Party's", "10")
```




```
plot_by_topics(dem_10top_terms, "Democratic Party's", "10")
```



Are there similar themes between the parties? Do you think that $k = 10$ likely picks up the differences more efficiently?

The complexity score that I computed produces a conflicting result against the eyeball inspection. In other words, the complexity score says that 5 topics better capture the underlying issues than 10 topics. However, the inspection of these topics says otherwise. I have already discussed potential issues with the scores and possible ways to rectify it in the last question. My answer is $k = 10$ picks up the topics more efficiently as we see clearer partition among the topic distributions.

Question 11 — Conclusion

Based on the general tones, sentiments and policy priorities, I would support the Democratic Party in the 2020 election. This is because I personally prefer a political party that prioritizes a wide range of policy issues other than the economy and national security. Issues such as climate change, healthcare, and minority rights are as important as traditional issues like those that the Republican Party focused in 2016. This does not necessarily mean that I fully agree with the Democratic Party's solutions to these issues which points to one of the main limitations of current techniques for mass text analysis. This is because although they give you an overview of topics in the documents, it is the voters who still have to dive into the different approaches that these parties take to tackle issues like healthcare and taxes.

```
# References
# https://rstudio-pubs-static.s3.amazonaws.com/163802_0f005a14bcfb4c4b8ee17ac8a8e6c3e9.html
# https://cfss.uchicago.edu/slides/text-analysis-fundamentals-and-sentiment-analysis/#24
# https://cfss.uchicago.edu/talk/
# https://www.datacamp.com/community/tutorials/sentiment-analysis-R
# https://cran.r-project.org/web/packages/tm/vignettes/tm.pdf
# https://eight2late.wordpress.com/2015/05/27/a-gentle-introduction-to-text-mining-using-r/
```