PS-5

Lilly

11/27/2019

##1. Load the platforms.csv file containing the 2016 Democratic and Republican party platforms. Note the 2X2 format, where each row is a document, with the party recorded as a separate feature. Also, load the individual party .txt files as a corpus.

```
platforms <- read.csv("platforms.csv", stringsAsFactors = FALSE)
```

```
## Warning in read.table(file = file, header = header, sep = sep, quote =
## quote, : incomplete final line found by readTableHeader on 'platforms.csv'
```

```
names(platforms)[1] <- "doc_id"
names(platforms)[2] <- "text"

platforms[2]
```

```
##
## 1
```

```
platformsVC <- VCorpus(DataframeSource(platforms))

summary(platformsVC)
```

```
##            Length Class             Mode
## democrat   2      PlainTextDocument list
## republican 2      PlainTextDocument list
```

```
docs <- tm_map(platformsVC, removePunctuation)
```

##2. Create a document-term matrix and preprocess the platforms by the following criteria (at a minimum):

Convert to lowercase Remove the stopwords Remove the numbers Remove all punctuation Remove the whitespace

```
for (j in seq(docs)) {
  docs[[j]] <- gsub("/", " ", docs[[j]])
  docs[[j]] <- gsub("â", " ", docs[[j]])
  docs[[j]] <- gsub("â", " ", docs[[j]])
  docs[[j]] <- gsub("\\|", " ", docs[[j]])
  docs[[j]] <- gsub("@", " ", docs[[j]])
  docs[[j]] <- gsub("\u2028", " ", docs[[j]])  # an ascii character that does not translate
  docs[[j]] <- gsub("\n", " ", docs[[j]])
  docs[[j]] <- gsub("â\200", " ", docs[[j]])
}
```

```r
docs <- tm_map(docs, removeNumbers)

# For consistency, we may also want to remove captialization
docs <- tm_map(docs, tolower)
docs <- tm_map(docs, PlainTextDocument)

# Next, remove superfluous words like articles or words with no
#substantive value for analysis
## For a list of the stopwords, run: stopwords("english")
#  And you can use this to see how many there are (174):
#length(stopwords("english"))
docs <- tm_map(docs,
               removeWords,
               stopwords("english"))
docs <- tm_map(docs, PlainTextDocument) # redefine

# manually removing words too for your specific purposes
#(e.g., our "representative" or "honourable" example Tuesday)
docs <- tm_map(docs, removeWords, c("will")) # trump says "will" a
#ton (try leaving "will" in; most frequently used)

# There are some words that tm pulls apart that should stay together;
#manually define for each document, j:
for (j in seq(docs)) {
  docs[[j]] <- gsub("fake news", "fake_news", docs[[j]])
  docs[[j]] <- gsub("inner city", "inner-city", docs[[j]])
  docs[[j]] <- gsub("I m", "I'm", docs[[j]])
  docs[[j]] <- gsub("politically correct", "politically_correct", docs[[j]])
  docs[[j]] <- gsub("Great Recession", "Great_Recession", docs[[j]])
}

docs <- tm_map(docs, PlainTextDocument) # redefine docs

#install.packages("SnowballC")
# We can also omit certain "stems" or common English word endings (e.g., ing, es)
docs <- tm_map(docs, stemDocument)
# note that we are storing this in a new corpus to give ourselves some
#options for analysis later
docs <- tm_map(docs, PlainTextDocument)

# Preprocessing leaves behind a lot of white space, or extra spaces between
#words or lines
docs <- tm_map(docs, stripWhitespace)
docs <- tm_map(docs, PlainTextDocument) # final redefine for retaining the
#latest preprocessing steps

#writeLines(as.character(docs[1]))
```

```r
dtm_dem <- DocumentTermMatrix(docs[1])
dtm_rep <- DocumentTermMatrix(docs[2])

tdm_dem <- TermDocumentMatrix(docs[1])
tdm_rep <- TermDocumentMatrix(docs[2])
```

```
# EXPLORE numerically
# with preprocessed and staged text, we can now explore trump's speeches

frequency_dem <- sort(colSums(as.matrix(dtm_dem)),
                  decreasing=TRUE) # add number of times each term is used,
#and sorting based on frequency of usage
frequency_rep <- sort(colSums(as.matrix(dtm_rep)),
                  decreasing=TRUE) # add number of times each term is used,
#and sorting based on frequency of usage
head(frequency_dem) # most frequently used words
```

```
## democrat american   worker   believ  support     work
##       52       44       37       32       30       30
```

```
head(frequency_rep) # most frequently used words
```

```
## american   govern    feder   nation  america     busi
##       40       30       27       25       24       22
```

```
# what is the most common term used?
#frequency_dem[1]
```

##3. Visually inspect your cleaned documents by creating a wordcloud for each major party's platform. Based on this naive visualization, offer a few-sentence-description of general patterns you see (e.g., What are commonly used words? What are less commonly used words? Can you get a sense of differences between the parties at this early stage?

```
# same thing, another way - verify that there are 14 words used over 100 times
wf_dem <- data.frame(word = names(frequency_dem),
                  freq = frequency_dem)
wf_rep <- data.frame(word = names(frequency_rep),
                  freq = frequency_rep)

# and of course, some color is good
# using brewer palette, we can first choose the color scheme we like best,
#and then specify it below in our word cloud
#display.brewer.all(n = NULL, type = "all", select = NULL, exact.n = TRUE,
#                   colorblindFriendly = FALSE)


DEM <- wf_dem %>%
  dplyr::select(c(2))
DEM <- as.matrix(DEM)

wordcloud(rownames(DEM), DEM, min.freq =10,
          random.order = FALSE, random.color = FALSE,
          colors= c("navy", "royalblue2", "midnightblue"))
```


```
GOP <- wf_rep %>%
  dplyr::select(c(2))
GOP <- as.matrix(GOP)

wordcloud(rownames(GOP), GOP, min.freq =10, scale=c(2.5, .3),
          random.order = FALSE, random.color = FALSE,
          colors= c("red4", "darkred", "firebrick4"))
```


```
#wordcloud(wf_rep)

#The commonly used terms by both parties are "American", "people", "create"
#and "America"; the terms used by Democrats that are less commonly used by
#Republicans include "Democrat", "worker", "support", "work" and "family."
#On the other hand, Republicans use "government", "nation", "tax", "economical"
#and "market" more frequently. Therfore, we may contrast the two parties more
#easily.For example, the Democrats focus on supporting workers versus the
#Republicans focusing more on lower taxes and economic well-being.
```

##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).

```
platforms_sentiment_afinn <- platforms %>%
    unnest_tokens(word, text) %>%
    inner_join(get_sentiments("afinn"))
```

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

```
platforms_sentiment_bing <- platforms %>%
    unnest_tokens(word, text) %>%
    inner_join(get_sentiments("bing"))
```

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

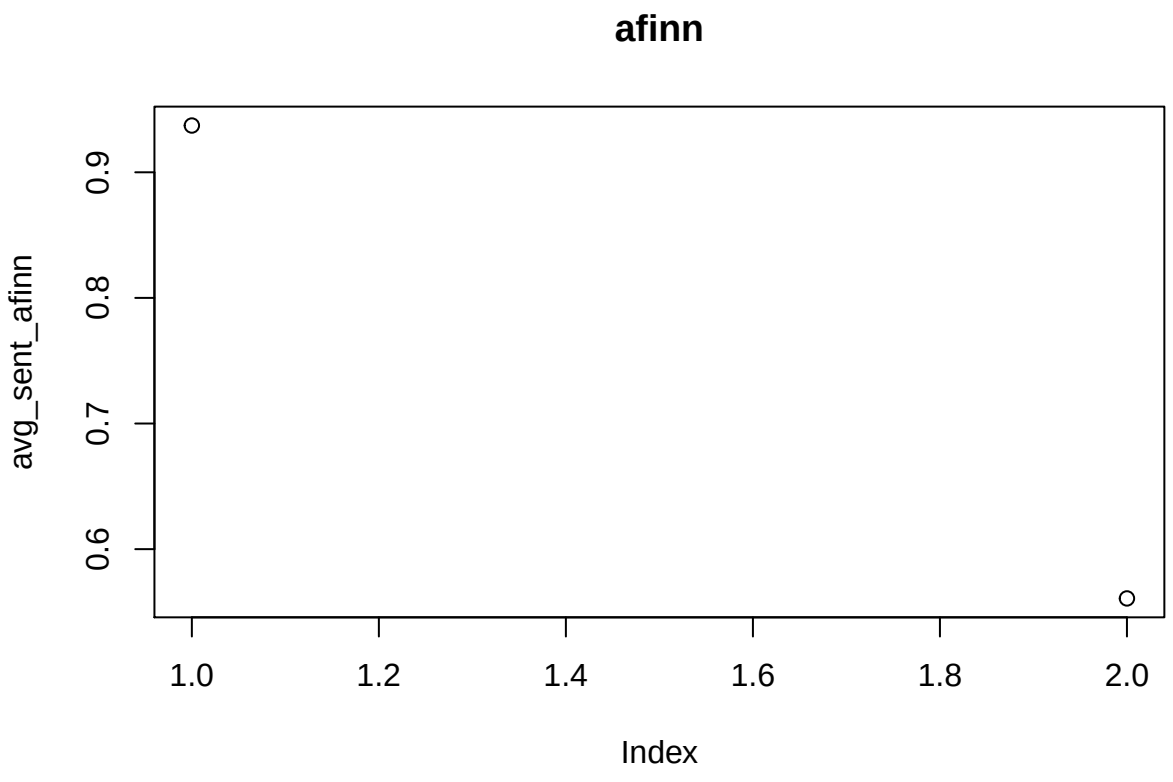
```r
platforms_sentiment_bing$sentiment[platforms_sentiment_bing$sentiment=="positive"] <- 1
platforms_sentiment_bing$sentiment[platforms_sentiment_bing$sentiment=="negative"] <- 0
platforms_sentiment_bing <- transform(platforms_sentiment_bing, sentiment = as.numeric(sentiment))
#platforms_sentiment_bing$sentiment
#platforms_sentiment_afinn$value

avg_sent_bing <- sapply(split(platforms_sentiment_bing$sentiment, platforms_sentiment_bing$doc_id), mea

avg_sent_afinn <- sapply(split(platforms_sentiment_afinn$value, platforms_sentiment_afinn$doc_id), mean

avg_sent_bing
```

```
##   democrat republican
##  0.7149644  0.5984456
```

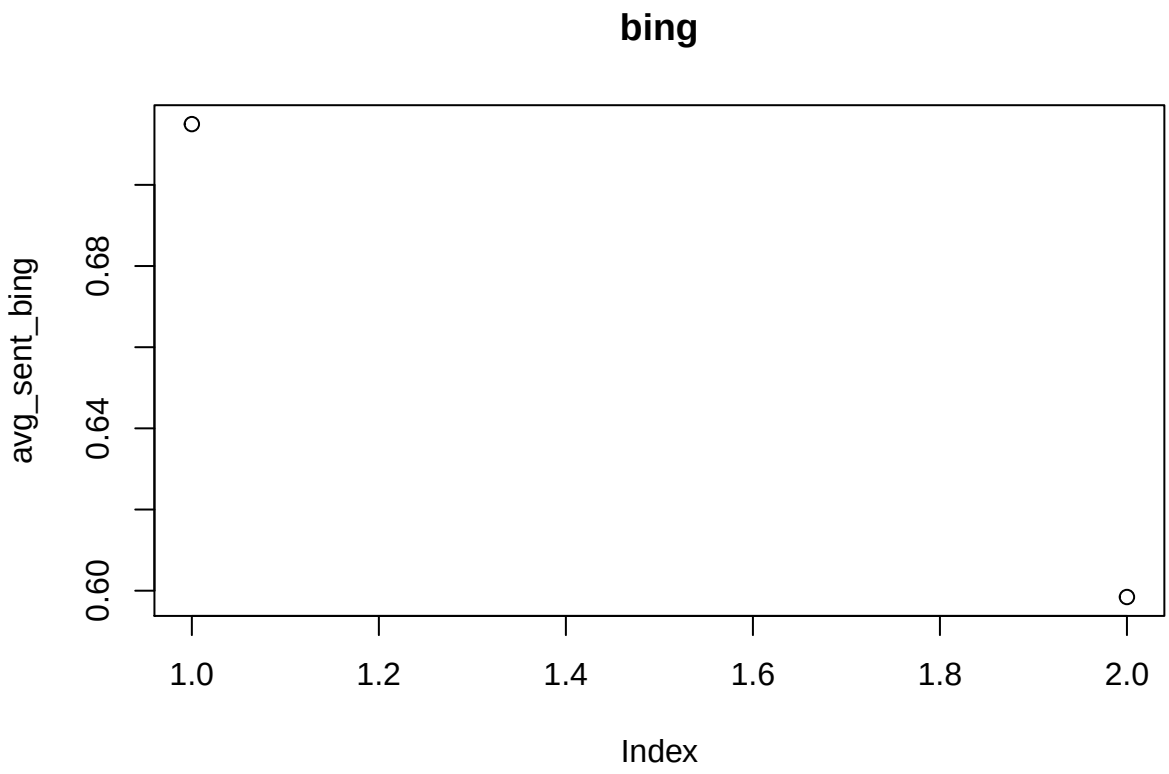
```r
avg_sent_afinn
```

```
##   democrat republican
##  0.9372197  0.5607735
```

```r
plot(avg_sent_afinn, main="afinn")
```



```
plot(avg_sent_bing, main="bing")
```



##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?

```
# The libraries visually display similar results for the sentiment analysis
#and both suggest that Democratic speech has more positive sentiments in
#comparison to the Republicans. We may also notice that both speeches are
#generally positive; therefore, this comports with my perceptions of both
#parties.
```

##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).

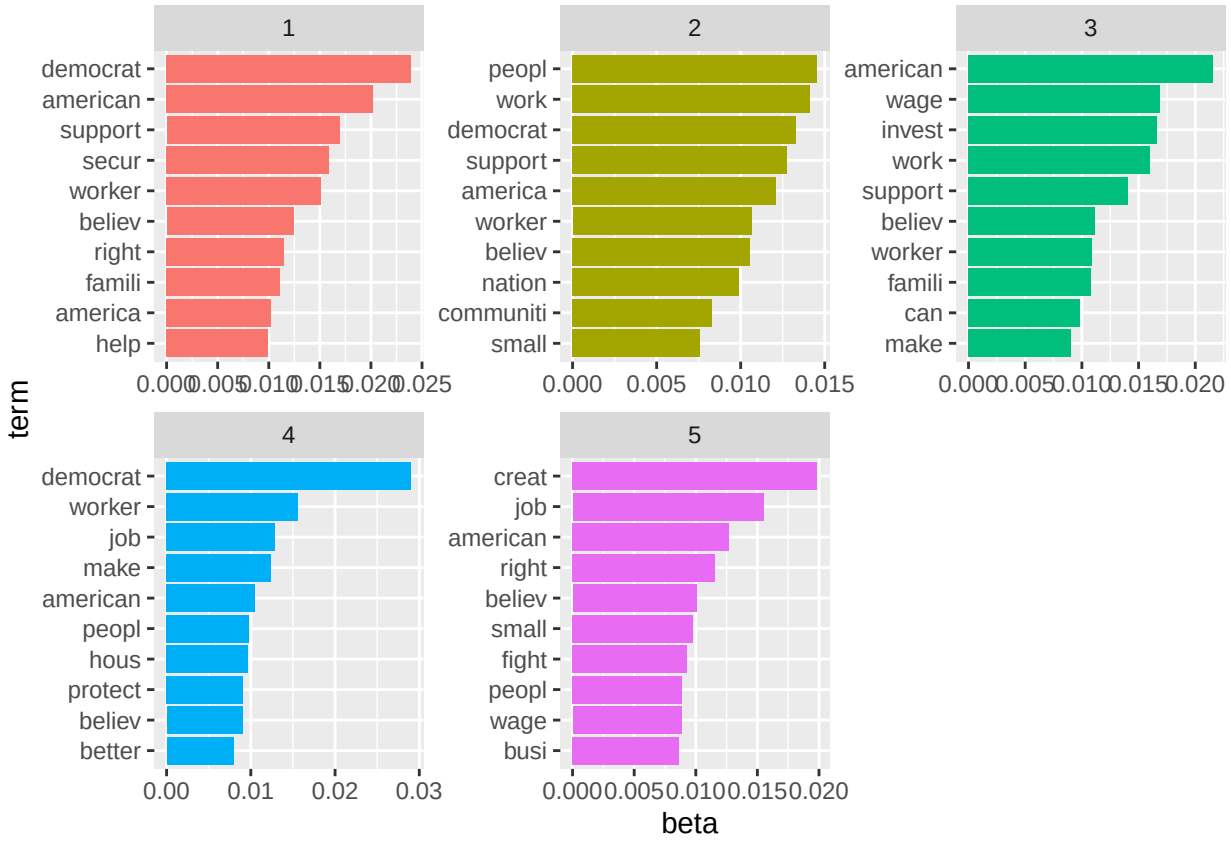
```
lda_dem <- LDA(dtm_dem, k = 5, control = list(seed = 1234))
lda_rep <- LDA(dtm_rep, k = 5, control = list(seed = 1234))

topics_dem <- tidy(lda_dem, matrix = "beta")
topics_rep <- tidy(lda_rep, matrix = "beta")

topics_dem
```

```
## # A tibble: 5,310 x 3
##    topic term       beta
##    <int> <chr>     <dbl>
##  1     1 aapi  0.000473
##  2     2 aapi  0.000232
##  3     3 aapi  0.000615
##  4     4 aapi  0.000110
##  5     5 aapi  0.000127
##  6     1 abil  0.000581
##  7     2 abil  0.000712
##  8     3 abil  0.00106
##  9     4 abil  0.0000164
## 10     5 abil  0.00125
## # ... with 5,300 more rows
```

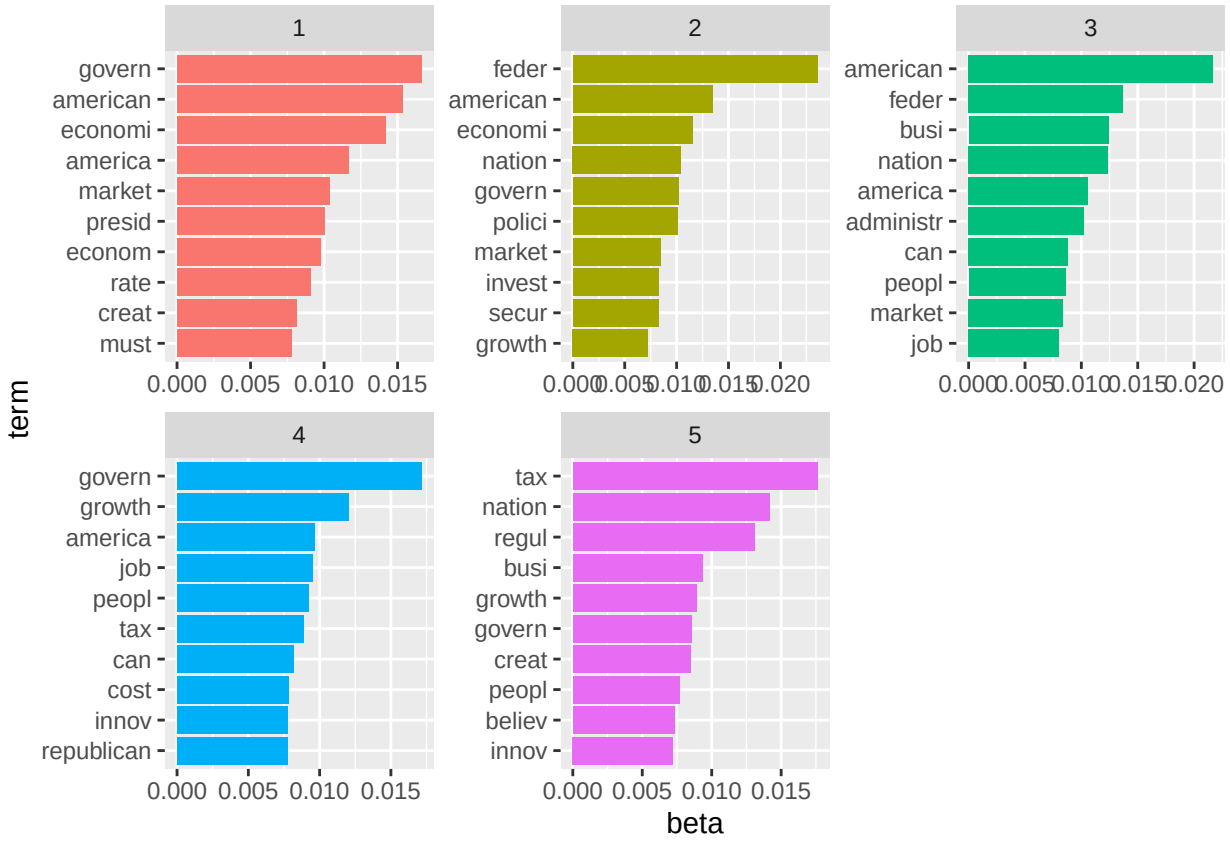
```
topics_rep
```

```
## # A tibble: 5,850 x 3
##    topic term         beta
##    <int> <chr>       <dbl>
##  1     1 abandon 0.000287
##  2     2 abandon 0.000559
##  3     3 abandon 0.000454
##  4     4 abandon 0.000201
##  5     5 abandon 0.000192
##  6     1 abil    0.00120
##  7     2 abil    0.000926
##  8     3 abil    0.00138
##  9     4 abil    0.0000890
## 10     5 abil    0.00148
## # ... with 5,840 more rows
```

```
dem_top_terms <- topics_dem %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

dem_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()
```




```
rep_top_terms <- topics_rep %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

rep_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()
```



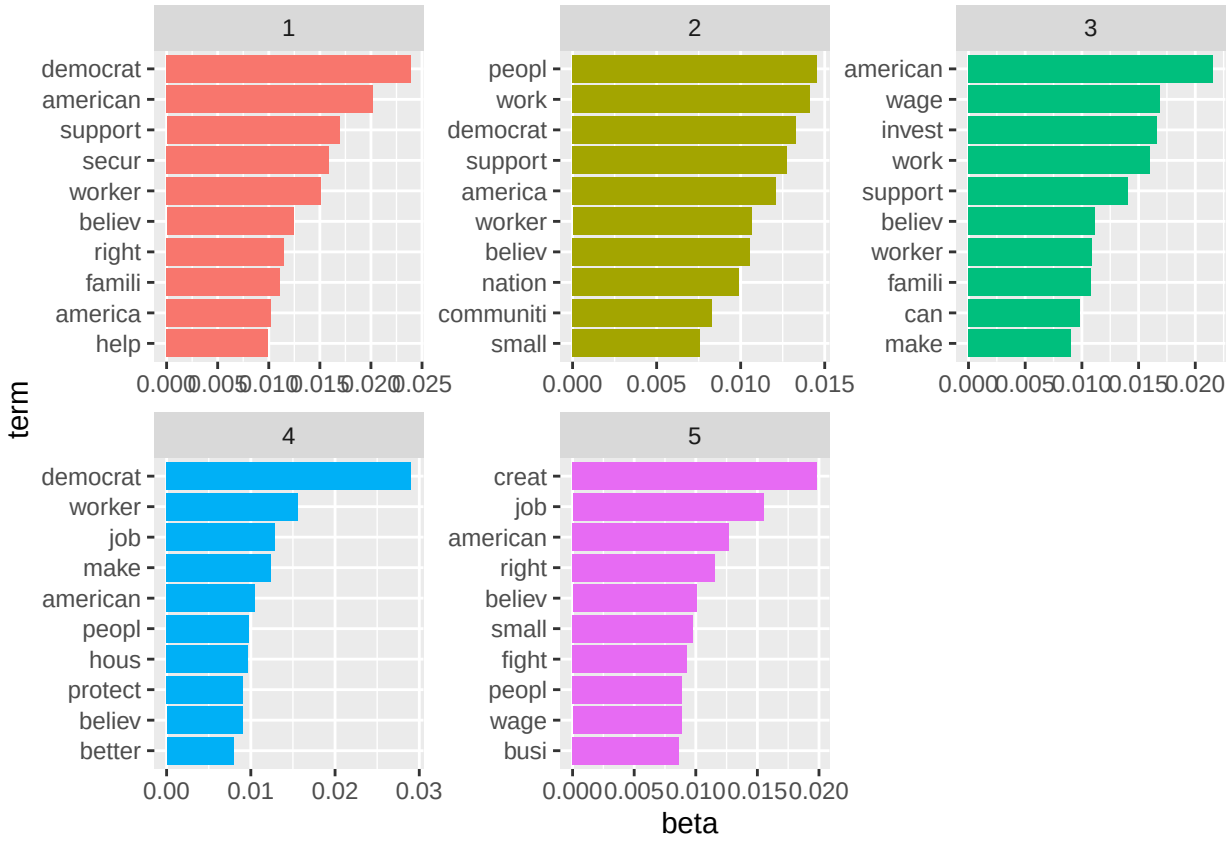
##7. Describe the general trends in topics that emerge from this stage. Are the parties focusing on similar or different topics, generally?

```
#Democrats
#topic 1: labor
#topic 2: Works Rights
#topic 3: Future Outlook
#topic 4: Economical Growth
#topic 5: Nation Building
# Overall, the Democrats emphasize Worker's rights and building a stable
#economy and strong communities. They also clearly have a positive view of
#the future

#Republicans
#topic 1: Federal Government
#topic 2: Economic Growth and Jobs
#topic 3: Housing
#topic 4: Taxes
#topic 5: Private Sector
# The republicans emphasize the importance of the federal government,
#economic growth and low taxes. Also, there are much less positive terms or
#verbs pertaining to the future when compared with the dems.

# Generally, the priorities revealed by the 5 topics appear to be different.
```

##8. Fit 6 more topic models at the follow levels of k for each party: 5, 10, 25. Present the results however

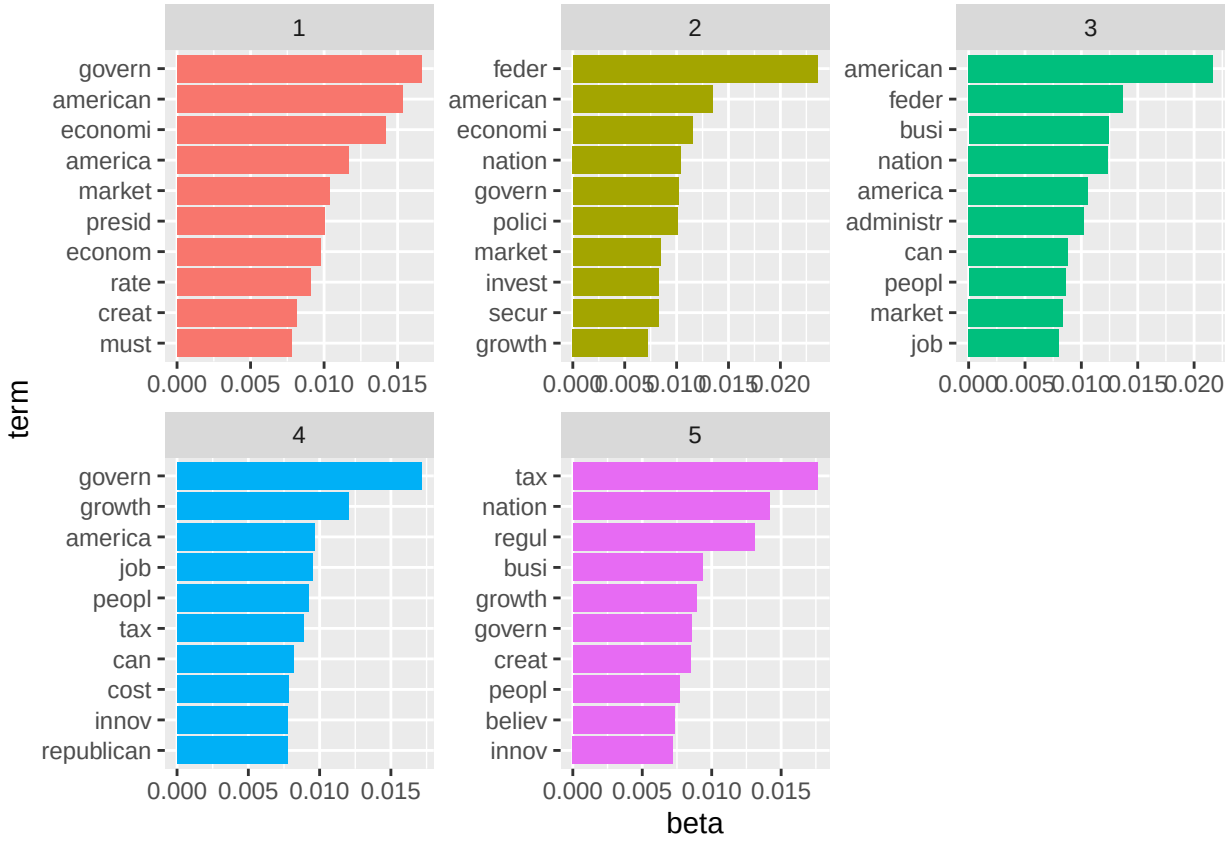
you'd like (e.g., visually and/or numerically).

```
lda_dem_5 <- LDA(dtm_dem, k = 5, control = list(seed = 1234))
lda_rep_5 <- LDA(dtm_rep, k = 5, control = list(seed = 1234))

topics_dem_5 <- tidy(lda_dem_5, matrix = "beta")
topics_rep_5 <- tidy(lda_rep_5, matrix = "beta")

dem_top_terms_5 <- topics_dem_5 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

dem_top_terms_5 %>%
  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()
```



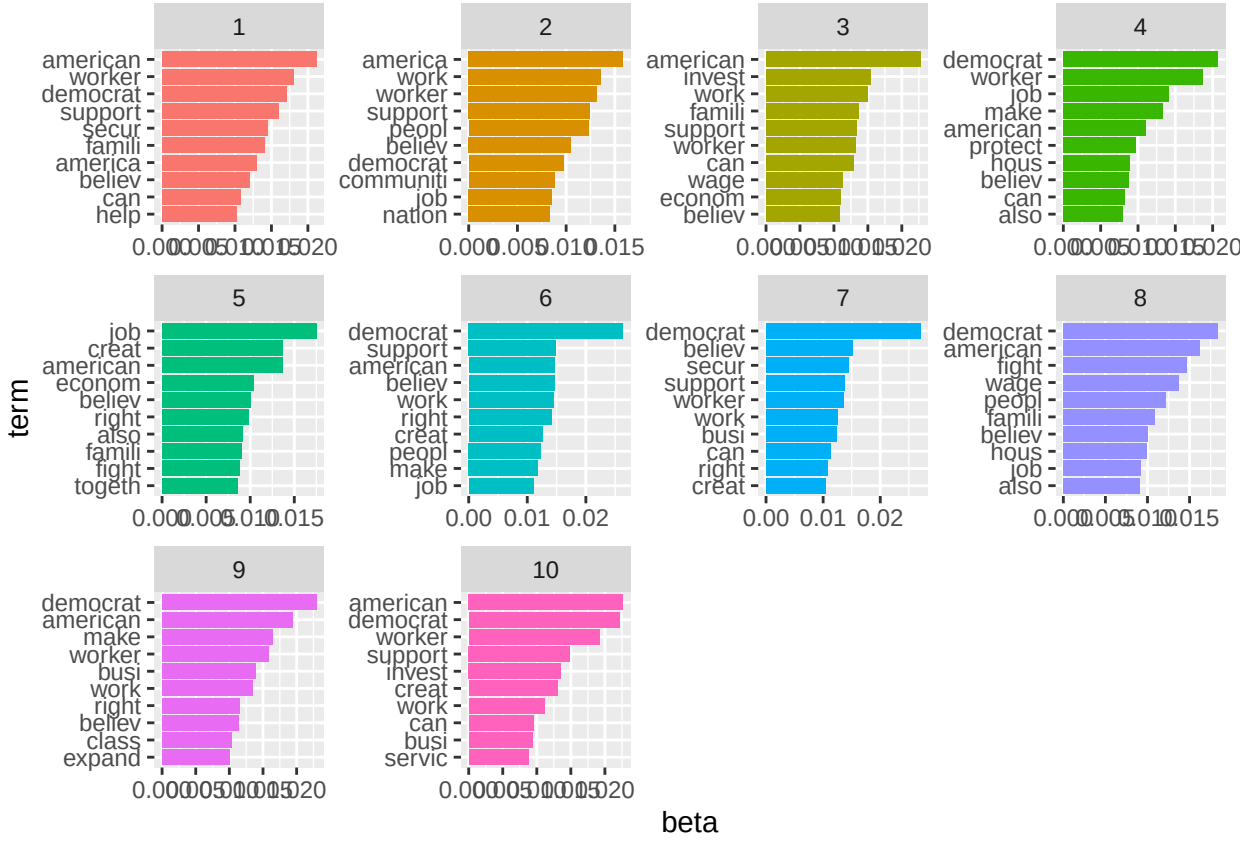

```
rep_top_terms_5 <- topics_rep_5 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
```

```
  ungroup() %>%
  arrange(topic, -beta)

rep_top_terms_5 %>%
  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()
```

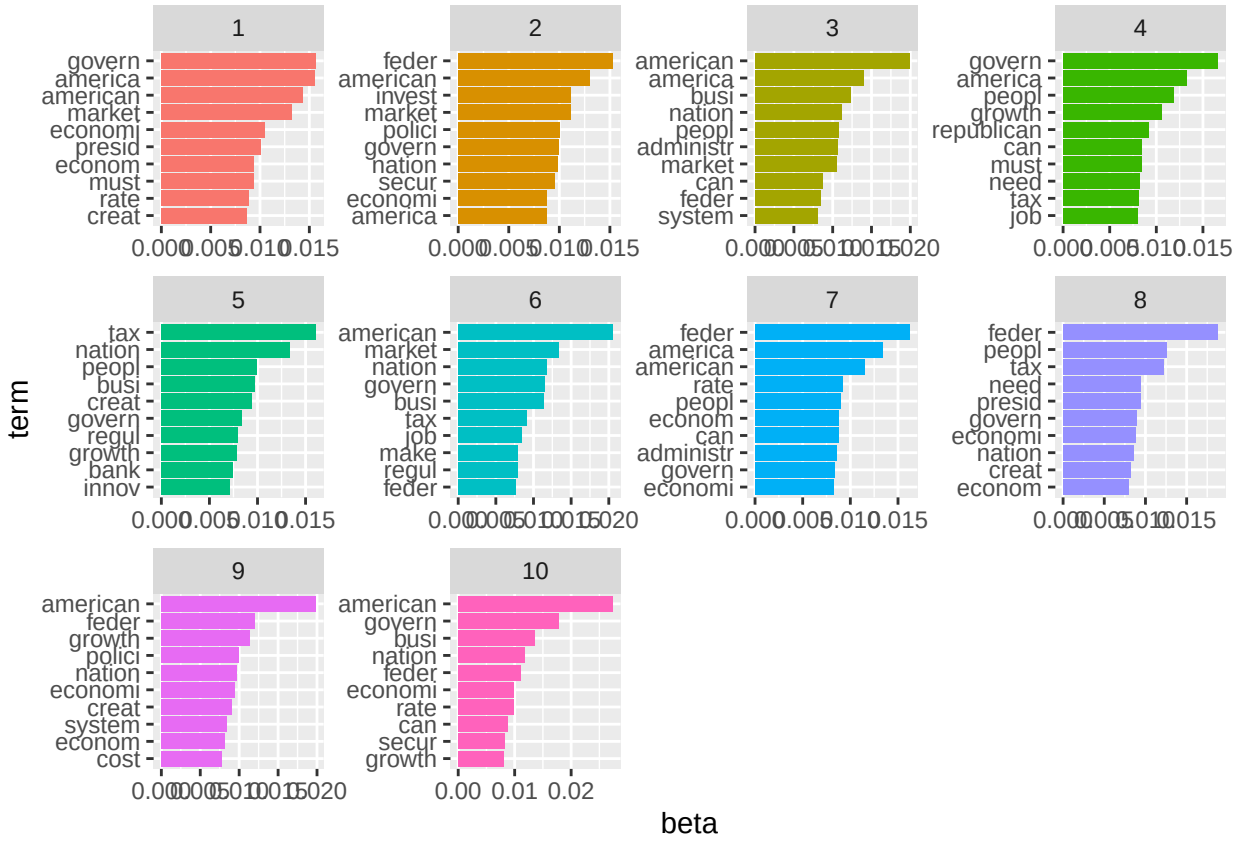



```
lda_dem_10 <- LDA(dtm_dem, k = 10, control = list(seed = 1234))
lda_rep_10 <- LDA(dtm_rep, k = 10, control = list(seed = 1234))

topics_dem_10 <- tidy(lda_dem_10, matrix = "beta")
topics_rep_10 <- tidy(lda_rep_10, matrix = "beta")

dem_top_terms_10 <- topics_dem_10 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

dem_top_terms_10 %>%
  mutate(term = reorder_within(term, beta, topic)) %>%
```

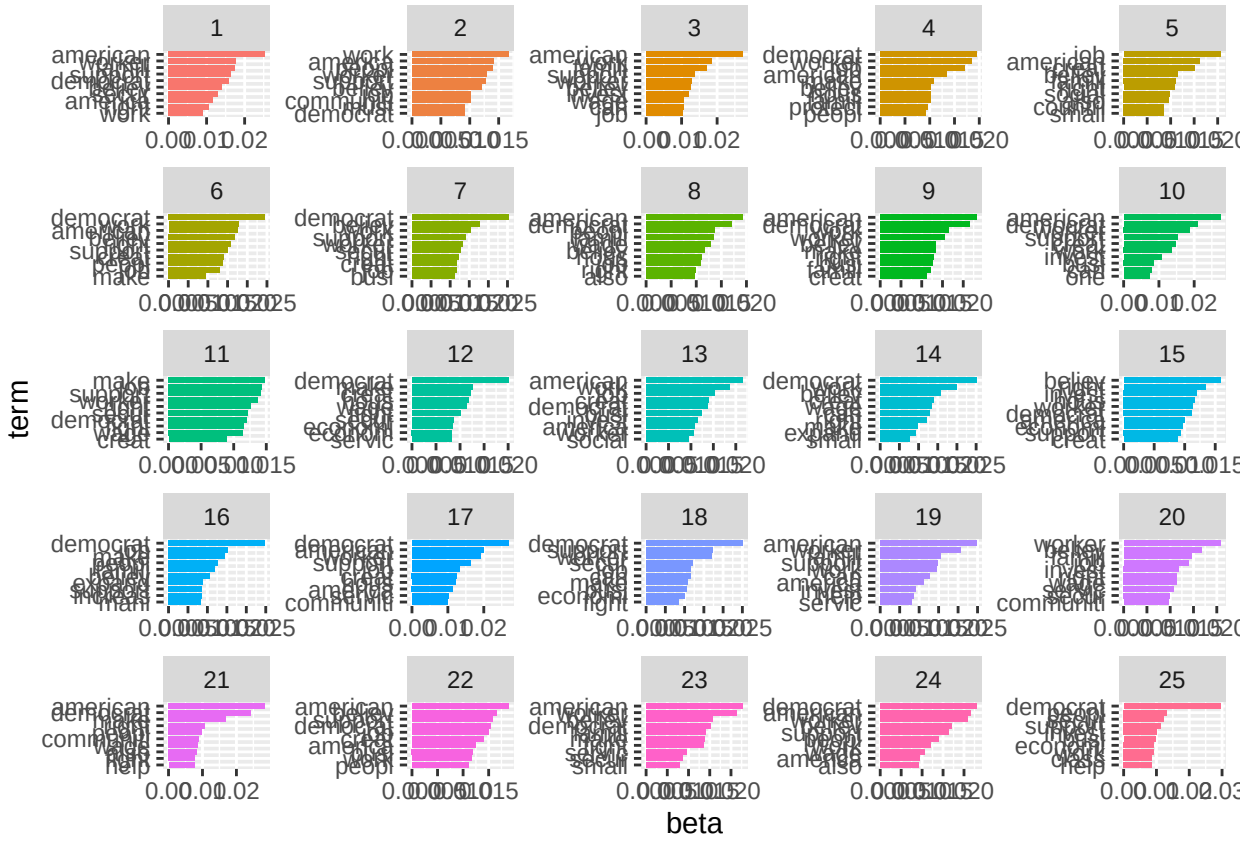
```
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip() +
  scale_x_reordered()
```



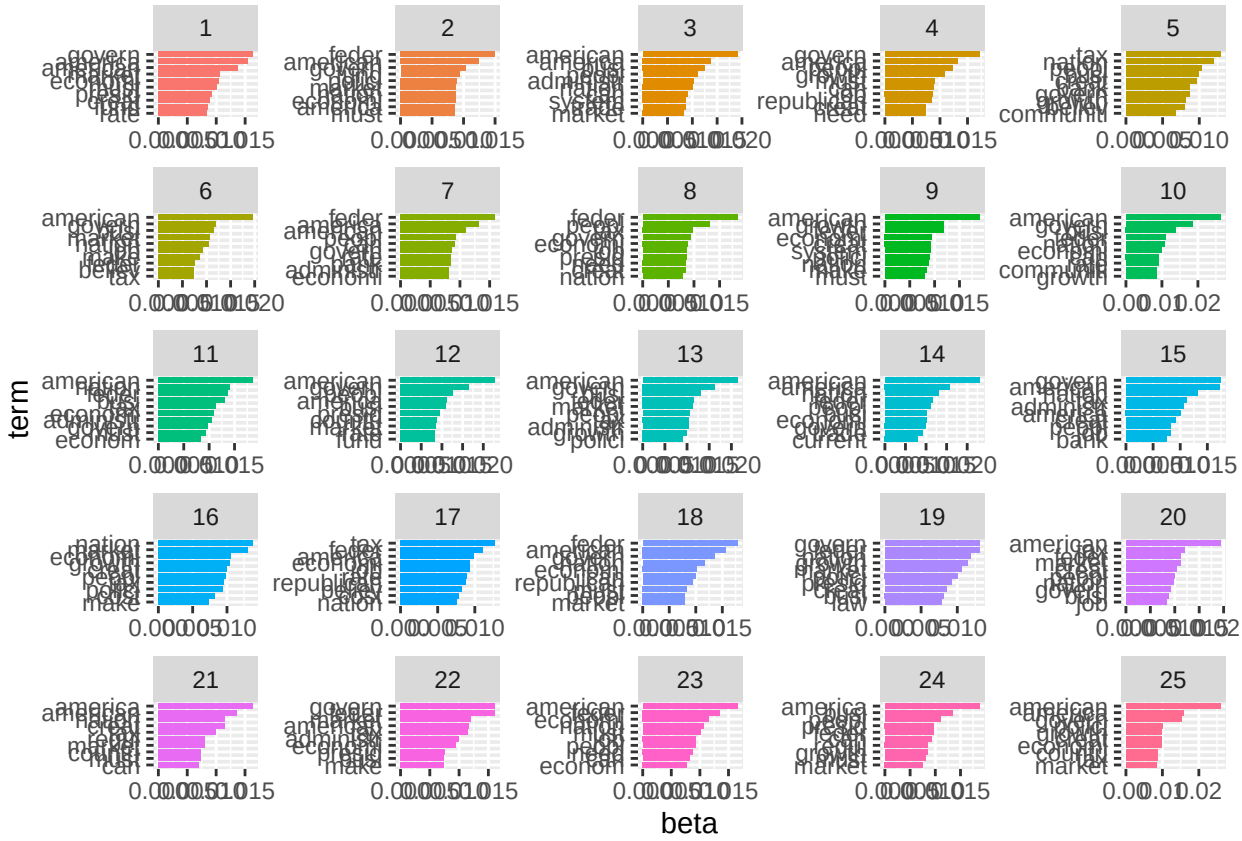

```
rep_top_terms_10 <- topics_rep_10 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

rep_top_terms_10 %>%
  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()
```




```
lda_dem_25 <- LDA(dtm_dem, k = 25, control = list(seed = 1234))
lda_rep_25 <- LDA(dtm_rep, k = 25, control = list(seed = 1234))

topics_dem_25 <- tidy(lda_dem_25, matrix = "beta")
topics_rep_25 <- tidy(lda_rep_25, matrix = "beta")

dem_top_terms_25 <- topics_dem_25 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

dem_top_terms_25 %>%
  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()
```




```
rep_top_terms_25 <- topics_rep_25 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

rep_top_terms_25 %>%
  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()
```



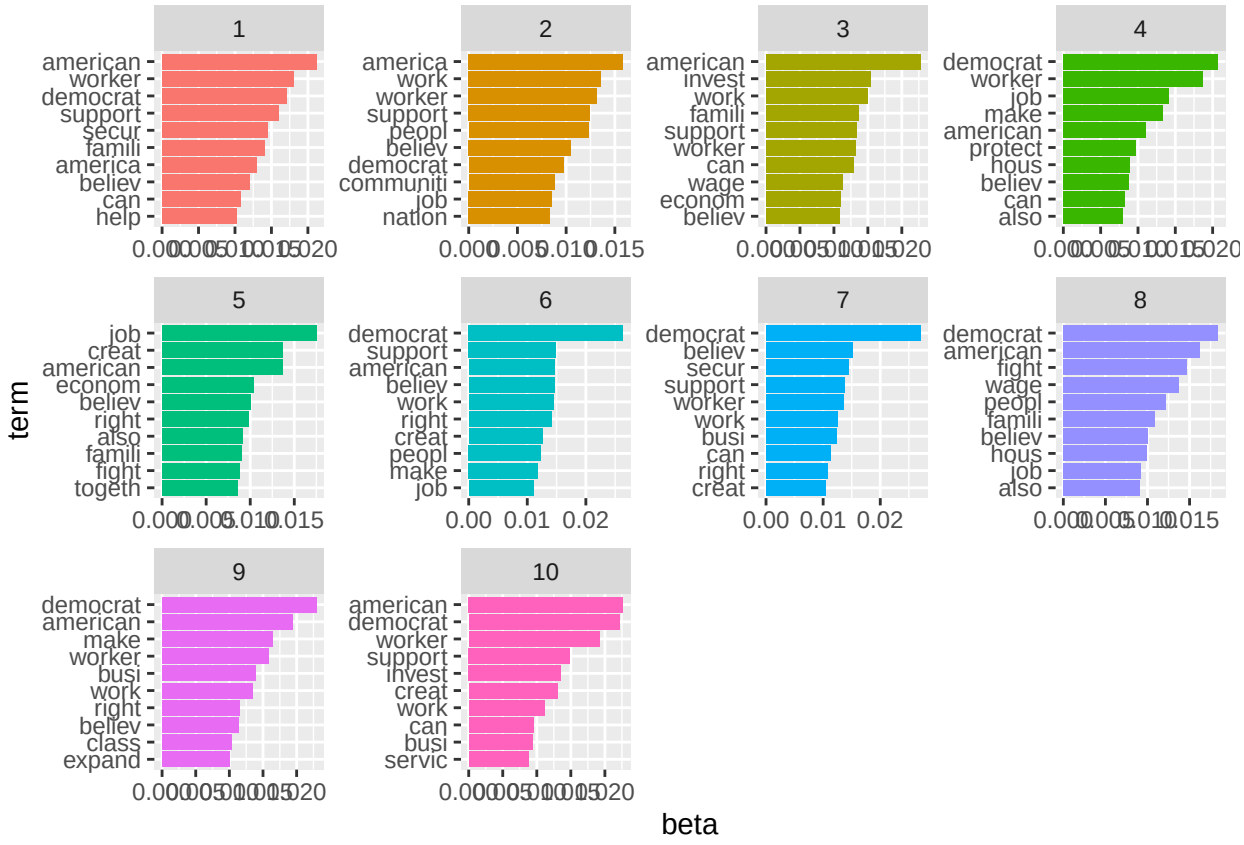
##9. Calculate the perplexity of each model iteration and describe which technically fits best.

```
perplex_dem_5 <- perplexity(lda_dem_5)
perplex_rep_5 <- perplexity(lda_rep_5)
perplex_dem_10 <- perplexity(lda_dem_10)
perplex_rep_10 <- perplexity(lda_rep_10)
perplex_dem_25 <- perplexity(lda_dem_25)
perplex_rep_25 <- perplexity(lda_rep_25)

# The perplexity for the democratic speech with k = 10 topic model was the
#lowest and th erefore the best fit. The perplexity for the republican
#speech with k = 5 topic model was the lowest and therefore the best fit.
```

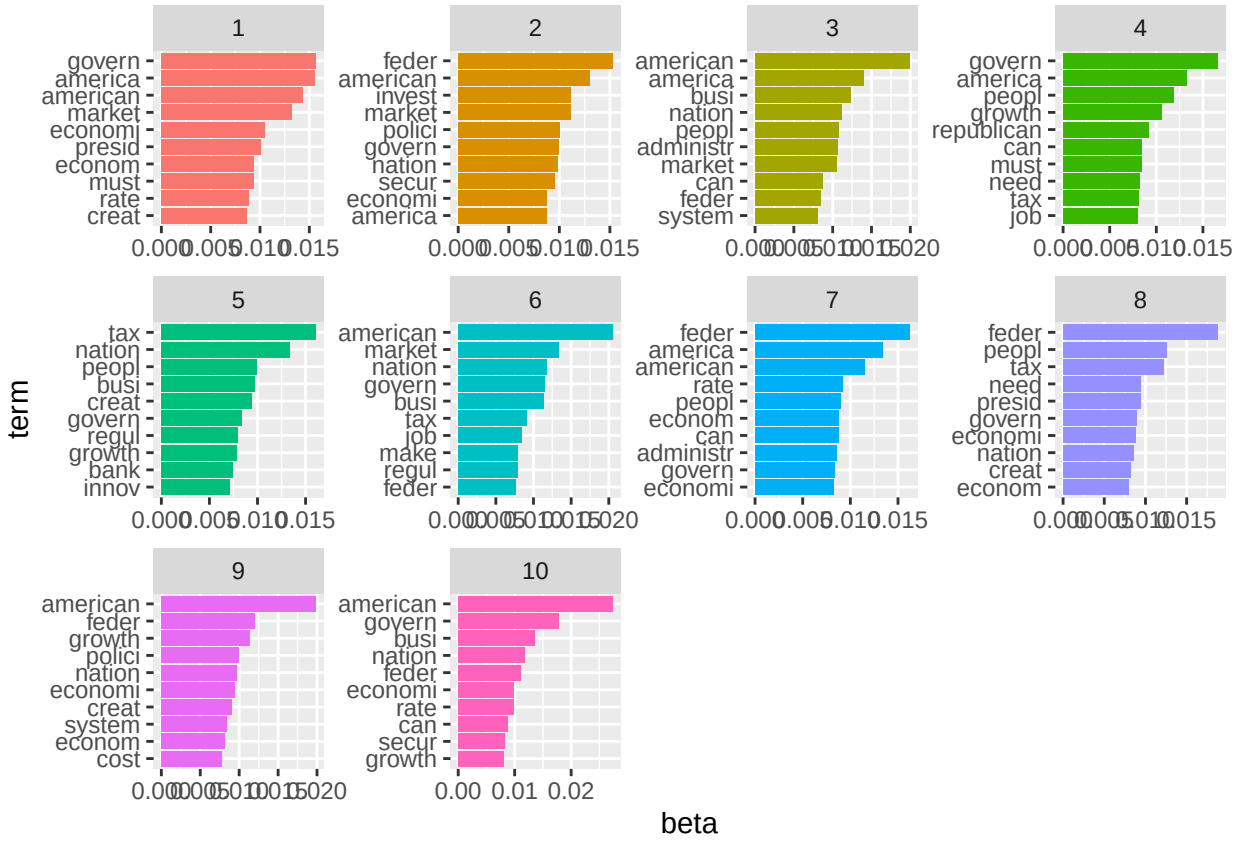
##10. Building on the previous question, display a barplot of the k = 10 model for each party, and offer some general inferences as to the main trends that emerge. Are there similar themes between the parties? Do you think k = 10 likely picks up differences more efficiently? Why or why not?

```
dem_top_terms_10 <- topics_dem_10 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

dem_top_terms_10 %>%
  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()
```




```
rep_top_terms_10 <- topics_rep_10 %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

rep_top_terms_10 %>%
  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 main trends that emerge within the democratic speech are worker rights,
#and positive sentiments such as believe, security and job creation.
#The main trends that emerge within the republican speech are the federal
#government, America and American, taxation, job creation, economic growth and
#stability. Similarly, both political parties include job creation.
#Indeed, the difference is more easily seen with k = 10.
#The k = 10 model allows for better grouping of words under topics and
#highlights the differences due to this extra degree of freedom.
```

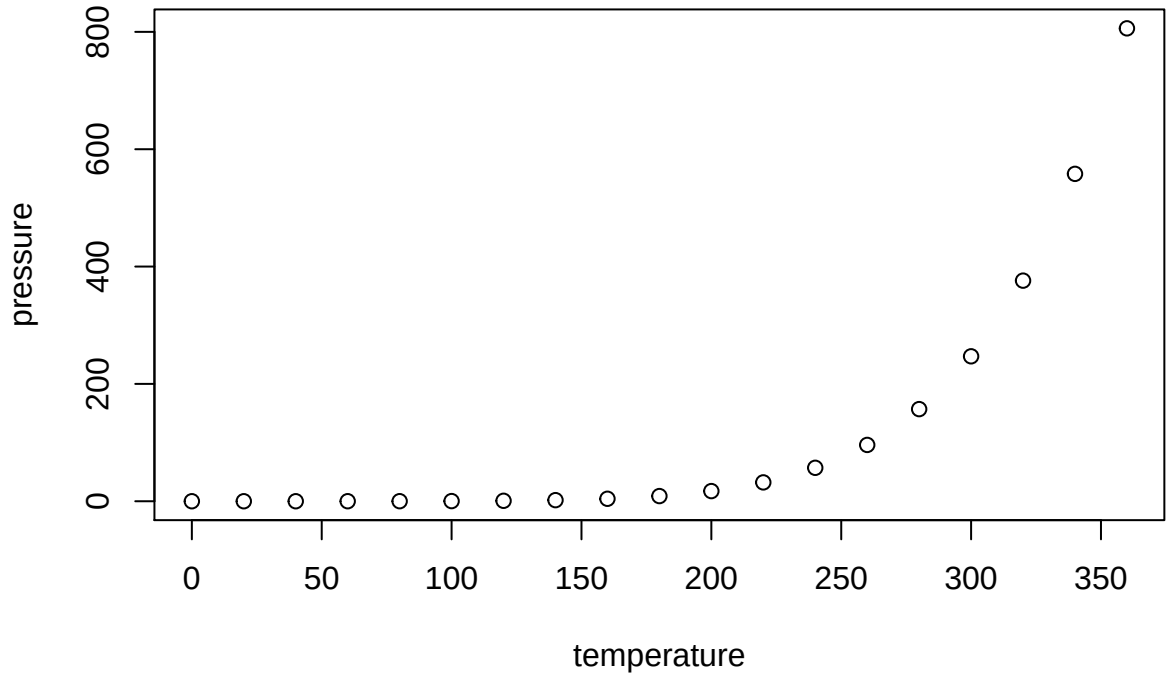
##11. Per the opening question, based on your analyses (including exploring party brands, general tones/sentiments, political outlook, and policy priorities), which party would you support in the 2020 election (again, this is hypothetical)?

```
# Based off the analyses, I would strongly support the Democratic Party
#for the 2020 election since they are more in tune with the Nation's issues.
```

Including Plots

You can also embed plots, for example:

```
plot(pressure)
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.