Assignment 3: Topic Analysis

Colleen McCamy

April 15, 2023

Reading in "Nuclear Energy" Nexis Article Data

```
text_file_path <- here("data/nuclear_news")

my_files <- list.files(pattern = ".docx", path = text_file_path,
                       full.names = TRUE, recursive = TRUE, ignore.case = TRUE)

# setting directory to save
setwd(here("data"))
saveRDS(my_files, "nuclear_news_data.RDS") # will need to do this four our portion

# reading in the data as an RDS
dat <- readRDS("/Users/colleenmccamy/Documents/MEDS/classes/spring/eds-231-text-analysis/text-sentiment-

# changing to a LNT output
dat <- lnt_read(dat)

# saving individual dataframes from the LNT output
meta <- dat@meta
articles_df <- dat@articles
paragraphs_df <- dat@paragraphs

write_csv(articles_df, here("data/nuclear_articles_df.csv"))
write_csv(meta, here("data/nuclear_meta_df.csv"))
```

Create a corpus from the nuclear articles

```
# reading in the data
articles_df <- read_csv("data/nuclear_articles_df.csv")
```

```
## Rows: 100 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Article
## dbl (1): ID
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

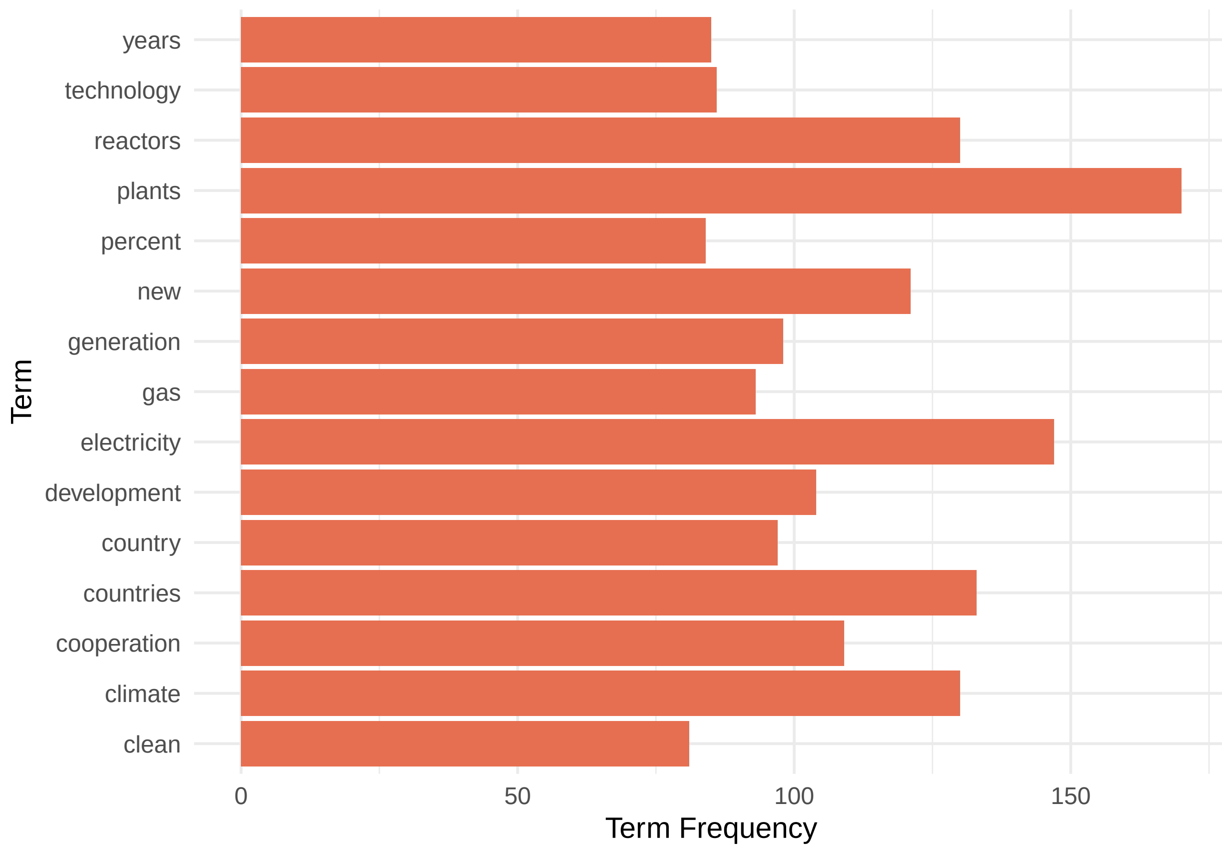
```r
### ------------------------
# cleaning Nexus text data
### ------------------------

# removing the query info from the paranthesis in the front
articles_df <- articles_df |>
  mutate(text_noq = gsub("^\\[[0-9]+\\] |\\([^\\(]*\\)", "", Article))

# remove the classification language sub-text at the end of each article
articles_df <- separate(articles_df, col = text_noq, into = c("Article", "Metadata"),
                     sep = "Classification Language: ", extra = "merge") |>
  select(ID, Article)

### ------------------------
# cleaning Nexus text data
### ------------------------

# creating a corpus object using the corpus function
corp_nuclear <- corpus(x = articles_df, text_field = "Article")

# creating a summary stats table
article_stats <- summary(corp_nuclear)
```

Cleaning & transforming the data for topic modeling

```r
# creating tokens of the data
toks <- tokens(corp_nuclear,
               remove_punct = T,
               remove_numbers = T)

# loading stop words and adding some words to it too
add_stops <- c(stopwords("en"), "nuclear", "energy", "photo",
               "said", "also", "two", "will", "can", "last", "one", "time", "year",
               "us", "including", "power", "plant")

# removing stopwords from the token lists
toks_clean <- tokens_select(toks,
                pattern = add_stops,
                selection = "remove")

# converting everything to lowercase and creating a dfm matrix
dfm_all <- dfm(toks_clean, tolower = T)

#trimming the matrix for only words that occur in at least two docs
dfm <- dfm_trim(dfm_all,
                min_docfreq = 2)

# creating a logical for words & reducing sparse rows
sel_idx <- slam::row_sums(dfm)>0
dfm <- dfm[sel_idx, ]
```

Plotting term frequency to explore the data

```
# term frequency
term_freq <- textmineR::TermDocFreq(dfm) |>
  arrange(desc(term_freq))

terms_top15 <- head(term_freq, 15)

# plotting term frequency
ggplot(terms_top15, aes(x = term,
                        y = term_freq)) +
  geom_col(fill = "#e76f51") +
  labs(x = "Term",
       y = "Term Frequency") +
  coord_flip() +
  theme_minimal()
```



Selecting the best value for k (using three models)

Model 1 - Three topics

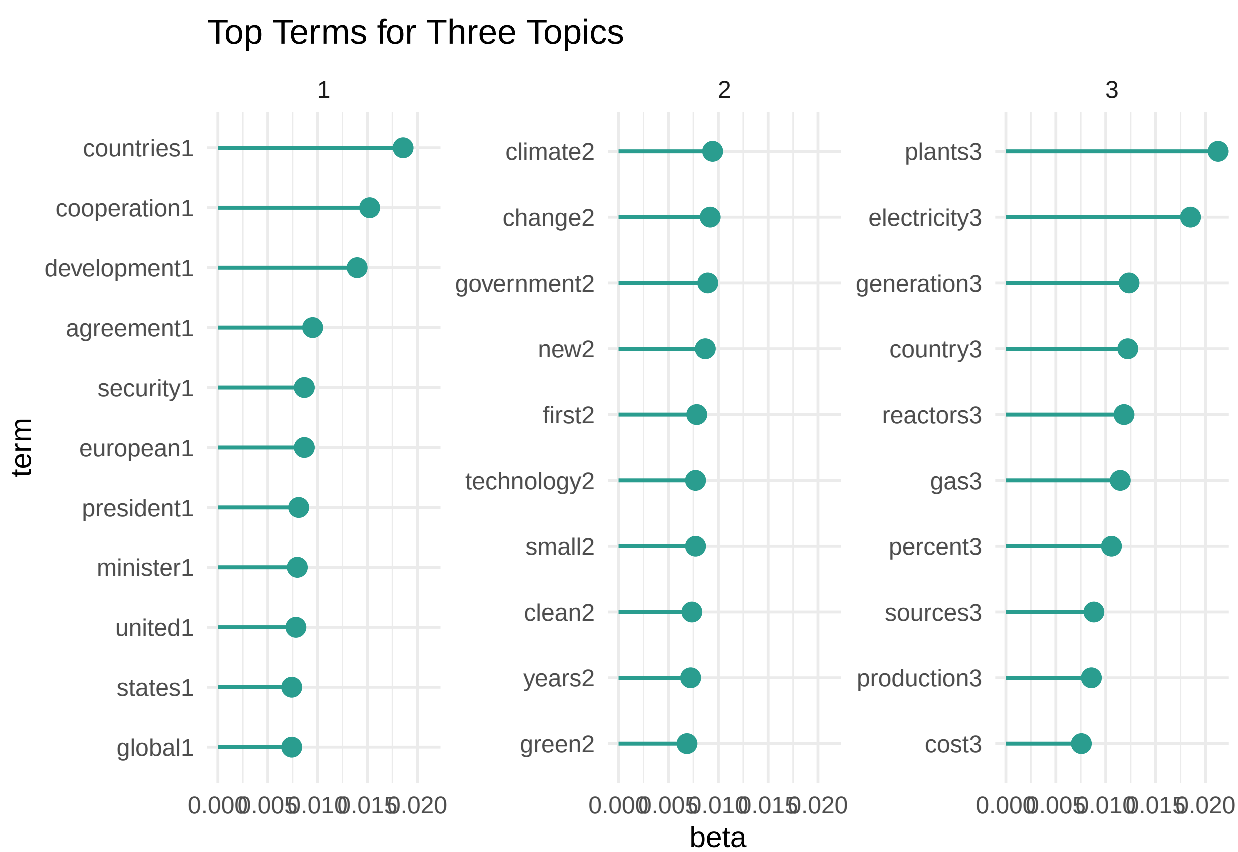
```
### ---------------------
# Creating the Model
### ---------------------
```

```r
k <- 3 # setting the number of topics

# setting model parameters
control <- list(iter = 10000, verbose = 1000)
alpha <- 0.1
beta <- 0.01

# running the LDA model and Gibbs to estimate joint distribution
topicModel_k3 <- LDA(dfm, k,
                     method = "Gibbs",
                     control = control,
                     alpha = alpha,
                     beta = beta)
```

```
## K = 3; V = 2865; M = 100
## Sampling 10000 iterations!
## Iteration 1000 ...
## Iteration 2000 ...
## Iteration 3000 ...
## Iteration 4000 ...
## Iteration 5000 ...
## Iteration 6000 ...
## Iteration 7000 ...
## Iteration 8000 ...
## Iteration 9000 ...
## Iteration 10000 ...
## Gibbs sampling completed!
```

```r
### ---------------------
# Evaluating Results
### ---------------------

# collecting the results from the model
result <- posterior(topicModel_k3)


# Looking at the different words in the topics
top_words_3topics <- tibble(terms(topicModel_k3, 10))
colnames(top_words_3topics)[1] <- "Topics"

#top_words_3topics

# tidying the model object and pulling the probability
nuclear_topics3 <- tidy(topicModel_k3,
                 matrix = "beta")

# picking words most relevant to the meaning of the topic
top_terms3 <- nuclear_topics3 |>
  group_by(topic) |>
  top_n(10, beta) |>
  ungroup() |>
  arrange(topic, -beta)
```

```r
### ---------------------
# Plotting
### ---------------------

# plotting top terms
top_terms3 |>
  mutate(term = reorder_within(term, beta, topic, sep = "")) %>%
  ggplot(aes(term, beta)) +
  geom_point(size = 3, shape = 21, color = "#2a9d8f",
           fill = "#2a9d8f") +
  geom_segment(aes(xend = term, yend = 0), color = "#2a9d8f",
             linewidth = 0.7) +
  facet_wrap(~ topic, scales = "free_y") +
  scale_x_reordered() +
  coord_flip() +
  theme_minimal() +
  labs(title = "Top Terms for Three Topics") +
  guides(fill = FALSE)
```



Model 2 - Five Topics

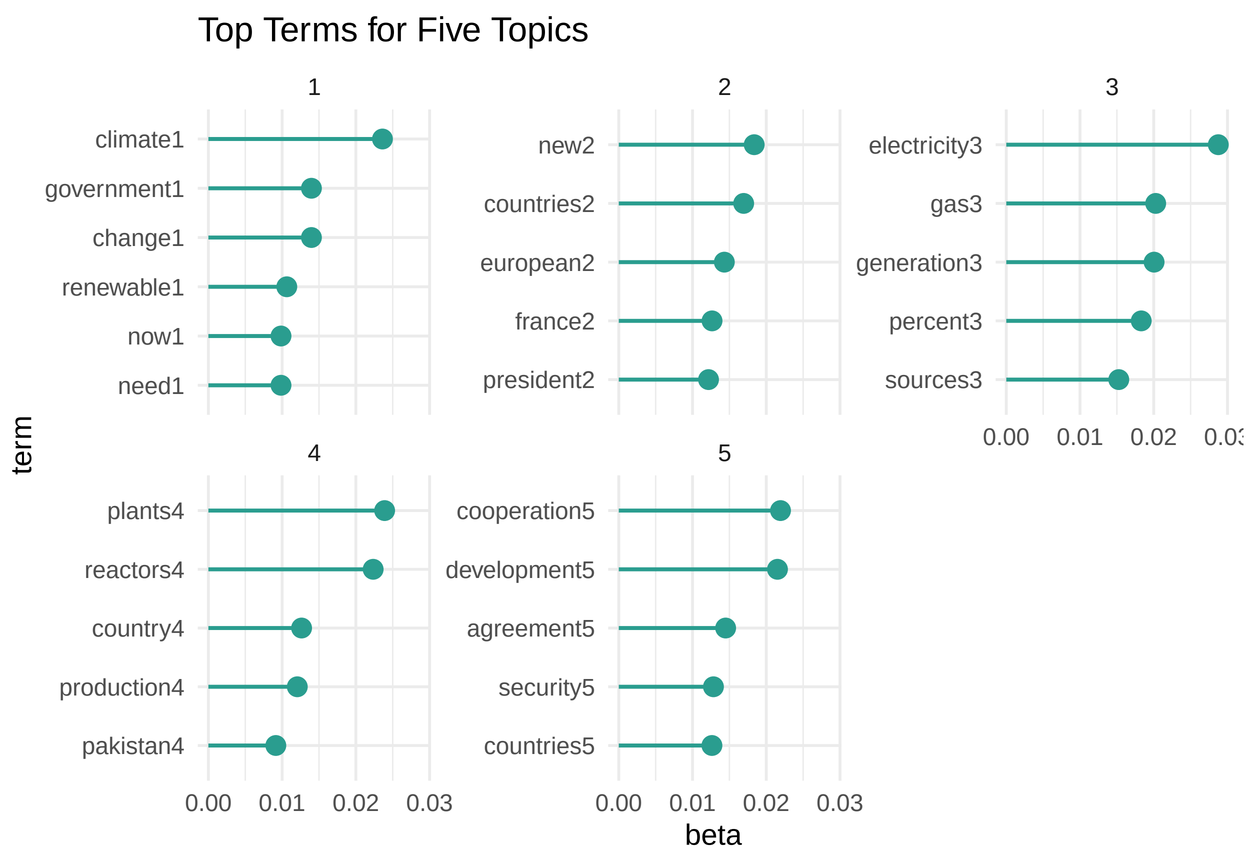
```r
### ---------------------
# Creating the Model
### ---------------------

k <- 5 # setting the number of topics
```

```r
# setting model parameters
control <- list(iter = 10000, verbose = 1000)
alpha <- 0.1
beta <- 0.01

# running the LDA model and Gibbs to estimate joint distribution
topicModel_k5 <- LDA(dfm, k,
                      method = "Gibbs",
                      control = control,
                      alpha = alpha,
                      beta = beta)
```

```
## K = 5; V = 2865; M = 100
## Sampling 10000 iterations!
## Iteration 1000 ...
## Iteration 2000 ...
## Iteration 3000 ...
## Iteration 4000 ...
## Iteration 5000 ...
## Iteration 6000 ...
## Iteration 7000 ...
## Iteration 8000 ...
## Iteration 9000 ...
## Iteration 10000 ...
## Gibbs sampling completed!
```

```r
### ---------------------
# Evaluating Results
### ---------------------

# collecting the results from the model
results5 <- posterior(topicModel_k5)


top_words_5topics <- tibble(terms(topicModel_k5, 10))
colnames(top_words_5topics)[1] <- "Topics"

#top_words_5topics

# tidying the model object and pulling the probability
nuclear_topics5 <- tidy(topicModel_k5,
                  matrix = "beta")

# pickign words most relevant to the meaning of the topic
top_terms5 <- nuclear_topics5 |>
  group_by(topic) |>
  top_n(5, beta) |>
  ungroup() |>
  arrange(topic, -beta)

### ---------------------
# Plotting
### ---------------------
```

```r
# plotting top terms
top_terms5 |>
  mutate(term = reorder_within(term, beta, topic, sep = "")) %>%
  ggplot(aes(term, beta)) +
  geom_point(size = 3, shape = 21, color = "#2a9d8f",
             fill = "#2a9d8f") +
  geom_segment(aes(xend = term, yend = 0), color = "#2a9d8f",
               size = 0.7) +
  facet_wrap(~ topic, scales = "free_y") +
  scale_x_reordered() +
  coord_flip() +
  theme_minimal() +
  labs(title = "Top Terms for Five Topics") +
  guides(fill = FALSE)
```



Model 3 - Ten Topics

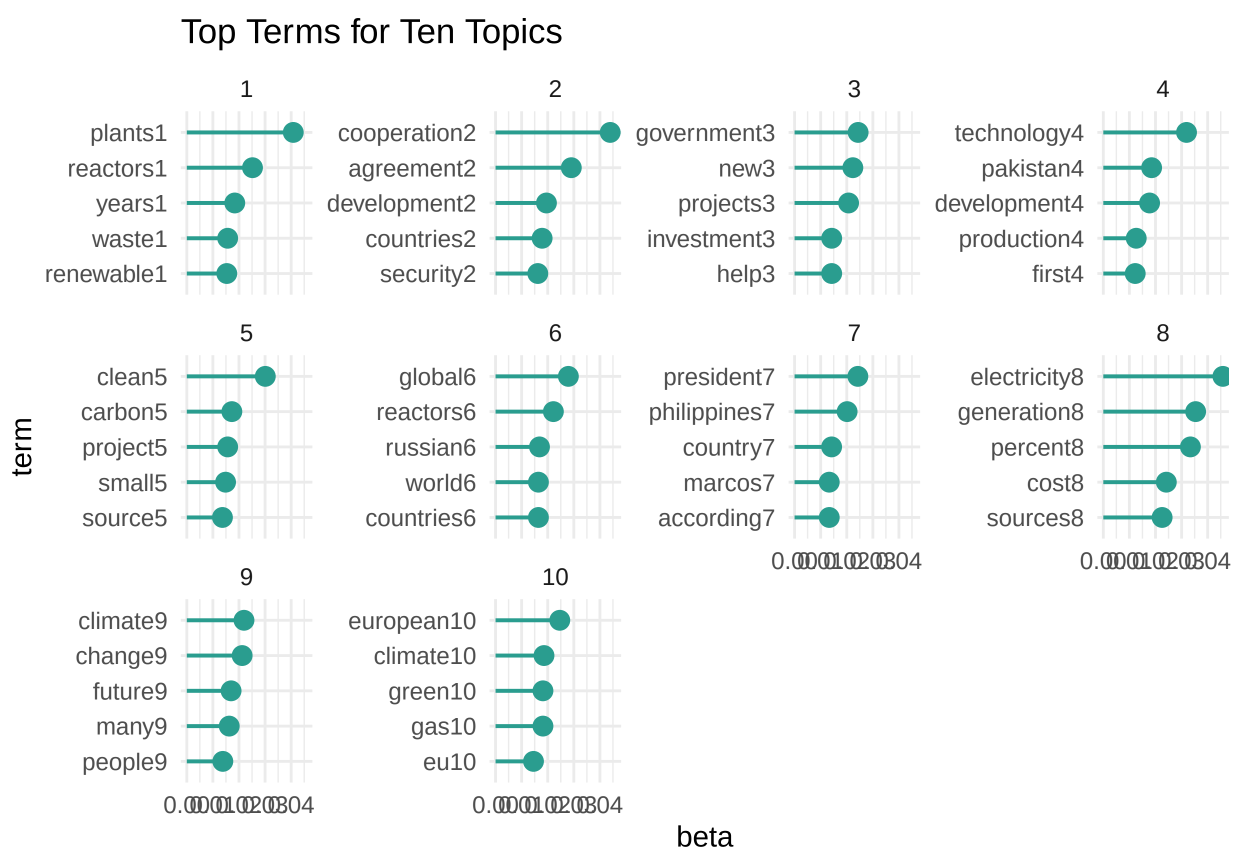
```r
### ---------------------
# Creating the Model
### ---------------------

k <- 10 # setting the number of topics

# setting model parameters
control <- list(iter = 10000, verbose = 1000)
```

```r
alpha <- 0.1
beta <- 0.01

# running the LDA model and Gibbs to estimate joint distribution
topicModel_k10 <- LDA(dfm, k,
                      method = "Gibbs",
                      control = control,
                      alpha = alpha,
                      beta = beta)
```

```
## K = 10; V = 2865; M = 100
## Sampling 10000 iterations!
## Iteration 1000 ...
## Iteration 2000 ...
## Iteration 3000 ...
## Iteration 4000 ...
## Iteration 5000 ...
## Iteration 6000 ...
## Iteration 7000 ...
## Iteration 8000 ...
## Iteration 9000 ...
## Iteration 10000 ...
## Gibbs sampling completed!
```

```r
### ---------------------
# Evaluating Results
### ---------------------

# collecting the results from the model
results10 <- posterior(topicModel_k10)


top_words_10topics <- tibble(terms(topicModel_k10, 10))
colnames(top_words_10topics)[1] <- "Topics"

#top_words_10topics

# tidying the model object and pulling the probability
nuclear_topics10 <- tidy(topicModel_k10,
                  matrix = "beta")

# pickign words most relevant to the meaning of the topic
top_terms10 <- nuclear_topics10 |>
  group_by(topic) |>
  top_n(5, beta) |>
  ungroup() |>
  arrange(topic, -beta)

### ---------------------
# Plotting
### ---------------------

# plotting top terms
```

```
top_terms10 |>
  mutate(term = reorder_within(term, beta, topic, sep = "")) |>
  ggplot(aes(term, beta)) +
  geom_point(size = 3, shape = 21, color = "#2a9d8f",
             fill = "#2a9d8f") +
  geom_segment(aes(xend = term, yend = 0), color = "#2a9d8f",
               size = 0.7) +
  facet_wrap(~ topic, scales = "free_y") +
  scale_x_reordered() +
  coord_flip() +
  theme_minimal() +
  labs(title = "Top Terms for Ten Topics") +
  guides(fill = FALSE)
```

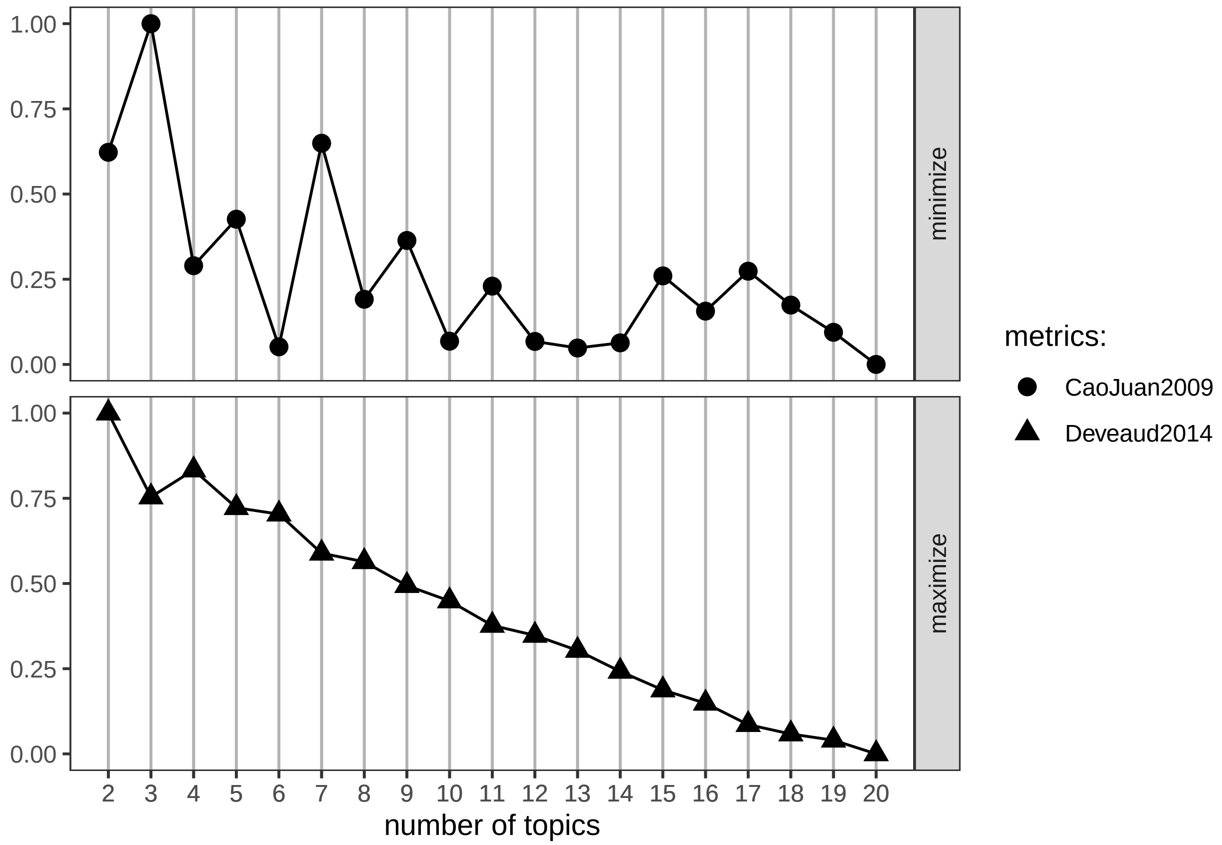


Choosing the best k with CaoJuan2009 and Deveaud2014 metrics

```
# selecting k from a range of values
results <- FindTopicsNumber(dfm,
                            topics = seq(from = 2,
                                         to = 20,
                                         by =1),
                            metrics = c("CaoJuan2009", "Deveaud2014"),
                            method = "Gibbs",
                            verbose = T)
```

```
## fit models... done.
## calculate metrics:
##   CaoJuan2009... done.
##   Deveaud2014... done.
```

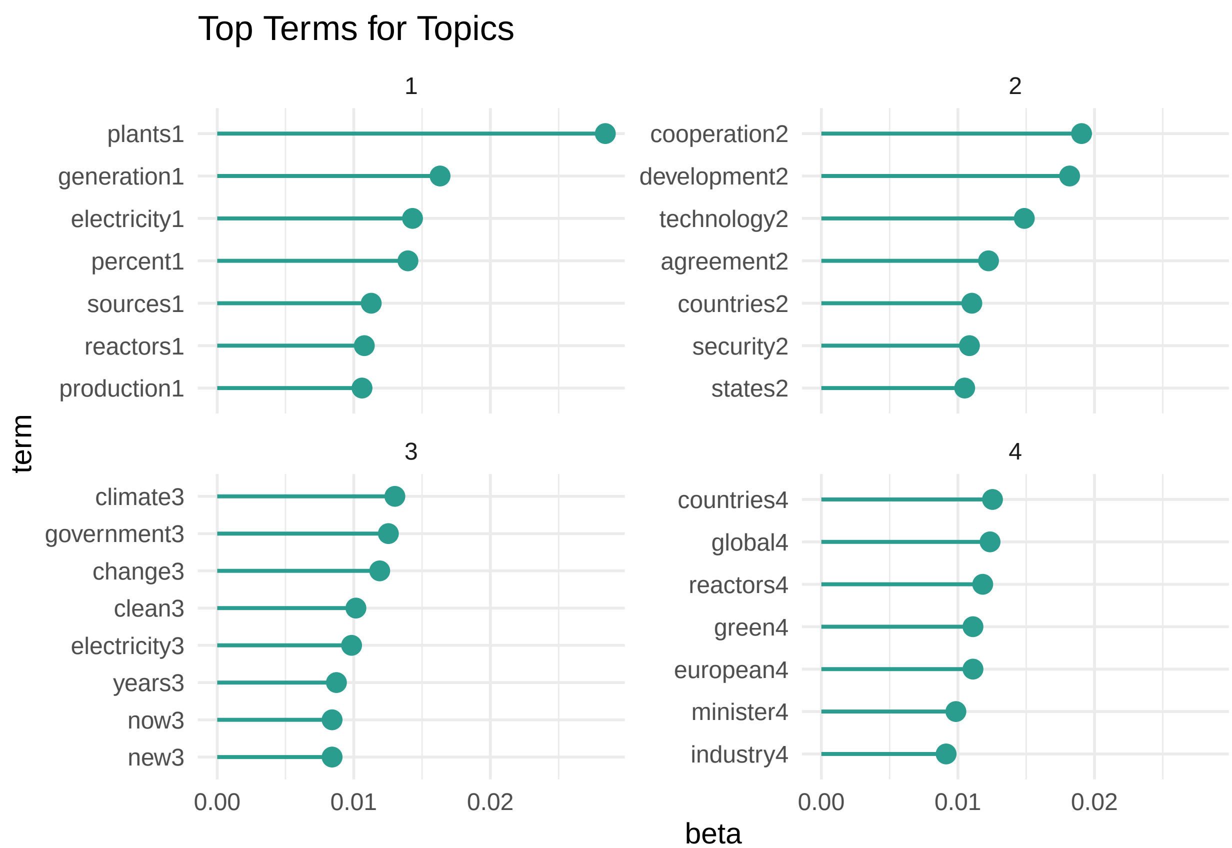
```
# plotting the results
FindTopicsNumber_plot(results)
```




```
### ---------------------
# Creating the Final Model
### ---------------------

k <- 4 # setting the number of topics

# setting model parameters
control <- list(iter = 10000, verbose = 1000)
alpha <- 0.1
beta <- 0.01

# running the LDA model and Gibbs to estimate joint distribution
topicModel_final <- LDA(dfm, k,
                      method = "Gibbs",
                      control = control,
                      alpha = alpha,
                      beta = beta)
```

```
## K = 4; V = 2865; M = 100
## Sampling 10000 iterations!
## Iteration 1000 ...
## Iteration 2000 ...
## Iteration 3000 ...
## Iteration 4000 ...
## Iteration 5000 ...
## Iteration 6000 ...
## Iteration 7000 ...
## Iteration 8000 ...
## Iteration 9000 ...
## Iteration 10000 ...
## Gibbs sampling completed!
```

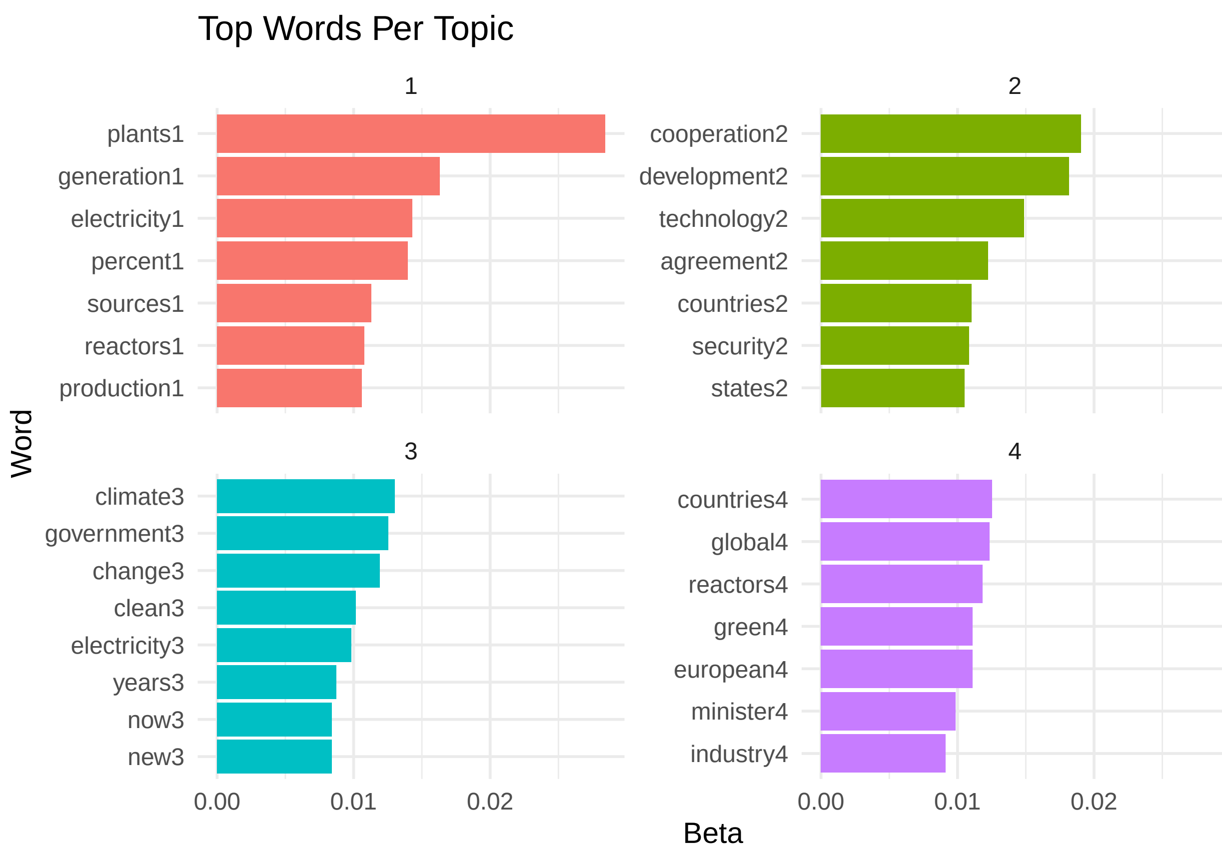
```r
### ---------------------
# Evaluating Results
### ---------------------

# collecting the results from the model
results_final <- posterior(topicModel_final)

# collecting top words
top_words <- tibble(terms(topicModel_final, 10))
colnames(top_words)[1] <- "Topics"

#top_words

# tidying the model object and pulling the probability
nuclear_topics <- tidy(topicModel_final,
                  matrix = "beta")

# picking words most relevant to the meaning of the topic
top_terms <- nuclear_topics |>
  group_by(topic) |>
  top_n(7, beta) |>
  ungroup() |>
  arrange(topic, -beta)

# plotting top terms
top_terms |>
  mutate(term = reorder_within(term, beta, topic, sep = "")) |>
  ggplot(aes(term, beta)) +
  geom_point(size = 3, shape = 21, color = "#2a9d8f",
             fill = "#2a9d8f") +
  geom_segment(aes(xend = term, yend = 0), color = "#2a9d8f",
               size = 0.7) +
  facet_wrap(~ topic, scales = "free_y") +
  scale_x_reordered() +
  coord_flip() +
  theme_minimal() +
  labs(title = "Top Terms for Topics") +
  guides(fill = FALSE)
```



RESPONSE: I choose the number of topics to be four based on the types of words within the topics were unique and this number of topics maximized the Deveaud2014 metric and minimized the CaoJuan2009 metric. Even though the CaoJuan2009 metric was lower with five topics I found that there was overlap between some topics with five topics. The Deveaud2014 metric was also higher at four topics compared to three.

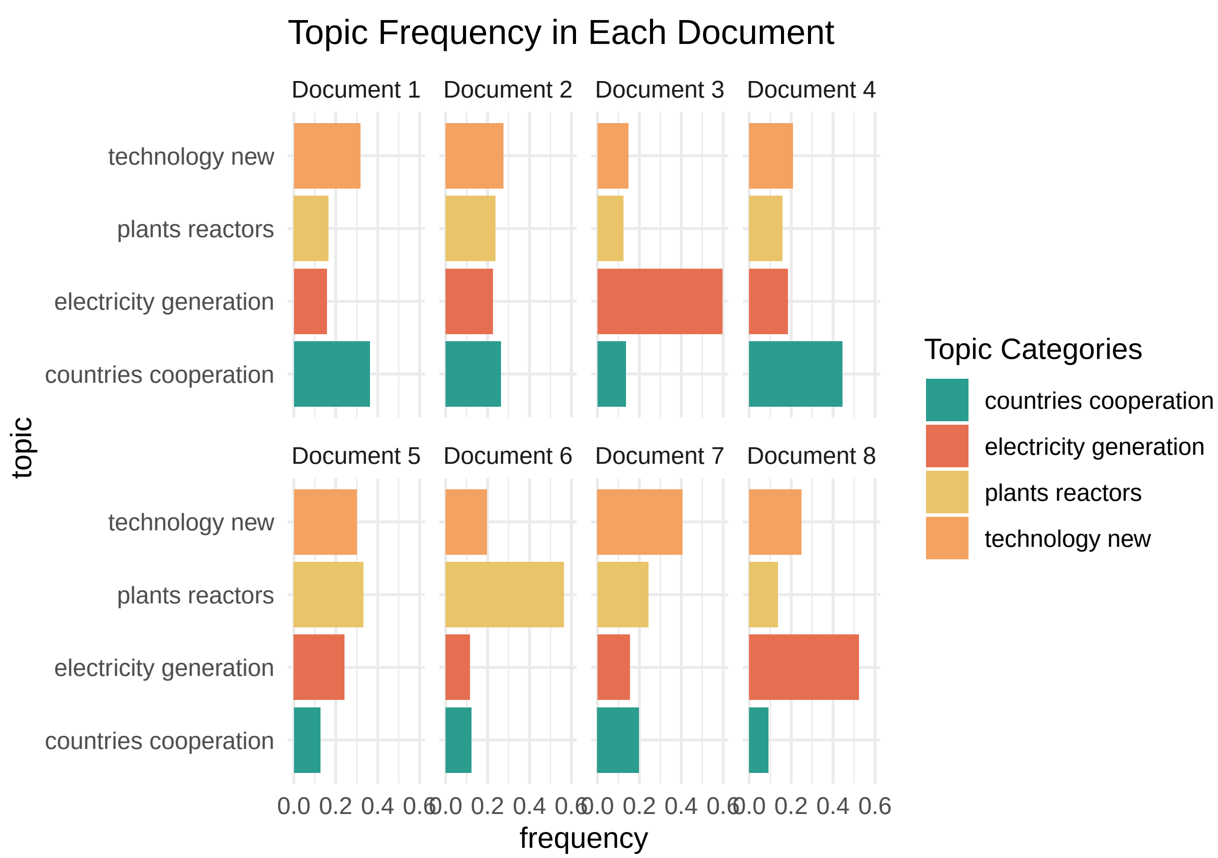
4. Plot the top terms in each topic and the distribution of topics across a sample of the documents (constrained by what looks good in the plot).

```
### ---------------------
# Plotting Top Words
### ---------------------

# plotting top words for the four topics
top_terms |>
  mutate(term = reorder_within(term, beta, topic, sep = "")) |>
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free_y") +
  scale_x_reordered()+
  coord_flip() +
  theme_minimal() +
  labs(title = "Top Words Per Topic",
       y = "Beta",
       x = "Word")
```

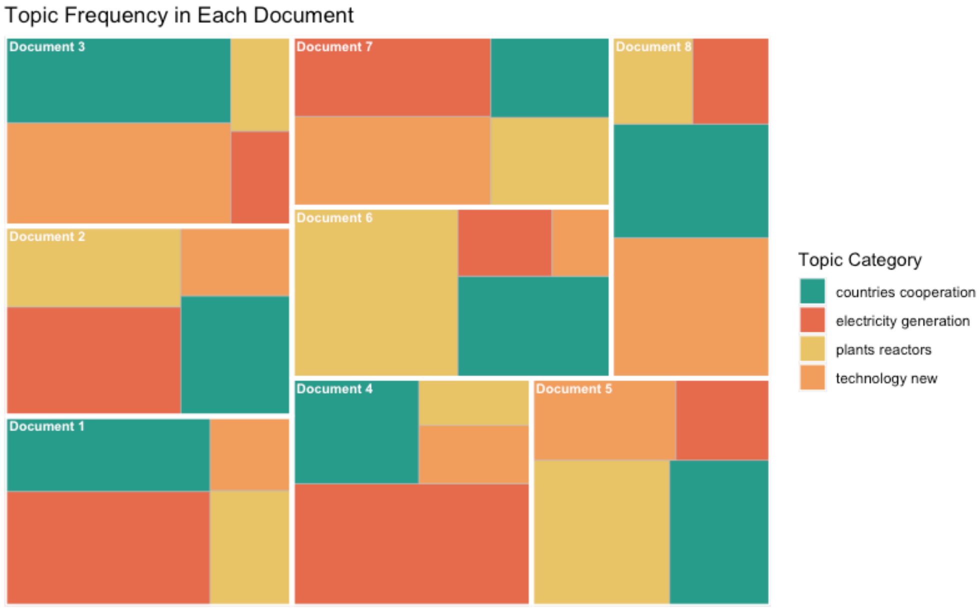



```r
### --------------------
# Data Cleaning for Plotting
### --------------------

# getting top words from the topics
topic_words <- terms(topicModel_final, 2)
topic_names <- apply(topic_words, 2, paste, collapse = " ")

# updating the topic names
names_df <- as_tibble(topic_names)
names_list <- as.list(names)

# setting up parameters for document frequency
theta <- results_final$topics
example_ids <- c(1:8)
n <- length(example_ids)

# selecting documents frequencies to sample from the theta results
example_props <- theta[example_ids, ]

# creating the dataframe for graphing
graph_df <- melt(cbind(data.frame(example_props),
                       document = factor(1:n),
                       variable.name = "topic",
                       id.vars = "document")) |>
  rename("topic" = "variable",
```

```
        "frequency" = "value") |>
  select(-c(variable.name, id.vars)) |>
  mutate(topic = case_when(topic == "X1" ~ "countries cooperation",
                           topic == "X2" ~ "electricity generation",
                           topic == "X3" ~ "plants reactors",
                           topic == "X4" ~ "technology new"),
         document = paste("Document ", document, sep = ""))

### ---------------------
# Generating Graphs
### ---------------------

# plotting a bar graph
ggplot(data = graph_df, aes(topic, frequency, fill = topic), ylab = "proportion") +
  geom_bar(stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  coord_flip() +
  facet_wrap(~ document, ncol = 4) +
  theme_minimal() +
  labs(title = "Topic Frequency in Each Document",
       fill = "Topic Categories") +
    scale_fill_manual(values = c("#2a9d8f", "#e76f51",
                                 "#e9c46a", "#f4a261"))
```



```
# plotting a treemap
ggplot(graph_df, aes(area = frequency, fill = topic,
                     label = document,
                     subgroup = document)) +
  geom_treemap() +
  geom_treemap_subgroup_border(color = "white") +
  geom_treemap_subgroup_text(size = 8, color = "white",
                              fontface = "bold", place = "topleft") +
  labs(title = "Topic Frequency in Each Document",
       fill = "Topic Category") +
  scale_fill_manual(values = c("#2a9d8f", "#e76f51",
                               "#e9c46a", "#f4a261"))
```




```
### ---------------------
# Summary Stats
### ---------------------

# setting up parameters for document frequency
theta2 <- results_final$topics
ids_100 <- c(1:100)
n <- length(ids_100)

# selecting documents frequencies to sample from the theta results
example_props <- theta[ids_100, ]

# creating the dataframe for graphing
summary_df <- melt(cbind(data.frame(example_props),
                     document = factor(1:n),
                     variable.name = "topic",
                     id.vars = "document")) |>
  rename("topic" = "variable",
         "frequency" = "value") |>
  select(-c(variable.name, id.vars)) |>
```

```
  mutate(topic = case_when(topic == "X1" ~ "countries cooperation",
                           topic == "X2" ~ "electricity generation",
                           topic == "X3" ~ "plants reactors",
                           topic == "X4" ~ "technology new"),
         document = paste("Document ", document, sep = "")) |>
  group_by(topic) |>
  summarize(avg_freq = mean(frequency))

gt(summary_df)
```

topic	avg_freq
countries cooperation	0.2362729
electricity generation	0.2638598
plants reactors	0.2540166
technology new	0.2458507

Interpreting the resulting topics. What are the key themes discussed in the articles in your data base?

RESPONSE: Each of the topics represent an interesting lens in which to look at nuclear energy. And looking at the average frequency for the topics within all of the documents it is interesting to see that they each have a similar frequency of about 0.25. This could signify that either there could be some overlap of words within topics or that nuclear energy is discussed in these specific topics equally within the corpus.

The top words within the topics also appear to be more similar to other words within the topic compared to top words in other topics. This can help indicate that the four topics are a good number in looking at topic modeling for this corpus. One topic appears to focuses on the development of clean energy, while another contains top words that describe the physical attributes of nuclear energy. Another topic contains top words that seem more future-focused on climate action and the other topic has words that contain a more governmental and jurisdiction-based lens of nuclear energy.

However, I would feel hesitant to provide this analysis as a full topic model for nuclear energy. As this corpus was limited in the number of documents and could be biased towards articles, publications and text that favor nuclear energy development and governmental relations.