Assignment 01: Text Data in R

Julia Parish

2022/04/12

Text Data in R

This text analysis was completed as an assignment for the course, Environmental Data Science 231: Text and Sentiment Analysis for Environmental Problems. The data was sourced from the New York Times using the New York Times API.

Load Packages

```
library(jsonlite) #convert results from API queries into R-friendly formats
library(tidyverse)
library(tidytext) #text data management and analysis
library(ggplot2) #plot word frequencies and publication dates
```

1. Create a free New York Times Developers account

In order to use New York Times APIs, I created an account, signed in and register the Article Search API.

```
api_key <- "qUngwwaLSVRerrzAB0TgUdXSHf93H6ea"
```

2. Key Word Selection & Query NY Times API

I have selected the search term, "coal ash". Coal ash is a term that refers to waste byproduct created by coal-fired power plants. Coal ash may contain numerous harmful substances, such as arsenic, chromium, chlorine, lead, and mercury. If the toxins found in coal ash are consumed by humans, they may cause cancer, reproductive issues, heart damage, and many more health issues. The EPA estimates that 110 million tons of coal ash are generated annually US EPA 2014). Coal ash is typically stored in landfills. In December 2008, there was 1.1 billion gallon coal ash slurry spill in Roane County, Tennessee. This environmental disaster spurred the creation and updating numerous federal regulatory and legislative policies. I anticipate that there will be a dearth of coal ash articles prior to 2008, and then a significant spike in articles on coal ash late 2008 and early 2009.

```
#set search parameters
term <- "coal+ash" # Need to use + to string together separate words
begin_date <- "20060101" # YYYYMMDD
end_date <- "20220409" # YYYYMMDD
```

```
# construct the query url using API operators
baseurl <- paste0("http://api.nytimes.com/svc/search/v2/articlesearch.json?q=",
                  term, "&begin_date=", begin_date,
                  "&end_date=", end_date, "&facet_filter=true&api-key=",
                  api_key,
                  sep = "")
baseurl
```

```
## [1] "http://api.nytimes.com/svc/search/v2/articlesearch.json?q=coal+ash&begin_date=20060101&end_date=
```

```
# this code allows for obtaining multiple pages of query results
initialQuery <- fromJSON(baseurl)

maxPages <- round((initialQuery$response$meta$hits[1] / 10) - 1)

pages <- list()
for(i in 0:maxPages){
  coalashSearch <- fromJSON(paste0(baseurl, "&page=", i), flatten = TRUE) %>%
    data.frame()
  message("Retrieving page ", i)
  pages[[i + 1]] <- coalashSearch
  Sys.sleep(6) # keeps you from hitting limit for API
}

# check class of coalashSearch object
class(coalashSearch)

# need to bind the pages and create a tibble
coalashData <- rbind_pages(pages)

dim(coalashData)

# save search results as a CSV
write_csv(x = coalashData, path = here::here("assignments/HW01_NYTimes/data/coalashData.csv"))
```

This query resulted in 41 pages, 417 articles, and 33 fields (variables).

```
coalashDF <- read.csv(here::here("assignments/HW01_NYTimes/data/coalashData.csv"))
```

```
snippet <- coalashDF$response.docs.snippet[11]

snippet
```

Review a snippet from one article

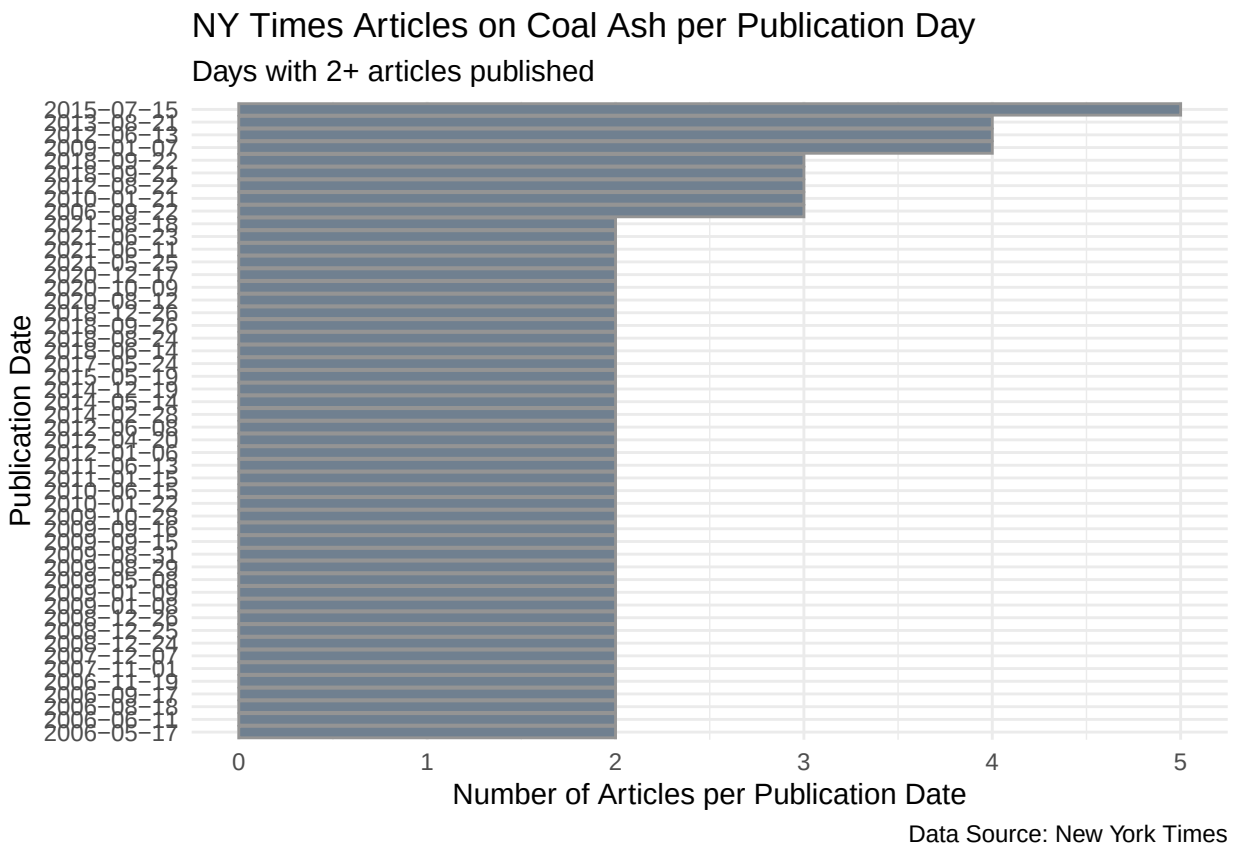
```
## [1] "The dumping of coal ash in a poor, mostly black county has generated a debate over revenue versu
```

3. Recreate the publications per day and word frequency plots using the first paragraph

3.A. Publications per day Plot

```
pubsday_plot <- coalashDF %>%
  mutate(pubDay = gsub("T.*", "", response.docs.pub_date)) %>% # replace "T." with "" - remove time but
  group_by(pubDay) %>%
  summarise(count = n()) %>%
  filter(count >= 2) %>%
  ggplot() +
  geom_bar(aes(x = reorder(pubDay, count),
               y = count),
           color = "grey58", fill = "slategrey",
           stat = "identity") +
  labs(x = "Publication Date", y = "Number of Articles per Publication Date",
```

```
       title = "NY Times Articles on Coal Ash per Publication Day",
       subtitle = "Days with 2+ articles published",
       caption = "Data Source: New York Times") +
  theme_minimal() +
  coord_flip()

pubsday_plot
```



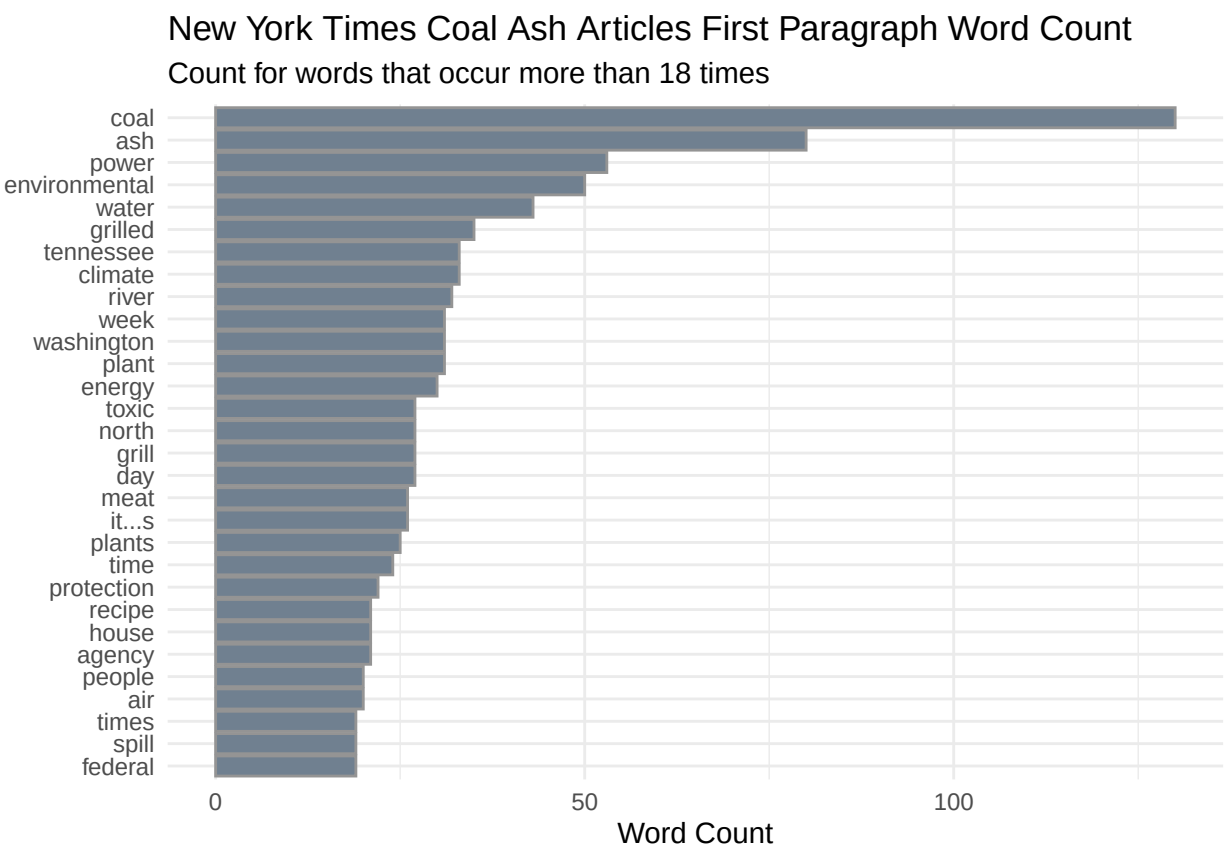
3.B. Word Frequency Plot

```
paragraph <- names(coalashDF)[6] # The 6th column, "response.doc.lead_paragraph", contains the first pa

tokenized <- coalashDF %>%
  unnest_tokens(word, paragraph)
```

```
data(stop_words)
# stop_words

tokenized <- tokenized %>%
  anti_join(stop_words) #removes the stop words
```

```
wordfreq_plot1 <- tokenized %>%
  count(word, sort = TRUE) %>%
  filter(n > 18) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
```

```
  geom_col(color = "grey58", fill = "slategrey") +
  labs(x = "Word Count", y = NULL,
       title = "New York Times Coal Ash Articles First Paragraph Word Count",
       subtitle = "Count for words that occur more than 18 times") +
  theme_minimal()

wordfreq_plot1
```



3.C. Make some (at least 3) transformations to the corpus (add stopword(s), stem a key term and its variants, remove numbers)

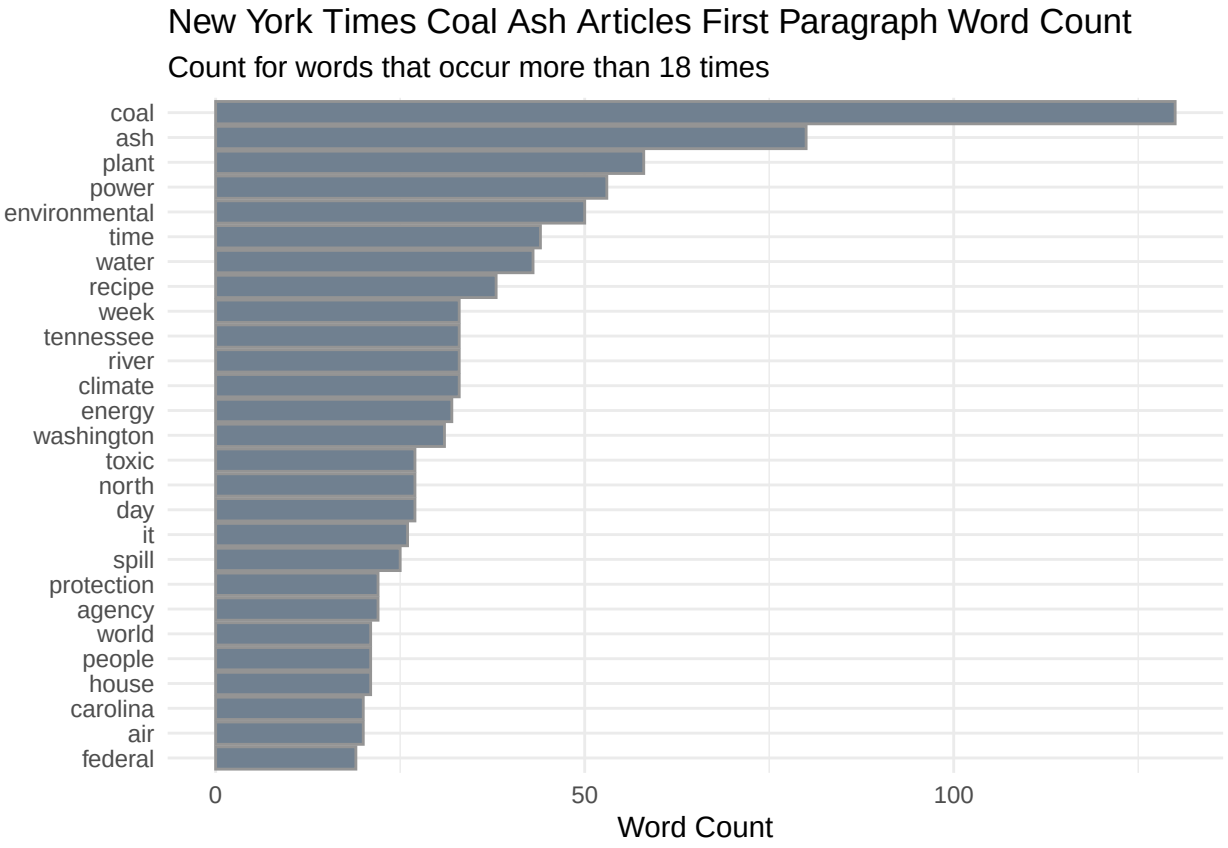
```
# inspect the list of tokens (words)
#tokenized$word

# remove all numbers
clean_tokens <- str_remove_all(tokenized$word, "[:digit:]")

# stem the token "plant" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "plant[a-z, A-Z]*",
                                replacement = "plant")

#clean_tokens <- str_replace_all(string = clean_tokens, pattern = "grill(?<=.)", replacement = "grill")

# stem the token "spill" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
```

```
                                pattern = "spill[a-z, A-Z]*",
                                replacement = "spill")

# stem the token "time" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "time[a-z, A-Z]*",
                                replacement = "time")

# stem the token "environmentalist" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "environmentalist[a-z, A-Z]*",
                                replacement = "environmentalist")

# stem the token "regulation" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "regulation[a-z, A-Z]*",
                                replacement = "regulation")

# stem the token "seam" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "seam[a-z, A-Z]*",
                                replacement = "seam")

# remove the word "recipe". Searching for coal ash resulted in a number of recipes.
clean_tokens <- str_remove_all(string = clean_tokens, pattern = "(?<=recipe).")

# remove 'ed' from the stem "grill". Searching for coal ash resulted in a number of recipes which involv
clean_tokens <- str_remove_all(string = clean_tokens, pattern = "(?<=grill)ed")

# remove 'ing' from the stem "grill". Searching for coal ash resulted in a number of recipes which invo
clean_tokens <- str_remove_all(string = clean_tokens, pattern = "(?<=grill)ing")

# remove the token "grill". Searching for coal ash resulted in a number of recipes which involve grilli
clean_tokens <- str_remove_all(string = clean_tokens, pattern = "grill")

# remove the token "meat". Searching for coal ash resulted in a number of meat recipes.
clean_tokens <- str_remove_all(string = clean_tokens, pattern = "meat")

# remove possessive and replace with blank
clean_tokens <- gsub("'s", '', clean_tokens)
```

```
tokenized$clean <- clean_tokens # put the cleaned tokens into the `tokenized` df `clean` column

#remove the empty strings
tib <-subset(tokenized, clean!="")

#reassign
tokenized <- tib
```

Word frequency plot after cleaning data

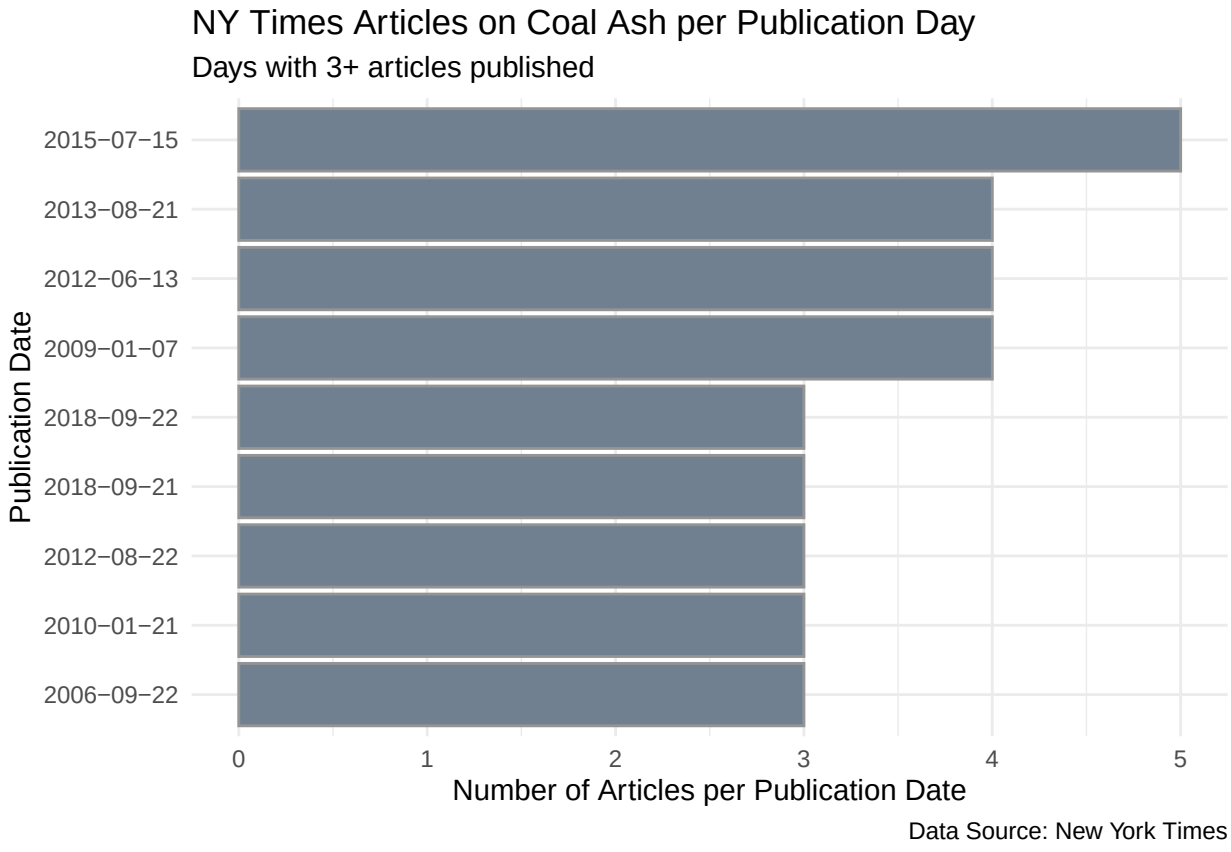
```
wordfreq_plot2 <- tokenized %>%
  count(clean, sort = TRUE) %>%
  filter(n > 18) %>%
  mutate(clean = reorder(clean, n)) %>%
  ggplot(aes(n, clean)) +
  geom_col(color = "grey58", fill = "slategrey") +
  labs(x = "Word Count", y = NULL,
       title = "New York Times Coal Ash Articles First Paragraph Word Count",
       subtitle = "Count for words that occur more than 18 times") +
  theme_minimal()

wordfreq_plot2
```



4. Recreate the publications per day and word frequency plots using the headlines variable (response.docs.headline.main).

4.A. Publications per day Plot

```
pubsday_plot <- coalashDF %>%
  mutate(pubDay = gsub("T.*", "", response.docs.pub_date)) %>% # replace "T." with "" - remove time but
  group_by(pubDay) %>%
  summarise(count = n()) %>%
  filter(count >= 3) %>%
  ggplot() +
  geom_bar(aes(x = reorder(pubDay, count),
               y = count),
           color = "grey58", fill = "slategrey",
           stat = "identity") +
  labs(x = "Publication Date", y = "Number of Articles per Publication Date",
       title = "NY Times Articles on Coal Ash per Publication Day",
       subtitle = "Days with 3+ articles published",
       caption = "Data Source: New York Times") +
  theme_minimal() +
  coord_flip()

pubsday_plot
```



4.B. Word Frequency Plot - Headlines

```
headline <- names(coalashDF)[22] # The 6th column, "response.doc.lead_paragraph", contains the first pa

tokenized_head <- coalashDF %>%
  unnest_tokens(word, headline)
```

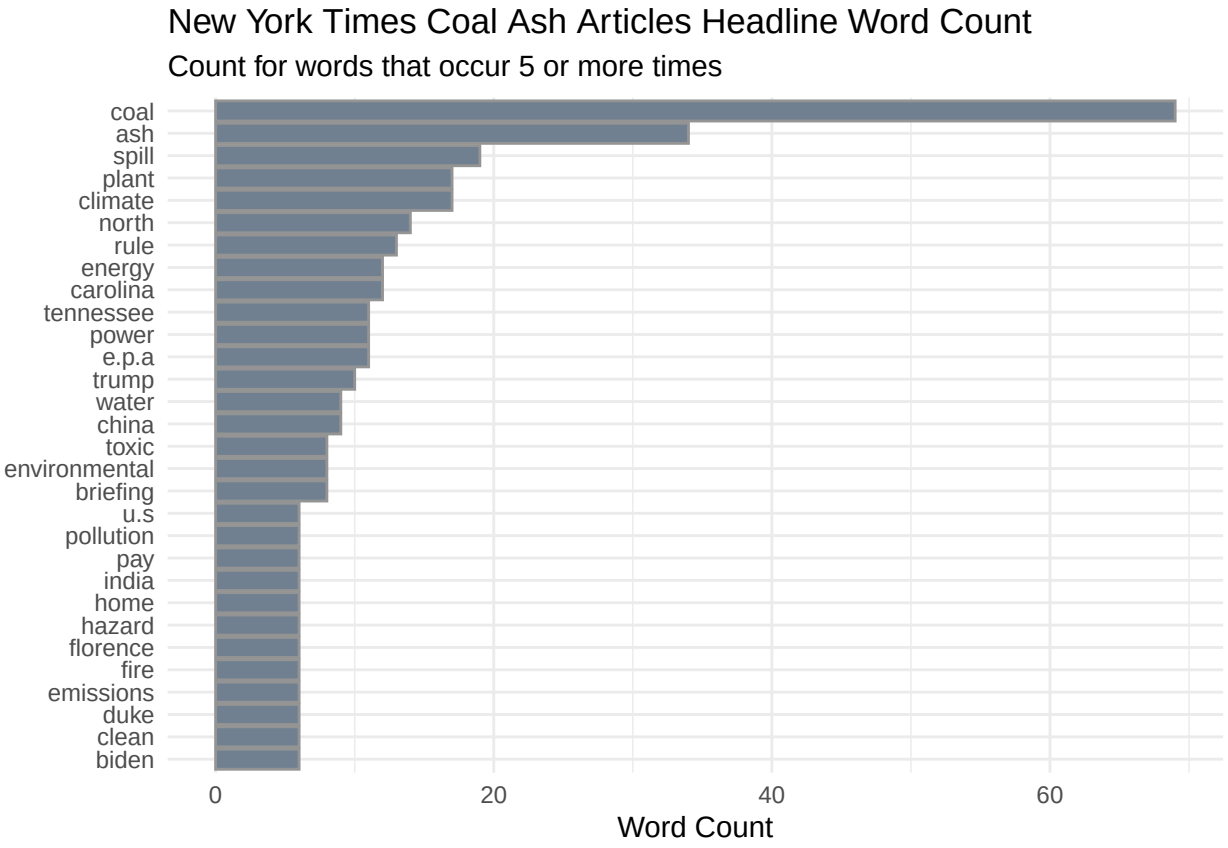
```
data(stop_words)
# stop_words

tokenized_head <- tokenized_head %>%
  anti_join(stop_words) #removes the stop words
```

```
# inspect the list of tokens (words)
# tokenized_head$word

# remove all numbers
clean_tokens <- str_remove_all(tokenized_head$word, "[:digit:]")

# stem the token "plant" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "plant[a-z, A-Z]*",
                                replacement = "plant")

# stem the token "spill" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "spill[a-z, A-Z]*",
                                replacement = "spill")

# stem the token "hazard" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "hazard[a-z, A-Z]*",
                                replacement = "hazard")

# stem the token "regulation" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "regulation[a-z, A-Z]*",
                                replacement = "regulation")

# stem the token "dump" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "dump[a-z, A-Z]*",
                                replacement = "dump")

# stem the token "rule" as it may occur in the plural form
clean_tokens <- str_replace_all(string = clean_tokens,
                                pattern = "rule[a-z, A-Z]*",
                                replacement = "rule")

# remove the word "joliet"
clean_tokens <- str_remove_all(string = clean_tokens,
                               pattern = "joliet")

# remove the word "recipe". Searching for coal ash resulted in a number of recipes.
```

```
clean_tokens <- str_remove_all(string = clean_tokens, pattern = "(?<=recipe).")

# remove 'ed' from the stem "grill". Searching for coal ash resulted in a number of recipes which involv
clean_tokens <- str_remove_all(string = clean_tokens, pattern = "(?<=grill)ed")

# remove 'ing' from the stem "grill". Searching for coal ash resulted in a number of recipes which invo
clean_tokens <- str_remove_all(string = clean_tokens, pattern = "(?<=grill)ing")

# remove the token "grill". Searching for coal ash resulted in a number of recipes which involve grilli
clean_tokens <- str_remove_all(string = clean_tokens, pattern = "grill")

# remove the token "meat". Searching for coal ash resulted in a number of meat recipes.
clean_tokens <- str_remove_all(string = clean_tokens, pattern = "meat")

# remove the token "chicken". Searching for coal ash resulted in a number of recipes.
clean_tokens <- str_remove_all(string = clean_tokens, pattern = "chicken")

# remove possessive and replace with blank
clean_tokens <- gsub("'s", '', clean_tokens)

# clean_tokens
```

```
tokenized_head$clean <- clean_tokens # put the cleaned tokens into the `tokenized` df `clean` column

#remove the empty strings
tib_head <-subset(tokenized_head, clean!="")

#reassign
tokenized_head <- tib_head
```

```
wordfreq_plot3 <- tokenized_head %>%
  count(clean, sort = TRUE) %>%
  filter(n > 5) %>%
  mutate(clean = reorder(clean, n)) %>%
  ggplot(aes(n, clean)) +
  geom_col(color = "grey58", fill = "slategrey") +
  labs(x = "Word Count", y = NULL,
       title = "New York Times Coal Ash Articles Headline Word Count",
       subtitle = "Count for words that occur 5 or more times") +
  theme_minimal()

wordfreq_plot3
```



4.C. Compare the distributions of word frequencies between the first paragraph and headlines. Do you see any difference?

For the first paragraph word count plot, the top 5 most common words were coal, ash, plant, power, and environmental. For the headlines word count plot, the top 5 most common words were coal, ash, spill, plant, and climate. It is self evident that coal and ash would be the most frequent words. It could be insightful to see what the results would be if I removed those two words from the search. I noticed that two presidents are listed as frequently occurring in the headlines word count vs the paragraph word count. It seems that for the headline words that buzzwords (ex: climate, toxic) or powerful names or countries are more frequent.

References

US EPA. (2014, December 16). Frequent Questions about the 2015 Coal Ash Disposal Rule [Other Policies and Guidance]. Accessed: 2022-04-10. https://www.epa.gov/coalash/frequent-questions-about-2015-coal-ash-disposal-rule