Day1

Shale

4/6/2022

NYT API

```
key <- "18y0Ab911Nz6AbLlBP0fFqmHTGlxAgOD"
term <- "Montana+mine" # Need to use + to string together separate words
begin_date <- "20200101"
end_date <- "20220401"

# 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=",key, sep="")
```

```
# 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){
  nytSearch <- fromJSON(paste0(baseurl, "&page=", i), flatten = TRUE) %>% data.frame()
  message("Retrieving page ", i)
  pages[[i+1]] <- nytSearch
  Sys.sleep(6) # only 10 queries allowed per minute, so have to slow down loop to once every 6 seconds
}
```

```
nytDat <- rbind_pages(pages)
write_csv(nytDat, "mt_text.csv")
```

Publications per day:

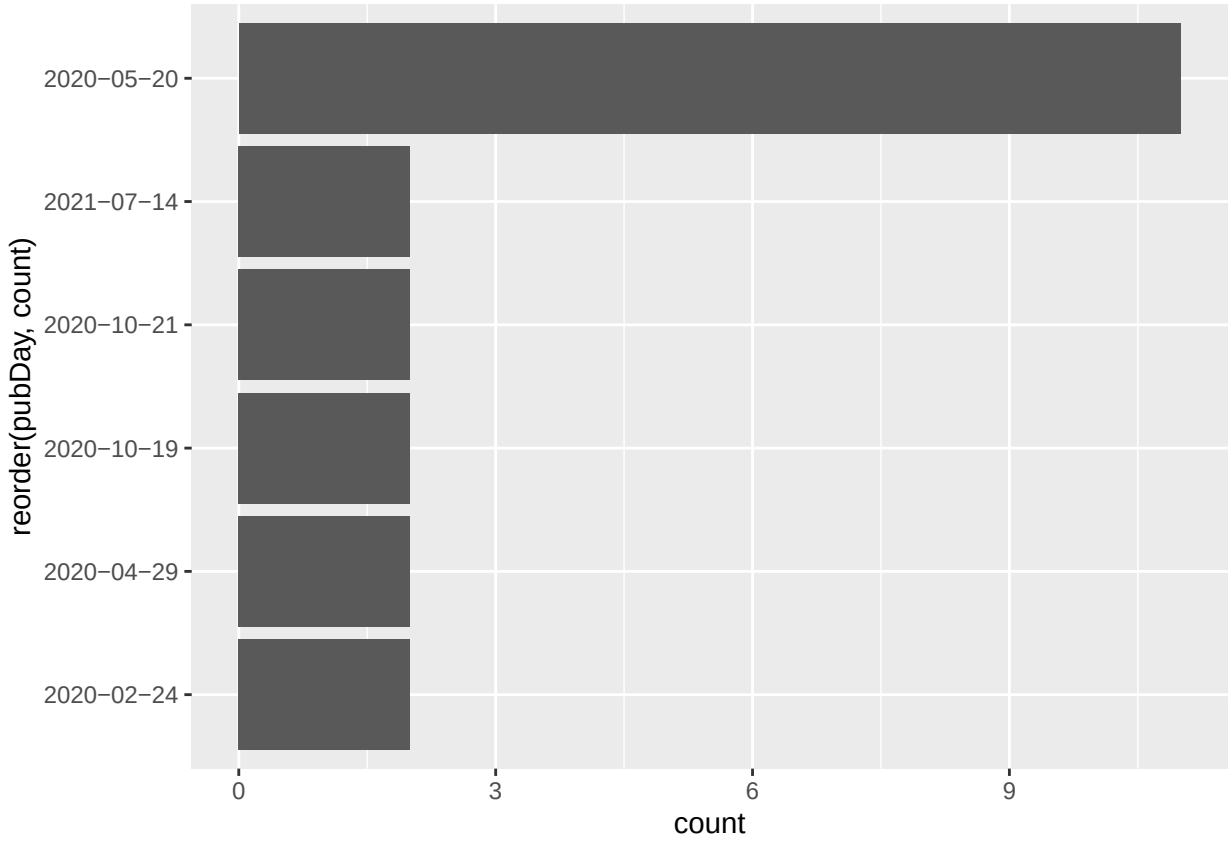
Looks like something happened with Montana mining in May of 2020?

```
nytDat = read_csv(here("mt_text.csv"))
```

```
## Rows: 85 Columns: 33-- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (20): status, copyright, response.docs.abstract, response.docs.web_url,...
## dbl   (5): response.docs.word_count, response.docs.print_page, response.meta...
## lgl   (7): response.docs.multimedia, response.docs.keywords, response.docs.h...
## dttm  (1): response.docs.pub_date
## 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.
```

```
nytDat %>%
  mutate(pubDay=gsub(" .*","",response.docs.pub_date)) %>% # exporting as csv removed the T
```

```
  group_by(pubDay) %>%
  summarise(count=n()) %>%
  filter(count > 1) %>%
  ggplot() +
  geom_bar(aes(x=reorder(pubDay, count), y=count), stat="identity") + coord_flip()
```



Word frequency in first paragraph:

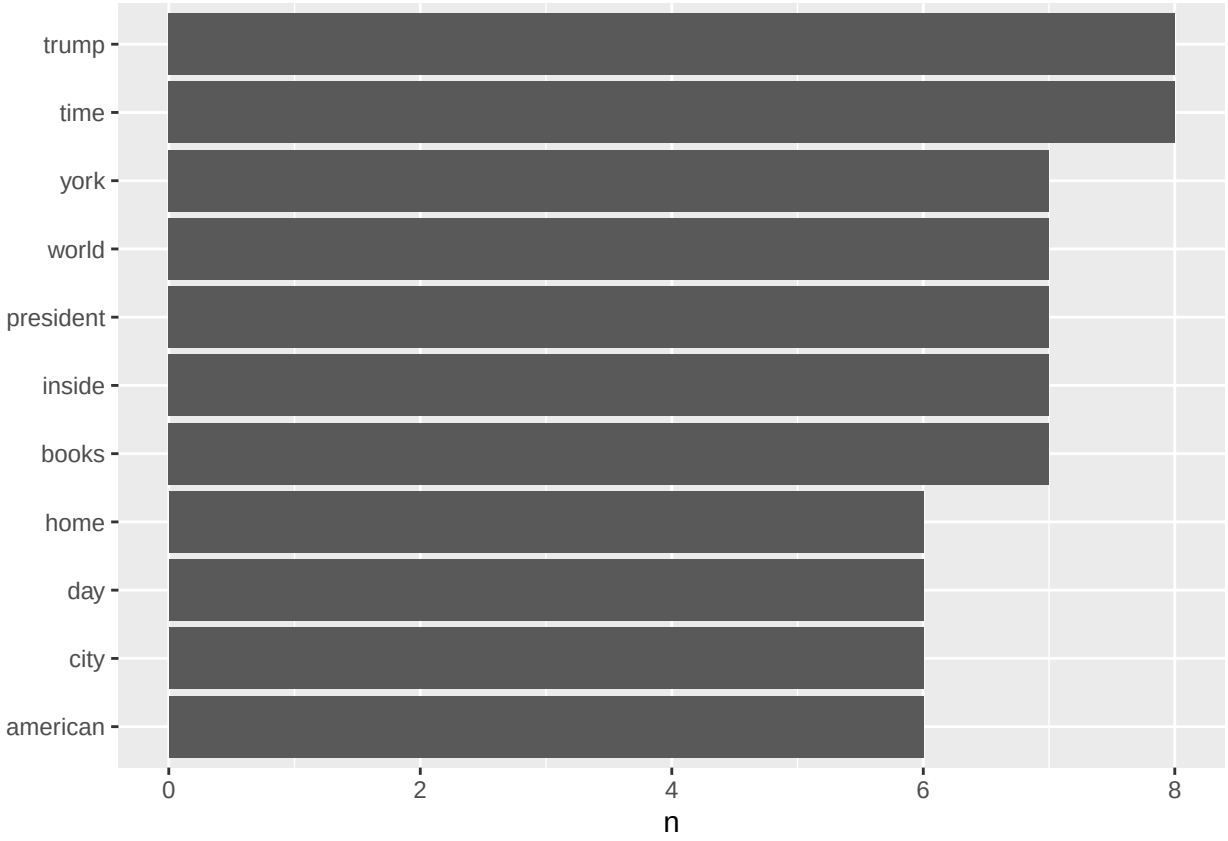
```
paragraph <- names(nytDat)[6] #The 6th column, "response.doc.lead_paragraph", is the one we want here.

tokenized <- nytDat %>%
  unnest_tokens(word, paragraph) %>%
  anti_join(stop_words)
```

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

```
tokenized$clean_tokens = tokenized$word %>%
  str_remove_all("'s$") %>% # remove 's
  str_remove_all("’s$") %>% # not all articles use the same apostrophe
  str_remove_all("[:digit:]") %>%
  str_remove_all("^.?.$") # all words with < 3 characters

tokenized %>% count(word, sort = TRUE) %>%
  filter(n > 5) %>% #illegible with all the words displayed
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = NULL)
```




```
# based off this it would mess some words up to stem words ending in 'ing'
ings = str_detect(string = tokenized$clean_tokens, pattern = "ing$")
wings = cbind(tokenized$clean_tokens, ings) %>%
  as.data.frame() %>%
  filter(ings == TRUE)
```

Word frequency using `headlines` (note that headlines doesn't depend on `word`, so will be the same no matter what)

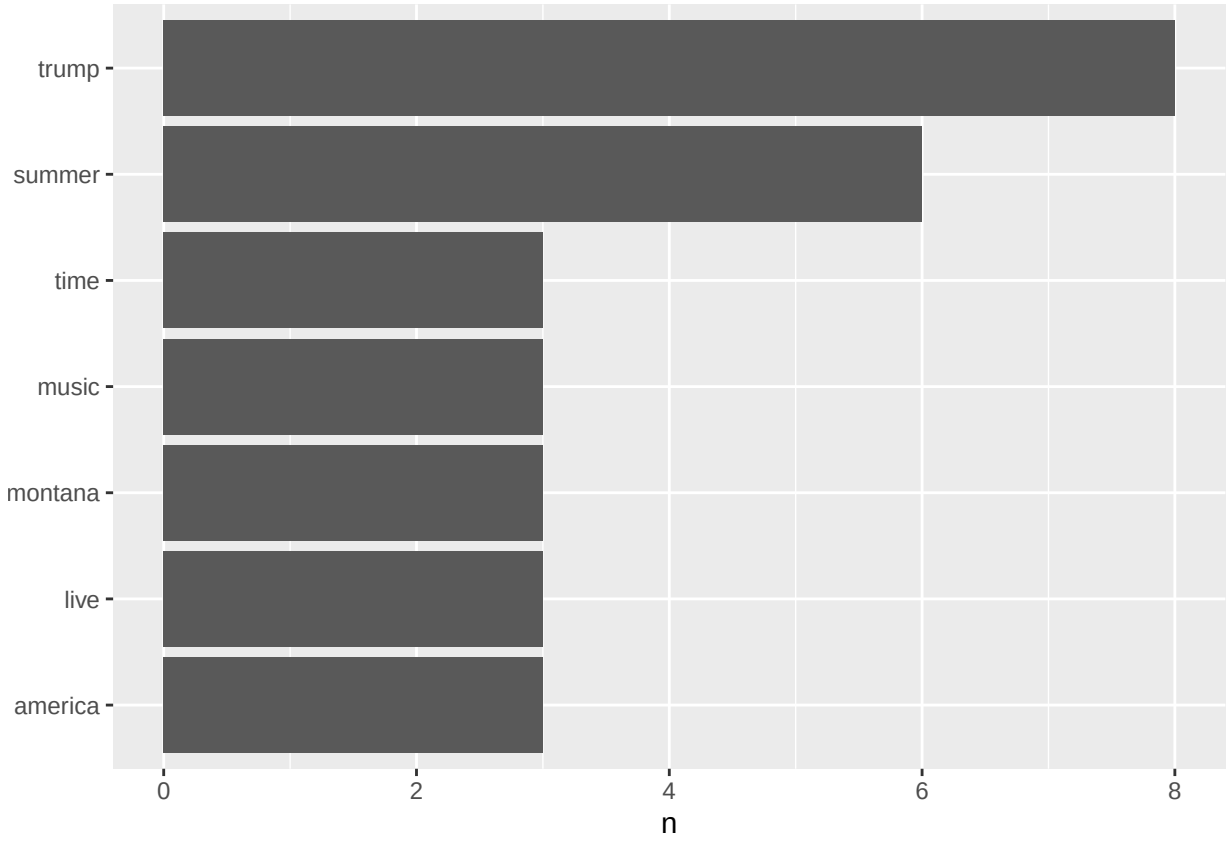
```
headline <- names(nytDat)[21] #The 6th column, "response.doc.lead_paragraph", is the one we want here.

tokenized_head <- nytDat %>%
  unnest_tokens(word, headline) %>%
  anti_join(stop_words)
```

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

```
tokenized_head$clean_tokens = tokenized_head$word %>%
  str_remove_all("'s$") %>% # remove 's
  str_remove_all("’s$") %>% # not all articles use the same apostrophe
  str_remove_all("[:digit:]") %>%
  str_remove_all("^.?.$") # all words with < 3 characters

tokenized_head %>% count(word, sort = TRUE) %>%
  filter(n > 2) %>% #illegible with all the words displayed
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = NULL)
```



Compared to the distribution of word frequencies in the first paragraph, there are fewer words that get repeated in headlines. In addition to the political keywords which appear in both lists, the headlines seem to emphasize live music more (perhaps a summer concert?) but the reduction in overall numbers makes it harder to trust any pattern that may seem to appear. There are 1782 tokens/words in the `paragraph` subset but only 416 tokens/words in the `headline` subset. As it turns out, mining vocabulary doesn't appear to be a central part of any of these articles, though the political polarization of the topic could cause mining topics to be hidden under more common political keywords like `trump` and `america`.