Assignment 1 - NY Times Text Analysis

Colleen McCamy

2023-04-09

1-2) Establishing the URL for the API Query

Created a free New York Times account to pull text data.

Chose 'decarbonization' as the environmental key word and used the {jsonlite} package to query the API.

```
# establishing the query terms
term1 <- "decarbonization"
begin_date <- "20190101"
end_date <- "20220101"

#construct the query url using API operators
baseurl <- paste0("http://api.nytimes.com/svc/search/v2/articlesearch.json?q=",
                  term1,
                  "&begin_date=", begin_date,
                  "&end_date=", end_date,
                  "&facet_filter=true&api-key=", API_KEY,
                  sep="")

#examine our query url
baseurl
```

Looping Through the Pages of Articles

```
# find the initial query to determine how many hits are available
initialQuery <- fromJSON(baseurl)

# determining the max pages
maxPages <- round((initialQuery$response$meta$hits[1] / 10) -1)

# creating a blank list to store each API result
pages <- list() # object that will store each result from the API looking for a list of length 20

#looping through URL for all different pages
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(12)
}
```

Saving & Loading the Data

```
# binding the rows
nyt_df <- do.call("bind_rows", pages) # binding all of the dataframes within the list as one big datafra

saveRDS(nyt_df, "data/nyt_df_decarbonization.rds") #saving it as an RDS
```

```
nytDat <- readRDS("data/nyt_df_decarbonization.rds")

# looking at the data
#dim(nytDat)
#unique(nytDat$status)
```

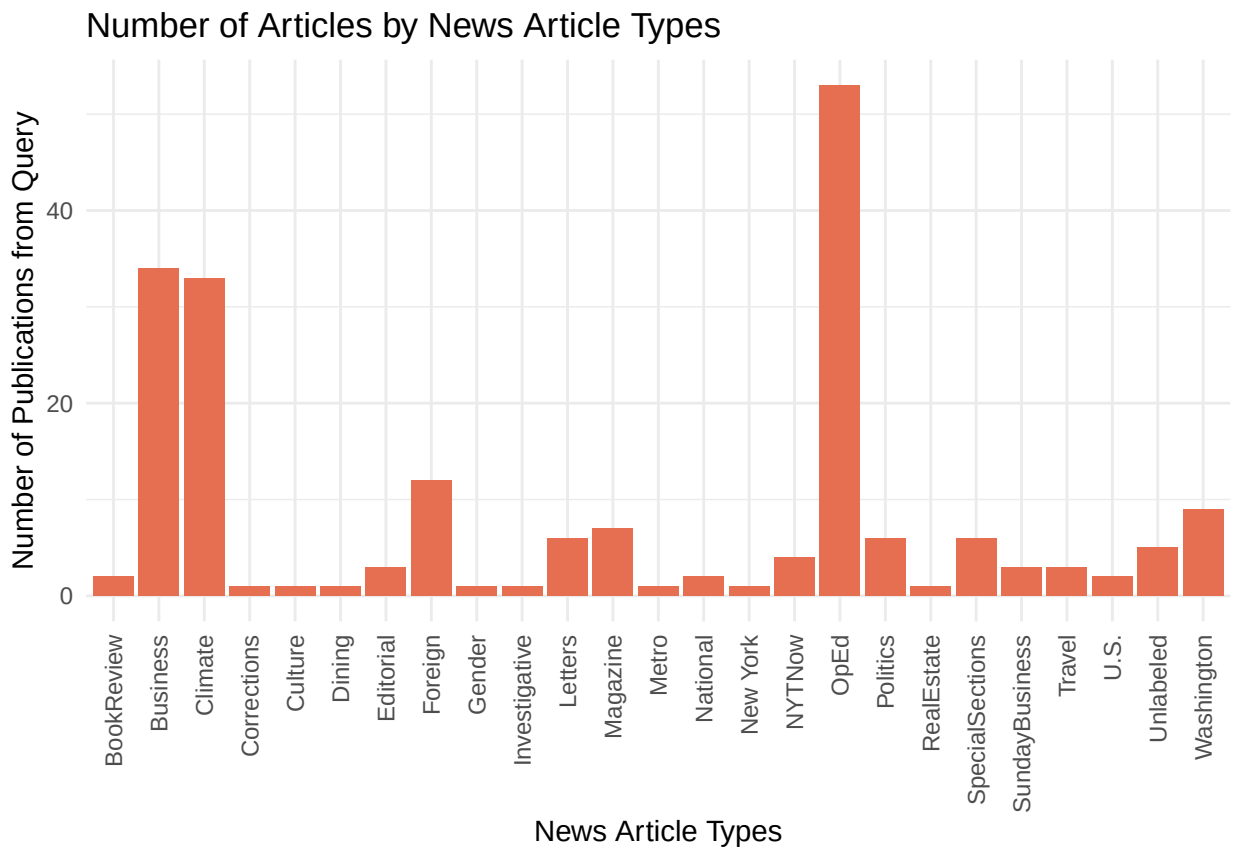
3) Date Published & Word Frequency Plots - Paragraph

Recreated the publications per day and word frequency plots using the first paragraph of each article pulled.

3a) Filtering for relevant news desks

Filtered for the new desks that have greater than 5 articles. These also correlate to news desk that would typically report on relevant decarbonization news.

```
df <- nytDat

# determining top article types
freq <- df |>
  group_by(response.docs.news_desk) |>
  summarize(count = n()) |>
  mutate(percentage = count/sum(count))

freq[1,1] <- "Unlabeled"


pfreq <- ggplot(data = freq, aes(x = response.docs.news_desk, y = count)) +
  geom_col(fill = "#e76f51") +
  labs(x = "News Article Types",
       y = "Number of Publications from Query",
       title = "Number of Articles by News Article Types") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))

# calling the plot back
pfreq
```




```
# creating a list to filter by for the larger dataframe
top_freq <- freq |>
  filter(count > 5 | count == 5)
top_types <- top_freq$response.docs.news_desk

# filtering the larger dataframe for news desks that have 5 or more articles on decarbonization
df_filtered <- df[df$response.docs.news_desk %in% top_types, ]

# ensuring that the filtering worked
unique(df_filtered$response.docs.news_desk)
```

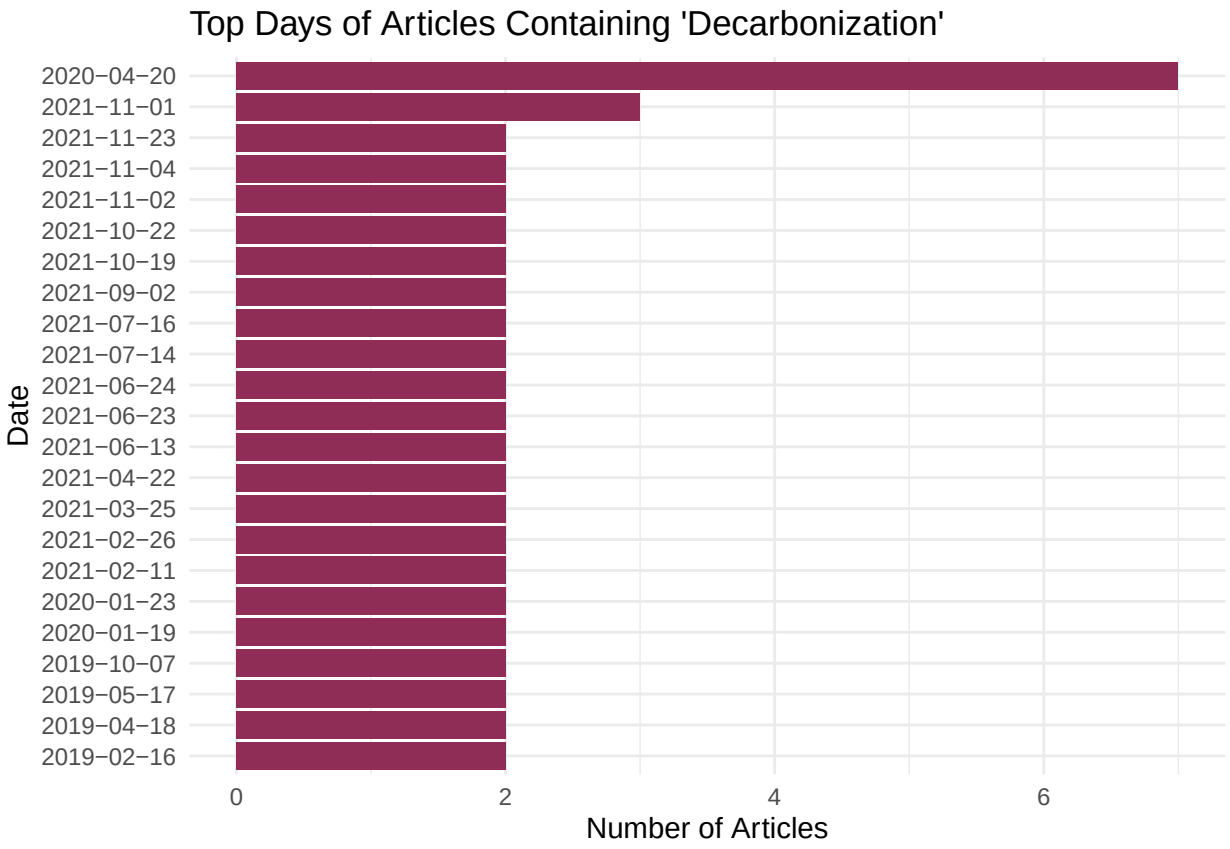
```
## [1] "OpEd"           "Climate"        "Magazine"       "Politics"
## [5] "Business"       "Washington"     "SpecialSections" "Foreign"
## [9] "Letters"
```

3b) Creating a date published graph

Created a graph of articles published for each day that had greater than two articles published with decarbonization within it.

```
# publications per date graph
df_filtered |>
  mutate(pubDay=gsub("T.*","",response.docs.pub_date)) |>
  group_by(pubDay) |>
```

```
  summarise(count=n()) |>
  filter(count >= 2) |>
  ggplot() +
  geom_bar(aes(x=reorder(pubDay, count), y=count),
           stat="identity",
           fill = "#8f2d56") +
  theme_minimal() +
  labs(y = "Number of Articles",
       x = "Date",
       title = "Top Days of Articles Containing 'Decarbonization'") +
  coord_flip()
```



3c) Creating a word frequency plot

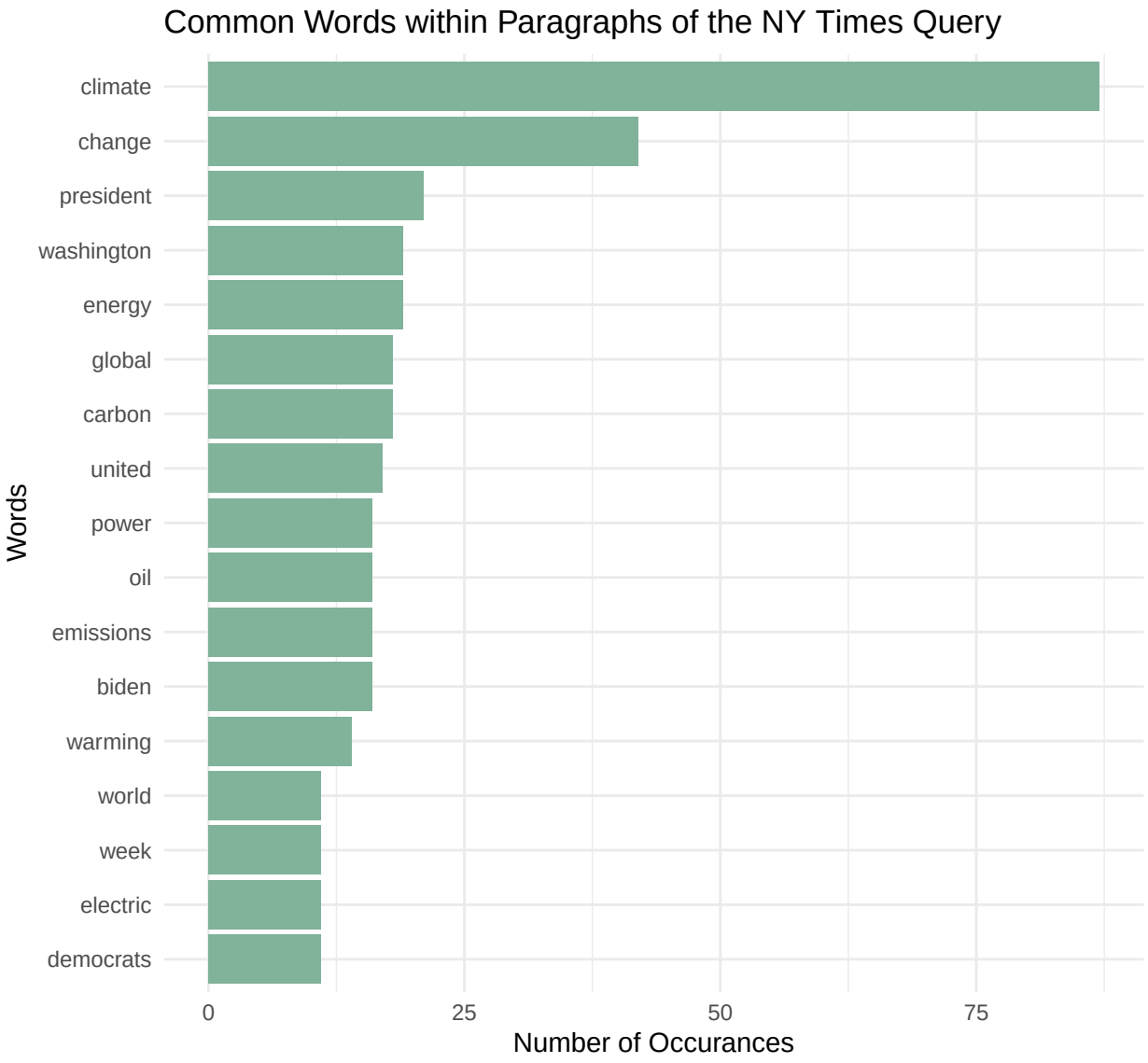
Updating the data to be "machine-usable" data with one row per observation and removing stop words.

```
# saving paragraph column name to use
paragraph <- names(df_filtered)[6]

# unnesting tokens from the tidytext:: package (turning each word to an observation)
tokenized <- df_filtered |>
  unnest_tokens(word, paragraph)

# can see the words split here
```

```
#tokenized[, "word"]

# joining stop words to the tokenized dataframe to remove these words
tokenized <- tokenized |>
  anti_join(stop_words, by = 'word')
```

Created a graph of words that appear greater than 10 times within the articles.

```
p_paragraph <- tokenized |>
  count(word, sort = TRUE) |>
  filter(n > 10) |>
  mutate(word = reorder(word, n)) |>
  ggplot(aes(n, word)) +
  geom_col(fill = "#81b29a") +
  labs(y = NULL) +
  theme_minimal() +
  labs(x = "Number of Occurances",
               y = "Words",
               title = "Common Words within Paragraphs of the NY Times Query")

p_paragraph
```



3.5) Corpus Transformations

Made transformations to the corpus and examined the differences.

Adding context-specific stopword

```
# appending stop words with new words
new_sws <- data.frame(word = c("united", "week"),
           lexicon = c("SMART", "SMART"))

sw_updated <- rbind(stop_words,
                    new_sws)
```

```
# joining the updated stop words to the tokenized dataframe
clean_tokens_join <- tokenized |>
  anti_join(sw_updated, by = 'word')
```

Removing digits from the words

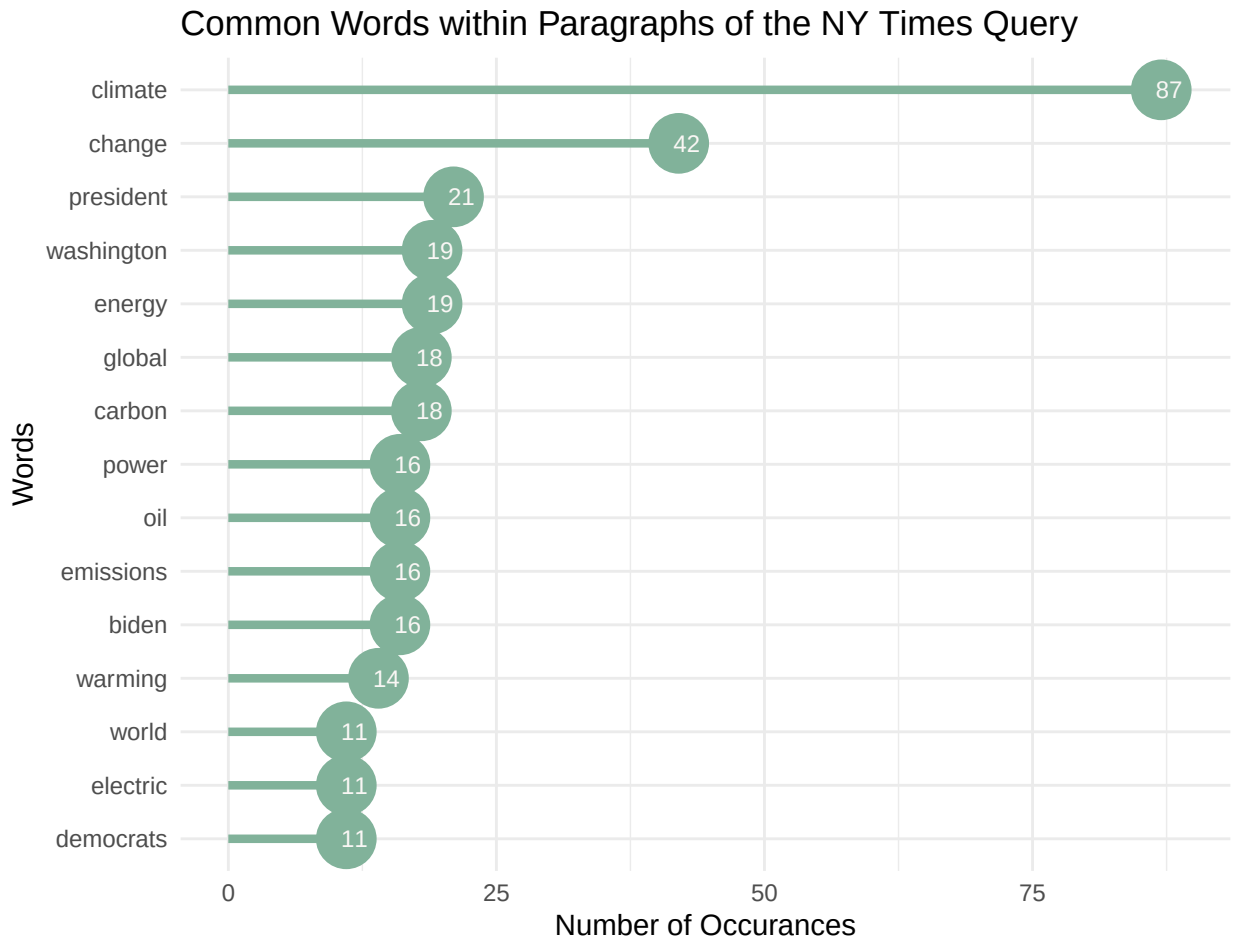
```
# removing all words that have numbers (digits)
clean_tokens <- str_remove_all(tokenized$word, "[:digit:]") # this is a wild card for all

# adding it to tokenized
tokenized$clean <- clean_tokens
```

Removing apostrophe's from the words

```
# removing aopostrophe's
clean_tokens <- gsub("'s", '', clean_tokens) # remove apostrophe 's

# adding the column back to the dataset
tokenized$clean <- clean_tokens
```

Graphing the frequency plot with the clean data

```
p_paragraph <- clean_tokens_join |>
  count(word, sort = TRUE) |>
  filter(n > 10) |>
  mutate(word = reorder(word, n)) |>
  ggplot(aes(n, word)) +
  geom_point(color = "#81b29a", size = 10) +
  geom_segment(aes(x = 0, xend = n, y = word, yend = word),
               color = "#81b29a",
               size = 1.5) +
  geom_text(aes(x = n + 2, y = word, label = n),
            vjust = 0.5,
            hjust = 1,
            size = 3,
            color = "#faf7f5") +
  labs(y = NULL) +
  theme_minimal() +
  labs(x = "Number of Occurances",
               y = "Words",
               title = "Common Words within Paragraphs of the NY Times Query")

p_paragraph
```

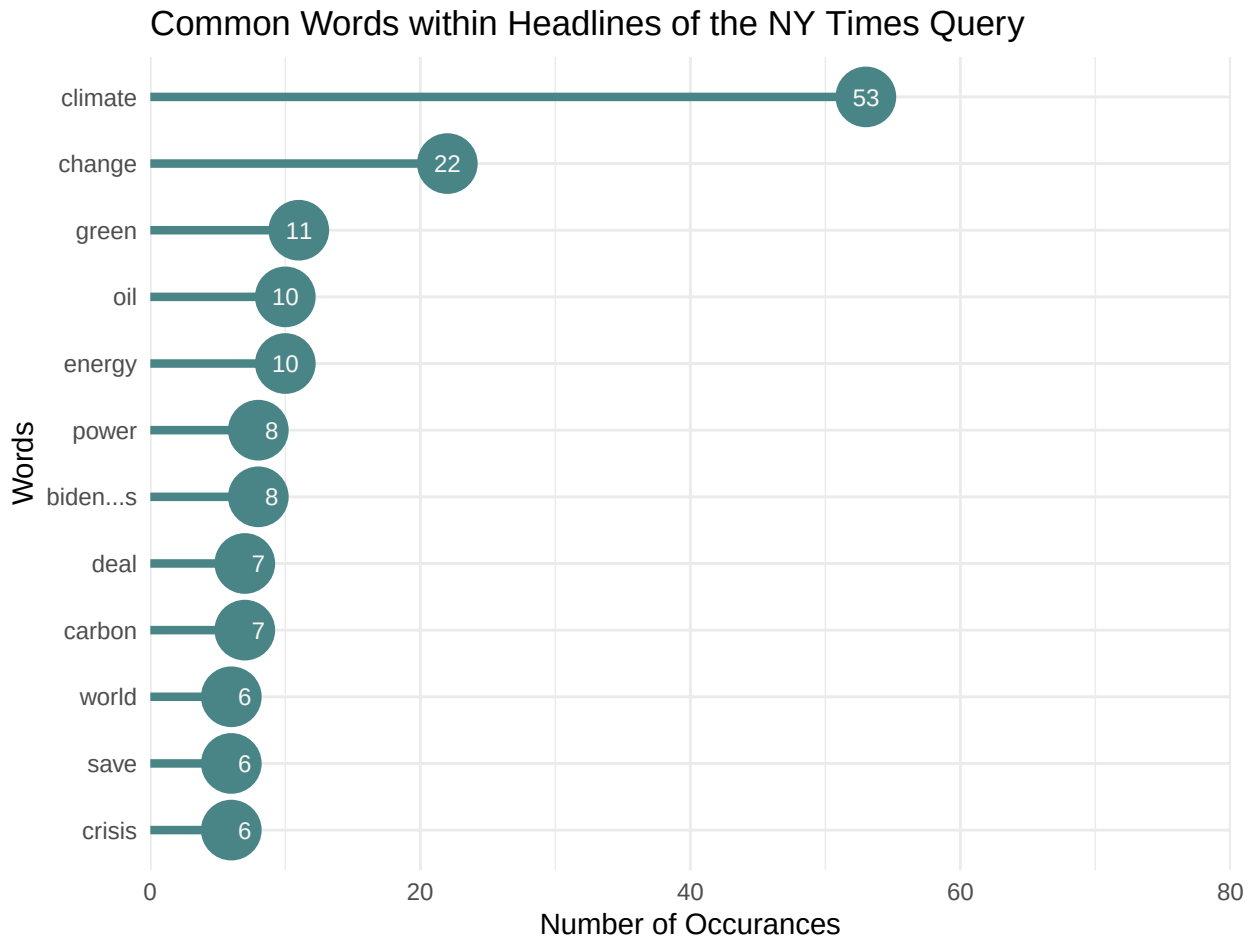


4) Recreating Frequency Plots with Headlines Variable

```
# saving headlines column name to use
headlines <- names(df_filtered)[21]

# unnesting tokens from the tidytext:: package (turning each word to an observation)
tokenized_headlines <- df_filtered |>
  unnest_tokens(word, headlines)

# joining stop words to the tokenized dataframe to remove these words
tokenized_headlines <- tokenized_headlines |>
  anti_join(stop_words, by = 'word')
```

```
# plotting the headlines
p_headline <- tokenized_headlines |>
  count(word, sort = TRUE) |>
  filter(n > 5) |>
  mutate(word = reorder(word, n)) |>
  ggplot(aes(n, word)) +
  geom_point(color = "#498587", size = 10) +
  geom_segment(aes(x = 0, xend = n, y = word, yend = word),
```

```
              color = "#498587",
              size = 1.5) +
  geom_text(aes(x = n + 2, y = word, label = n),
            vjust = 0.5,
            hjust = 1.5,
            size = 3,
            color = "#faf7f5") +
  labs(y = NULL) +
  theme_minimal() +
  labs(x = "Number of Occurances",
       y = "Words",
       title = "Common Words within Headlines of the NY Times Query") +
  scale_x_continuous(limits = c(0, 80), expand = c(0, 0))


p_headline
```

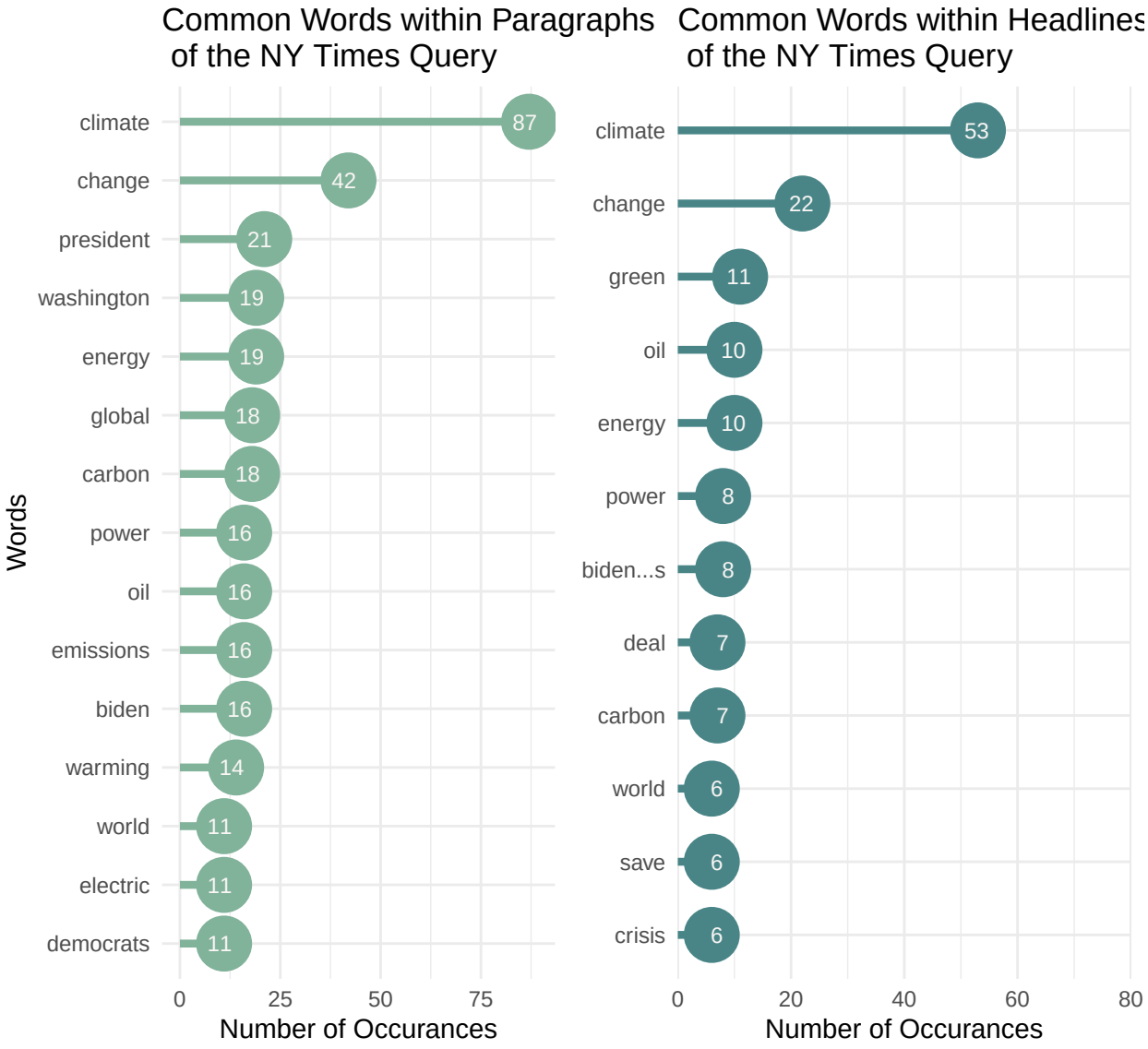


Comparing the Graphs

Answer: We can see that the top two words 'climate' and 'change' take the first and second place for most frequent words within both the paragraph text and headlines. It is interesting to see that the headlines

tend to be a more emotion-driven with words like “save”, “crisis”, and “world.” While the top words for the paragraph text are more jargon-focused with “emissions” and “global”.

```
# updating minor aesthetics for the comparison graph
p_paragraph2 <- clean_tokens_join |>
  count(word, sort = TRUE) |>
  filter(n > 10) |>
  mutate(word = reorder(word, n)) |>
  ggplot(aes(n, word)) +
  geom_point(color = "#81b29a", size = 10) +
  geom_segment(aes(x = 0, xend = n, y = word, yend = word),
               color = "#81b29a",
               size = 1.5) +
  geom_text(aes(x = n + 2, y = word, label = n),
            vjust = 0.5,
            hjust = 1,
            size = 3,
            color = "#faf7f5") +
  labs(y = NULL) +
  theme_minimal() +
  labs(x = "Number of Occurances",
                y = "Words",
                title = "Common Words within Paragraphs \n of the NY Times Query")

# updating minor aesthetics for the comparison graph
p_headline2 <- tokenized_headlines |>
  count(word, sort = TRUE) |>
  filter(n > 5) |>
  mutate(word = reorder(word, n)) |>
  ggplot(aes(n, word)) +
  geom_point(color = "#498587", size = 10) +
  geom_segment(aes(x = 0, xend = n, y = word, yend = word),
               color = "#498587",
               size = 1.5) +
  geom_text(aes(x = n + 2, y = word, label = n),
            vjust = 0.5,
            hjust = 1,
            size = 3,
            color = "#faf7f5") +
  labs(y = NULL) +
  theme_minimal() +
  labs(x = "Number of Occurances",
       title = "Common Words within Headlines \n of the NY Times Query") +
  scale_x_continuous(limits = c(0, 80), expand = c(0, 0))

ggarrange(p_paragraph2, p_headline2,
          ncol = 2, nrow = 1)
```


Common Words within Paragraphs of the NY Times Query
Words
climate
change
president
washington
energy
global
carbon
power
oil
emissions
biden
warming
world
electric
democrats
87
42
21
19
19
18
18
16
16
16
16
14
11
11
11
0
25
50
75
Number of Occurances
Common Words within Headlines of the NY Times Query
climate
change
green
oil
energy
power
biden...s
deal
carbon
world
save
crisis
53
22
11
10
10
8
8
7
7
6
6
6
0
20
40
60
80
Number of Occurances