Lab 1

Felicia Cruz

4/13/2022

```
library(jsonlite) #convert results from API queries into R-friendly formats
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()  masks stats::filter()
## x purrr::flatten() masks jsonlite::flatten()
## x dplyr::lag()     masks stats::lag()
```

```
library(tidytext) #text data management and analysis
library(ggplot2) #plot word frequencies and publication dates
```

```
# query NYT for the word "decarbonization"
t <- fromJSON("http://api.nytimes.com/svc/search/v2/articlesearch.json?q=decarbonization&api-key=e2lN9c
# key = e2lN9cy24RayOLq5H5wASYsn1vBUwtGt

t <- t %>%
  data.frame()

# class(t) # what is it?
# dim(t) # how big is it?
# names(t) # what variables are we working with?
```

Narrowing Down the Query

```
term <- "decarbonization"
begin_date <- "20210101" # chose the entire year 2021
end_date <- "20220101"

# 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=","e2lN9cy24Ray0Lq5H5wASYsn1vBUwtGt", sep="")

# examine the query url
# 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)
}
```

```
## Retrieving page 0

## Retrieving page 1

## Retrieving page 2

## Retrieving page 3

## Retrieving page 4

## Retrieving page 5

## Retrieving page 6

## Retrieving page 7

## Retrieving page 8

## Retrieving page 9
```

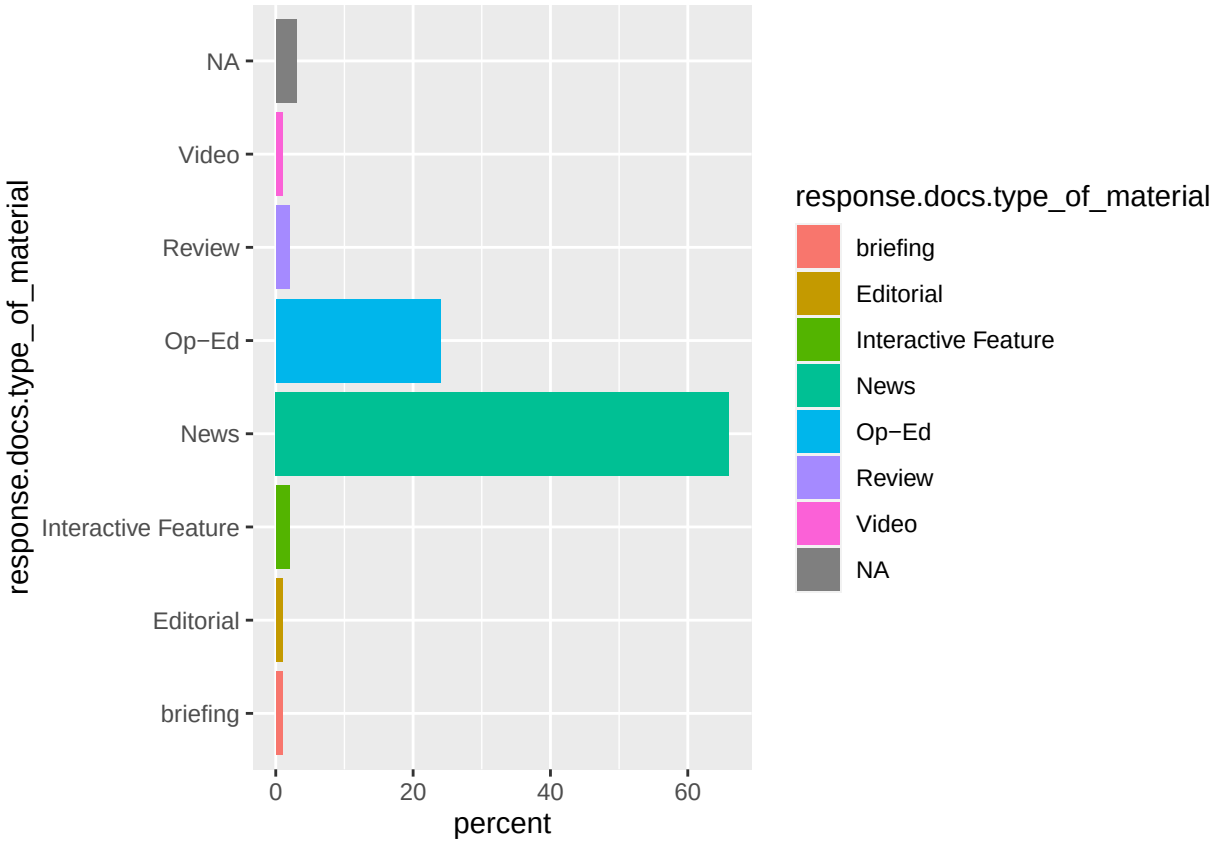
```
class(nytSearch)
```

```
## [1] "data.frame"
```

```
nytDat <- rbind_pages(pages) %>%
  as.tibble() # need to bind the pages and create a tibble from nytDa
```

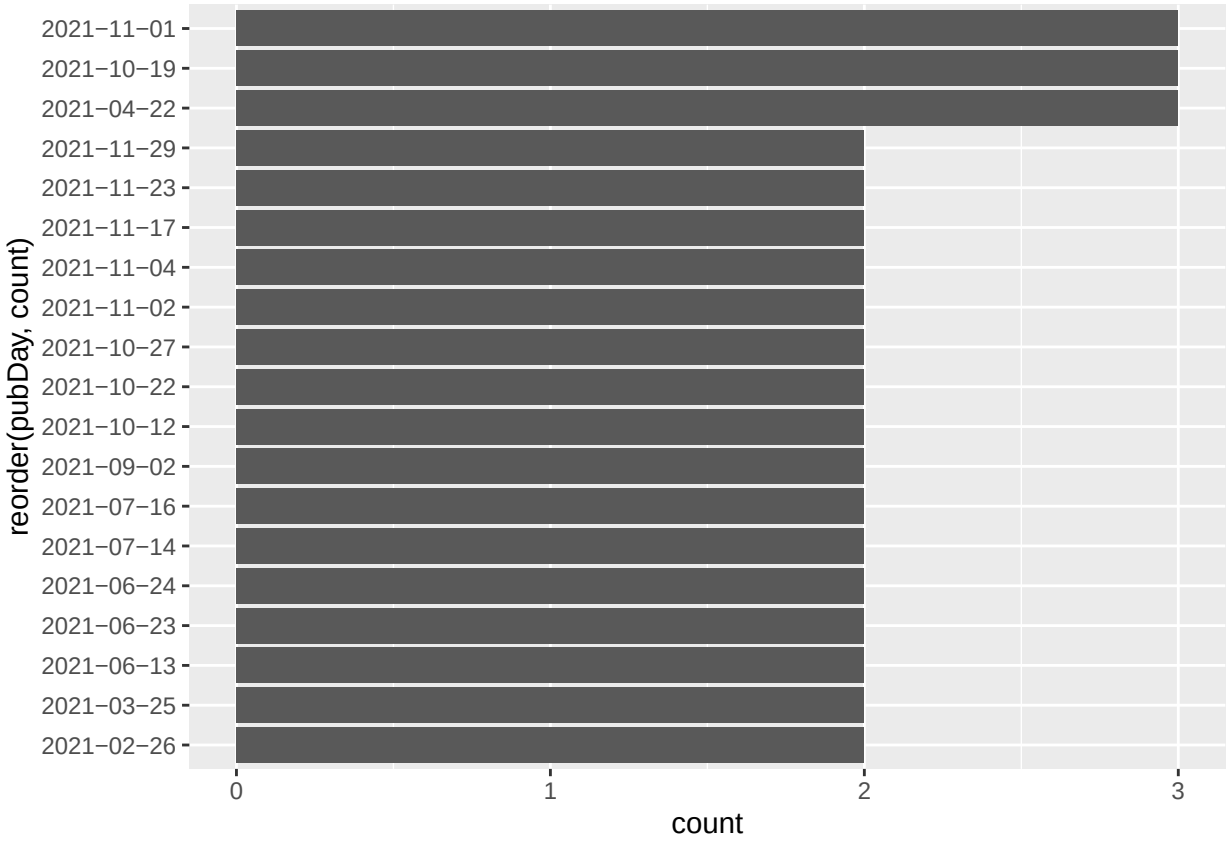
```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
# look at the types of material where the word comes up

nytDat %>%
  group_by(response.docs.type_of_material) %>%
  summarize(count=n()) %>%
  mutate(percent = (count / sum(count))*100) %>%
  ggplot() +
  geom_bar(aes(y=percent, x=response.docs.type_of_material, fill=response.docs.type_of_material), stat =
```



Publications per Day

```
nytDat %>%
  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") + coord_flip()
```



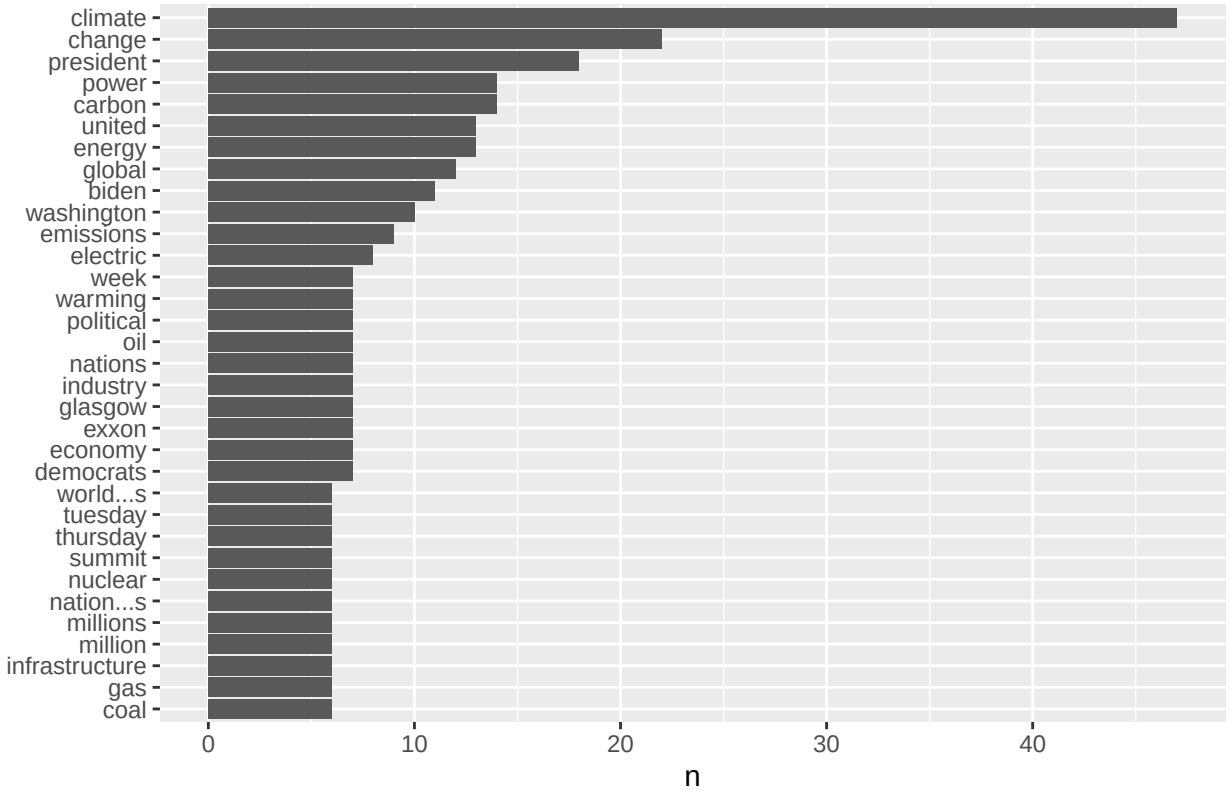
Word Frequency Plots

Using the lead paragraph variable

```
paragraph <- names(nytDat)[6] # The 6th column is "response.doc.lead_paragraph"
tokenized <- nytDat %>%
  unnest_tokens(word, paragraph)

data(stop_words)

tokenized <- tokenized %>%
  anti_join(stop_words) # remove all stop words
```

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

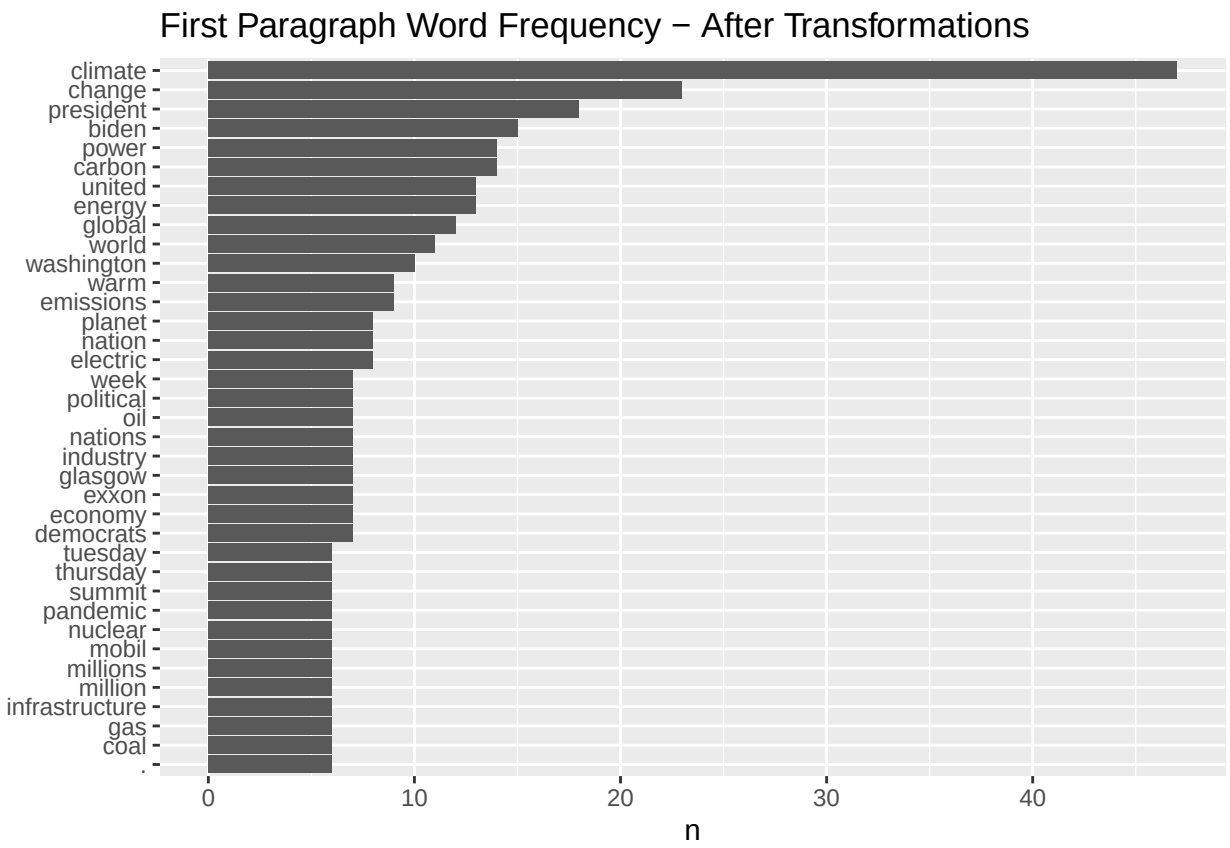
```
tokenized %>%
  count(word, sort = TRUE) %>%
  filter(n > 5) %>% # illegible with all the words displayed
  mutate(word = reorder(word, n)) %>% # show most frequent words at the top
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = NULL,
       title = "First Paragraph Word Frequency - Before Transformations")
```




```
# inspect tokens list
# tokenized$word

# stem words, warming or warm will be common
clean_tokens <- str_replace_all(tokenized$word,"warm[a-z,A-Z]*","warm")

# remove all numbers
clean_tokens <- str_replace_all(clean_tokens,"temperatur[a-z,A-Z]*","temperature")

clean_tokens <- str_remove_all(clean_tokens, "[:digit:]")

# remove 's wherever it occur
clean_tokens <- gsub("'s", '', clean_tokens)


tokenized$clean <- clean_tokens
```

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

#reassign
tokenized <- tib

# plot the word frequencies
tokenized %>%
  count(clean, sort = TRUE) %>%
```

```
  filter(n > 5) %>% #illegible with all the words displayed
  mutate(clean = reorder(clean, n)) %>%
  ggplot(aes(n, clean)) +
  geom_col() +
  labs(y = NULL,
       title = "First Paragraph Word Frequency - After Transformations")
```



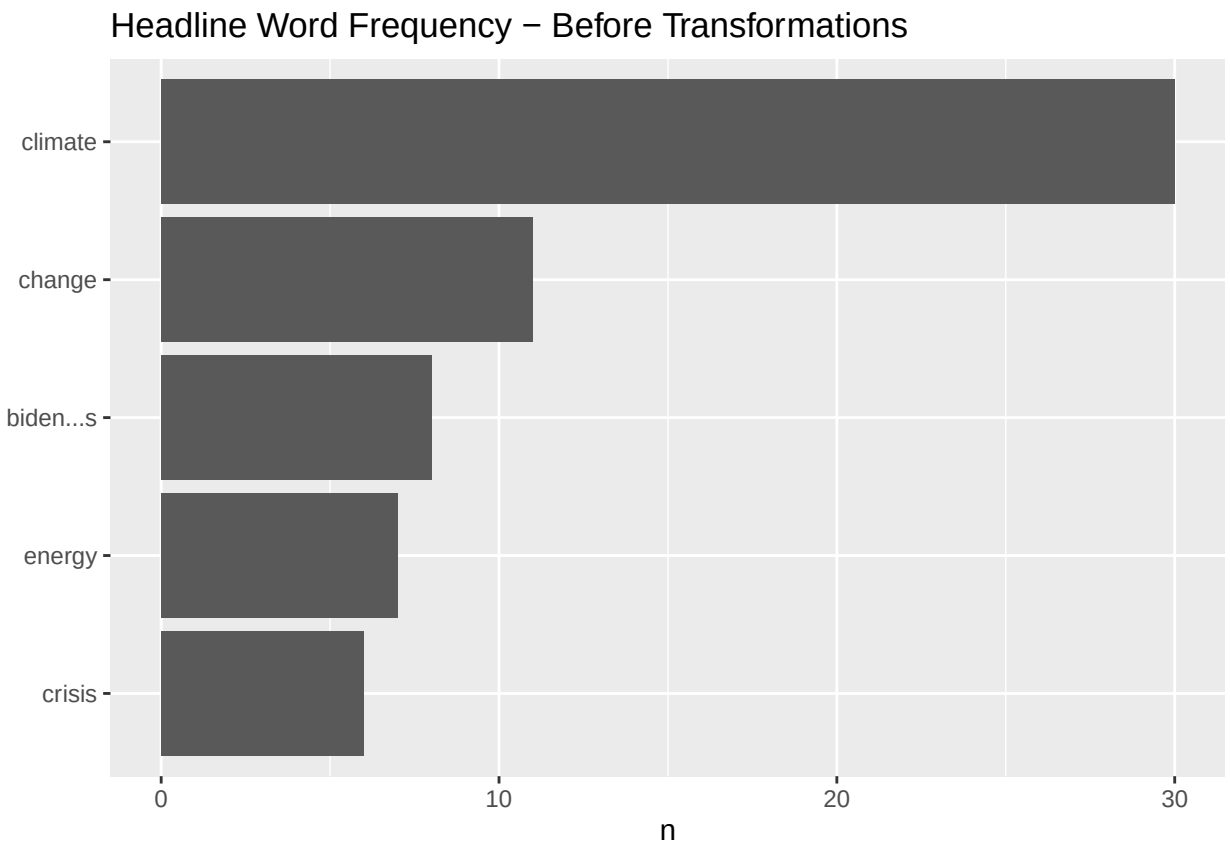
Using the headlines variable

```
headline <- names(nytDat)[21] # use the 21st column, "response.doc.headline.main"

tokenized_headline <- nytDat %>%
  unnest_tokens(word, headline)

tokenized_headline <- tokenized_headline %>%
  anti_join(stop_words) # remove all stop words
```

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

```
tokenized_headline %>%
  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,
       title = "Headline Word Frequency - Before Transformations")
```

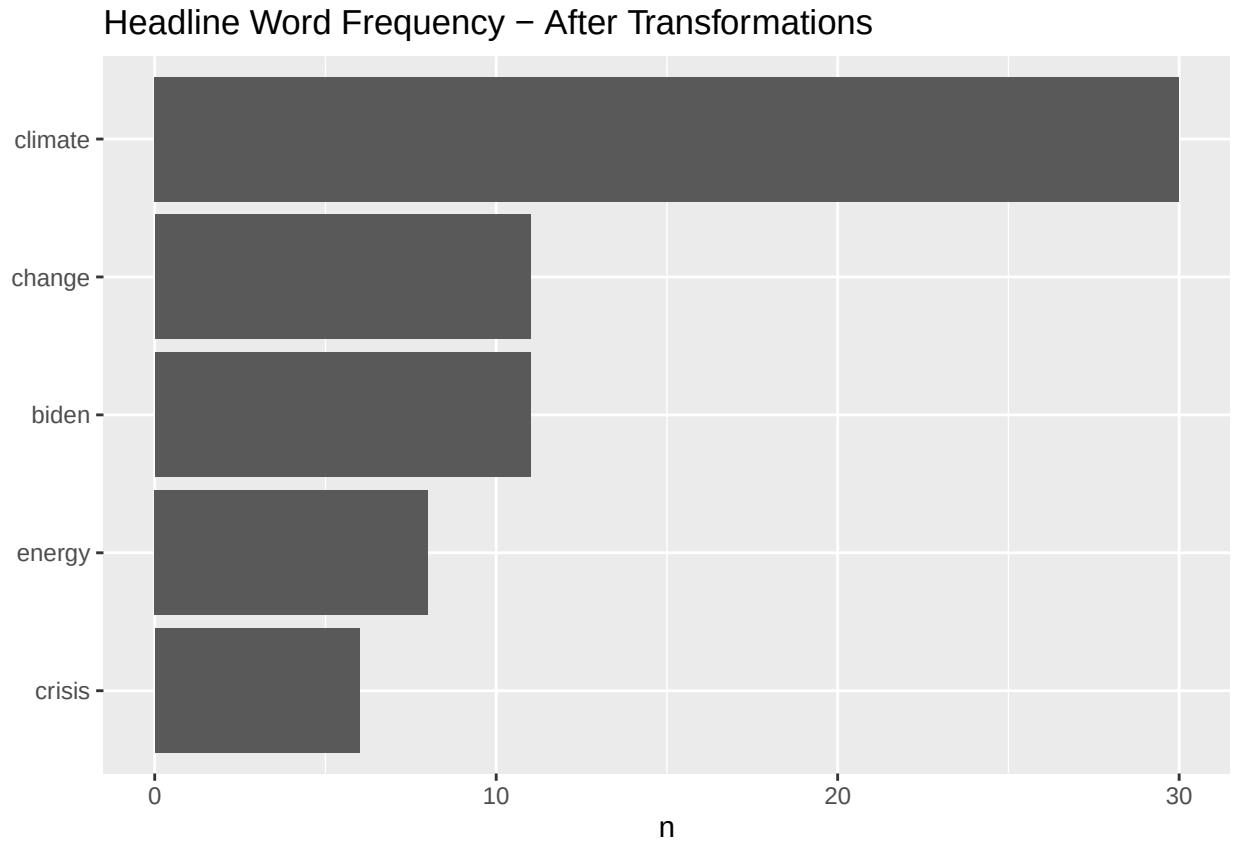



```
# inspect token list
# tokenized_headline$word
```

```
# stem words, warming or warm will be common
clean_tokens <- str_replace_all(tokenized_headline$word,"warm[a-z,A-Z]*","warm")

clean_tokens <- str_replace_all(clean_tokens,"temperatur[a-z,A-Z]*","temperature")

clean_tokens <- str_replace_all(clean_tokens, "fuel[a-z, A-Z]*", "fuel")

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

# remove 's wherever it occur
clean_tokens <- gsub("'s", '', clean_tokens)
```

```
tokenized_headline$clean <- clean_tokens
```

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

# reassign
tokenized_headline <- tib

# plot
tokenized_headline %>%
  count(clean, sort = TRUE) %>%
  filter(n > 5) %>% # illegible with all the words displayed
  mutate(clean = reorder(clean, n)) %>%
  ggplot(aes(n, clean)) +
  geom_col() +
  labs(y = NULL,
       title = "Headline Word Frequency - After Transformations")
```



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

For both the first paragraphs and headlines, the top 2 words, “climate” and “change” are the same. The words “biden” and “energy” are also frequent words for both first paragraphs and headlines, but the 5th most frequent word in headlines, “crisis” does not even appear in the plot for the first paragraph. Additionally, after removing stop words and doing some transformations to the headlines dataframe, there are only 5 words that appear in the plot, compared to over 30 for the first paragraph. In both instances, “climate” is much more frequent than all the other words.