Assignment 1

Paloma Cartwright

2022-04-11

Contents

Set-Up

```
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
```

Querying the NY Times for the word "hurricane"

This assignment I will be querying the word hurricane to see what coverage there is. I also want to see how prevalent The Bahamas is in the main search that I do which is for a date range where a major hurricane impacted the island I live on.

Practicing working with text strings from a query

```
#create an object called t with the results of our query ("hurricanes")
#from JSON flattens the JSON object
t <- fromJSON("http://api.nytimes.com/svc/search/v2/articlesearch.json?q=hurricane&api-key=mIFzdD9Pmlmb

#convert to a data frame
t <- t %>%
  data.frame()
```

```
#what variables are we working with?
names(t)

t$response.docs.snippet[2]

#assign a snippet to x to use as fodder for stringr functions.
x <- "The island's fragile electrical grid was slowly recovering from its latest loss of power, potentia

tolower(x)
str_split(x, ','); str_split(x, 't')
str_replace(x, 'potentially', '')
str_replace(x, ' ', '_') #first one
str_replace_all(x,' ', '_') # all of them
str_detect(x, 't'); str_detect(x, 'tive') ### is pattern in the string? T/F
str_locate(x, 't'); str_locate_all(x, 'as')
```

Publication Count Graphs

I will be querying the api for the word "hurricane" from September 15th 2015 - October 15th 2015. This was the year when Hurricane Joaquin destroyed my home and island so I'm curious about the New York Times' coverage of that.

```
term <- "hurricane"
begin_date <- "20150915"
end_date <- "20151015"

#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=", "mIFzdD9Pmlmb2hMFkZTol7ObtrmKYlQp",
                  sep="")

#examine our query url
baseurl
```

```
## [1] "http://api.nytimes.com/svc/search/v2/articlesearch.json?q=hurricane&begin_date=20150915&end_dat
```

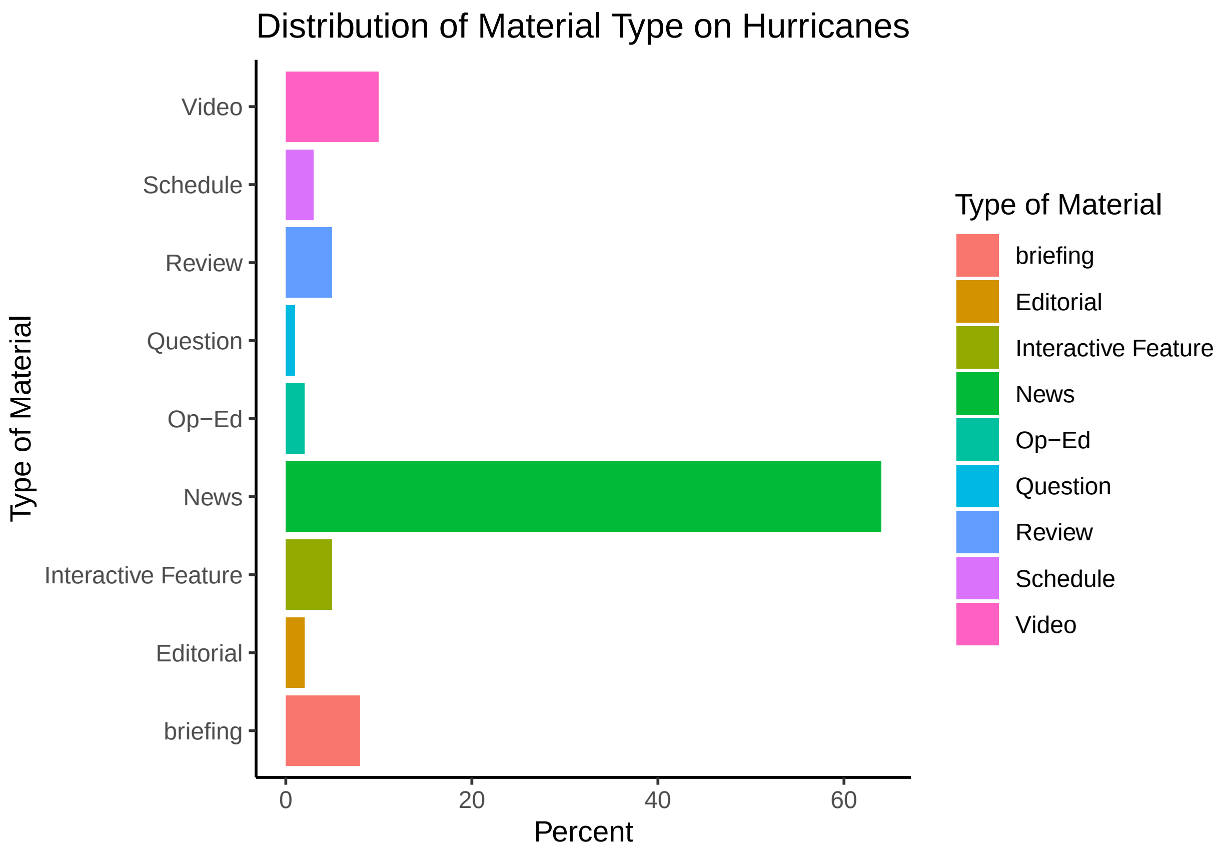
```
#this code allows for obtaining multiple pages of query results
initialQuery <- fromJSON(baseurl)
maxPages <- round((initialQuery$response$meta$hits[1] / 10)-1)

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)
}
class(nytSearch)
```

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

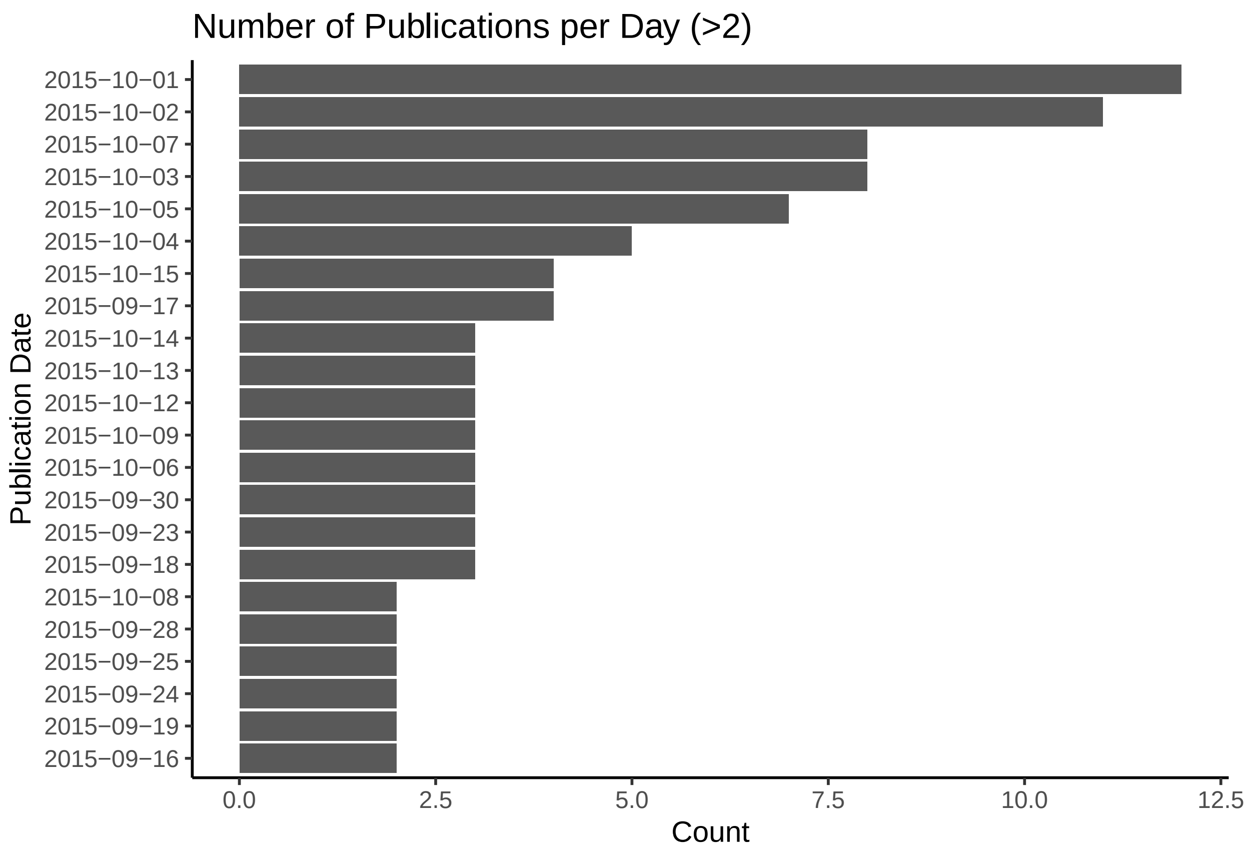
```
#need to bind the pages and create a tibble from nytDa
nytDat <- rbind_pages(pages)
dim(nytDat)
```

```
## [1] 100  33
```

```
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 = "identity") +
  coord_flip() +
  labs(y = "Percent",
       x = "Type of Material",
       title = "Distribution of Material Type on Hurricanes",
       fill = "Type of Material") +
  theme_classic()
```



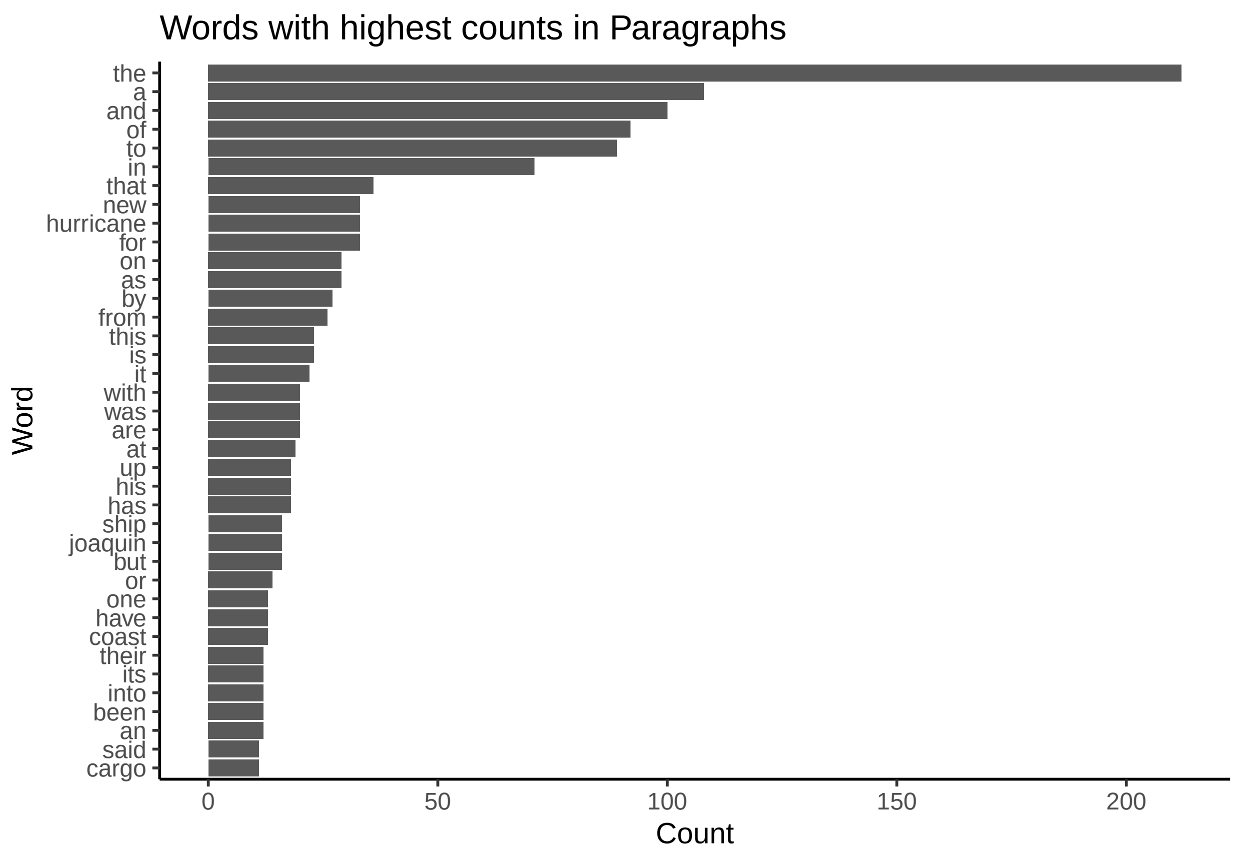
```
nytDat %>%
  mutate(pubDay=gsub("T.*","",response.docs.pub_date)) %>%
  group_by(pubDay) %>%
  summarise(count=n()) %>%
  filter(count >= 2) %>%
  ggplot() +
  geom_bar(aes(x=reorder(pubDay, count), y=count), stat="identity") +
  coord_flip() +
  labs(x = "Publication Date",
       y = "Count",
       title = "Number of Publications per Day (>2)") +
  theme_classic()
```



Querying Paragraphs

```
# names(nytDat)

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

# tokenized[, 34]
```

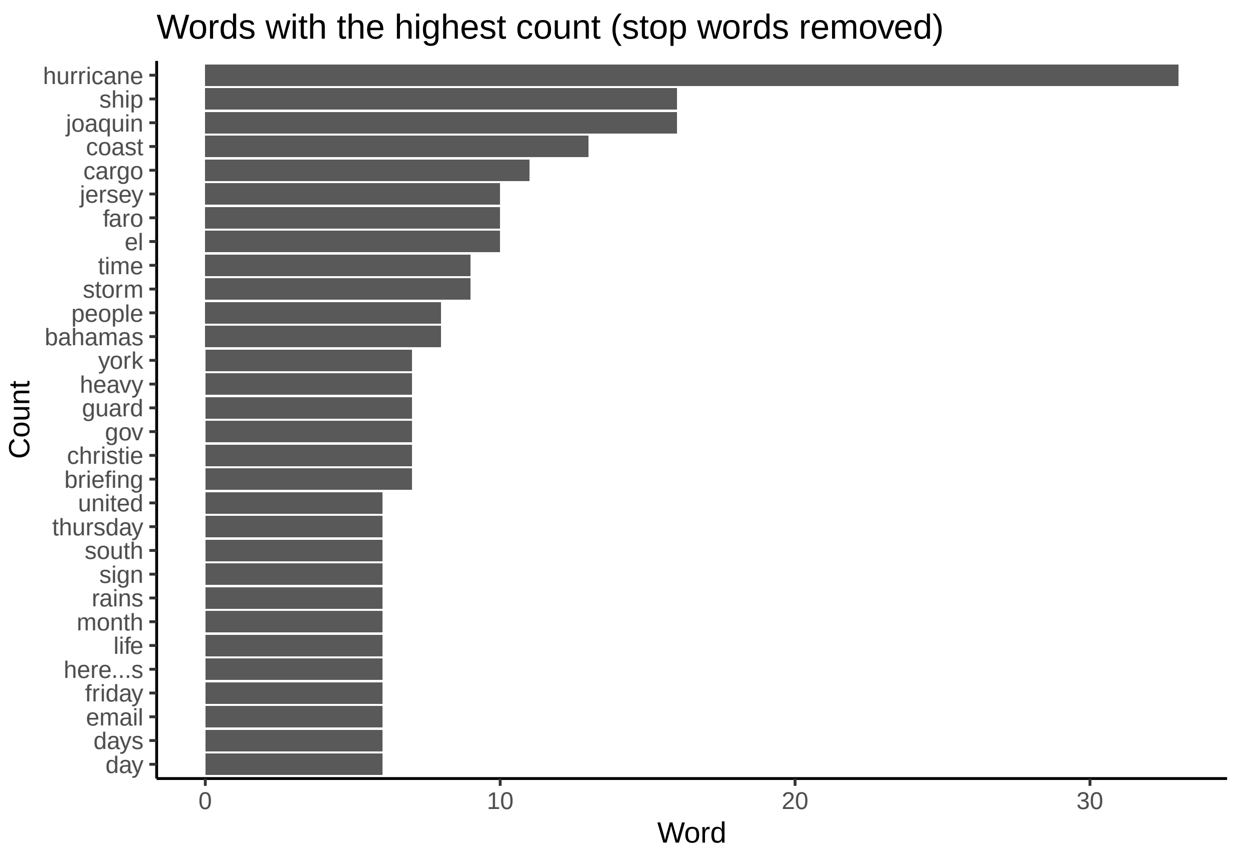
```
tokenized %>%
  count(word, sort = TRUE) %>%
  filter(n > 10) %>% #illegible with all the words displayed
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = "Word",
       x = "Count",
       title = "Words with highest counts in Paragraphs") +
  theme_classic()
```



```
tokenized <- tokenized %>%
  anti_join(stop_words)
```

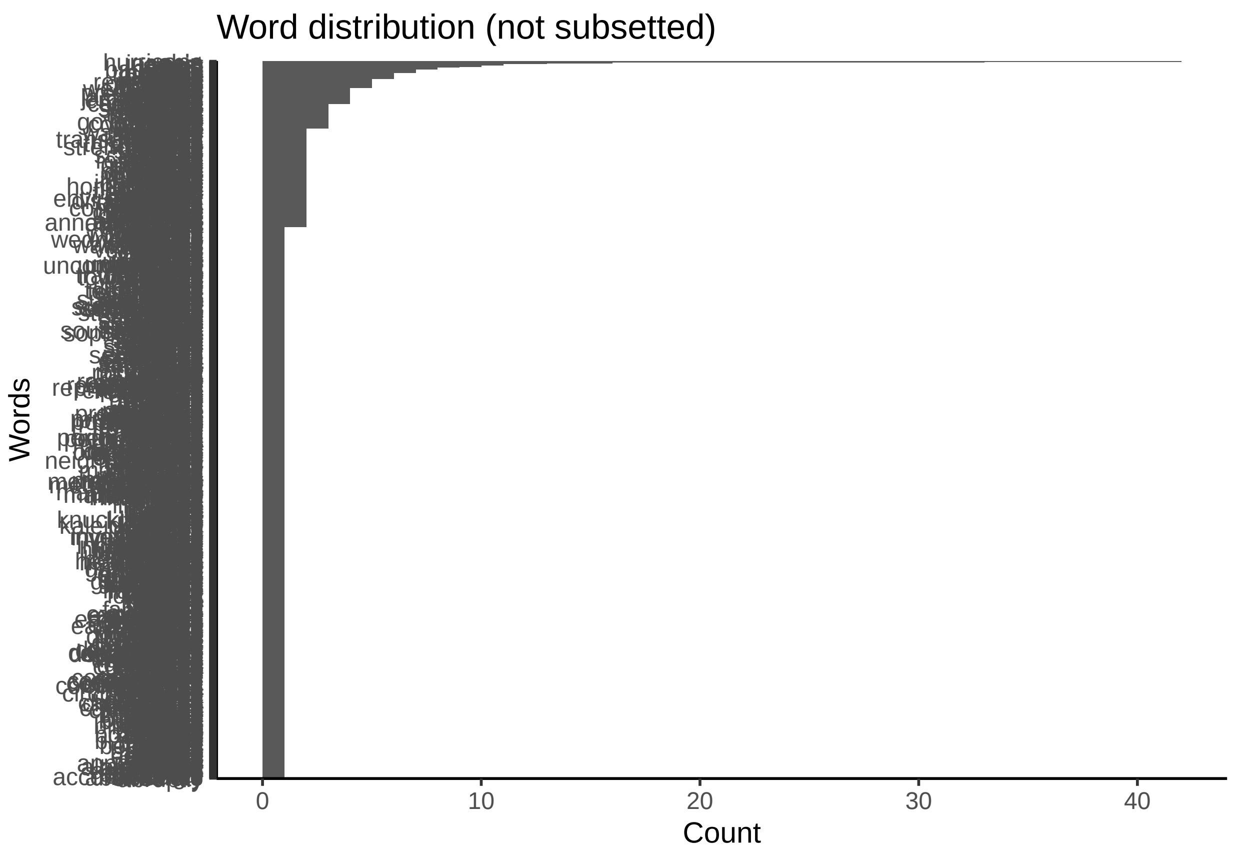
```
## Joining, by = "word"
```

```
tokenized %>%
  count(word, sort = TRUE) %>%
  filter(n > 5) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = "Count",
       x = "Word",
       title = "Words with the highest count (stop words removed)") +
  theme_classic()
```

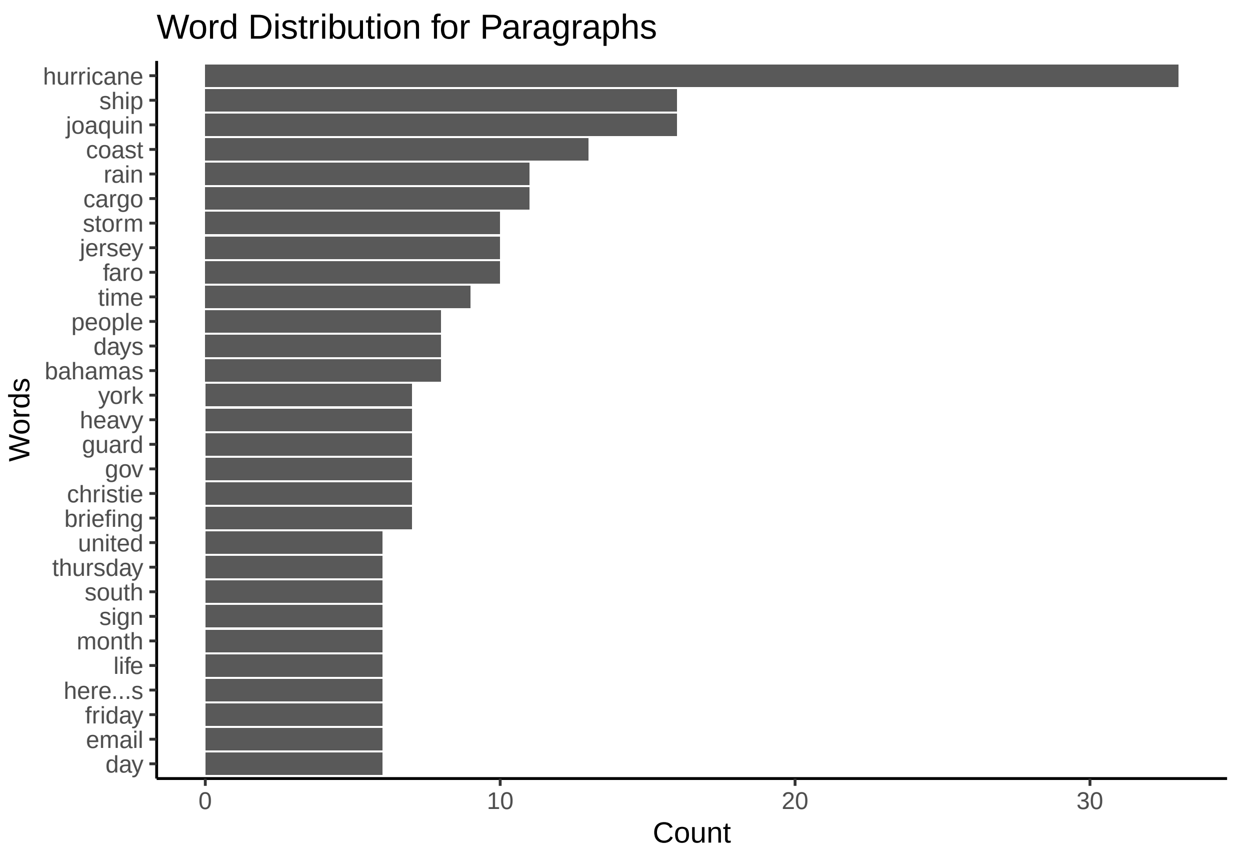


# Cleaning the paragraphs data

```
clean_tokens <- str_remove_all(tokenized$word, "[:digit:]")
clean_tokens <- str_remove_all(clean_tokens, "el")
clean_tokens <- str_replace_all(clean_tokens, "rain[a-z,A-Z]*", "rain")
clean_tokens <- gsub("'s", '', clean_tokens)

tokenized$clean <- clean_tokens

tokenized %>%
  count(clean, sort = TRUE) %>%
  mutate(clean = reorder(clean, n)) %>%
  ggplot(aes(n, clean)) +
  geom_col() +
  labs(y = "Words",
       x = "Count",
       title = "Word distribution (not subsetted)") +
  theme_classic()
```



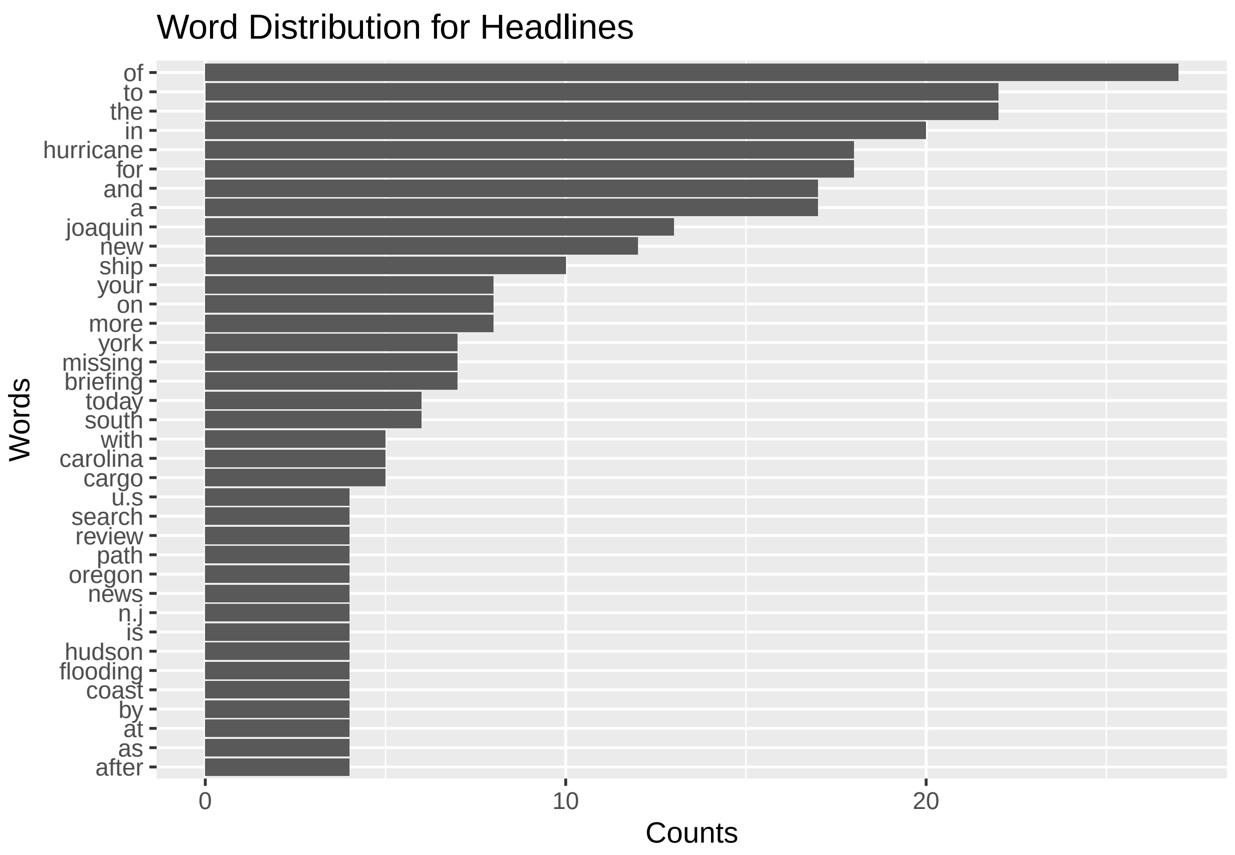
```r
#remove the empty strings
tokenized <- subset(tokenized, clean!="")

#try again
tokenized %>%
  count(clean, sort = TRUE) %>%
  filter(n > 5) %>%
  mutate(clean = reorder(clean, n)) %>%
  ggplot(aes(n, clean)) +
  geom_col() +
  labs(y = "Words",
       x = "Count",
       title = "Word Distribution for Paragraphs") +
  theme_classic()
```



Querying Headlines

```
names(nytDat)

headline <- names(nytDat)[21] #The 21st column, "response.doc.headline_main", is the one we want here.
tokenized_headline <- nytDat %>%
  unnest_tokens(word, headline)

#tokenized_headline[, 34]
```

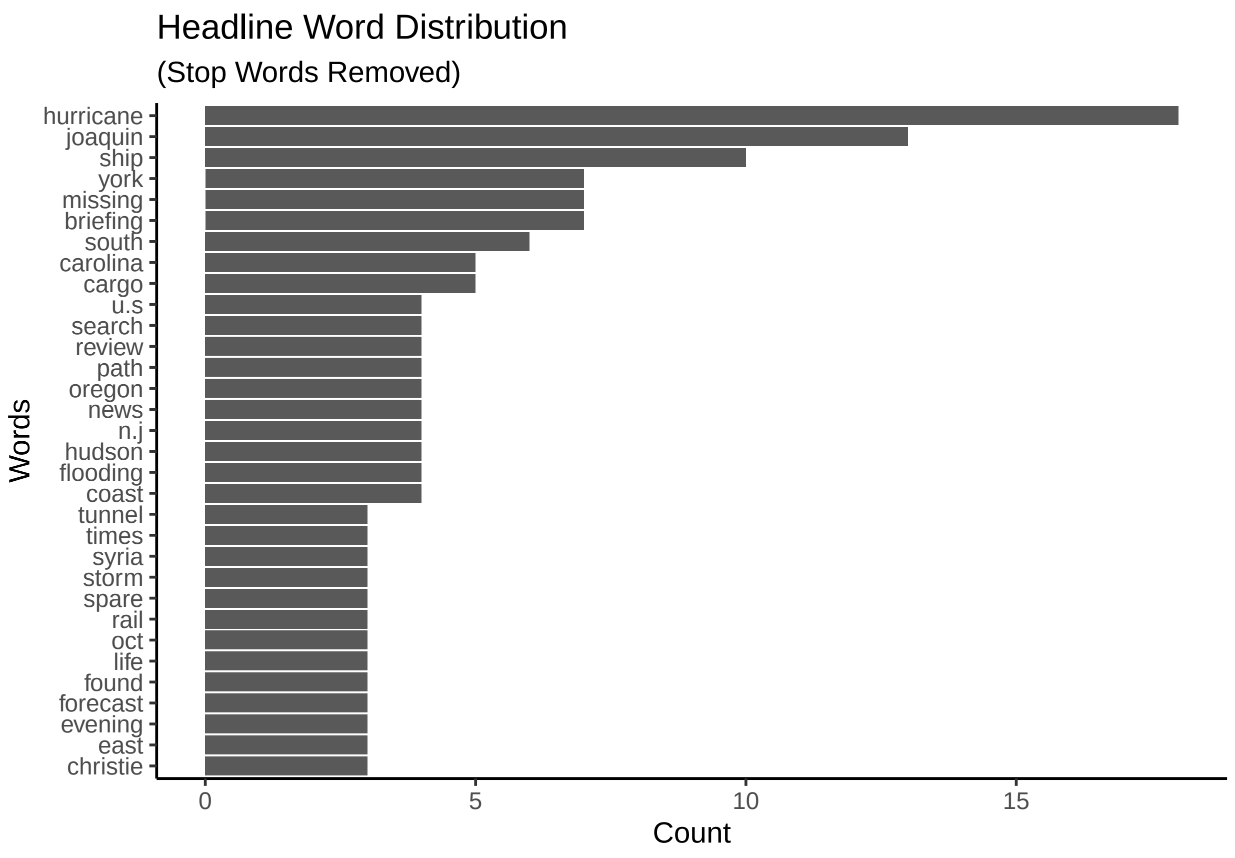
```
tokenized_headline %>%
  count(word, sort = TRUE) %>%
  filter(n > 3) %>% #illegible with all the words displayed
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = "Words",
       x = "Counts",
       title = "Word Distribution for Headlines")
```



```
tokenized_headline <- tokenized_headline %>%
  anti_join(stop_words)
```

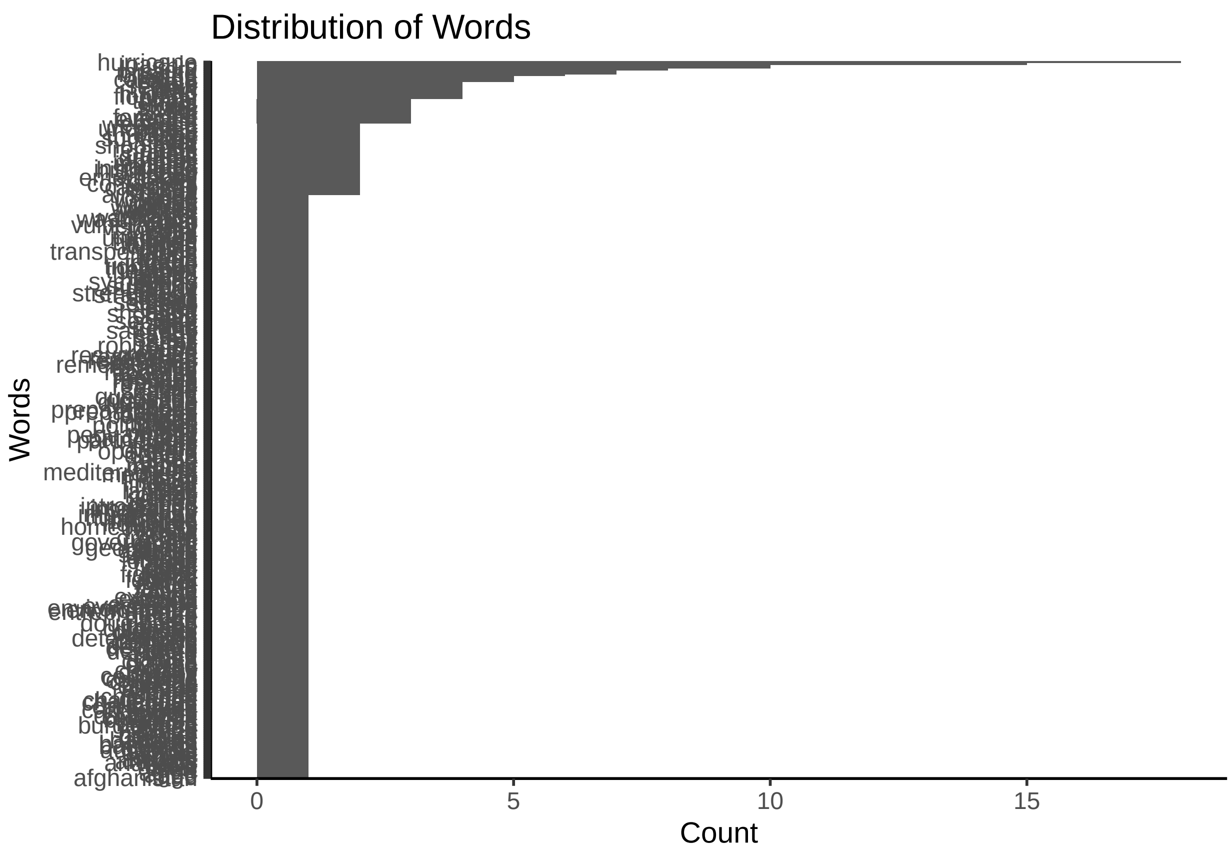
```
## Joining, by = "word"
```

```
tokenized_headline %>%
  count(word, sort = TRUE) %>%
  filter(n > 2) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = "Words",
       x = "Count",
       title = "Headline Word Distribution",
       subtitle = "(Stop Words Removed)") +
  theme_classic()
```

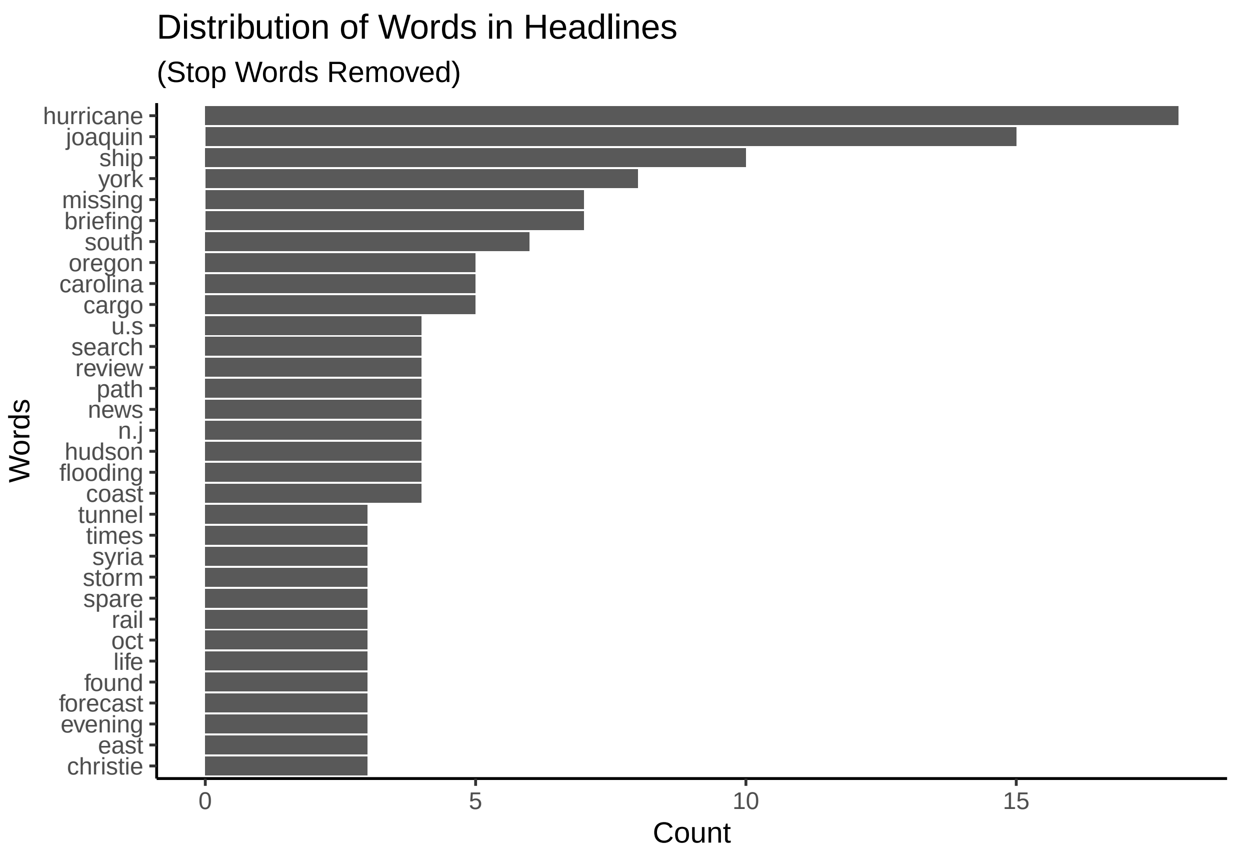


Cleaning the headlines data

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

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

clean_tokens_headline <- gsub("’s", '', clean_tokens_headline)

tokenized_headline$clean <- clean_tokens_headline

tokenized_headline %>%
  count(clean, sort = TRUE) %>%
  mutate(clean = reorder(clean, n)) %>%
  ggplot(aes(n, clean)) +
  geom_col() +
  labs(y = "Words",
       x = "Count",
       title = "Distribution of Words") +
  theme_classic()
```



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

#try again
tokenized_headline %>%
  count(clean, sort = TRUE) %>%
  filter(n > 2) %>%
  mutate(clean = reorder(clean, n)) %>%
  ggplot(aes(n, clean)) +
  geom_col() +
  labs(y = "Words",
       x = "Count",
       title = "Distribution of Words in Headlines",
       subtitle = "(Stop Words Removed)") +
  theme_classic()
```



Discussion

The biggest difference between the tokenized paragraphs and the tokenized headlines is the change in frequencies of words with paragraphs having hurricanes appear over 30 times but headlines only saw hurricane appear over 15 times. This would be due to the increased amount of content in a paragraph when comparing that with a headline.