HW_3

AUTHOR
Tyler Gallagher

Homework 3

```
file_path <- "/Users/TylerGallagher13/Desktop/pubmed.csv"
abstracts <- read.table(file_path, header = TRUE, sep = "\t")
```

```
Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec,
: EOF within quoted string
```

Question 1

```
library(tm)
```

```
Loading required package: NLP
```

```
library(tidytext)
library(dplyr)
```

```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(stopwords)
```

```

Attaching package: 'stopwords'

The following object is masked from 'package:tm':

    stopwords
```

```
abstracts_tokens <- abstracts %>%
  unnest_tokens(word, abstract.term)
top_words <- abstracts_tokens %>%
  count(word, sort = TRUE) %>%
  head(10)
print(top_words)
```

```
    word     n
1    the 28126
2     of 24760
3    and 19993
4     in 14653
5     to 10920
6  covid  8256
7      a  8245
8   with  8038
9     19  7080
10    is  5649
```

```
abstracts_tokens <- abstracts %>%
  unnest_tokens(word, abstract.term)
top_words <- abstracts_tokens %>%
  count(word, sort = TRUE) %>%
  head(10)
print(top_words)
```

```
    word     n
1    the 28126
2     of 24760
3    and 19993
4     in 14653
5     to 10920
6  covid  8256
7      a  8245
8   with  8038
9     19  7080
10    is  5649
```

```
stop_words <- data.frame(word = stopwords("en"))
abstracts_tokens_no_stop <- abstracts_tokens %>%
  anti_join(stop_words)
```

```
Joining with `by = join_by(word)`
```

```
top_words_no_stop <- abstracts_tokens_no_stop %>%
  count(word, sort = TRUE) %>%
  head(10)
print(top_words_no_stop)
```

```
           word    n
1         covid 8256
2            19 7080
```

```
3        cancer 4786
4      patients 4684
5      prostate 4619
6  preeclampsia 2643
7       disease 2574
8           pre 2165
9     eclampsia 2005
10    treatment 1841
```

Before considering removal of stop words, the five most common words are the (28,126 observations), of (24,760), and (19,993), in (14,653), and to (10,920). This changes substantially after removing stop words. Now, the most common words are covid (8,256), 19 (7,080), cancer (4,786), patients (4,684), and prostate (4,619). It appears that this set of abstracts focuses on COVID-19, cancer, and the prostate.

Question 2

```
abstracts_bigrams <- abstracts %>%
  unnest_tokens(bigram, abstract.term, token = "ngrams", n = 2)
stop_words <- data.frame(word = stopwords("en"))
abstracts_bigrams_no_stop <- abstracts_bigrams %>%
  filter(!bigram %in% paste(stop_words$word, collapse = "|"))
top_bigrams_no_stop <- abstracts_bigrams_no_stop %>%
  count(bigram, sort = TRUE) %>%
  head(10)
print(top_bigrams_no_stop)
```

```
            bigram    n
1         covid 19 6969
2  prostate cancer 4009
3           of the 3883
4           in the 3418
5    pre eclampsia 1854
6    patients with 1587
7         of covid 1519
8  cystic fibrosis 1236
9          and the 1154
10          to the 1061
```

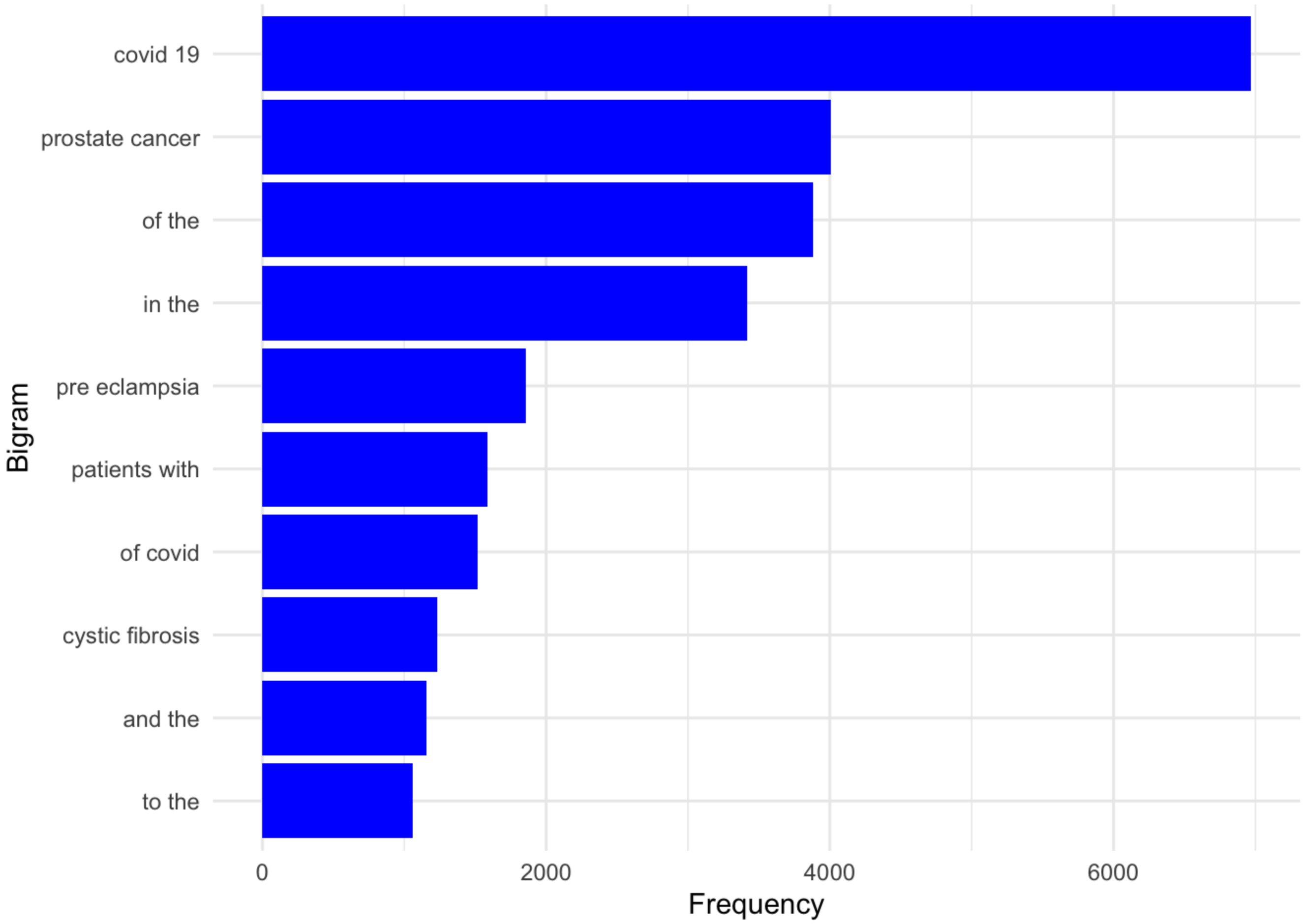
```
library(ggplot2)
```

```
Attaching package: 'ggplot2'

The following object is masked from 'package:NLP':

    annotate
```

```
ggplot(top_bigrams_no_stop, aes(x = reorder(bigram, n), y = n)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(x = "Bigram", y = "Frequency") +
  coord_flip() +
  theme_minimal()
```



Question 3

```
# Assuming you have a dataset named "abstracts" with a column "abstract.term"

# Tokenize the abstracts and remove stop words
abstracts_tokens <- abstracts %>%
  unnest_tokens(word, abstract.term) %>%
  anti_join(stop_words)
```

```
Joining with `by = join_by(word)`
```

```
# Calculate term frequency (TF)
tf <- abstracts_tokens %>%
  group_by(word) %>%
  summarise(tf = n())
```

```
# Calculate document frequency (DF)
df <- abstracts_tokens %>%
  distinct(word) %>%
  group_by(word) %>%
  summarise(df = n())

# Calculate inverse document frequency (IDF)
N <- nrow(abstracts)
idf <- df %>%
  mutate(idf = log(N / df))

# Calculate TF-IDF
tfidf <- tf %>%
  left_join(idf, by = "word") %>%
  mutate(tfidf = tf * idf)

# Find the top 5 tokens with the highest TF-IDF values
top_tokens <- tfidf %>%
  arrange(desc(tfidf)) %>%
  top_n(5)
```

```
Selecting by tfidf
```

```
# Print the result
print(top_tokens)
```

```
# A tibble: 5 × 5
  word        tf    df   idf  tfidf
  <chr>    <int> <int> <dbl>  <dbl>
1 covid     8256     1  7.70 63611.
2 19        7080     1  7.70 54550.
3 cancer    4786     1  7.70 36875.
4 patients  4684     1  7.70 36089.
5 prostate  4619     1  7.70 35589.
```

The TD-IDF demonstrates the relative overall importance and frequency of words weighted together. The five with the most value include covid (TF-IDF=63,611), 19 (54,550), cancer (36,875), patients (36,089), and prostate (35,589). Interestingly, the TF-IDF top-5 is the same top-5 in the same order as the individually tokenized words. Additionally, the TF-IDF values seem similar in scale to the raw n of the words.