Lab 6

AUTHOR
Breyonne Williams

Data Setup

```
knitr::opts_chunk$set(eval = FALSE, include  = TRUE)
library(readr)
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
```

```
mt_samples <- read_csv("https://raw.githubusercontent.com/USCbiostats/data-science-data/
```

```
New names:
• `` -> `...1`

Rows: 4999 Columns: 6
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (5): description, medical_specialty, sample_name, transcription, keywords
dbl (1): ...1

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.
```

```
mt_samples <- mt_samples %>%
  select(description, medical_specialty, transcription)

head(mt_samples)
```

```
# A tibble: 6 × 3
  description                                     medical_specialty transcription
  <chr>                                           <chr>             <chr>
1 A 23-year-old white female presents with comp… Allergy / Immuno… "SUBJECTIVE:…
2 Consult for laparoscopic gastric bypass.       Bariatrics        "PAST MEDICA…
3 Consult for laparoscopic gastric bypass.       Bariatrics        "HISTORY OF …
4 2-D M-Mode. Doppler.                           Cardiovascular / … "2-D M-MODE:…
5 2-D Echocardiogram                             Cardiovascular / … "1.  The lef…
6 Morbid obesity.  Laparoscopic antecolic anteg… Bariatrics        "PREOPERATIV…
```

Question 1

```
library(dplyr)
mt_samples %>%
  count(medical_specialty, sort = TRUE)
# surgery is the most repeated category among the medical specialties
```

Question 2

```
library(tm)
library(ggplot2)

corpus <- Corpus(VectorSource(mt_samples$transcription))

corpus <- tm_map(corpus, content_transformer(tolower))

corpus <- tm_map(corpus, removePunctuation)

corpus <- tm_map(corpus, removeNumbers)

corpus <- tm_map(corpus, removeWords, stopwords("english"))

corpus <- tm_map(corpus, stripWhitespace)

# from this we get shortened versions of what happnened, smiliar to short hand form when

library(Matrix)
dtm <- DocumentTermMatrix(corpus)
word_freq <- rowSums(as.matrix(dtm))

word_freq_df <- data.frame(word = names(word_freq), freq = word_freq)
word_freq_df <- word_freq_df[order(-word_freq_df$freq), ]

top_words <- head(word_freq_df, 20)
ggplot(top_words, aes(x = reorder(word, -freq), y = freq)) +
  geom_bar(stat = "identity", fill = "blue") +
  coord_flip() +
  labs(x = "Words", y = "Frequency", title = "Top 20 Most Frequent Words") +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 12))
```

Question 3

```
corpus <- tm_map(corpus, removeWords, stopwords("english"))
corpus <- tm_map(corpus, removeNumbers)

dtm <- DocumentTermMatrix(corpus)
word_freq <- rowSums(as.matrix(dtm))

word_freq_df <- data.frame(word = names(word_freq), freq = word_freq)
word_freq_df <- word_freq_df[order(-word_freq_df$freq), ]

top_words <- head(word_freq_df, 20)

ggplot(top_words, aes(x = reorder(word, -freq), y = freq)) +
  geom_bar(stat = "identity", fill = "blue") +
  coord_flip() +
  labs(x = "Words", y = "Frequency", title = "Top 20 Most Frequent Words (Stopwords and
  theme_minimal() +
  theme(axis.text.y = element_text(size = 12))
```

Question 4

```
bigram_tokenizer <- function(x) {
  unlist(lapply(ngrams(words(x), 2), paste, collapse = " "))
}
corpus_bigrams <- Corpus(VectorSource(corpus))
corpus_bigrams <- tm_map(corpus_bigrams, content_transformer(tolower))
corpus_bigrams <- tm_map(corpus_bigrams, removePunctuation)
corpus_bigrams <- tm_map(corpus_bigrams, removeNumbers)
corpus_bigrams <- tm_map(corpus_bigrams, removeWords, stopwords("english"))
corpus_bigrams <- tm_map(corpus_bigrams, stripWhitespace)
corpus_bigrams <- tm_map(corpus_bigrams, content_transformer(bigram_tokenizer))

trigram_tokenizer <- function(x) {
  unlist(lapply(ngrams(words(x), 3), paste, collapse = " "))
}
corpus_trigrams <- Corpus(VectorSource(corpus))
corpus_trigrams <- tm_map(corpus_trigrams, content_transformer(tolower))
corpus_trigrams <- tm_map(corpus_trigrams, removePunctuation)
corpus_trigrams <- tm_map(corpus_trigrams, removeNumbers)
corpus_trigrams <- tm_map(corpus_trigrams, removeWords, stopwords("english"))
corpus_trigrams <- tm_map(corpus_trigrams, stripWhitespace)
corpus_trigrams <- tm_map(corpus_trigrams, content_transformer(trigram_tokenizer))

# the bi gram gives very little information outside the race and sex of the person, the
```

Question 5

```
count_words_around_target <- function(corpus, female) {

  before_counts <- numeric(0)
  after_counts <- numeric(0)


  for (doc in corpus) {
    tokens <- unlist(strsplit(as.character(doc$content), " "))


    target_positions <- which(tokens == target_word)


    for (position in target_positions) {
      if (position > 1) {
        before_word <- tokens[position - 1]
        before_counts <- append(before_counts, before_word)
      }
      if (position < length(tokens)) {
        after_word <- tokens[position + 1]
        after_counts <- append(after_counts, after_word)
      }
    }
  }}
```

Question 6

```
library(dplyr)
library(tm)

corpus <- Corpus(VectorSource(mt_samples))


corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords, stopwords("english"))
corpus <- tm_map(corpus, stripWhitespace)

corpus <- tm_map(corpus, content_transformer(tokenize))

top_words_5 <- head(corpus, 5)
```

Question 7

```
library(dplyr)
library(tm)

corpus <- Corpus(VectorSource(corpus$content))

corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removeWords, stopwords("english"))
corpus <- tm_map(corpus, stripWhitespace)

top_words_2 <- head(corpus, 2)
```