

Lab 6

Jazmin Hernandez

Lab 06 - Text Mining

```
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(ggplot2)
library(tidytext)
library(readr)
mt_samples <- read_csv("https://raw.githubusercontent.com/USCbiostats/data-science-data/master/mt_samples.csv")
```

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 x 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: What specialties do we have?

There are 40 different medical specialties. Specialties such as Cosmetic / Plastic Surgery and Dentistry both have 27 counts. Diets and Nutrition and Rheumatology specialties both have counts of 10. Autopsy and Lab Medicine - Pathology specialties both have counts of 8. There does not appear to be an even distribution between the medical specialties as we can see that Surgery has 1103 counts compared to Hospice - Palliative Care with only 6 counts.

```
med_specialty_counts <- mt_samples |>
  count(medical_specialty, name = "n", sort = TRUE)
print(med_specialty_counts)
```

```
# A tibble: 40 x 2
  medical_specialty      n
  <chr>              <int>
1 Surgery            1103
2 Consult - History and Phy.    516
3 Cardiovascular / Pulmonary    372
4 Orthopedic          355
5 Radiology           273
6 General Medicine      259
7 Gastroenterology      230
8 Neurology            223
9 SOAP / Chart / Progress Notes  166
```

```
10 Obstetrics / Gynecology          160
# i 30 more rows
```

```
overlap_counts <- mt_samples |>
rowwise() |>
mutate(num_specialties = sum(c_across(starts_with("specialty_")), na.rm = TRUE)) |>
count(num_specialties)

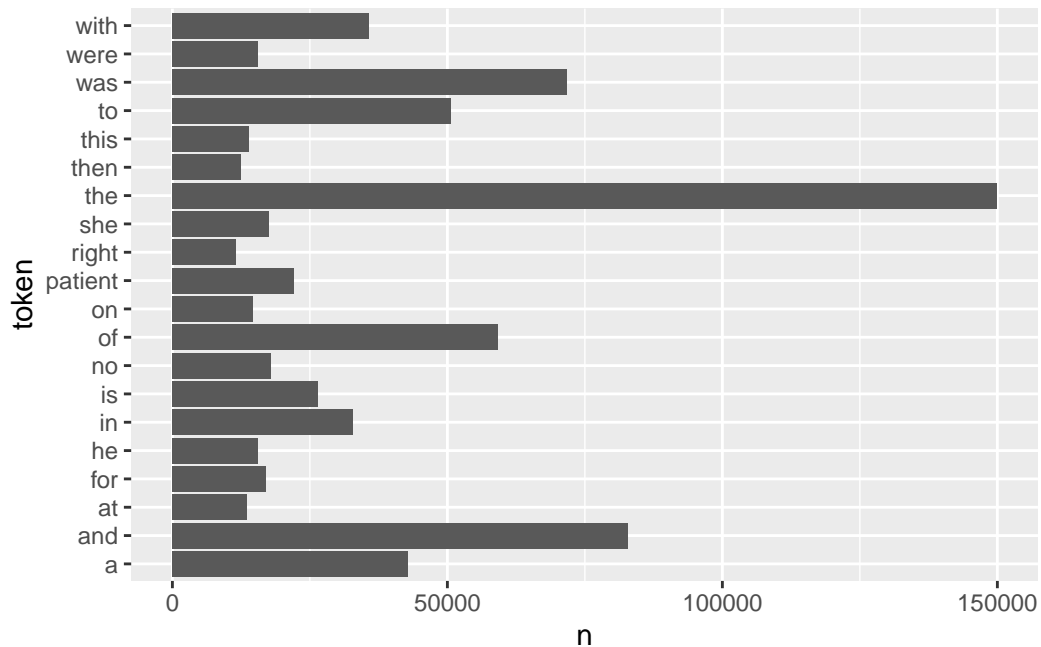
print(overlap_counts)
```

```
# A tibble: 1 x 2
# Rowwise:
  num_specialties      n
      <int> <int>
1             0 4999
```

Question 2

The list shows that the word “the” appears the most (149888 times) in the text. This makes sense because stop words usually appear the most in English text. Looking at the top tenth word that appears the most, patient, which appears 22065 times, it does give us an insight that the text is focused on medical transcripts mainly revolving around patient interactions.

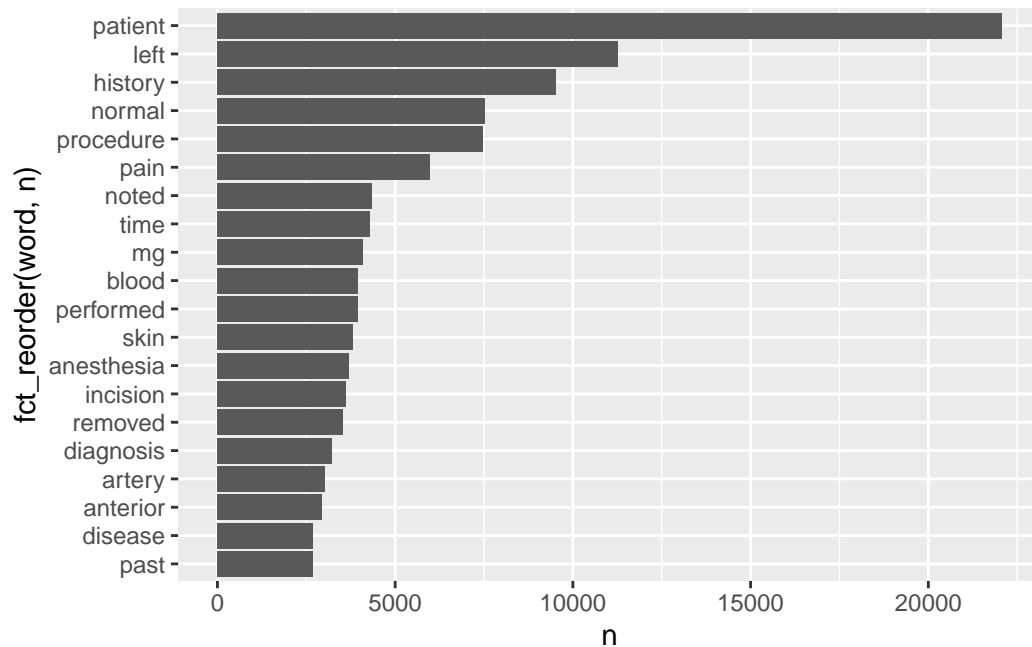
```
mt_samples |>
  unnest_tokens(token, transcription) |>
  count(token) |>
  top_n(20, n) |>
  ggplot(aes(n, token)) +
  geom_col()
```



Question 3

Now that we have removed stop words, we can see that the word 'patient' appears the most which is more fitting given that this is a medical transcript. Looking at the rest of the top 20 words, it is clear that this text is about patient procedures or charting.

```
library(forcats)
library(tidytext)
mt_samples |>
  unnest_tokens(word, transcription) |>
  anti_join(stop_words, by = "word") |>
  filter(!grepl("[0-9]", word)) |>
  count(word, sort = TRUE) |>
  top_n(20, n) |>
  ggplot(aes(n, fct_reorder(word, n))) +
  geom_col()
```



Question 4

We have a lot more insight into what the text is about when tokenizing into tri-grams rather than bi-grams. Bi-grams is mostly stop words but looking at tri-grams, we can see more insight into procedures and even patient symptoms.

```
mt_samples |>
unnest_tokens(bigram, transcription, token = "ngrams", n = 2) |>
count(bigram, sort = TRUE)
```

A tibble: 301,415 x 2

	bigram	n
	<chr>	<int>
1	the patient	20307
2	of the	19062
3	in the	12790
4	to the	12374
5	was then	6956
6	and the	6350
7	patient was	6293
8	the right	5509
9	on the	5241

```
10 the left      4860
# i 301,405 more rows
```

```
mt_samples |>
  unnest_tokens(trigram, transcription, token = "ngrams", n = 3) |>
  count(trigram, sort = TRUE)
```

```
# A tibble: 655,441 x 2
  trigram          n
  <chr>          <int>
1 the patient was 6104
2 the patient is  3075
3 as well as     2243
4 there is no    1678
5 the operating room 1532
6 patient is a   1491
7 prepped and draped 1490
8 was used to    1480
9 and draped in   1372
10 at this time   1333
# i 655,431 more rows
```

Question 5

```
library(stringr)
library(tidyr)
word_to_analyze <- "patient"
bi_grams <- mt_samples|>
unnest_tokens(bigram, transcription, token = "ngrams", n = 2)
print(head(bi_grams))
```

```
# A tibble: 6 x 3
  description                                medical_specialty bigram
  <chr>                                <chr>          <chr>
1 A 23-year-old white female presents with complaint o~ Allergy / Immuno~ subje~
2 A 23-year-old white female presents with complaint o~ Allergy / Immuno~ this ~
3 A 23-year-old white female presents with complaint o~ Allergy / Immuno~ 23 ye~
4 A 23-year-old white female presents with complaint o~ Allergy / Immuno~ year ~
5 A 23-year-old white female presents with complaint o~ Allergy / Immuno~ old w~
6 A 23-year-old white female presents with complaint o~ Allergy / Immuno~ white~
```

```
before_after <- bi_grams|>
filter(str_detect(bigram, word_to_analyze))
before_after <- before_after |>
separate(bigram, into = c("word1", "word2"), sep = " ")
```

```
before_count <- before_after |>
filter(word2 == word_to_analyze) |>
count(word1, sort = TRUE) |>
rename(before = word1)
```

```
after_count <- before_after |>
filter(word1 == word_to_analyze) |>
count(word2, sort = TRUE) |>
rename(after = word2)
```

```
print("Words Before 'patient':")
```

```
[1] "Words Before 'patient':"
```

```
print(before_count)
```

```
# A tibble: 269 x 2
  before      n
  <chr>    <int>
1 the      20307
2 this       470
3 history   101
4 a          67
5 and        47
6 procedure   32
7 female     26
8 with       25
9 use        24
10 old       23
# i 259 more rows
```

```
print("Words After 'patient':")
```

```
[1] "Words After 'patient':"
```

```
print(after_count)
```

```
# A tibble: 588 x 2
  after      n
  <chr>    <int>
1 was      6293
2 is       3332
3 has      1417
4 tolerated  994
5 had       888
6 will      616
7 denies    552
8 and       377
9 states    363
10 does     334
# i 578 more rows
```

Question 6

The most used word in allergy/immunology is 'history.' Autopsy is 'left,' Bariatrics is 'patient,' etc. The top 5 most used words include 'patient' 'left' 'history' '2', and '1'.

```
most_used_words <- mt_samples |>
  unnest_tokens(word, transcription) |>
  anti_join(stop_words, by = "word") |>
  group_by(medical_specialty, word) |>
  count(n = n(), sort = TRUE) |>
  arrange(medical_specialty, desc(n))
```

Storing counts in `nn`, as `n` already present in input
i Use `name = "new_name"` to pick a new name.

```
print (most_used_words)
```

```
# A tibble: 149,973 x 4
# Groups:   medical_specialty, word [149,973]
  medical_specialty word      n    nn
  <chr>            <chr>    <int> <int>
1 Allergy / Immunology history 1263045 38
```


2	Allergy / Immunology noted	1263045	23
3	Allergy / Immunology patient	1263045	22
4	Allergy / Immunology allergies	1263045	21
5	Allergy / Immunology nasal	1263045	13
6	Allergy / Immunology past	1263045	13
7	Allergy / Immunology bilaterally	1263045	12
8	Allergy / Immunology masses	1263045	12
9	Allergy / Immunology asthma	1263045	11
10	Allergy / Immunology medical	1263045	11

i 149,963 more rows

```
most_used_words <- mt_samples|>
  unnest_tokens(word, transcription)|>
  anti_join(stop_words, by = "word") |>
  count(word, sort = TRUE) |>
  top_n(5, n)
print(most_used_words)
```

```
# A tibble: 5 x 2
  word      n
  <chr>  <int>
1 patient 22065
2 left    11258
3 history  9509
4 2        8864
5 1         8396
```