Assignment 2

Rashik Mahmud Orchi-B00968298

02/06/2024

```
rm(list=ls())
```

```
knitr::opts_chunk$set(echo = TRUE)
suppressMessages({
  suppressWarnings({
    library(readr)
    library(dplyr)
    library(tidyr)
    library(ggplot2)
    library(scales)
    library(tidytext)
    library(textstem)
    library(clinspacy)
    library(topicmodels)
    library(reshape2)
    library(stringr)
  })
})
```

This practical is based on exploratory data analysis, named entity recognition, and topic modelling of unstructured medical note free-text data derived from electronic medical records (EMR). Real EMR data is very difficult to access without a specific need/request so this data set is derived from medical transcription data instead. I'll also caveat that the options of natural language processing (NLP) in R are far inferior to those available in Python.

First, install the packages in the setup block (`install.packages(c("readr", "dplyr", "tidyr", "ggplot2", "tidtext", "textstem", "clinspacy", "topicmodels", "reshape2"))`).

Note: To try and make it clearer which library certain functions are coming from clearer, I'll try to do explicit imports throughout this notebook.

Data Parsing

After that we can grab the dataset directly from the `clinspacy` library.

```
raw.data <- clinspacy::dataset_mtsamples()
dplyr::glimpse(raw.data)
```

```
## Rows: 4,999
## Columns: 6
## $ note_id           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
```

```
## $ description       <chr> "A 23-year-old white female presents with complaint ~
## $ medical_specialty <chr> "Allergy / Immunology", "Bariatrics", "Bariatrics", ~
## $ sample_name       <chr> "Allergic Rhinitis", "Laparoscopic Gastric Bypass Co~
## $ transcription     <chr> "SUBJECTIVE:,  This 23-year-old white female present~
## $ keywords          <chr> "allergy / immunology, allergic rhinitis, allergies,~
```

There is no explanation or data dictionary with this dataset, which is a surprisingly common and frustrating turn of events!

1 Using the output of dplyr's `glimpse` command (or rstudio's data viewer by clicking on `raw.data` in the Environment pane) provide a description of what you think each variable in this dataset contains.

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

Variable	Description	Data Type
note_id	A unique identifier for each note	Integer
description	Describes the patient's main complaint or condition.That means it gives us a brief overview of the patient's presentation or the primary reason for the medical note.	chr
medical_specialty	It categorizes the each note under specific medical fields, such as Allergy / Immunology, Bariatrics, radiology, orthopedic and so on	chr
sample_name	It provides a concise title or label for the medical note where the values represent specific conditions, procedures, or types of consultations	chr
transcription	This variable is a character string containing the full transcription of the medical note. It includes detailed clinical information, such as subjective and objective findings, medical history, medications, allergies, assessment, and recommnedations.	chr
keywords	Relevent keywords associated with each medical note are listed here	chr

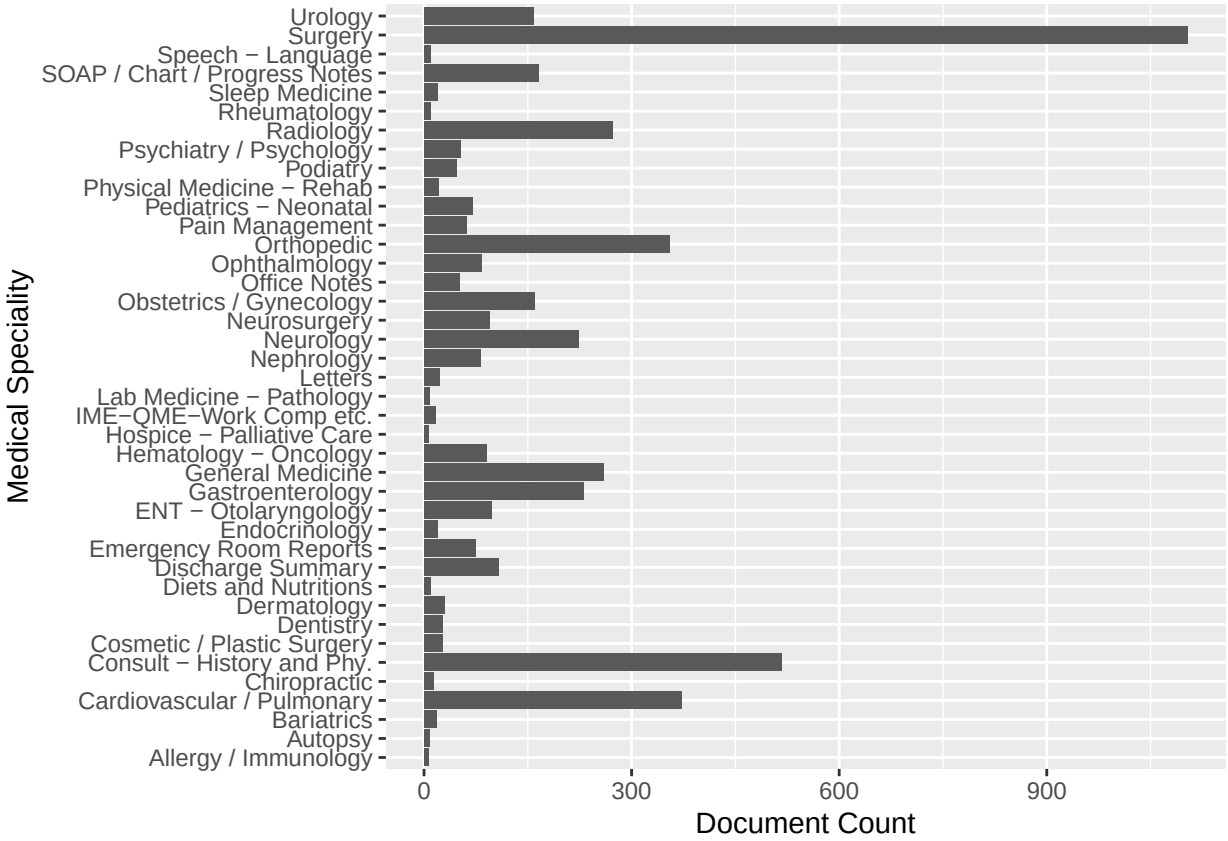
Let's see how many different medical specialties are featured in these notes:

```
raw.data %>% dplyr::select(medical_specialty) %>% dplyr::n_distinct()
```

```
## [1] 40
```

So, how many transcripts are there from each specialty:

```
ggplot2::ggplot(raw.data, ggplot2::aes(y=medical_specialty)) + ggplot2::geom_bar() + labs(x="Document C
```



Let's make our life easier and filter down to 3 specialties: a diagonstic/lab, a medical, and a surgical specialty

```
filtered.data <- raw.data %>% dplyr::filter(medical_specialty %in% c("Orthopedic", "Radiology", "Surgery
```

Text Processing

Let's now apply our standard pre-processing to the transcripts from these specialties.
We are going to use the `tidytext` package to tokenise the transcript free-text.
Let's remove stop words first. e.g., "the", "of", "to", and so forth. These are known as stop words and we can remove them relative easily using a list from `tidytext::stop_words` and `dplyr::anti_join()`

```
analysis.data <- filtered.data %>%
  unnest_tokens(word, transcription) %>%
  mutate(word = str_replace_all(word, "[^[:alnum:]]", "")) %>%
  filter(!str_detect(word, "[0-9]")) %>%
  anti_join(stop_words) %>%
  group_by(note_id) %>%
  summarise(transcription = paste(word, collapse = " ")) %>%
  left_join(select(filtered.data, -transcription), by = "note_id")
```

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

Now let's tokenize the `transcription` to words (unigram) By default this tokenises to words but other options include characters, n-grams, sentences, lines, paragraphs, or separation around a regular expression.

```
tokenized.data.unigram <- analysis.data %>% tidytext::unnest_tokens(word, transcription, to_lower=TRUE)
```

You can also do bi-grams

```
tokenized.data <- analysis.data %>% tidytext::unnest_tokens(ngram, transcription, token = "ngrams", n=2
```

How many stop words are there in `tidytext::stop_words` from each lexicon?

```
tidytext::stop_words %>% dplyr::group_by(lexicon) %>% dplyr::distinct(word) %>% dplyr::summarise(n=dplyr
```

```
## # A tibble: 3 x 2
##   lexicon      n
##   <chr>    <int>
## 1 SMART      570
## 2 onix       398
## 3 snowball   174
```

2 How many unique unigrams are there in the transcripts from each specialty:

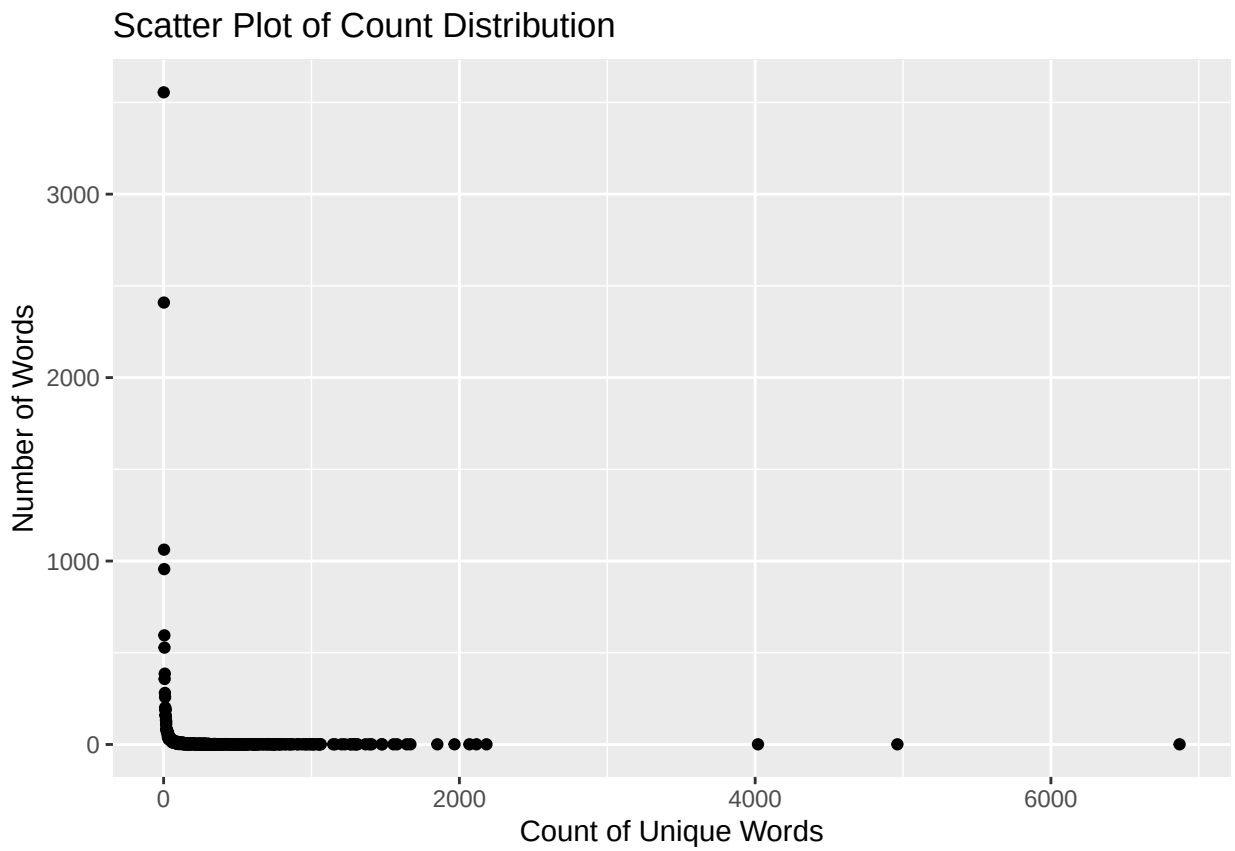
Unique unigrams in the transcripts are provided below:

```
tokenized.data.unigram%>% dplyr::group_by(medical_specialty) %>% dplyr::distinct(word) %>% dplyr::summar
```

```
## # A tibble: 3 x 2
##   medical_specialty     n
##   <chr>             <int>
## 1 Orthopedic         7681
## 2 Radiology          5933
## 3 Surgery           11977
```

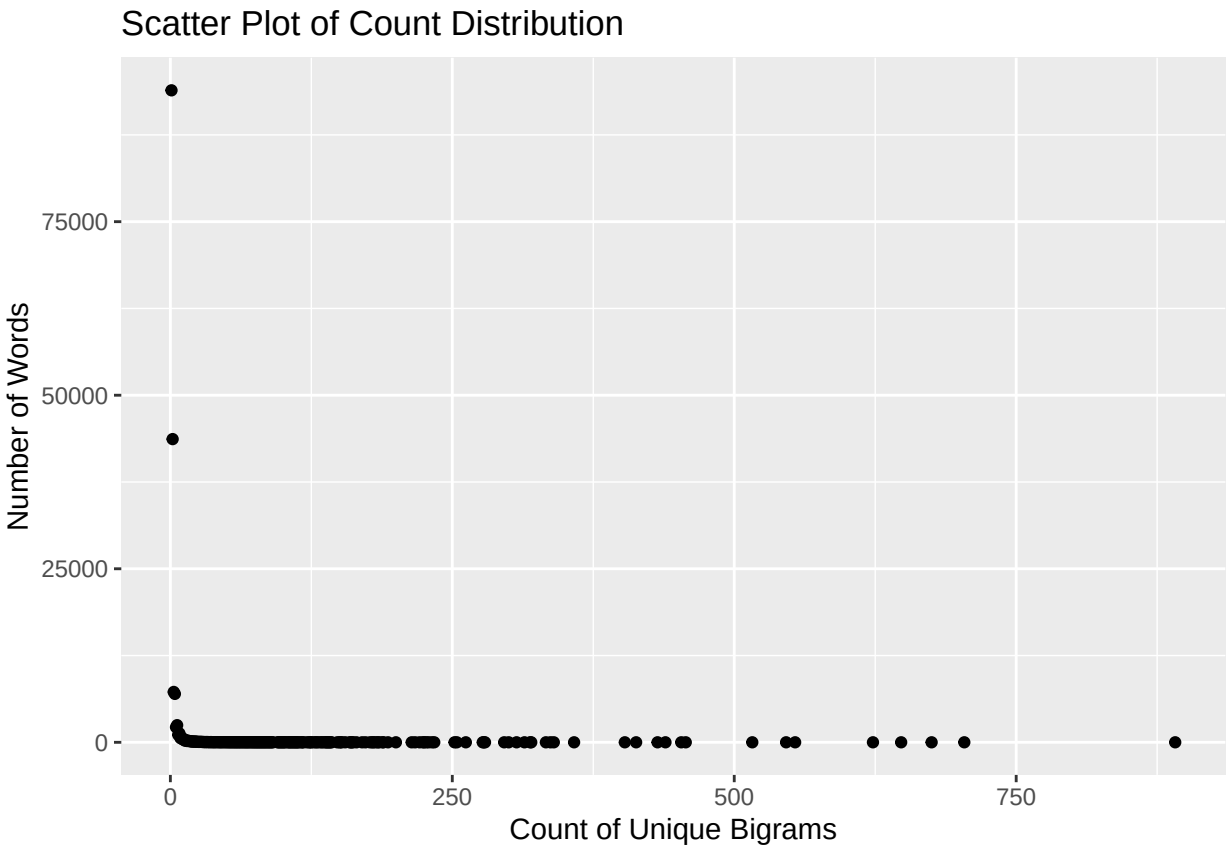
Let's plot some distribution of unigram tokens (words)

```
word_counts <- tokenized.data.unigram %>%
    group_by(word) %>%
    summarise(count = n()) %>%
    ungroup() %>%
    arrange(desc(count))

count_distribution <- word_counts %>%
  group_by(count) %>%
  summarise(num_words = n()) %>%
  ungroup()

 ggplot2::ggplot(count_distribution, aes(x = count, y = num_words)) +
  geom_point() +
  labs(title = "Scatter Plot of Count Distribution",
       x = "Count of Unique Words",
       y = "Number of Words")
```



Let's plot some distribution of bigram tokens (words)

```
word_counts <- tokenized.data %>%
    group_by(ngram) %>%
    summarise(count = n()) %>%
    ungroup() %>%
    arrange(desc(count))

count_distribution <- word_counts %>%
  group_by(count) %>%
  summarise(num_words = n()) %>%
  ungroup()

 ggplot2::ggplot(count_distribution, aes(x = count, y = num_words)) +
  geom_point() +
  labs(title = "Scatter Plot of Count Distribution",
       x = "Count of Unique Bigrams",
       y = "Number of Words")
```



3 How many unique bi-grams are there in each category without stop words and numbers?

The uniue bigrams without stop words and numbers are :

```
tokenized.data%>% dplyr::group_by(medical_specialty) %>% dplyr::distinct(ngram) %>% dplyr::summarise(Cou
```

```
## # A tibble: 3 x 2
##   medical_specialty Count_unique_bigrams
##   <chr>                            <int>
## 1 Orthopedic                       55730
## 2 Radiology                        28294
## 3 Surgery                         130404
```

Sometimes we are interested in tokenising/segmenting things other than words like whole sentences or paragraphs.
4 How many unique sentences are there in each category? Hint: use `?tidytext::unnest_tokens` to see the documentation for this function.

Unique sentences in each category without stop words and numbers are given below

```
analysis.data %>%
tidytext::unnest_tokens(sentence, transcription,token = 'sentences', to_lower=TRUE) %>%
group_by(medical_specialty) %>%
  dplyr::distinct(sentence) %>%
  dplyr::summarise(count_unique_sentence=dplyr::n())
```

```
## # A tibble: 3 x 2
##   medical_specialty count_unique_sentence
##   <chr>                             <int>
## 1 Orthopedic                          354
## 2 Radiology                           273
## 3 Surgery                            1087
```

Unique sentences in each category with stop words and numbers are given below:

```
filtered.data%>%
tidytext::unnest_tokens(sentence, transcription,token = 'sentences', to_lower=TRUE) %>%
group_by(medical_specialty) %>%
  dplyr::distinct(sentence) %>%
  dplyr::summarise(count_unique_sentence=dplyr::n())
```

```
## # A tibble: 3 x 2
##   medical_specialty count_unique_sentence
##   <chr>                             <int>
## 1 Orthopedic                        11174
## 2 Radiology                          4565
## 3 Surgery                           29652
```

Now that we've tokenized to words and removed stop words, we can find the most commonly word used within each category:

```
tokenized.data %>%
  dplyr::group_by(medical_specialty) %>%
  dplyr::count(ngram, sort = TRUE) %>%
  dplyr::top_n(5)
```

```
## Selecting by n
```

```
## # A tibble: 16 x 3
## # Groups:   medical_specialty [3]
##    medical_specialty ngram                      n
##    <chr>             <chr>                  <int>
##  1 Surgery           prepped draped           696
##  2 Surgery           preoperative diagnosis   555
##  3 Surgery           procedure patient        551
##  4 Surgery           postoperative diagnosis  518
##  5 Surgery           tolerated procedure      515
##  6 Orthopedic        prepped draped           183
##  7 Orthopedic        preoperative diagnosis   141
##  8 Orthopedic        lower extremity          139
##  9 Orthopedic        range motion             139
## 10 Orthopedic        postoperative diagnosis  124
## 11 Radiology         carotid artery            59
## 12 Radiology         heart rate                52
## 13 Radiology         reason exam               51
## 14 Radiology         left ventricular          50
## 15 Radiology         coronary artery           43
## 16 Radiology         exam unremarkable         43
```

We should lemmatize the tokenized words to prevent over counting of similar words before further analyses. Annoyingly, `tidytext` doesn't have a built-in lemmatizer.

5 Do you think a general purpose lemmatizer will work well for medical data? Why might it not?

A general purpose lemmatizer may not be highly effective for medical data due to several reasons. While lemmatizers are designed to reduce words to their base or dictionary forms, medical terminology presents unique challenges that can limit the effectiveness of a general lemmatizer. Some of the reasons why genereal purpose lemmatizer will **not** work well for medical data are discussed below:

a. One significant issue is the specialized vocabulary found in medical texts. Medical terminology often includes complex, multi-part words and phrases, many of which are derived from Latin and Greek. These terms might not be well-represented in the training data used for general lemmatizers, which are typically designed for more general language use. For instance, a general lemmatizer might struggle with accurately processing terms like "osteoporosis," "myocardial infarction," or "gastroesophageal reflux disease." Without specialized knowledge of these terms, a general lemmatizer may not be able to reduce them to their correct base forms.

b. Another challenge is the presence of acronyms and abbreviations, which are prevalent in medical texts. Medical professionals frequently use acronyms such as "MRI" (Magnetic Resonance Imaging) or "ECG" (Electrocardiogram). A general lemmatizer may not recognize these as distinct entities with specific meanings, potentially leading to errors in text processing.

c. Furthermore, medical data often includes patient notes and other narrative texts that can contain a mix of formal terminology, shorthand, and colloquial expressions. This variability can confuse general lemmatizers, which might not be equipped to handle such diverse linguistic inputs effectively. For example, a patient's record might use shorthand like "pt" for "patient" or "hx" for "history," which a general lemmatizer might not correctly interpret.

d. Additionally, the context in which medical terms are used can vary widely. A term like "hypertension" might appear in various forms, such as "hypertensive," "hypertension-related," or "anti-hypertensive." A general lemmatizer might not be able to recognize that these variations refer to the same underlying condition, leading to inconsistencies in data processing and analysis.

To address these issues, a lemmatizer specifically trained on medical corpora is often necessary. Such a lemmatizer would be more familiar with medical terminology, acronyms, and the context in which different terms are used, leading to more accurate and consistent lemmatization. This specialized training enables the lemmatizer to handle the complexities and nuances of medical language more effectively than a general-purpose tool. So, while a general purpose lemmatizer might provide some basic functionality for medical data, it is likely to fall short in accuracy and consistency due to the unique characteristics of medical terminology and text. A lemmatizer tailored to the medical domain is essential for achieving reliable and meaningful text processing in healthcare applications.

Unfortunately, a specialised lemmatizer like in `clinspacy` is going to be very painful to install so we will just use a simple lemmatizer for now:

```
lemmatized.data <- tokenized.data %>% dplyr::mutate(lemma=textstem::lemmatize_words(ngram))
```

We can now calculate the frequency of lemmas within each specialty and note.

```
lemma.freq <- lemmatized.data %>%
  dplyr::count(medical_specialty, lemma) %>%
  dplyr::group_by(medical_specialty) %>%
  dplyr::mutate(proportion = n / sum(n)) %>%
  tidyr::pivot_wider(names_from = medical_specialty, values_from = proportion) %>%
  tidyr::pivot_longer(`Surgery`:`Radiology`,
               names_to = "medical_specialty", values_to = "proportion")
```

```
head(lemma.freq)
```

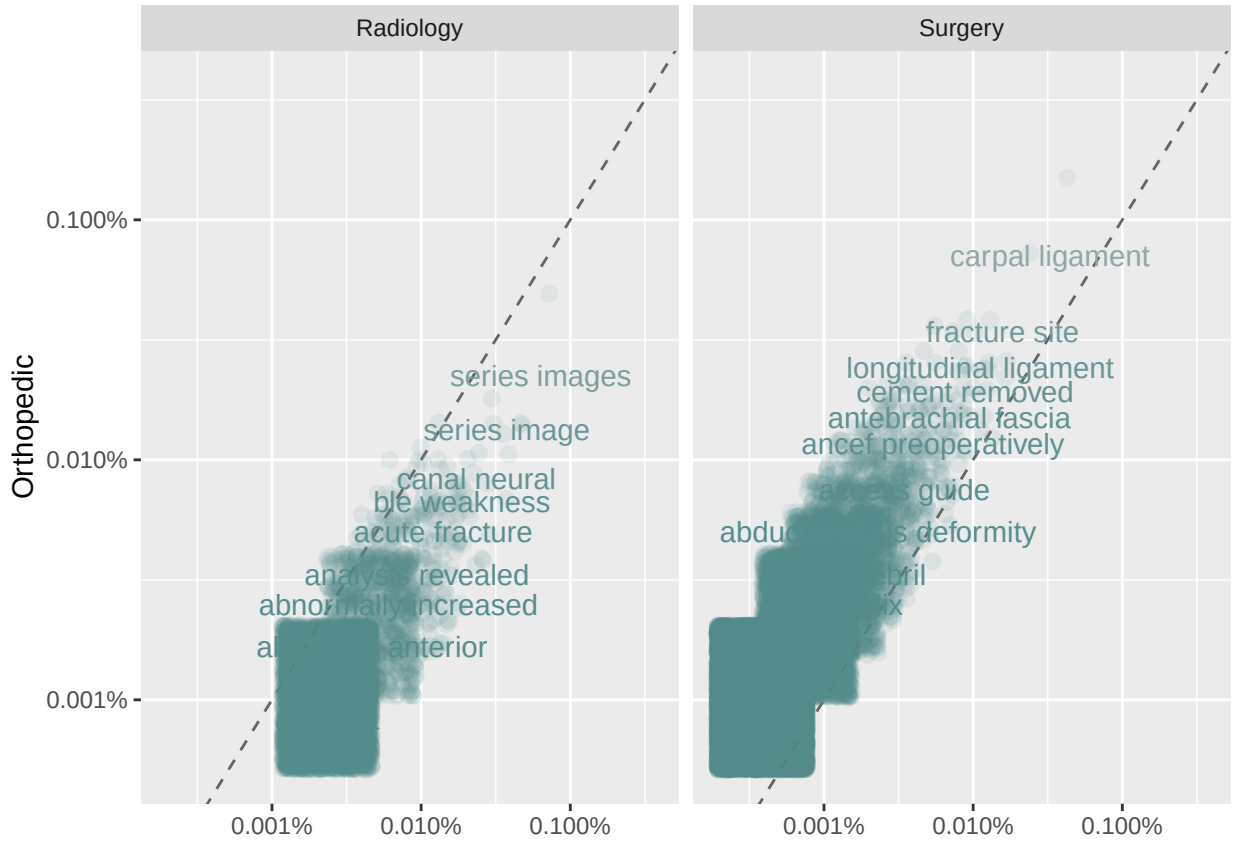
```
## # A tibble: 6 x 5
##   lemma          n Orthopedic medical_specialty  proportion
##   <chr>      <int>      <dbl> <chr>                     <dbl>
## 1 aa body        1  0.0000102 Surgery            0.00000385
## 2 aa body        1  0.0000102 Radiology         NA
## 3 aa femoral     1  0.0000102 Surgery            0.00000385
## 4 aa femoral     1  0.0000102 Radiology         NA
## 5 aa trial       1  0.0000102 Surgery            0.00000385
## 6 aa trial       1  0.0000102 Radiology         NA
```

And plot the relative proportions

```
ggplot2::ggplot(lemma.freq, ggplot2::aes(x=proportion,
                                         y=`Orthopedic`,
                                         color=abs(`Orthopedic` - proportion))) +
  ggplot2::geom_abline(color="gray40", lty=2) +
  ggplot2::geom_jitter(alpha=0.1, size=2.5, width=0.3, height=0.3) +
  ggplot2::geom_text(ggplot2::aes(label=lemma), check_overlap=TRUE, vjust=1.5) +
  ggplot2::scale_x_log10(labels=scales::percent_format()) +
  ggplot2::scale_y_log10(labels=scales::percent_format()) +
  ggplot2::scale_color_gradient(limits=c(0, 0.001), low="darkslategray4", high="gray75") +
  ggplot2::facet_wrap(~medical_specialty, ncol = 2) +
  ggplot2::theme(legend.position="none") +
  ggplot2:: labs(y="Orthopedic", x = NULL)
```

```
## Warning: Removed 314400 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 314400 rows containing missing values or values outside the scale range
## (`geom_text()`).
```



6 What does this plot tell you about the relative similarity of lemma frequencies between Surgery and Orthopedic and between radiology and Orthopedic? Based on what these specialties involve, is this what you would expect?

In the plot on the right (Surgery vs. Orthopedic), many data points fall above the diagonal line. This means that many words appear more frequently in Orthopedic texts than in Surgery texts.Again in the plot on the left (Radiology vs. Orthopedic), there are fewer data points above the diagonal, and those that are there are closer to the line. This suggests a higher overall similarity in word use between **Radiology and Orthopedic**.

Without these plots, I expected that orthopedic and surgery would have higher relative similarity of lemma frequencies.Given their shared focus on performing procedures, particularly on the musculoskeletal system, it's natural to expect Orthopedic and Surgery to have a higher relative similarity in lemma frequencies.While common sense suggested a strong similarity between orthopedic and surgery terminology, the data presented in these plots suggests otherwise.

Since Radiology is heavily involved in imaging and diagnostic processes that are crucial for orthopedic diagnoses and treatments, this shared focus on diagnosis likely results in a higher overlap in terminology, which is reflected in the plot. While Surgery and Orthopedic share some common ground, Orthopedic surgery is a specialized field within Surgery.Thus, there are more specific terms unique to Orthopedic surgery that are not as common in general surgical contexts, leading to a greater difference in lemma frequencies.

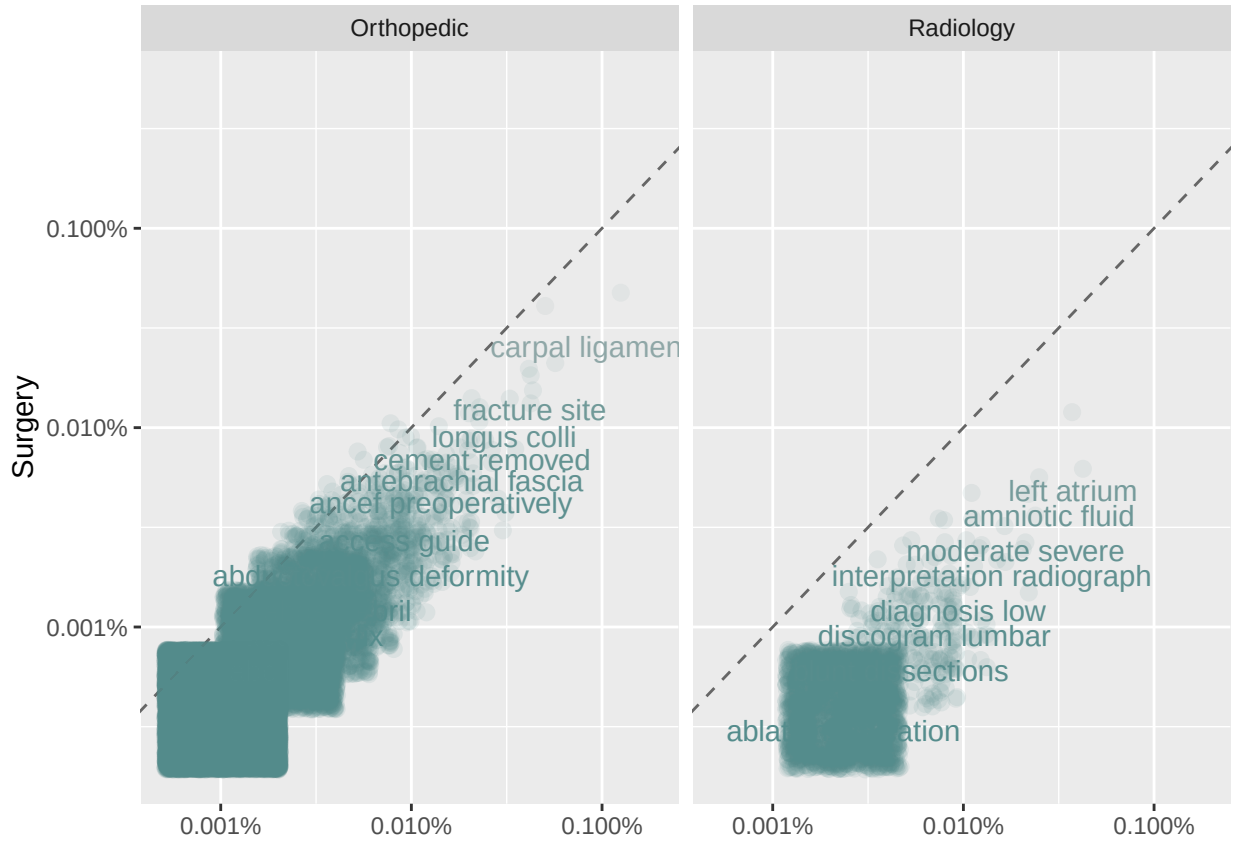
7 Modify the above plotting code to do a direct comparison of Surgery and Radiology (i.e., have Surgery or Radiology on the Y-axis and the other 2 specialties as the X facets)

```
lemma.freq_surgery <- lemmatized.data %>%
  dplyr::count(medical_specialty, lemma) %>%
  dplyr::group_by(medical_specialty) %>%
  dplyr::mutate(proportion = n / sum(n)) %>%
```

```
  tidyr::pivot_wider(names_from = medical_specialty, values_from = proportion) %>%
  tidyr::pivot_longer(cols = c("Radiology", "Orthopedic"), names_to = "medical_specialty", values_to ="
# Plot with Surgery
ggplot(lemma.freq_surgery , aes(x = proportion, y = Surgery, color = abs(Surgery - proportion))) +
  geom_abline(color = "gray40", lty = 2) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +
  geom_text(aes(label = lemma), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = scales::percent_format()) +
  scale_y_log10(labels = scales::percent_format()) +
  scale_color_gradient(limits = c(0, 0.001), low = "darkslategray4", high = "gray75") +
  facet_wrap(~ medical_specialty, ncol = 2) +
  theme(legend.position = "none") +
  labs(y = "Surgery", x = NULL)
```

```
## Warning: Removed 317813 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 317814 rows containing missing values or values outside the scale range
## (`geom_text()`).
```



The plot on the left (Orthopedic vs. Surgery) indicates a higher similarity, as many data points are closer to the diagonal line, suggesting similar word usage between Orthopedic and Surgery. In contrast, the plot on the right (Radiology vs. Surgery) shows fewer data points near the diagonal line, indicating lower similarity in terminology between Radiology and Surgery.

TF-IDF Normalisation

Maybe looking at lemmas across all notes in a specialty is misleading, what if we look at lemma frequencies across a specialty.

```
lemma.counts <- lemmatized.data %>% dplyr::count(medical_specialty, lemma)
total.counts <- lemma.counts %>%
                      dplyr::group_by(medical_specialty) %>%
                      dplyr::summarise(total=sum(n))

all.counts <- dplyr::left_join(lemma.counts, total.counts)
```

```
## Joining with `by = join_by(medical_specialty)`
```

Now we can calculate the term frequency / invariant document frequency (tf-idf):

```
all.counts.tfidf <- tidytext::bind_tf_idf(all.counts, lemma, medical_specialty, n)
```

We can then look at the top 10 lemma by tf-idf within each specialty:

```
all.counts.tfidf %>% dplyr::group_by(medical_specialty) %>% dplyr::slice_max(order_by=tf_idf, n=10)
```

```
## # A tibble: 30 x 7
## # Groups:   medical_specialty [3]
##    medical_specialty lemma                    n total       tf   idf   tf_idf
##    <chr>             <chr>                <int> <int>    <dbl> <dbl>    <dbl>
##  1 Orthopedic        range motion           139 97774 0.00142  0.405 0.000576
##  2 Orthopedic        carpal ligament         85 97774 0.000869 0.405 0.000352
##  3 Orthopedic        transverse carpal       81 97774 0.000828 0.405 0.000336
##  4 Orthopedic        extremity prepped       79 97774 0.000808 0.405 0.000328
##  5 Orthopedic        proximal phalanx        75 97774 0.000767 0.405 0.000311
##  6 Orthopedic        department anesthesia   63 97774 0.000644 0.405 0.000261
##  7 Orthopedic        dissection carried      63 97774 0.000644 0.405 0.000261
##  8 Orthopedic        steri strips            59 97774 0.000603 0.405 0.000245
##  9 Orthopedic        closed vicryl           58 97774 0.000593 0.405 0.000241
## 10 Orthopedic        dressing applied        58 97774 0.000593 0.405 0.000241
## # i 20 more rows
```

8 Are there any lemmas that stand out in these lists? Why or why not?

Based on TF-IDF value notable Lemmas by Specialty are given below:
Orthopedic:

a. Range motion (tf_idf = 0.0005764278)

b. Carpal ligament (tf_idf = 0.0003524918)

c. Transverse carpal (tf_idf = 0.0003359040)

The provided terms hold significant weight within Orthopedic documents, reflecting the core anatomical and procedural focus of the specialty. Terms like "range of motion" stand out as crucial for assessing and discussing joint and limb function. Similarly, "carpal ligament" and "transverse carpal" pinpoint specific wrist anatomy, frequently encountered in orthopedic treatments and surgeries
Radiology:

a. Myocardial perfusion (tf_idf = 0.0008513375)

b. Motor units (tf_idf = 0.0006707507)

c. Calcific plaque (tf_idf = 0.0005933564)

In Radiology, terms like "myocardial perfusion" and "motor units" are noteworthy. These terms are critical in diagnostic imaging and assessments performed in Radiology, highlighting their importance. "Myocardial perfusion" is key in cardiac imaging, while "motor units" and "calcific plaque" are crucial in understanding neuromuscular function and vascular health through imaging studies. **Surgery:**

a. Anterior chamber (tf_idf = 0.0004271531)

b. Lithotomy position (tf_idf = 0.0003425683)

c. External oblique (tf_idf = 0.0002875882)

For Surgery, terms like "anterior chamber" and "lithotomy position" stand out. These terms are specific to surgical procedures and patient positioning, which are key aspects of surgical practice. The "anterior chamber" is an important term in ophthalmic surgery, while the "lithotomy position" is a common patient position used in various surgical procedures. "External oblique" refers to a muscle often involved in abdominal surgeries.

The reasons why these lemmas stand out are provided below:

a. Each term is highly specialized and relevant to the procedures and focus areas of its respective medical specialty. For instance, "range motion" is critical in Orthopedic for assessing joint function, whereas "myocardial perfusion" is crucial in Radiology for evaluating heart function.

b. These terms have higher tf-idf values compared to others within the same specialty. This means they are more unique and carry more weight in their respective contexts.

c. The standout terms reflect procedures, anatomical terms, and diagnostic measures that are central to the practice and literature of the respective specialties. For example, "anterior chamber" is significant in surgical procedures involving the eye, while "calcific plaque" is important in imaging diagnostics in Radiology.

The lemmas that stand out in these lists are notable due to their high tf-idf values, indicating their relative importance and uniqueness within their respective specialties. These terms are specialized and reflect the core focus areas and procedures pertinent to Orthopedic, Radiology, and Surgery, thereby highlighting their significance in medical documentation and practice.

We can look at transcriptions in full using these lemmas to check how they are used with `stringr::str_detect`

```
analysis.data %>% dplyr::select(medical_specialty, transcription) %>% dplyr::filter(stringr::str_detect
```

```
## # A tibble: 1 x 2
##   medical_specialty transcription
##   <chr>             <chr>
## 1 Surgery           preoperative diagnoses hallux rigidus left foot elevated me~
```

9 Extract an example of one of the other "top lemmas" by modifying the above code

```
analysis.data %>% dplyr::select(medical_specialty, transcription) %>% dplyr::filter(stringr::str_detect
```

```
## # A tibble: 1 x 2
##   medical_specialty transcription
##   <chr>             <chr>
## 1 Radiology         indications chest pain hypertension type ii diabetes mellit~
```

Topic Modelling

In NLP, we often have collections of documents (in our case EMR transcriptions) that we'd like to divide into groups so that we can understand them separately. Topic modeling is a method for unsupervised classification of such documents, similar to clustering on numeric data.

Latent Dirichlet allocation (LDA) is a particularly popular method for fitting a topic model. It treats each document as a mixture of topics, and each topic as a mixture of words. This allows documents to "overlap" each other in terms of content, rather than being separated into discrete groups, in a way that mirrors typical use of natural language.

- Every document is a mixture of topics. We imagine that each document may contain words from several topics in particular proportions. For example, in a two-topic model we could say "Document 1 is 90% topic A and 10% topic B, while Document 2 is 30% topic A and 70% topic B."
- Every topic is a mixture of words. For example, we could imagine a two-topic model of American news, with one topic for "politics" and one for "entertainment." The most common words in the politics topic might be "President", "Congress", and "government", while the entertainment topic may be made up of words such as "movies", "television", and "actor". Importantly, words can be shared between topics; a word like "budget" might appear in both equally.

LDA is a mathematical method for estimating both of these at the same time: finding the mixture of words that is associated with each topic, while also determining the mixture of topics that describes each document. There are a number of existing implementations of this algorithm, and we'll explore one of them in depth.

First lets calculate a term frequency matrix for each transcription:

```
lemma.counts <- lemmatized.data %>% dplyr::count(note_id, lemma)
total.counts <- lemma.counts %>%
                      dplyr::group_by(note_id) %>%
                      dplyr::summarise(total=sum(n))

all.counts <- dplyr::left_join(lemma.counts, total.counts)
```

```
## Joining with `by = join_by(note_id)`
```

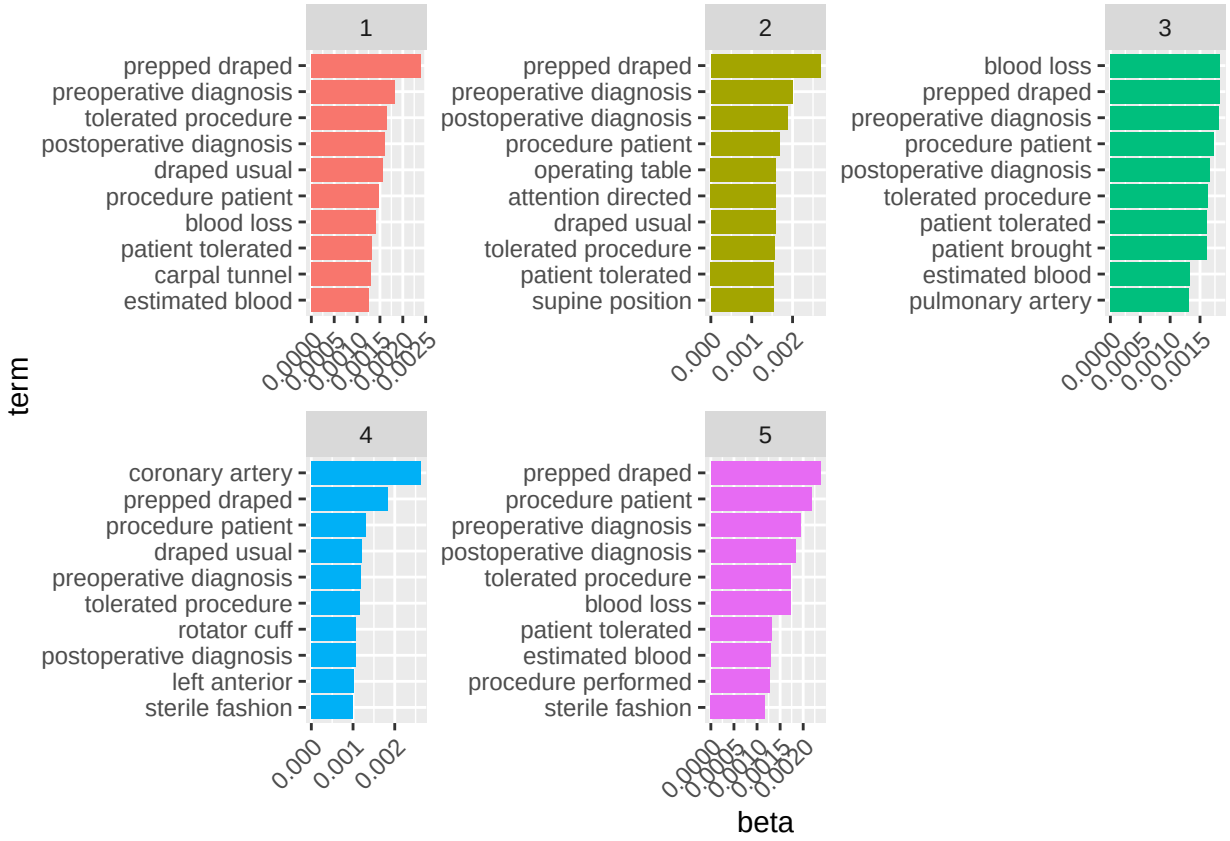
```
emr.dcm <- all.counts %>% tidytext::cast_dtm(note_id, lemma, n)
```

Then we can use LDA function to fit a 5 topic (`k=5`) LDA-model

```
emr.lda <- topicmodels::LDA(emr.dcm, k=5, control=list(seed=42))
emr.topics <- tidytext::tidy(emr.lda, matrix='beta')
```

Then we can extract the top terms per assigned topic:

```
top.terms <- emr.topics %>% dplyr::group_by(topic) %>%
  dplyr::slice_max(beta, n=10) %>%
  dplyr::ungroup() %>%
  dplyr::arrange(topic, -beta)

top.terms %>%
  dplyr::mutate(term=tidytext::reorder_within(term, beta, topic)) %>%
  ggplot2::ggplot(ggplot2::aes(beta, term, fill=factor(topic))) +
    ggplot2::geom_col(show.legend=FALSE) +
    ggplot2::facet_wrap(~ topic, scales='free')  +
    ggplot2::theme(axis.text.x = element_text(angle = 45,vjust = 1,hjust = 1)) +
    tidytext::scale_y_reordered()
```

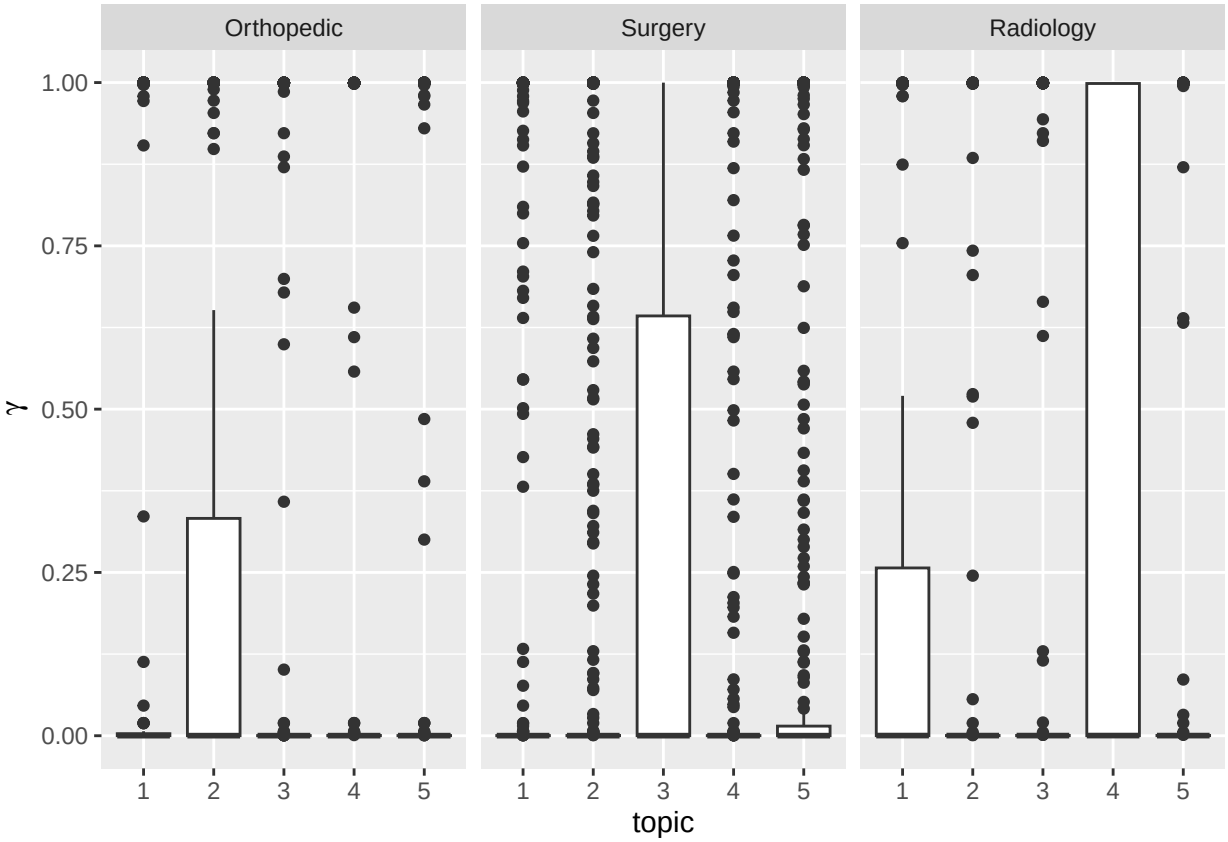


Now we can ask how well do these assigned topics match up to the medical specialties from which each of these transcripts was derived.

```
specialty_gamma <- tidytext::tidy(emr.lda, matrix='gamma')

# we need to join in the specialty from the note_id
note_id_specialty_mapping <- lemmatized.data %>%
  dplyr::mutate(document=as.character(note_id)) %>%
  dplyr::select(document, medical_specialty) %>%
  dplyr::distinct()

specialty_gamma <- dplyr::left_join(specialty_gamma, note_id_specialty_mapping)
```

```
## Joining with `by = join_by(document)`
```

```
specialty_gamma %>%
  dplyr::mutate(medical_specialty = reorder(medical_specialty, gamma * topic)) %>%
  ggplot2::ggplot(ggplot2::aes(factor(topic), gamma)) +
  ggplot2::geom_boxplot() +
  ggplot2::facet_wrap(~ medical_specialty) +
  ggplot2::labs(x = "topic", y = expression(gamma))
```



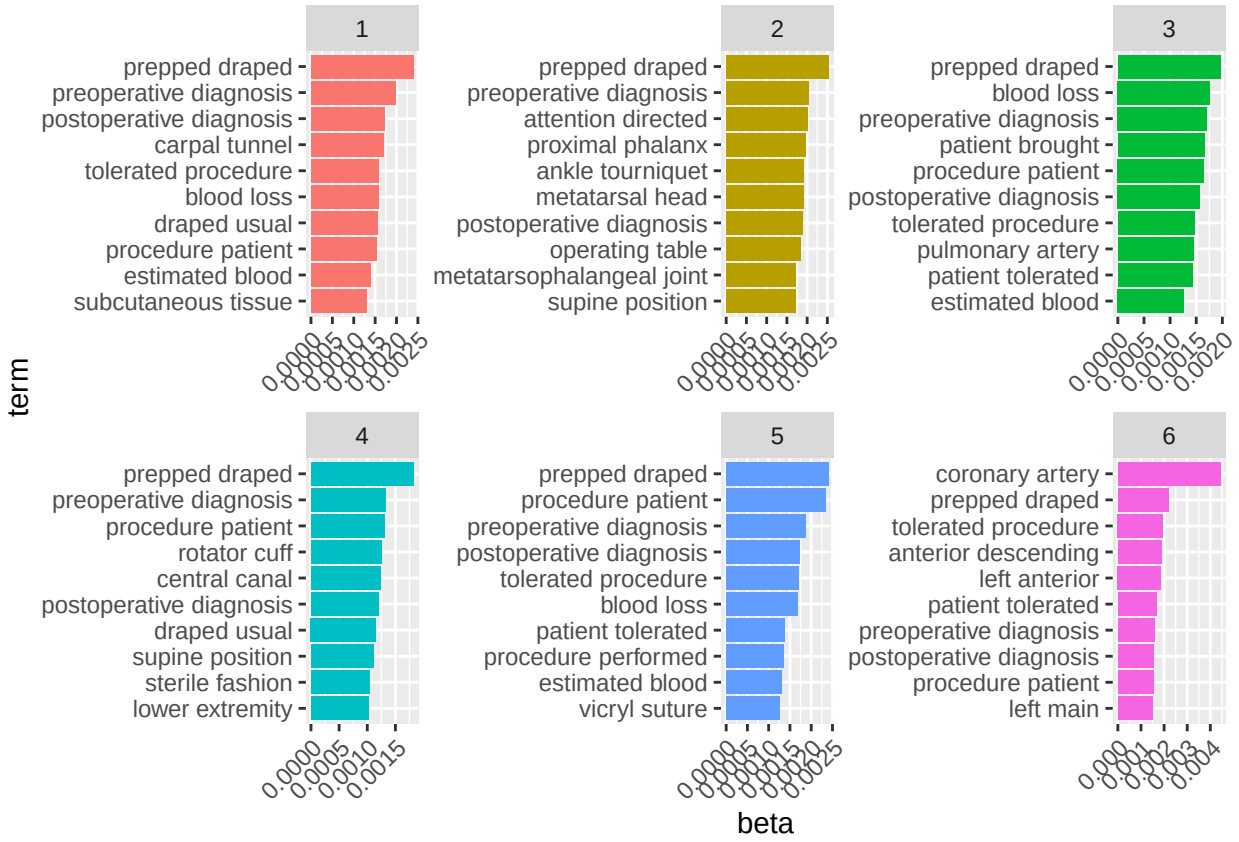
Interestingly, Surgery, Orthopedic, and Radiology assign mostly to a single topics. We'd possibly expect this from radiology due to referring to imaging for many different diagnoses/reasons. However, this may all just reflect we are using too few topics in our LDA to capture the range of possible assignments.

10 Repeat this with a 6 topic LDA, do the top terms from the 3 topic LDA still turn up? How do the specialties get split into sub-topics?

```
emr.lda <- topicmodels::LDA(emr.dcm, k=6, control=list(seed=42))
emr.topics <- tidytext::tidy(emr.lda, matrix='beta')
```

```
top.terms <- emr.topics %>% dplyr::group_by(topic) %>%
  dplyr::slice_max(beta, n=10) %>%
  dplyr::ungroup() %>%
  dplyr::arrange(topic, -beta)


top.terms %>%
```

```
  dplyr::mutate(term=tidytext::reorder_within(term, beta, topic)) %>%
  ggplot2::ggplot(ggplot2::aes(beta, term, fill=factor(topic))) +
    ggplot2::geom_col(show.legend=FALSE) +
    ggplot2::facet_wrap(~ topic, scales='free')  +
    ggplot2::theme(axis.text.x = element_text(angle = 45,vjust = 1,hjust = 1)) +
    tidytext::scale_y_reordered()
```



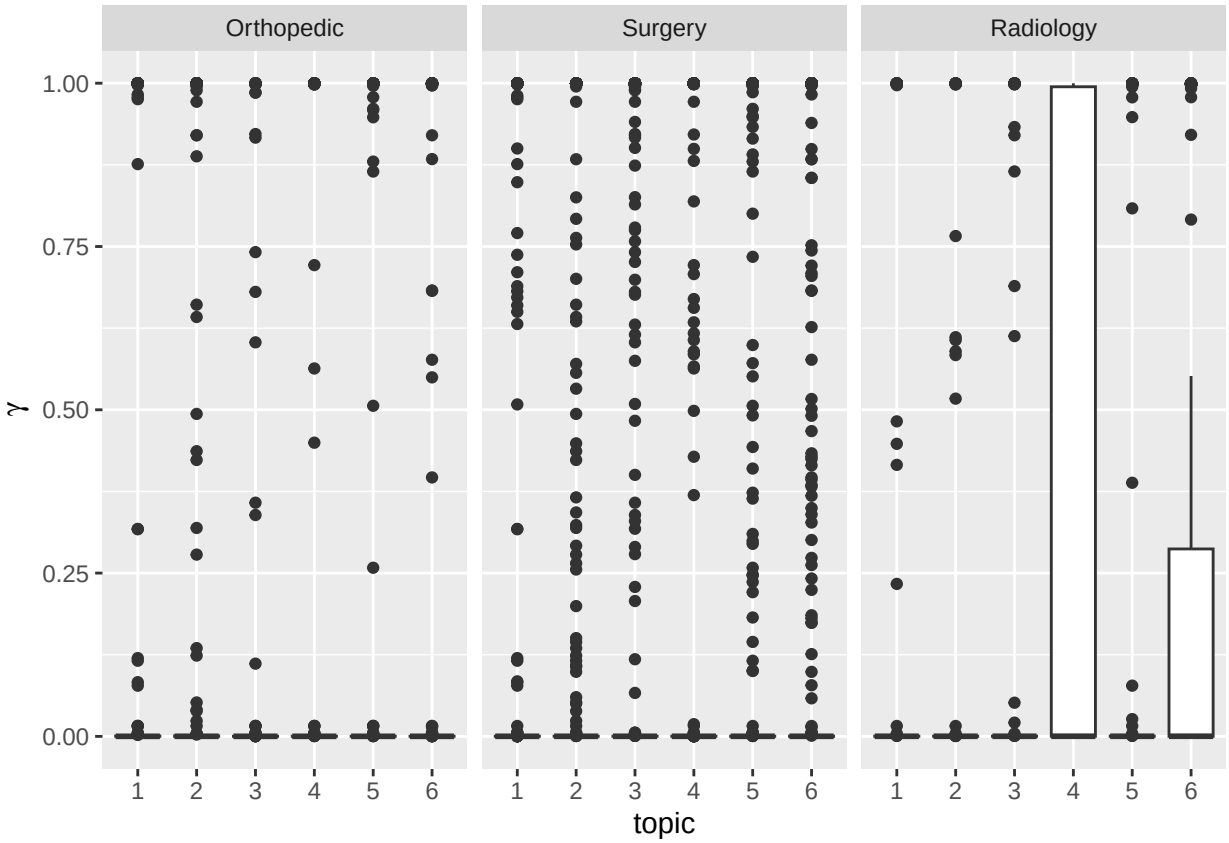
Out of 6 topics, in five of them we got the **"prepped draped"** in remaining topic we "coronary artery". But when we trained the using 5 topics earlier then we got "prepped draped" three times, coronary artery once and "blood loss" once . The "blood loss" is missing when we trained with six topics.

```
specialty_gamma<- tidytext::tidy(emr.lda, matrix='gamma')

# we need to join in the specialty from the note_id
note_id_specialty_mapping <- lemmatized.data %>%
  dplyr::mutate(document=as.character(note_id)) %>%
  dplyr::select(document, medical_specialty) %>%
  dplyr::distinct()

specialty_gamma <- dplyr::left_join(specialty_gamma, note_id_specialty_mapping)
```

```
## Joining with `by = join_by(document)`
```

```
specialty_gamma%>%
  dplyr::mutate(medical_specialty = reorder(medical_specialty, gamma * topic)) %>%
```

```
ggplot2::ggplot(ggplot2::aes(factor(topic), gamma)) +
ggplot2::geom_boxplot() +
ggplot2::facet_wrap(~ medical_specialty) +
ggplot2::labs(x = "topic", y = expression(gamma))
```



Orthopedic: The documents related to Orthopedic are distributed across all six topics (1 to 6).There is no single dominant topic, as the distribution is quite spread out. This indicates a diverse range of sub-topics within Orthopedic documents.

Surgery: Similar to Orthopedic, Surgery documents are also spread across all six topics. There are many points with lower gamma values, suggesting that many documents cover multiple topics rather than being focused on a single topic.

Radiology: Radiology documents show a different pattern compared to Orthopedic and Surgery. There is a high concentration of data points in topics 4 and 6, with topic 4 having a particularly large box indicating it is a major focus in Radiology notes. So, Topic 4 appears to be significantly more dominant for Radiology, as many documents have high gamma values close to 1 for this topic. The other topics(1,2,3,5) have fewer data points, suggesting they are less frequently addressed in Radiology notes.

Orthopedic and Surgery both specialties have documents that are spread across multiple topics, indicating that these fields cover a broad range of sub-topics.On the other hand, Radiology has a strong association with a specific topic (Topic 4), indicating a focused sub-topic within Radiology that is distinct from those in Orthopedic and Surgery. Other topics are also present but are less prominent.

Credits

Examples draw heavily on material (and directly quotes/copies text) from Julia Slige's `tidytext` textbook.