Medical Data

2024-12

This dataset is from Kaggle and contains medical data. I did edit the dataset with AI to help establish random symptoms. It was difficult to find a dataset with the needed 3 rows of a category (medical condition), word (symptom), and id. I settled on this dataset and just created a row for symptoms and assigned random symptoms to each id varying from 1-30 symptoms index. The uploaded dataset will be in the Kaggle Datasets folder if curious for examination.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
##
## 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(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.4.2
```

```
library(purrr)
library(readr)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.4.2
```

```
library(stringr)
```

```
#read my file
data <- read.csv("C:\\Users\\knc5576\\Downloads\\hello3.csv")
```

9.1.1 Pre-processing text

Cleaning my data with janitor.

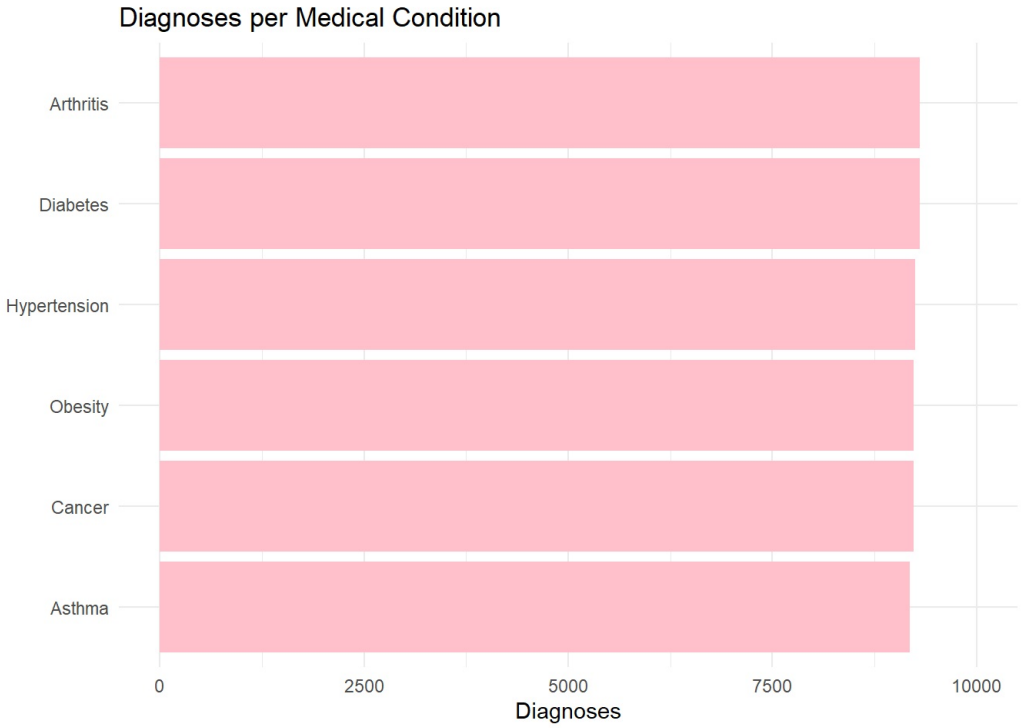
I also tokenized the symptoms column while keeping all original columns. I wanted to also remove stop words like “or”, “and”, “the”, etc.

I also displayed the final data after cleaning and tokenizing.

```
data <- data %>%
  mutate(id = row_number() - 1)
data <- janitor::clean_names(data)

tokenized_data <- data %>%
  unnest_tokens(word, symptoms, drop = FALSE)
data("stop_words")
filtered_data <- tokenized_data %>%
  filter(!word %in% stop_words$word)

final_data <- data %>%
  select(-symptoms) %>%
  left_join(
    filtered_data %>%
      group_by(medical_condition, id) %>%
      summarize(symptoms = paste(word, collapse = " "), .groups = "drop"),
    by = c("medical_condition", "id")
  ) %>%
  mutate(symptoms = ifelse(is.na(symptoms), "No valid symptoms", symptoms)) %>%
  select(medical_condition, id, symptoms)


#remove "of" as a stopword and display the final data
stop_words <- stop_words %>%
  filter(!word %in% c("of"))

### For the sake of my document not being 3000 pages long I am going to leave out me printing the final final_dat
a
```

9.1/9.2 (sorta stil pre-processing) Displaying the amount of each Diagnoses per Medical Condition

```
# Summarize the data: Count the number of Diagnoses per medical condition
summarized_data <- data %>%
  group_by(medical_condition) %>%
  summarize(diagnoses = n_distinct(id))

# Create the bar chart
ggplot(summarized_data, aes(x = diagnoses, y = reorder(medical_condition, diagnoses))) +
  geom_col(fill = "pink") +
  labs(
    x = "Diagnoses",
    y = NULL,
    title = "Diagnoses per Medical Condition"
  ) +
  scale_x_continuous(limits = c(0, 10000)) +
  theme_minimal()
```



```
## Finding most common symptoms
# Tokenize the symptoms column into individual words
symptom_counts <- data %>%
  unnest_tokens(word, symptoms) %>%
  count(word, sort = TRUE)
# View the top symptoms (did in rscript not in markup for small page size purposes)


#Find the most common symptoms by medical condition
words_by_condition <- filtered_data %>%
  count(medical_condition, word, sort = TRUE) %>%
  ungroup()
```

From the output, Arthritis is the most shown condition with “joint” being the most common symptom. The top symptom in general is “pain.” The next common symptom is fatigue, then loss.

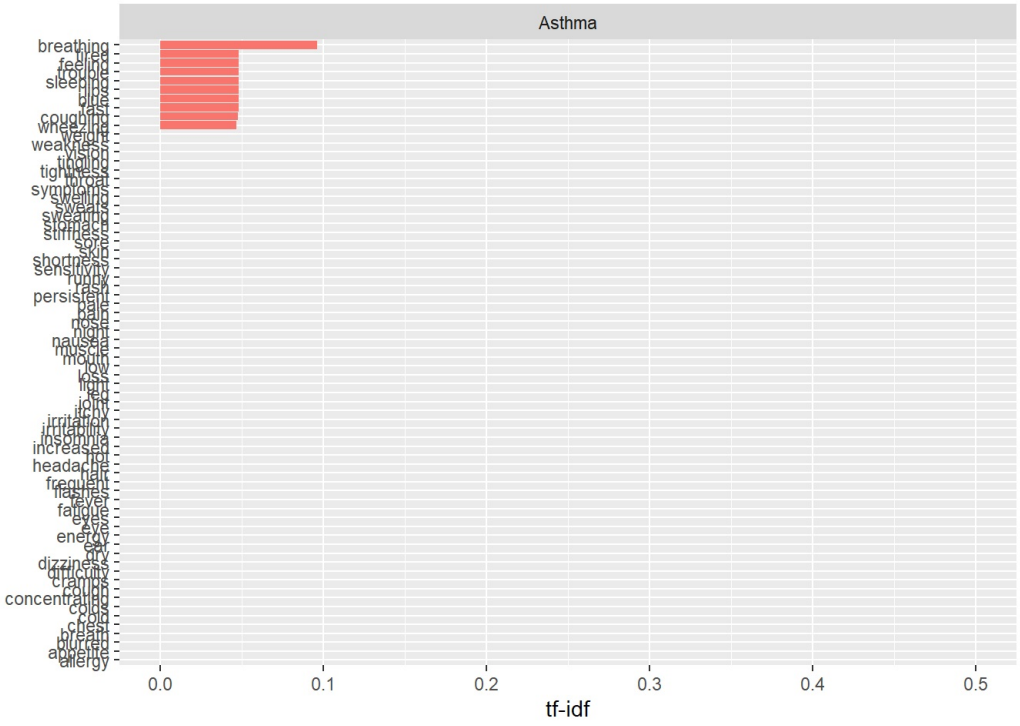
9.2.1 Finding tf-idf within medical condition

```
tf_idf <- words_by_condition %>%
  bind_tf_idf(word, medical_condition, n) %>%
  arrange(desc(tf_idf))
```

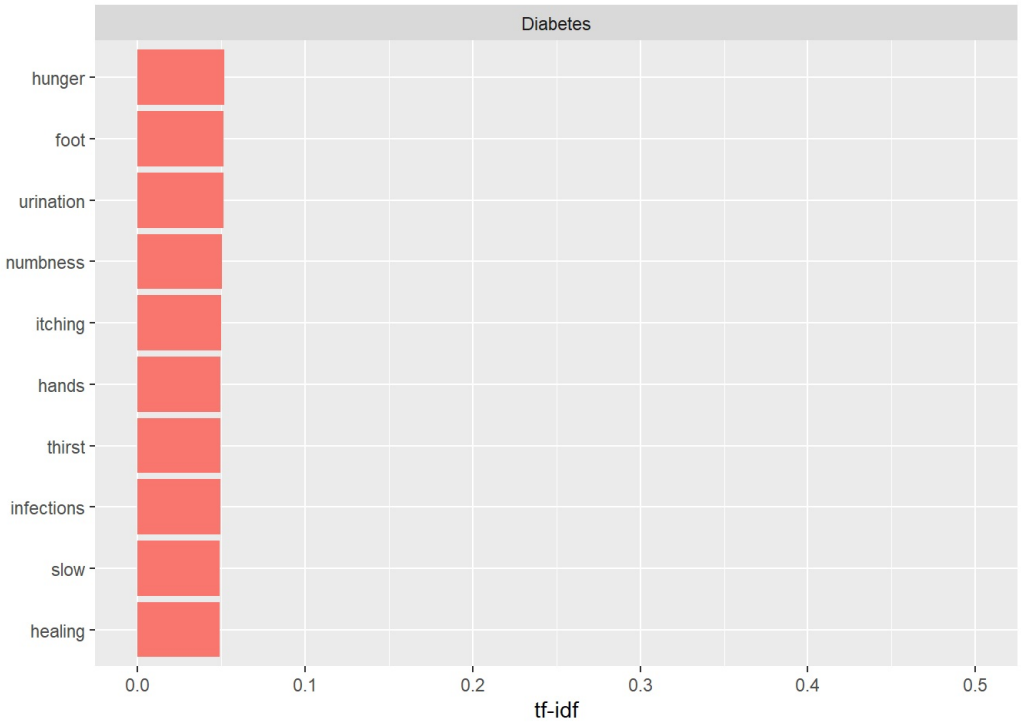
The output shows the top tf-idf words for each medical condition along with its top symptom. It essentially displayed 408 rows of data for medical conditions, word, n, tf, idf, and tf_idf. the idf’s were all about at 1.79 and the tf_idf less than 1, with Asthma having the highest at .096

Displaying every medical condition and it’s tf idf

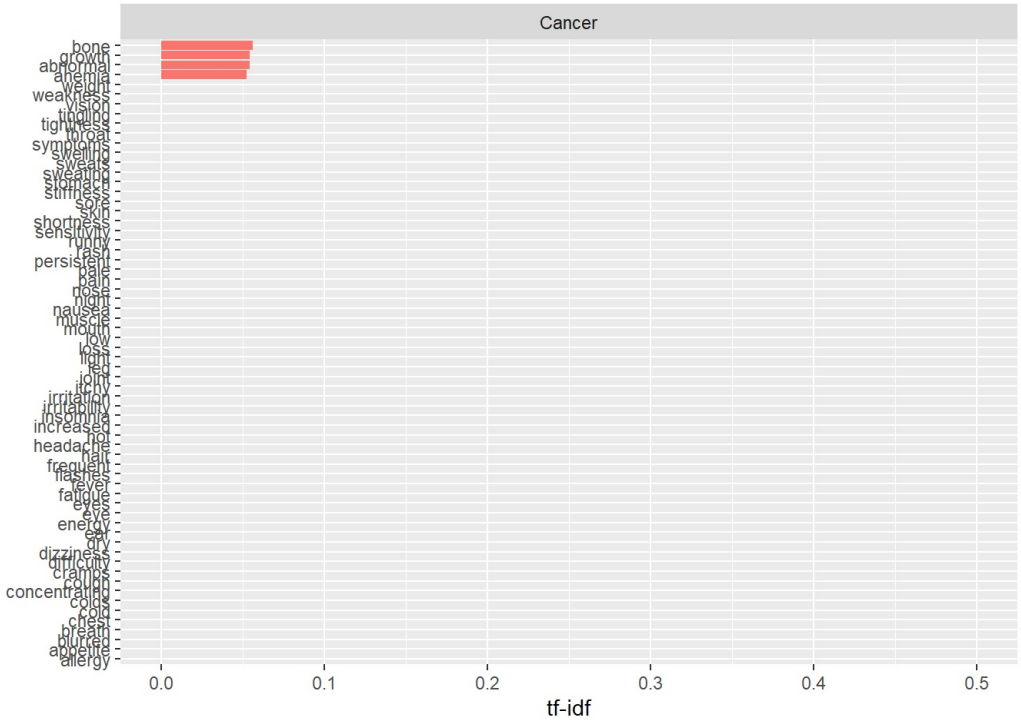
```
# Asthma
tf_idf %>%
  filter(str_detect(medical_condition, "Asthma")) %>%
  group_by(medical_condition) %>%
  slice_max(tf_idf, n = 12) %>%
  ungroup() %>%
  mutate(word = reorder(word, tf_idf)) %>%
  ggplot(aes(tf_idf, word, fill = medical_condition)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ medical_condition, scales = "free") +
  labs(x = "tf-idf", y = NULL) +
  scale_x_continuous(limits = c(0, .5))
```



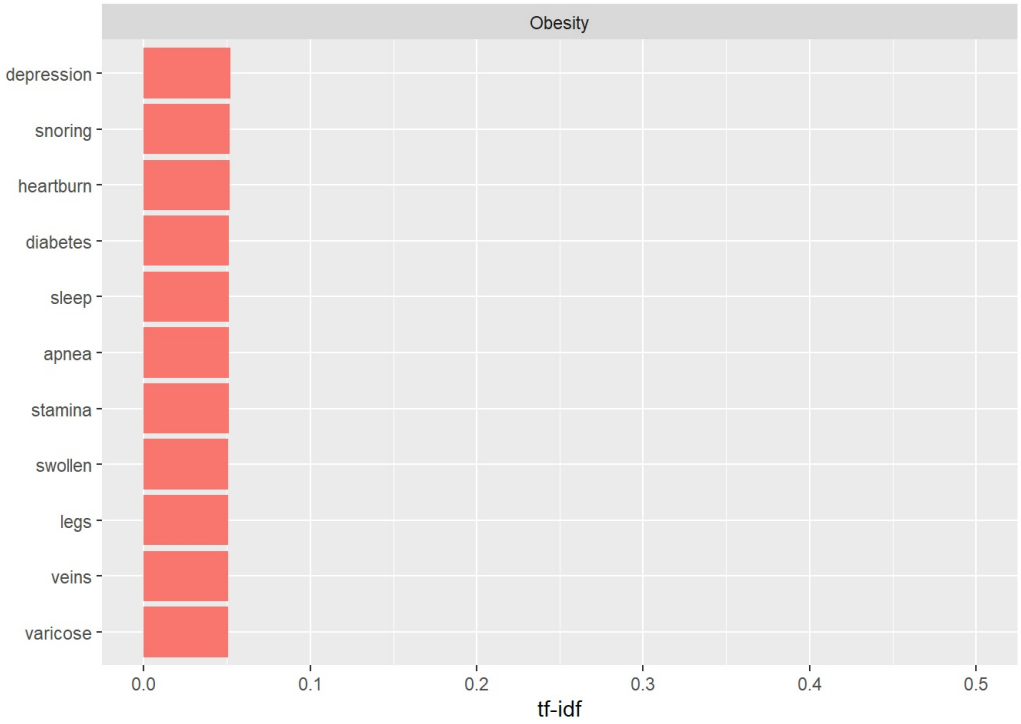
```
# Diabetes
tf_idf %>%
  filter(str_detect(medical_condition, "Diabetes")) %>%
  group_by(medical_condition) %>%
  slice_max(tf_idf, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, tf_idf)) %>%
  ggplot(aes(tf_idf, word, fill = medical_condition)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ medical_condition, scales = "free") +
  labs(x = "tf-idf", y = NULL) +
scale_x_continuous(limits = c(0, .5))
```




```
# Cancer
tf_idf %>%
  filter(str_detect(medical_condition, "Cancer")) %>%
  group_by(medical_condition) %>%
  slice_max(tf_idf, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, tf_idf)) %>%
  ggplot(aes(tf_idf, word, fill = medical_condition)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ medical_condition, scales = "free") +
  labs(x = "tf-idf", y = NULL) +
  scale_x_continuous(limits = c(0, .5))
```




```
# Obesity
tf_idf %>%
  filter(str_detect(medical_condition, "Obesity")) %>%
  group_by(medical_condition) %>%
  slice_max(tf_idf, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, tf_idf)) %>%
  ggplot(aes(tf_idf, word, fill = medical_condition)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ medical_condition, scales = "free") +
  labs(x = "tf-idf", y = NULL) +
  scale_x_continuous(limits = c(0, .5))
```



You may see that for every medical condition the graphs look sort of unison with the other symptoms. This is probably because my symptom data was generated by code and not actual data taken from the dataset. This part is a little unfortunate for variety reasons, but later we see some variety so it's not a bad dataset to view differences

9.2.2 Finding the correlation between words and medical conditions

```
library (widyr)
```

```
## Warning: package 'widyr' was built under R version 4.4.2
```

```
medical_condition_cors <- words_by_condition %>%
  pairwise_cor(word, medical_condition, n, sort = TRUE)
```

The correlations were all 1 or about near it.

Displaying the correlations between medical conditions and their symptoms

```
library(ggraph)
```

```
## Warning: package 'ggraph' was built under R version 4.4.2
```

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 4.4.2
```

```
##
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:purrr':
##
##     compose, simplify
```

```
## The following object is masked from 'package:tidyr':
##
##     crossing
```

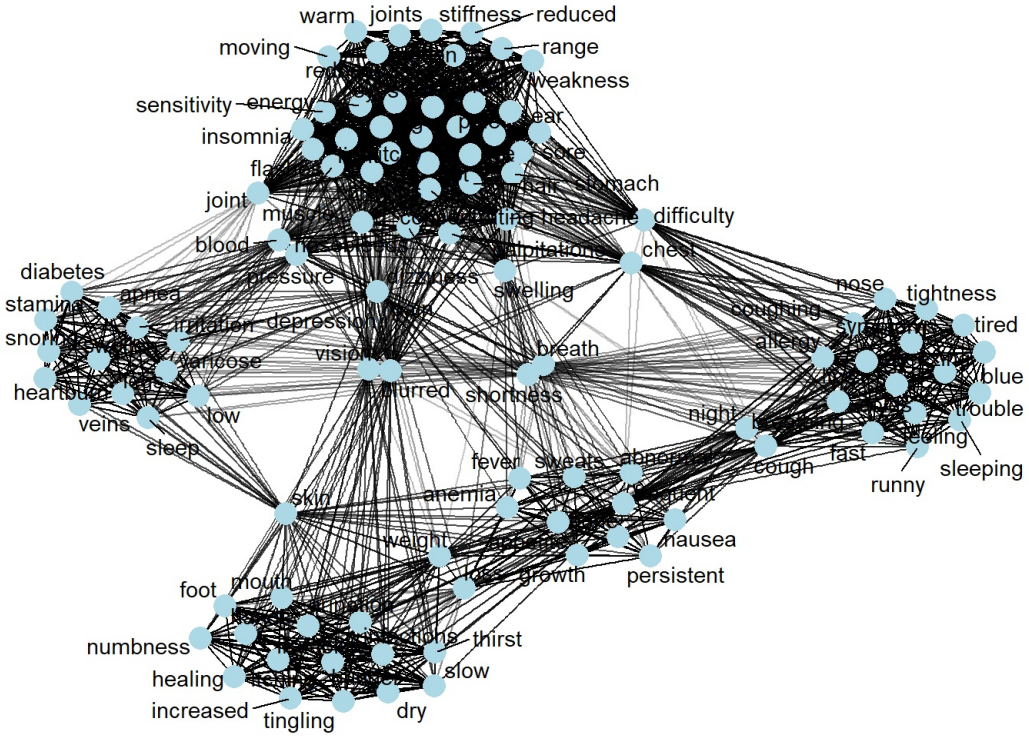
```
## The following objects are masked from 'package:dplyr':
##
##     as_data_frame, groups, union
```

```
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
```

```
## The following object is masked from 'package:base':
##
##     union
```

```
set.seed(2017)

medical_condition_cors %>%
  filter(correlation > 0.1) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "fr") +
  geom_edge_link(aes(edge_alpha = correlation), show.legend = FALSE) +
  geom_node_point(color = "lightblue", size = 5) +
  geom_node_text(aes(label = name), repel = TRUE) +
  theme_void()
```

```
## Warning: ggrepel: 3 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



This does look a bit like a mess, but it's interesting to see the correlation between medical conditions and their symptoms. The correlation is not very strong.

9.2.2 Topic Modeling

Filter words occurring at least 50 times across all conditions and create a document-term matrix (DTM)

```
word_medical <- filtered_data %>%
  group_by(word) %>%
  mutate(word_total = n()) %>%
  ungroup() %>%
  filter(word_total > 50)
medical_dtm <- word_medical %>%
  unite(document, medical_condition, id, sep = "_") %>%
  count(document, word) %>%
  cast_dtm(document, word, n)
```

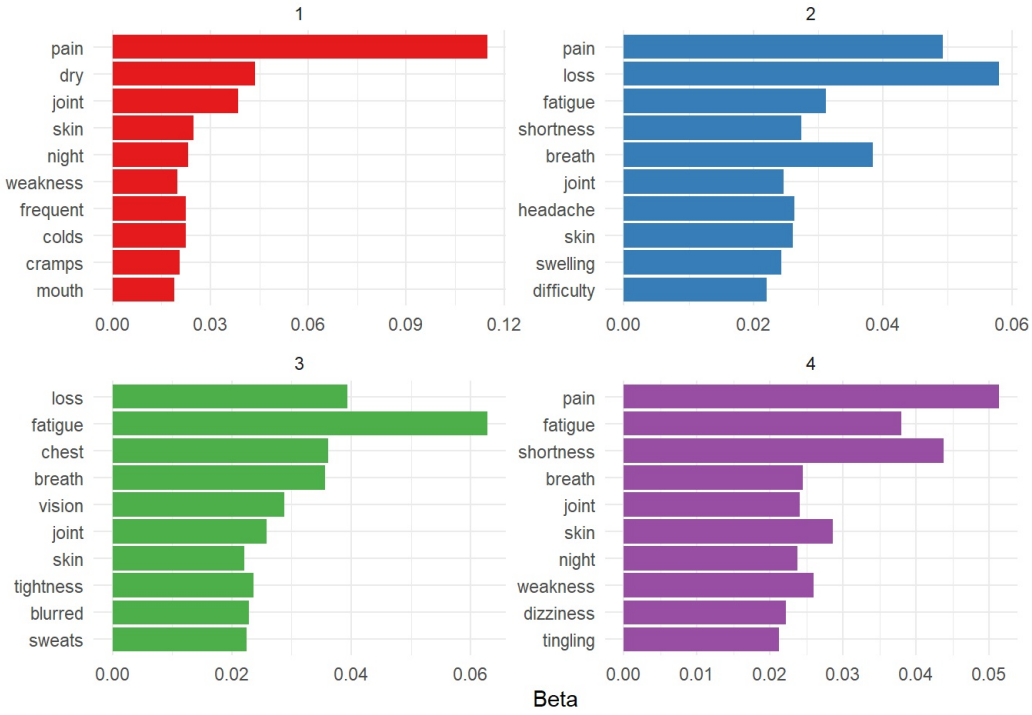
Discovering what each conditions result in which symptoms the most

Fit the LDA model with 4 topics, # View the topics, and visualize the topics

```
library(topicmodels)
```

```
## Warning: package 'topicmodels' was built under R version 4.4.2
```

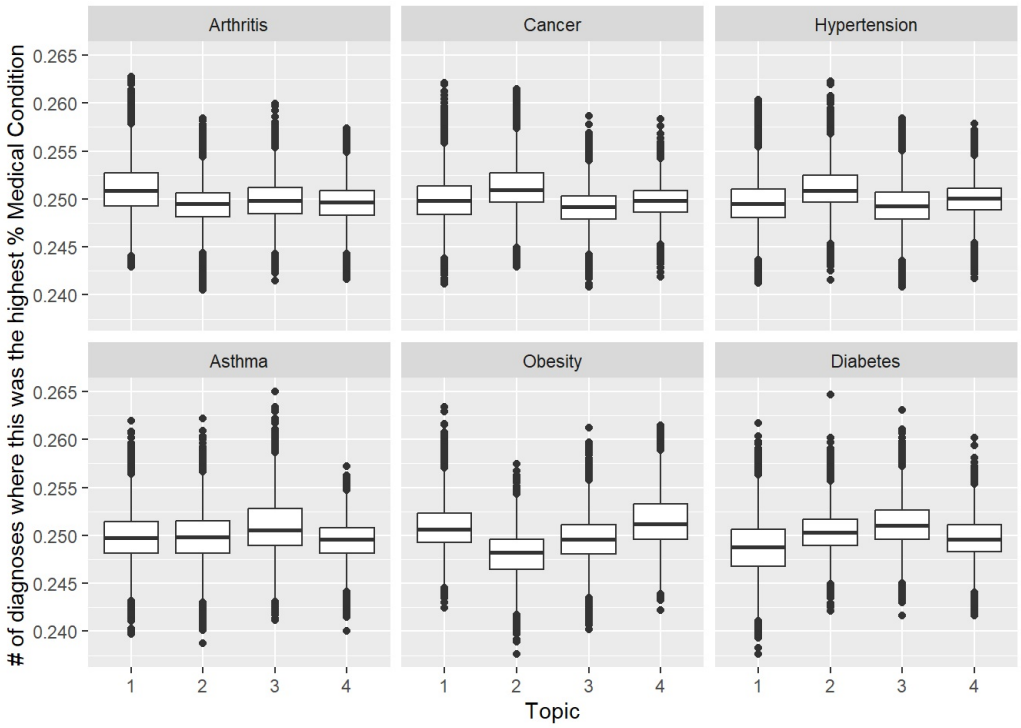
```
medical_lda <- LDA(medical_dtm, k = 4, control = list(seed = 2016))
medical_topics <- tidy(medical_lda, matrix = "beta")
medical_lda %>%
  tidy(matrix = "beta") %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip() +
  labs(x = NULL, y = "Beta") +
  scale_fill_brewer(palette = "Set1") +
  theme_minimal()
```



These bar graphs show the top 10 symptoms for each topic. The topics are 1-4. The symptoms are not very different from each other, but they do have some variety.

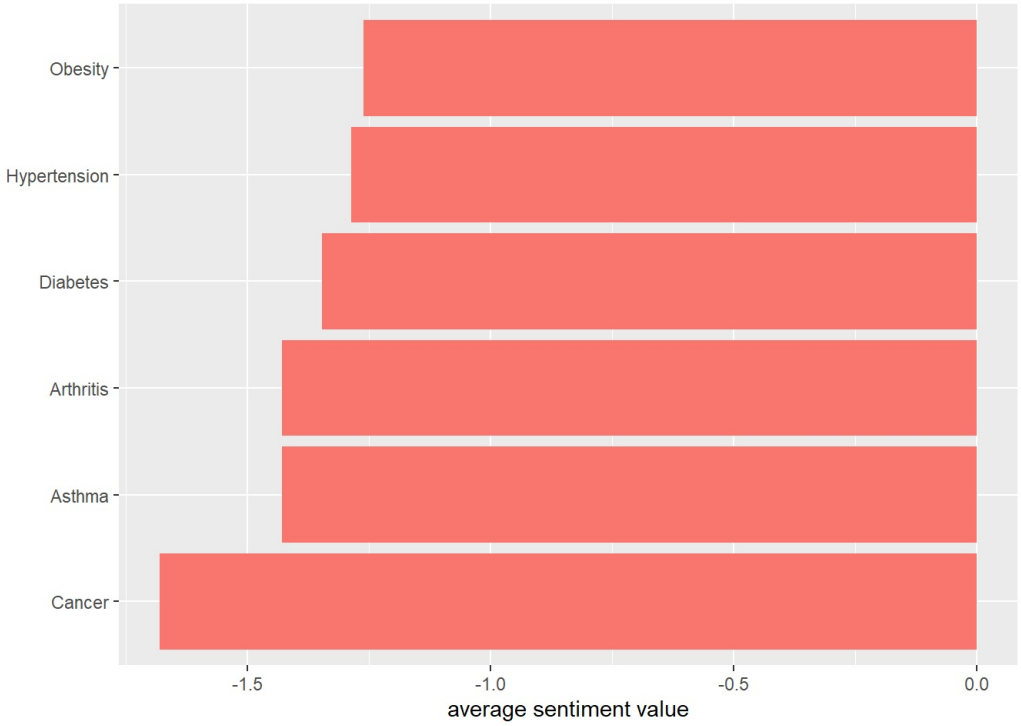
Examining how many diagnoses from each medical condition we have and which has higher gamma for each condition

```
medical_lda %>%
tidy(matrix = "gamma") %>%
  separate(document, c("medical_condition", "id"), sep = "_") %>%
  mutate(medical_condition = reorder(medical_condition, gamma * topic)) %>%
  ggplot(aes(factor(topic), gamma)) +
  geom_boxplot() +
  facet_wrap(~ medical_condition) +
  labs(x = "Topic",
       y = "# of diagnoses where this was the highest % Medical Condition")
```



9.3 Sentiment Analysis

```
medical_condition_sentiments <- words_by_condition %>%
  inner_join(get_sentiments("afinn"), by = "word") %>%
  group_by(medical_condition) %>%
  summarize(value=sum(value * n) / sum(n))



medical_condition_sentiments %>%
  mutate(medical_condition = reorder(medical_condition, value)) %>%
  ggplot(aes(value, medical_condition, fill = value > 0)) +
  geom_col(show.legend = FALSE) +
  labs(x= "average sentiment value",
       y = NULL)
```



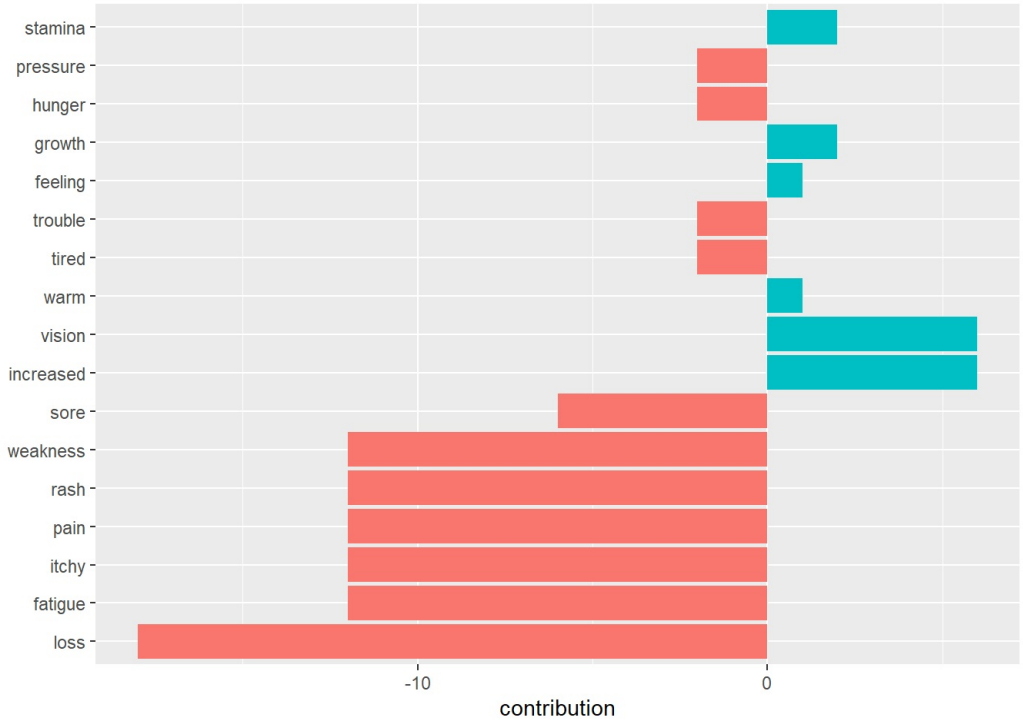
Obesity has the better sentiment value, while Cancer has the worst sentiment value.

Sentiment Analysis by word

```
contributions <- words_by_condition %>%
  inner_join(get_sentiments("afinn"), by = "word") %>%
  group_by(medical_condition, word) %>%
  summarize(occurences = n(),
            contribution = sum(value))
```

```
## `summarise()` has grouped output by 'medical_condition'. You can override using
## the `.groups` argument.
```

```
contributions %>%
  slice_max(abs(contribution), n = 10) %>%
  mutate(word = reorder(word, contribution)) %>%
  ggplot(aes(contribution, word, fill = contribution > 0)) +
  geom_col(show.legend = FALSE) +
  labs(y = NULL)
```



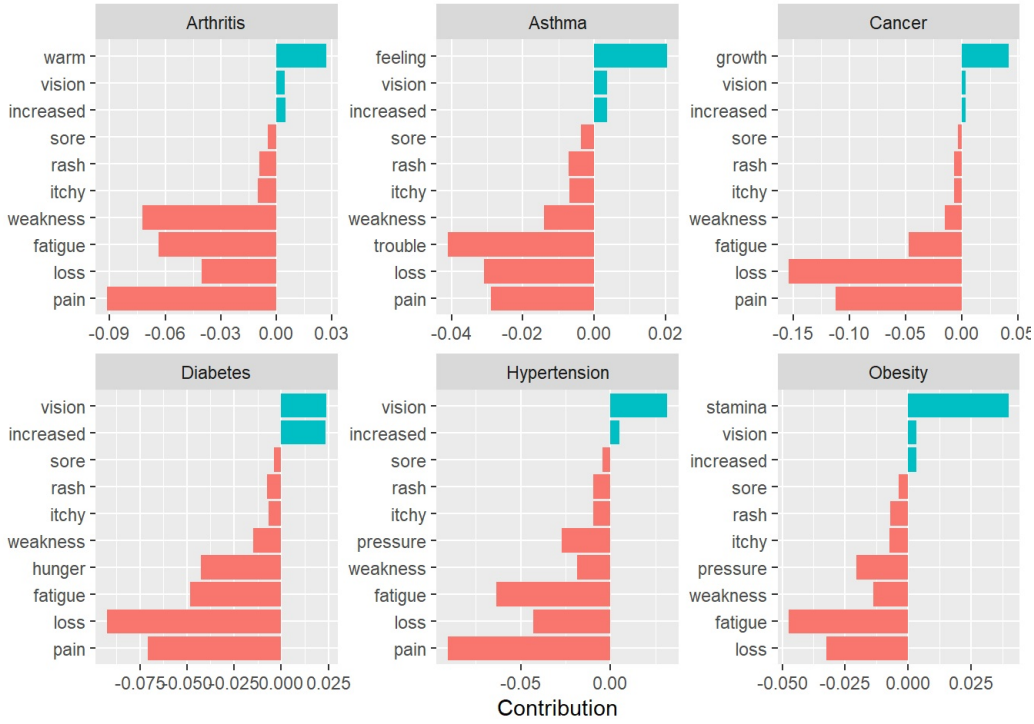
The top contributors are shown through my graph. Stamina, growth, feeling, warm, vision, and increased are positive contributors, while the rest appear to be vastly negative.

```
# Top sentiment words
top_sentiment_words <- words_by_condition %>%
  inner_join(get_sentiments("afinn"), by = "word") %>%
  mutate(contribution = value * n / sum(n))
```

From this tibble, the top sentiment words are shown with their contribution to the sentiment value. They range from pain being highest, then loss, then fatigue, and so on. Cancer and obesity come hand in hand with these symptoms

Display each medical condition and their top sentiment words

```
top_sentiment_words %>%
  group_by(medical_condition) %>%
  slice_max(contribution, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, contribution)) %>%
  ggplot(aes(contribution, word, fill = contribution > 0)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ medical_condition, scales = "free") +
  labs(x = "Contribution",
       y = NULL)
```



Pain seems to be the most negative contribution, while vision and stamina are the highest. It's a bit difficult to consider "increased" as positive because it could be paired with "pain" or "fatigue." This dataset is a little more confusing when considering that aspect.

Sentiment analysis by medical conditions

```
sentiment_conditions <- words_by_condition %>%
  inner_join(get_sentiments("afinn"), by = "word") %>%
  group_by(medical_condition) %>%
  summarize(sentiment = mean(value),
            words = n()) %>%
      ungroup() %>%
        filter(words >= 5)

sentiment_conditions %>%
  arrange(desc(sentiment))
```

```
## # A tibble: 6 × 3
##   medical_condition sentiment words
##   <chr>                <dbl> <int>
## 1 Cancer               -1       10
## 2 Obesity              -1       11
## 3 Arthritis            -1.1     10
## 4 Asthma               -1.25    12
## 5 Hypertension         -1.3     10
## 6 Diabetes             -1.4     10
```

```
print_condition <- function(group, data) {
  data %>%
    filter(medical_condition == group) %>%
    inner_join(get_sentiments("afinn"), by = "word") %>%
    group_by(medical_condition) %>%
    summarize(sentiment = mean(value),
              words = n()) %>%
    ungroup() %>%
    filter(words >= 5)
}
print_condition("Cancer", words_by_condition)
```

```
## # A tibble: 1 × 3
##   medical_condition sentiment words
##   <chr>                <dbl> <int>
## 1 Cancer                  -1    10
```

Overall, the sentiment analysis shows that Obesity has the best sentiment value, while Cancer has the worst sentiment value. Pain seems to big a very big symptom of any of the medical conditions.