Shakeer Sarwar Project #7
2/25/2022
DA 420

Sentiment Analysis with R (Choice #1)

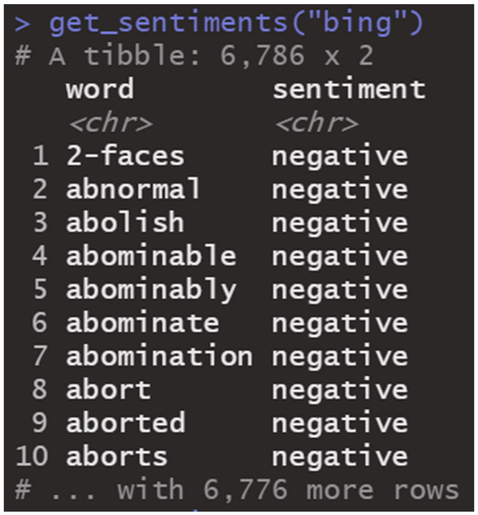
Code

```
if (packageVersion("devtools") < 1.6) {
  install.packages("devtools")
}

devtools::install_github("bradleyboehmke/harrypotter")
install.packages("tidyverse")
install.packages("tidytext")
install.packages("textdata")

library(dplyr)          # required package
library(tidyverse)      # data manipulation & plotting
library(stringr)        # text cleaning and regular expressions
library(tidytext)       # provides additional text mining functions
library(textdata)       # required package
library(harrypotter)    # provides the first seven novels of the Harry Potter series

# to see the individual lexicons try
get_sentiments("afinn")
get_sentiments("bing")
get_sentiments("nrc")

titles <- c("Philosopher's Stone", "Chamber of Secrets", "Prisoner of Azkaban",
            "Goblet of Fire", "Order of the Phoenix", "Half-Blood Prince",
            "Deathly Hallows")

books <- list(philosophers_stone, chamber_of_secrets, prisoner_of_azkaban,
              goblet_of_fire, order_of_the_phoenix, half_blood_prince,
              deathly_hallows)

series <- tibble()

for(i in seq_along(titles)) {

  clean <- tibble(chapter = seq_along(books[[i]]),
                  text = books[[i]]) %>%
    unnest_tokens(word, text) %>%
    mutate(book = titles[i]) %>%
    select(book, everything())
  series <- rbind(series, clean)
}

# set factor to keep books in order of publication
series$book <- factor(series$book, levels = rev(titles))

series

series %>%
  right_join(get_sentiments("nrc")) %>%
  filter(!is.na(sentiment)) %>%
  count(sentiment, sort = TRUE)

series %>%
  group_by(book) %>%
  mutate(word_count = 1:n(),
         index = word_count %/% 500 + 1) %>%
  inner_join(get_sentiments("bing")) %>%
  count(book, index = index , sentiment) %>%
  ungroup() %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative,
         book = factor(book, levels = titles)) %>%
  ggplot(aes(index, sentiment, fill = book)) +
  geom_bar(alpha = 0.5, stat = "identity", show.legend = FALSE) +
  facet_wrap(~ book, ncol = 2, scales = "free_x")

afinn <- series %>%
  group_by(book) %>%
  mutate(word_count = 1:n(),
         index = word_count %/% 500 + 1) %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(book, index) %>%
  summarise(sentiment = sum(value)) %>%
  mutate(method = "AFINN")

bing_and_nrc <- bind_rows(series %>%
                            group_by(book) %>%
                            mutate(word_count = 1:n(),
                                   index = word_count %/% 500 + 1) %>%
                            inner_join(get_sentiments("bing")) %>%
                            mutate(method = "Bing"),
                          series %>%
                            group_by(book) %>%
                            mutate(word_count = 1:n(),
                                   index = word_count %/% 500 + 1) %>%
                            inner_join(get_sentiments("nrc") %>%
                                         filter(sentiment %in% c("positive", "negative"))) %>%
                          mutate(method = "NRC")) %>%
  count(book, method, index = index , sentiment) %>%
  ungroup() %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative) %>%
  select(book, index, method, sentiment)

bind_rows(afinn,
          bing_and_nrc) %>%
  ungroup() %>%
  mutate(book = factor(book, levels = titles)) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_bar(alpha = 0.8, stat = "identity", show.legend = FALSE) +
  facet_grid(book ~ method)

bing_word_counts <- series %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()

bing_word_counts

bing_word_counts %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ggplot(aes(reorder(word, n), n, fill = sentiment)) +
  geom_bar(alpha = 0.8, stat = "identity", show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to sentiment", x = NULL) +
  coord_flip()

tibble(text = philosophers_stone) %>%
  unnest_tokens(sentence, text, token = "sentences")

ps_sentences <- tibble(chapter = 1:length(philosophers_stone),
                       text = philosophers_stone) %>%
  unnest_tokens(sentence, text, token = "sentences")

book_sent <- ps_sentences %>%
  group_by(chapter) %>%
  mutate(sentence_num = 1:n(),
         index = round(sentence_num / n(), 2)) %>%
  unnest_tokens(word, sentence) %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(chapter, index) %>%
  summarise(sentiment = sum(score, na.rm = TRUE)) %>%
  arrange(desc(sentiment))
```

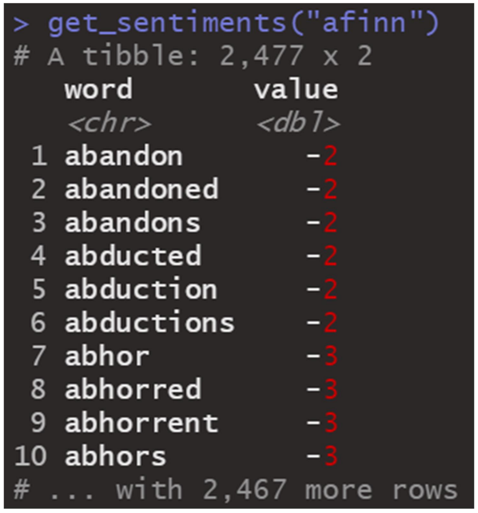
book_sent

# Continued receiving errors even after loading the dplyr package as recommended
# in the Projecr Notes Announcement. I will analyze the heat map shown in the
# tutorial website.

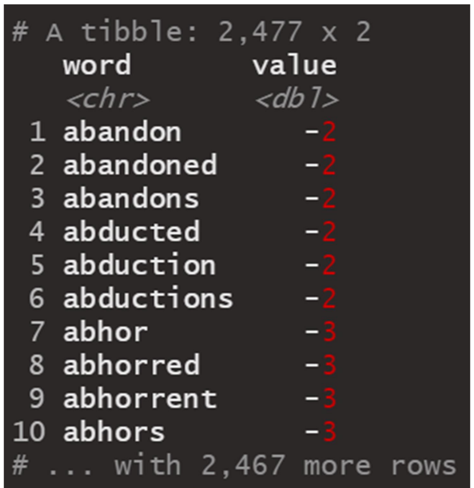
Output



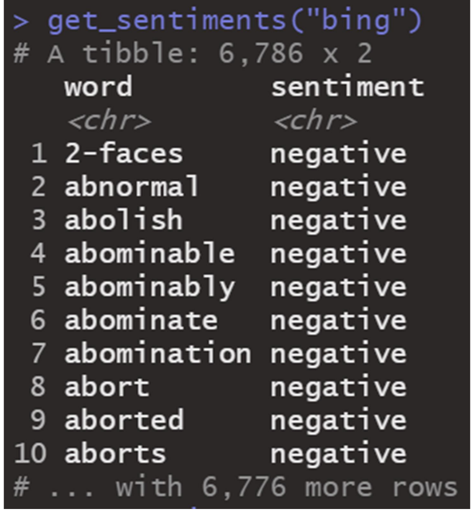
```
> get_sentiments("bing")
# A tibble: 6,786 x 2
   word        sentiment
   <chr>       <chr>
 1 2-faces     negative
 2 abnormal    negative
 3 abolish     negative
 4 abominable  negative
 5 abominably  negative
 6 abominate   negative
 7 abomination negative
 8 abort       negative
 9 aborted     negative
10 aborts      negative
# ... with 6,776 more rows
```



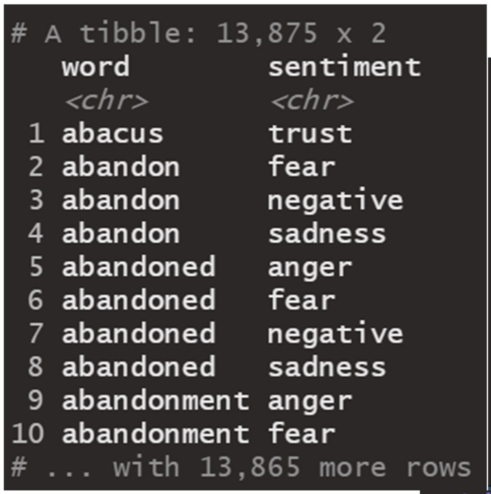
```
> get_sentiments("afinn")
# A tibble: 2,477 x 2
   word       value
   <chr>      <dbl>
 1 abandon       -2
 2 abandoned     -2
 3 abandons      -2
 4 abducted      -2
 5 abduction     -2
 6 abductions    -2
 7 abhor         -3
 8 abhorred      -3
 9 abhorrent     -3
10 abhors        -3
# ... with 2,467 more rows
```



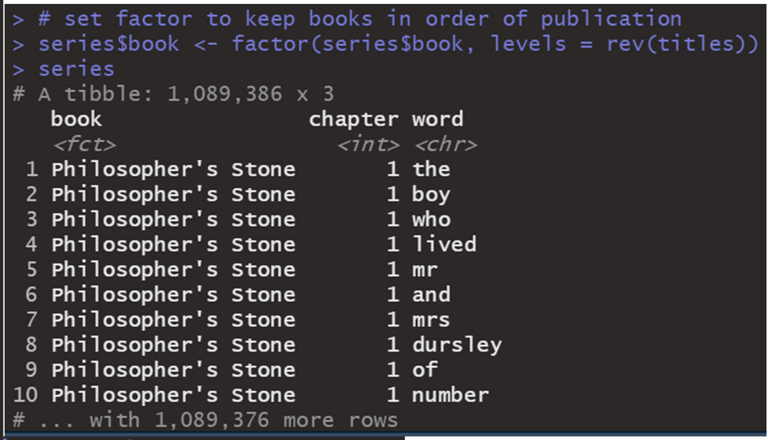
```
# A tibble: 2,477 x 2
   word       value
   <chr>      <dbl>
 1 abandon       -2
 2 abandoned     -2
 3 abandons      -2
 4 abducted      -2
 5 abduction     -2
 6 abductions    -2
 7 abhor         -3
 8 abhorred      -3
 9 abhorrent     -3
10 abhors        -3
# ... with 2,467 more rows
```



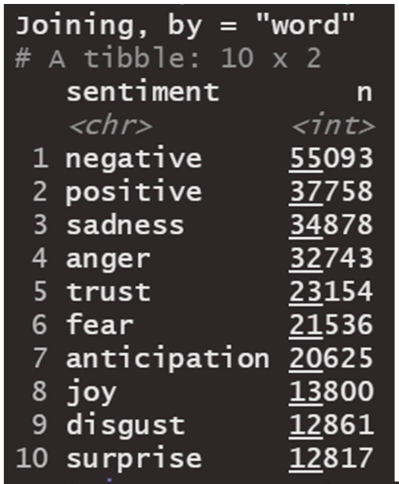
```
> get_sentiments("bing")
# A tibble: 6,786 x 2
   word        sentiment
   <chr>       <chr>
 1 2-faces     negative
 2 abnormal    negative
 3 abolish     negative
 4 abominable  negative
 5 abominably  negative
 6 abominate   negative
 7 abomination negative
 8 abort       negative
 9 aborted     negative
10 aborts      negative
# ... with 6,776 more rows
```



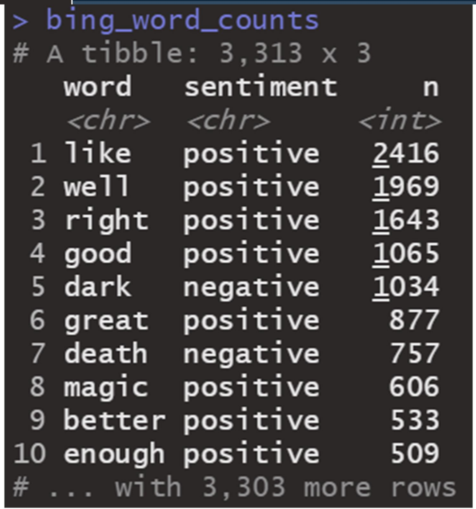
```
# A tibble: 13,875 x 2
   word        sentiment
   <chr>       <chr>    
 1 abacus      trust    
 2 abandon     fear     
 3 abandon     negative 
 4 abandon     sadness  
 5 abandoned   anger    
 6 abandoned   fear     
 7 abandoned   negative 
 8 abandoned   sadness  
 9 abandonment anger    
10 abandonment fear     
# ... with 13,865 more rows
```



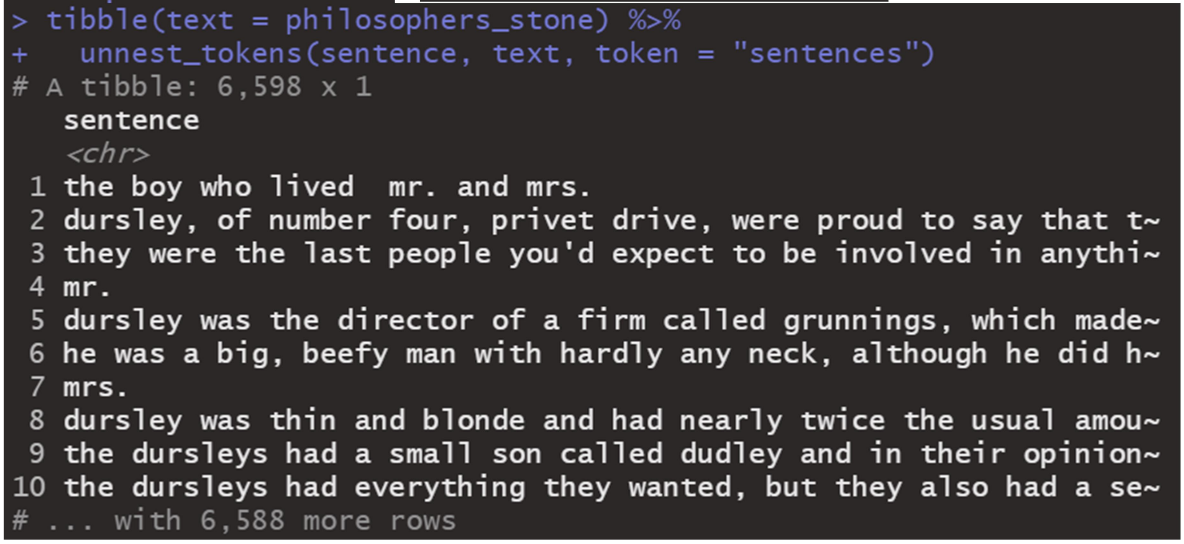
```
> # set factor to keep books in order of publication
> series$book <- factor(series$book, levels = rev(titles))
> series
# A tibble: 1,089,386 x 3
   book                chapter word   
   <fct>                 <int> <chr>  
 1 Philosopher's Stone       1 the    
 2 Philosopher's Stone       1 boy    
 3 Philosopher's Stone       1 who    
 4 Philosopher's Stone       1 lived  
 5 Philosopher's Stone       1 mr     
 6 Philosopher's Stone       1 and    
 7 Philosopher's Stone       1 mrs    
 8 Philosopher's Stone       1 dursley
 9 Philosopher's Stone       1 of     
10 Philosopher's Stone       1 number 
# ... with 1,089,376 more rows
```



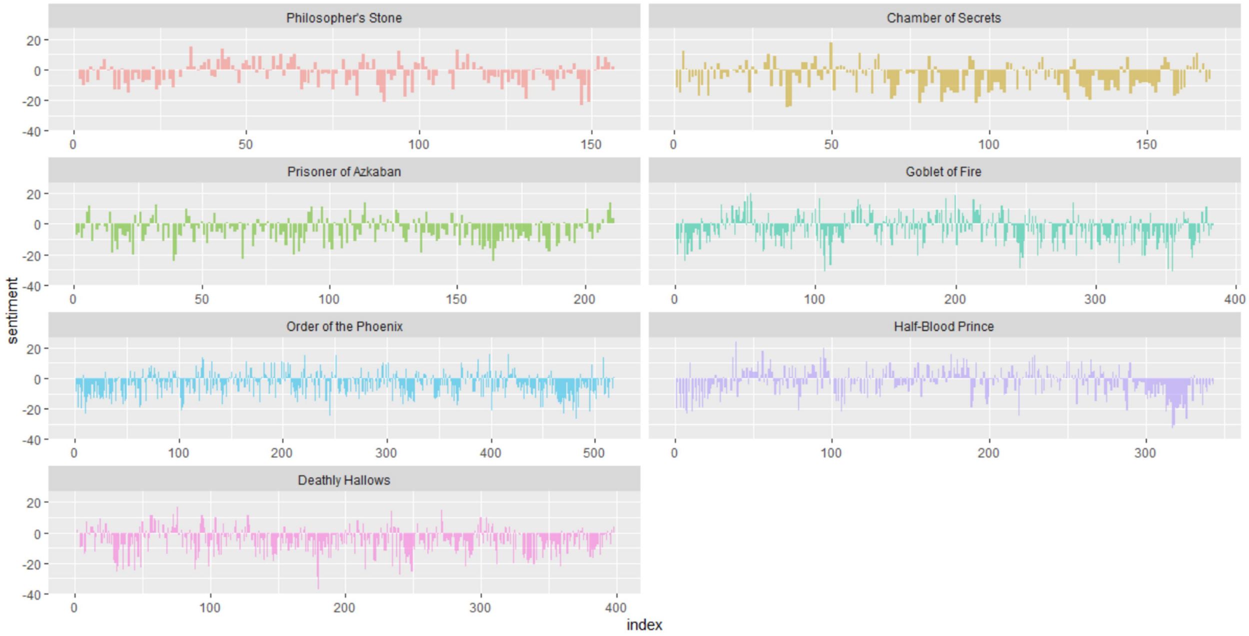
```
Joining, by = "word"
# A tibble: 10 x 2
   sentiment        n
   <chr>        <int>
 1 negative     55093
 2 positive     37758
 3 sadness      34878
 4 anger        32743
 5 trust        23154
 6 fear         21536
 7 anticipation 20625
 8 joy          13800
 9 disgust      12861
10 surprise     12817
```



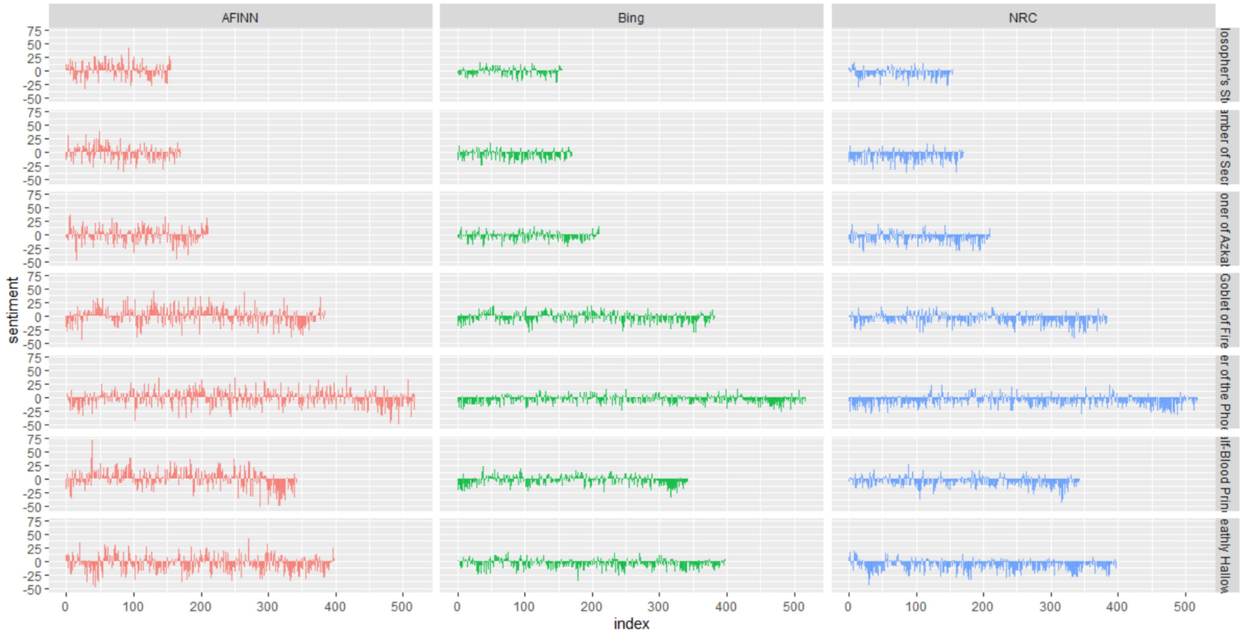
```
> bing_word_counts
# A tibble: 3,313 x 3
   word   sentiment     n
   <chr>  <chr>     <int>
 1 like   positive   2416
 2 well   positive   1969
 3 right  positive   1643
 4 good   positive   1065
 5 dark   negative   1034
 6 great  positive    877
 7 death  negative    757
 8 magic  positive    606
 9 better positive    533
10 enough positive    509
# ... with 3,303 more rows
```



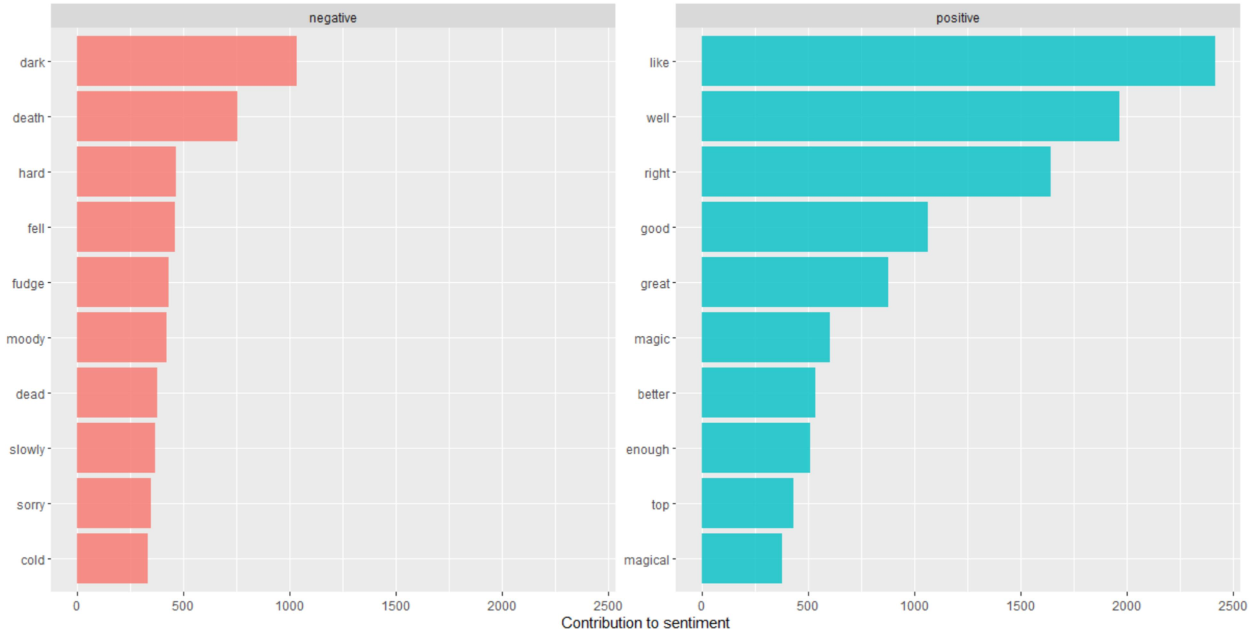
```
> tibble(text = philosophers_stone) %>%
+   unnest_tokens(sentence, text, token = "sentences")
# A tibble: 6,598 x 1
   sentence                                                            
   <chr>                                                               
 1 the boy who lived  mr. and mrs.                                     
 2 dursley, of number four, privet drive, were proud to say that t~
 3 they were the last people you'd expect to be involved in anythi~
 4 mr.                                                                 
 5 dursley was the director of a firm called grunnings, which made~
 6 he was a big, beefy man with hardly any neck, although he did h~
 7 mrs.                                                                
 8 dursley was thin and blonde and had nearly twice the usual amou~
 9 the dursleys had a small son called dudley and in their opinion~
10 the dursleys had everything they wanted, but they also had a se~
# ... with 6,588 more rows
```



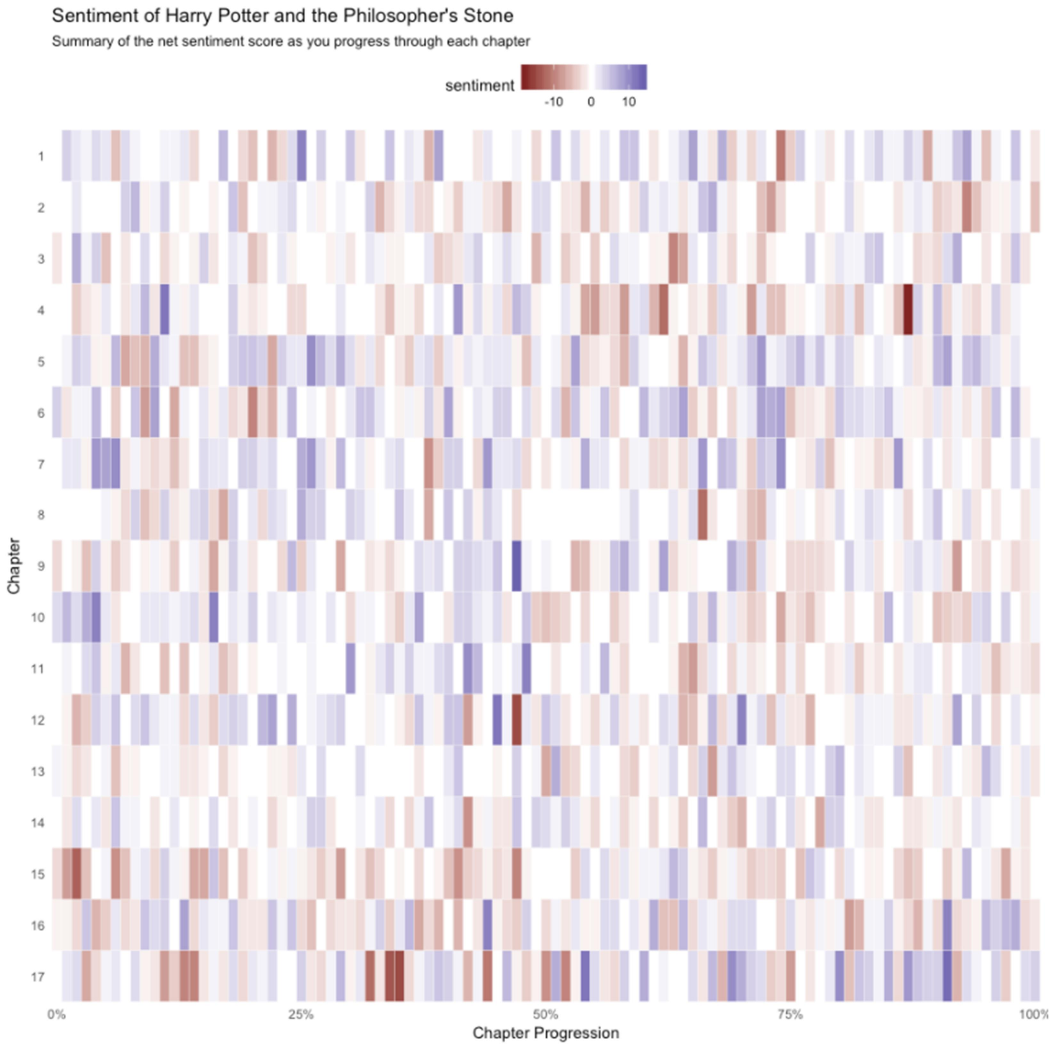
The above plot shows the overall sentiment of each Harry Potter novel. The x-axis shows the number of pages and the y-axis shows the sentiment index, with positive values representing net positive sentiment of words per interval recorded and negative values representing net negative sentiment of words per recorded interval. This index is recorded at an interval every 500 words, or approximately 2 pages. From our plots, we notice that all the novels have a greater number of net negative sentiments per 500 word intervals. The most positive novel in the series seems to be the first one, The Philosopher's Stone. It is also interesting to note that only The Philosopher's Stone, Prisoner of Azkaban, and Deathly Hallows seem to end their novels in a positive sentiment. Among these three "happy endings", The Prisoner of Azkaban has the highest peak positive sentiment ending while the Deathly Hallows has the lowest peak positive sentiment ending. The last 50 pages of the Half-Blood Prince seem to have the most negative ending along with the longest duration of net negative sentiment among all the books as that is when (spoilers) the beloved Dumbledore dies.



The above plot compares the three different sentiment lexicons – AFINN, Bing, and NRC – across all seven novels in the Harry Potter series so that we may observe the differences among the three lexicon options. Once again, the x-axis represents the number of pages while the y-axis represents the net sentiment per recorded interval, which is once again each 500 words. We immediately observe that AFINN has by far the most recorded intervals of net positive sentiment throughout the Harry Potter series while conversely, NRC has the fewest recorded intervals of net positive sentiment. Bing seems to be a good balance between these two as it does not veer to an extreme positive or extreme negative. AFFIN seems to also be the most volatile of the three options as it has the highest observed peak positive sentiments per recorded interval as well as the lowest observed peak negatives per recorded interval. Lastly, this plot also gives us a good idea of the length of each book, as we can see that The Order of the Phoenix is by far the longest novel in the series.



The above graph shows the most common negative sentiment and positive sentiment words and their respective number of reappearances throughout the Harry Potter series. Of the negative sentiment words, 'dark' and 'death' are the only words that reoccur over 500 times throughout the series, with over 1000 and over 750 reoccurrences each respectively. It is interesting that 'fudge' was categorized as a negative sentiment word with around 400 reoccurrences. The reoccurrence of positive sentiment words is much greater than that of the negative variety as we see 'like' topping the positives chart with nearly 2400 reoccurrences throughout the series and 'well' with nearly 2000 reoccurrences itself. Based upon these top words sentiment lists, it seems that there is a much greater volume of positive sentiment words in the novels than there are negative sentiments words.



This last plot is a heatmap that I was unfortunately not able to reproduce in my own R code, as I continuously received the error when working with arrange(desc(sentiment)) saying that Column `index` is not found. I followed the advice from the notes provided in the announcement for this assignment by loading the dplyr package but that still did not resolve the issue. Thus, as per your recommendation, I copied the heatmap shown in the end of the R tutorial webpage to analyze it.

The heatmap above indexes our net sentiment for The Philosopher's Stone into a red-blue color scale in which red is extreme negative and blue is extreme positive sentiment. Here, the x-axis represents the percent of progression through a single chapter while the y-axis represents the chapter of interest within the novel. We can observe from this heatmap that most of the positive sentiments tend to take place in the earlier half of the novel as well as in the earlier half of the chapters themselves. Conversely, most of the negative sentiments seem to appear in the latter half of the novel as well as the latter half of the chapters. Chapter 15 seems to be the most negative while Chapter 17 seems to have the greatest peak of negative sentiment. Chapter 17 is also an exception in that it ends with the highest positive sentiment observed, but that is expected as it is the end of the novel. Chapter 4 also stands out as one of the earlier chapters with high negative sentiment throughout.