Week 10 Assignment

waheeb Algabri and Farhana Akther

Introduction

Chapter 2 of Text Mining with R focuses on Sentiment Analysis. Our task for this assignment is to first obtain the main example code from the chapter and ensure it works in an R Markdown document, with a citation to the original code. Then, we are required to expand the code in two ways:

Using a different corpus of our choice. Adding at least one extra sentiment lexicon, which we can discover through research, potentially from another R package.

```
library(tidyverse)
library(tidytext)
library(textdata)
library(janeaustenr)
library(wordcloud)
library(reshape2)
library(gutenbergr)
```

Loading required Libraries

The sentiments datasets Obtain sentiment lexicons from three different sources: AFINN, Bing, and NRC.

```
afinn<- get_sentiments("afinn")
bing<- get_sentiments("bing")
nrc<-get_sentiments("nrc")
```

Sentiment analysis with inner join We use the `austen_books()` function from the `janeaustenr` package to extract text from Jane Austen's novels and prepare it for analysis by splitting it into individual words using the `unnest_tokens()` function.

```
tidy_books <- austen_books() %>%
  group_by(book) %>%
  mutate(
    linenumber = row_number(),
    chapter = cumsum(str_detect(text,
                                regex("^chapter [\\divxlc]",
                                      ignore_case = TRUE)))) %>%
  ungroup() %>%
  unnest_tokens(word, text)
```

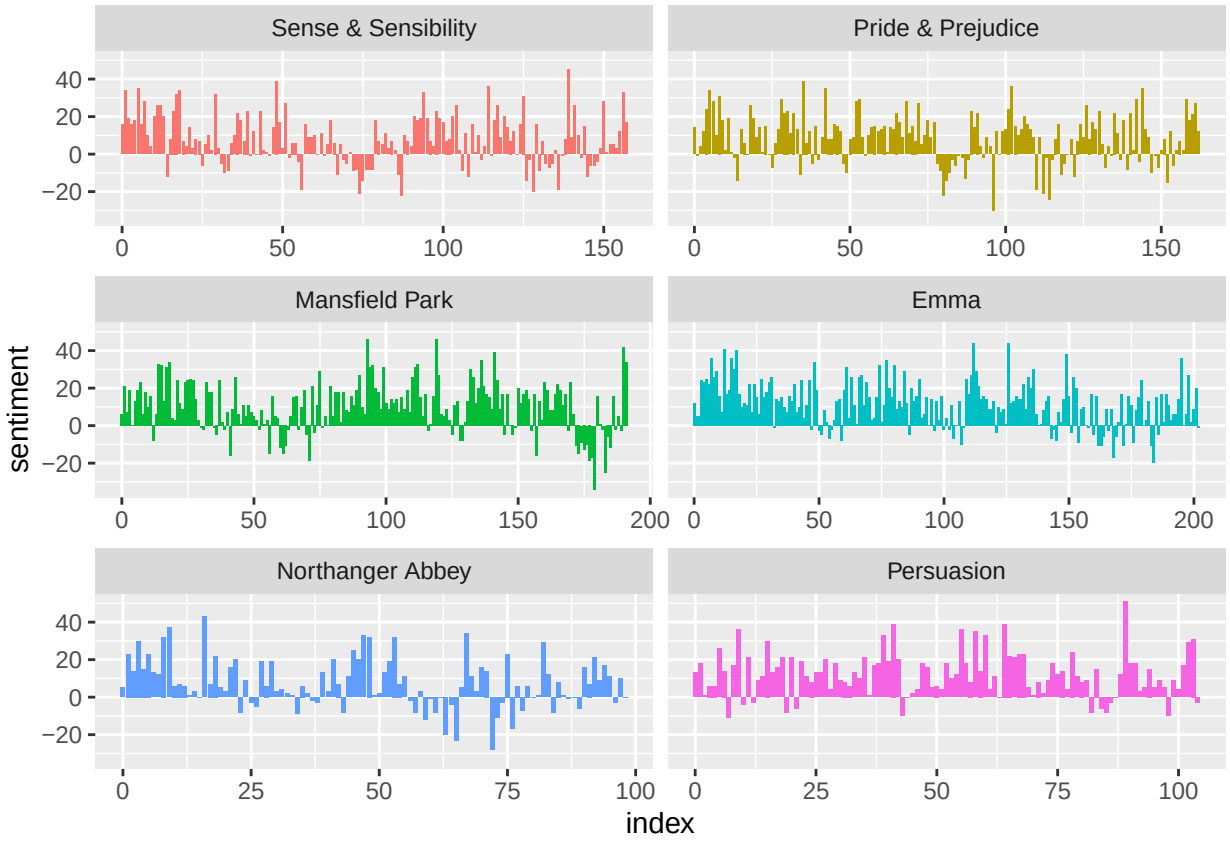
We filter the NRC sentiment lexicon to include only words with a "joy" sentiment, then use the `inner_join()` function to merge this lexicon with the tidy text data frame. The resulting data frame is then filtered to include only words from "Emma" and is counted using `count()` to show the frequency of words with a "joy" sentiment.

```
nrc_joy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")

tidy_books %>%
  filter(book == "Emma") %>%
  inner_join(nrc_joy) %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 301 x 2
##    word          n
##    <chr>     <int>
##  1 good        359
##  2 friend      166
##  3 hope        143
##  4 happy       125
##  5 love        117
##  6 deal         92
##  7 found        92
##  8 present      89
##  9 kind         82
## 10 happiness    76
## # ... with 291 more rows
```

We join the tidy text data frame with the Bing sentiment lexicon using `inner_join()`. We then use `count()` and `pivot_wider()` functions to count the number of positive and negative words in each book, grouped by sections of 80 lines. Finally, the `ggplot()` function is used to create a bar chart that shows the sentiment score over the plot trajectory of each novel. The chart is facet-wrapped by book, and the sentiment score is calculated as the difference between the number of positive and negative words.

```
jane_austen_sentiment <- tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(book, index = linenumber %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(sentiment = positive - negative)

ggplot(jane_austen_sentiment, aes(index, sentiment, fill = book)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~book, ncol = 2, scales = "free_x")
```



Comparing the three sentiment dictionaries

- Filter Data

```
pride_prejudice <- tidy_books %>%
  filter(book == "Pride & Prejudice")
```

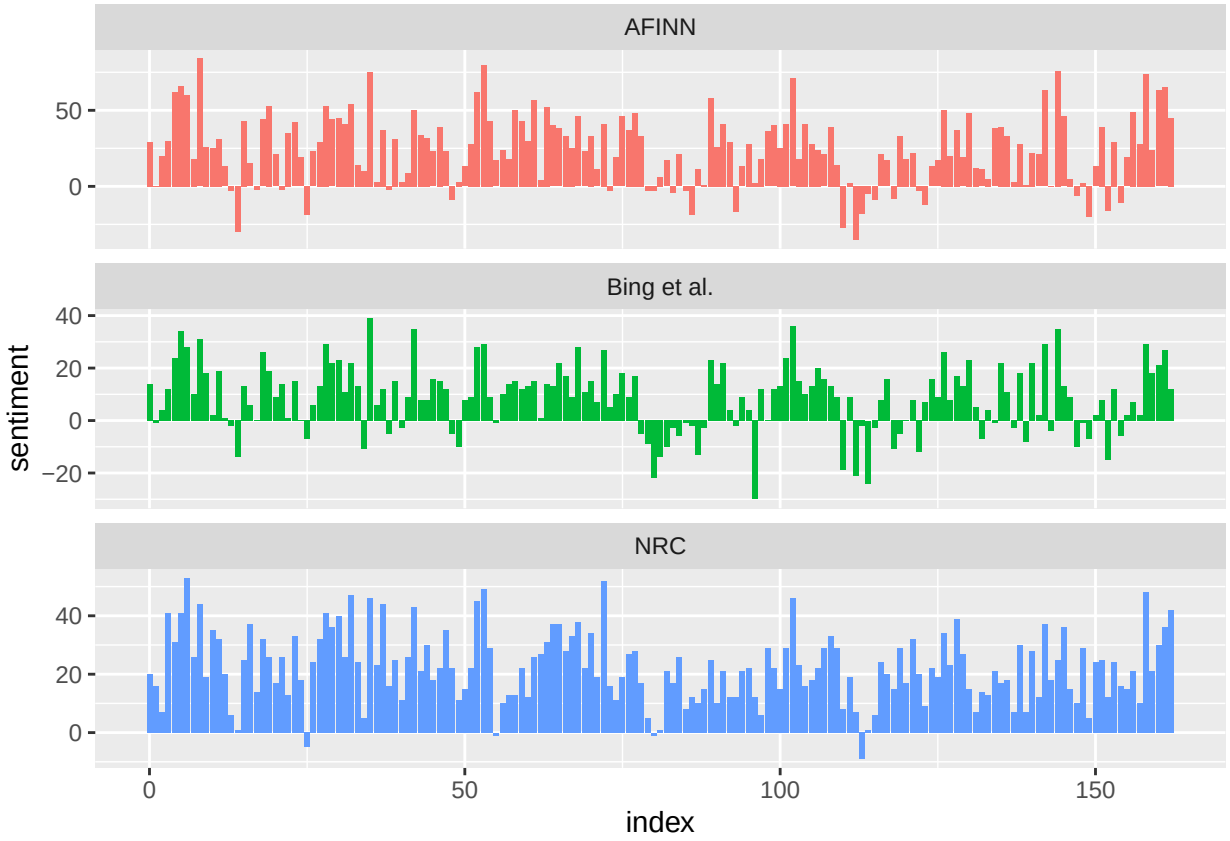
- Calculate Sentiment Scores

```
afinn <- pride_prejudice %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(index = linenumber %/% 80) %>%
  summarise(sentiment = sum(value)) %>%
  mutate(method = "AFINN")

bing_and_nrc <- bind_rows(
  pride_prejudice %>%
    inner_join(get_sentiments("bing")) %>%
    mutate(method = "Bing et al."),
  pride_prejudice %>%
    inner_join(get_sentiments("nrc") %>%
                 filter(sentiment %in% c("positive",
                                         "negative"))
    ) %>%
```

```
    mutate(method = "NRC")) %>%
  count(method, index = linenumber %/% 80, sentiment) %>%
  pivot_wider(names_from = sentiment,
              values_from = n,
              values_fill = 0) %>%
  mutate(sentiment = positive - negative)
```

- Visualize Sentiment Scores

```
bind_rows(afinn,
          bing_and_nrc) %>%
  ggplot(aes(index, sentiment, fill = method)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~method, ncol = 1, scales = "free_y")
```

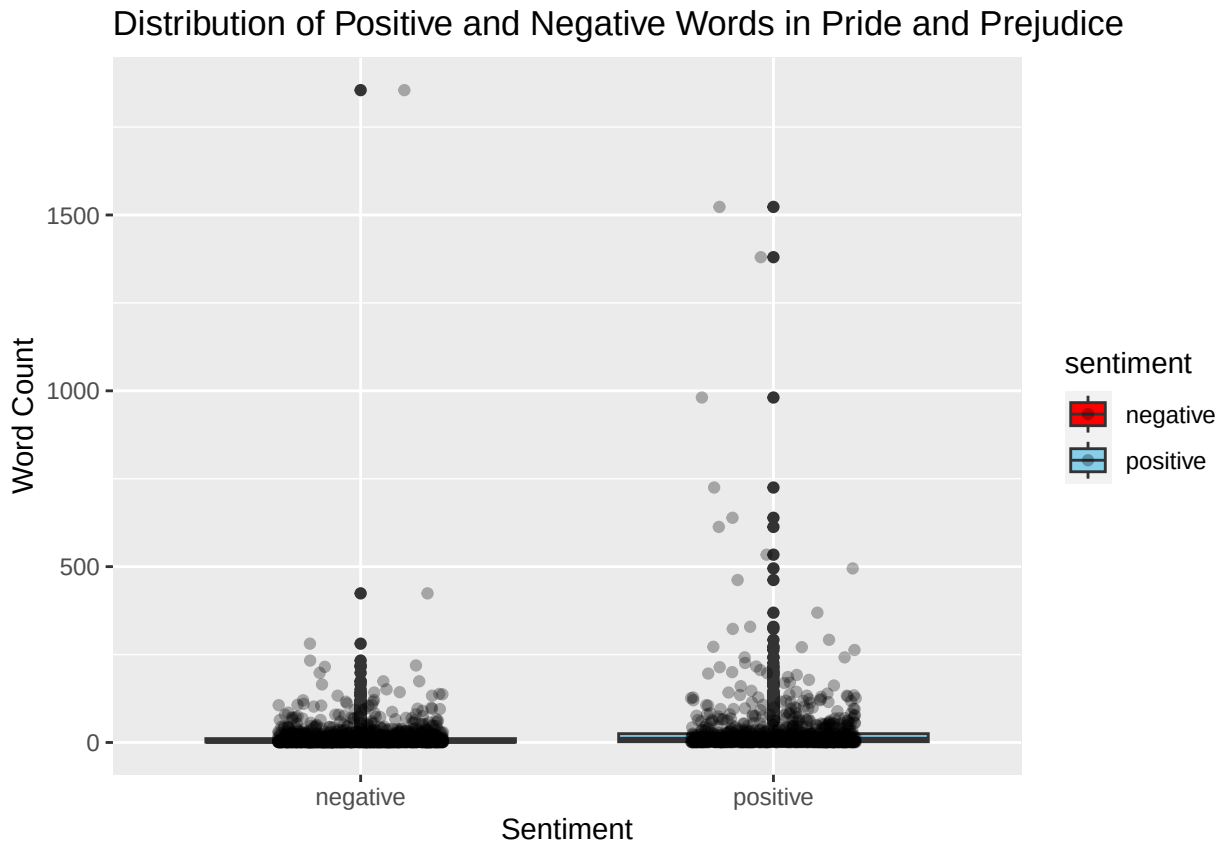


Most common positive and negative words Counts the frequency of words in a text dataset categorized by sentiment using the Bing lexicon

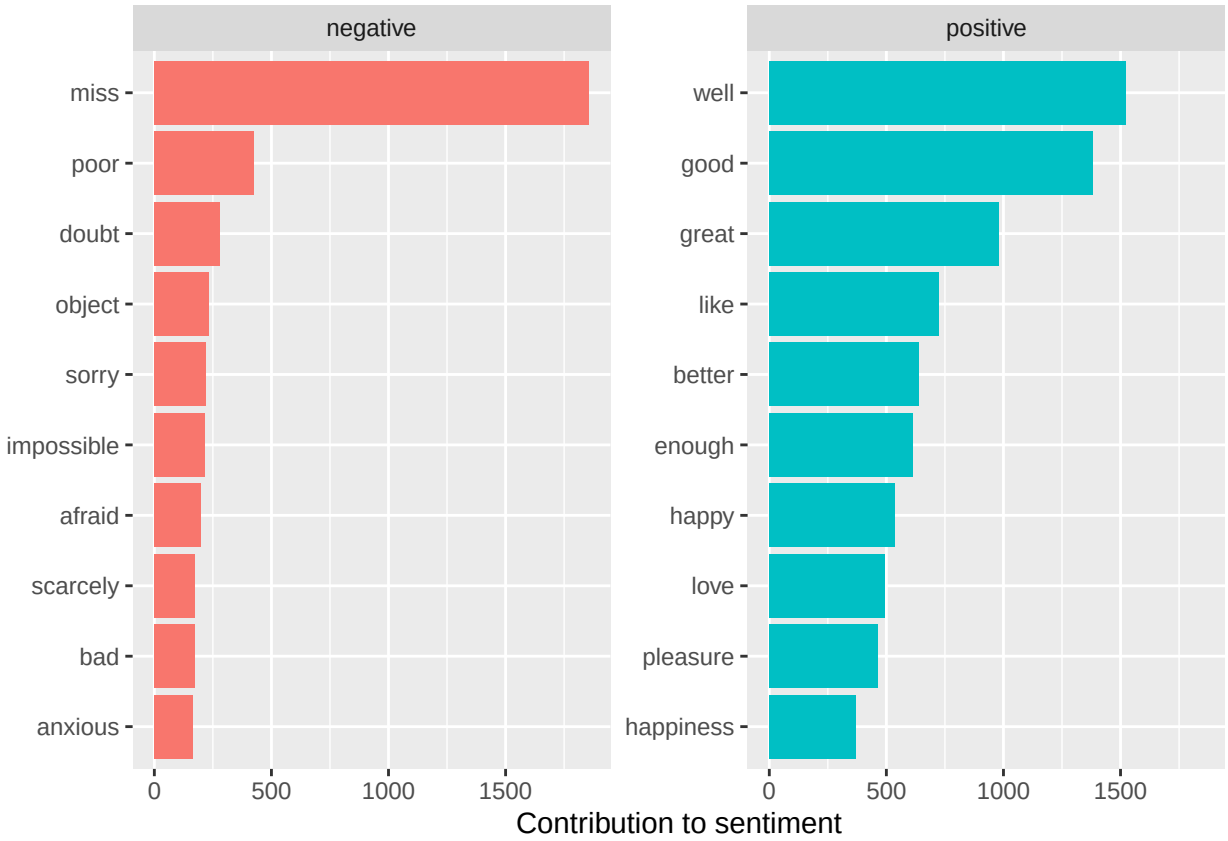
```
bing_word_counts <- tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

Visualizes the top 10 positive and negative words using the Bing lexicon in a bar plot.

```
bing_word_counts %>%
  filter(sentiment %in% c("positive", "negative")) %>%
  ggplot(aes(x = sentiment, y = n, fill = sentiment)) +
  geom_boxplot() +
  geom_jitter(width = 0.2, height = 0, alpha = 0.3) +
  scale_fill_manual(values = c("positive" = "skyblue", "negative" = "red")) +
  labs(title = "Distribution of Positive and Negative Words in Pride and Prejudice",
       x = "Sentiment",
       y = "Word Count")
```




```
bing_word_counts %>%
  group_by(sentiment) %>%
  slice_max(n, n = 10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(x = "Contribution to sentiment",
       y = NULL)
```



Creates a custom list of stop words that includes the words “well”, ““, and”miss” by binding together a tibble of these words with the standard list of stop words.

```
custom_stop_words <- bind_rows(tibble(word = c("well", "", "miss"),
                                      lexicon = c("custom")),
                               stop_words)
```

```
library(RColorBrewer)

# Color palette for the wordclouds
colors <- brewer.pal(8, "Dark2")

# Wordcloud of non-stopwords
tidy_books %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100, color = colors))
```

Wordclouds

```
# Sentiment analysis to tag positive and negative words using an inner join, then find the most common
tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = colors,
                   max.words = 100)
```


```
p_and_p_sentences <- tibble(text = pridepredjudice) %>%
  unnest_tokens(sentence, text, token = "sentences")
p_and_p_sentences$sentence[2]
```

Looking at units beyond just words

```
## [1] "by jane austen"
```

```
austen_chapters <- austen_books() %>%
  group_by(book) %>%
  unnest_tokens(chapter, text, token = "regex",
                pattern = "Chapter|CHAPTER [\\dIVXLC]") %>%
  ungroup()
austen_chapters %>%
  group_by(book) %>%
  summarise(chapters = n())
```

```
## # A tibble: 6 x 2
##   book                chapters
##   <fct>                  <int>
## 1 Sense & Sensibility       51
## 2 Pride & Prejudice         62
```

```
## 3 Mansfield Park            49
## 4 Emma                      56
## 5 Northanger Abbey          32
## 6 Persuasion                25
```

```
bingnegative <- get_sentiments("bing") %>%
  filter(sentiment == "negative")
wordcounts <- tidy_books %>%
  group_by(book, chapter) %>%
  summarize(words = n())
tidy_books %>%
  semi_join(bingnegative) %>%
  group_by(book, chapter) %>%
  summarize(negativewords = n()) %>%
  left_join(wordcounts, by = c("book", "chapter")) %>%
  mutate(ratio = negativewords/words) %>%
  filter(chapter != 0) %>%
  slice_max(ratio, n = 1) %>%
  ungroup()
```

```
## # A tibble: 6 x 5
##   book                chapter negativewords words  ratio
##   <fct>                 <int>         <int> <int>  <dbl>
## 1 Sense & Sensibility      43           161  3405 0.0473
## 2 Pride & Prejudice        34           111  2104 0.0528
## 3 Mansfield Park           46           173  3685 0.0469
## 4 Emma                     15           151  3340 0.0452
## 5 Northanger Abbey         21           149  2982 0.0500
## 6 Persuasion                4            62  1807 0.0343
```

Extension

My Bondage and My Freedom by Frederick Douglass

we will analyze text `My Bondage and My Freedom` by Frederick Douglass. and we will use the gutenbergr library to search and download it.

```
bondage_count <- gutenberg_download(202)
bondage_count
```

The sentiments datasets

```
## # A tibble: 12,324 x 2
##    gutenberg_id text
##           <int> <chr>
##  1          202 "MY BONDAGE and MY FREEDOM"
##  2          202 ""
##  3          202 "By Frederick Douglass"
##  4          202 ""
```

```
##  5          202 ""
##  6          202 "By a principle essential to Christianity, a PERSON is eternall~
##  7          202 "differenced from a THING; so that the idea of a HUMAN BEING,"
##  8          202 "necessarily excludes the idea of PROPERTY IN THAT BEING. -COLE~
##  9          202 ""
## 10          202 "Entered according to Act of Congress in 1855 by Frederick Doug~
## # ... with 12,314 more rows
```

Tidy the data

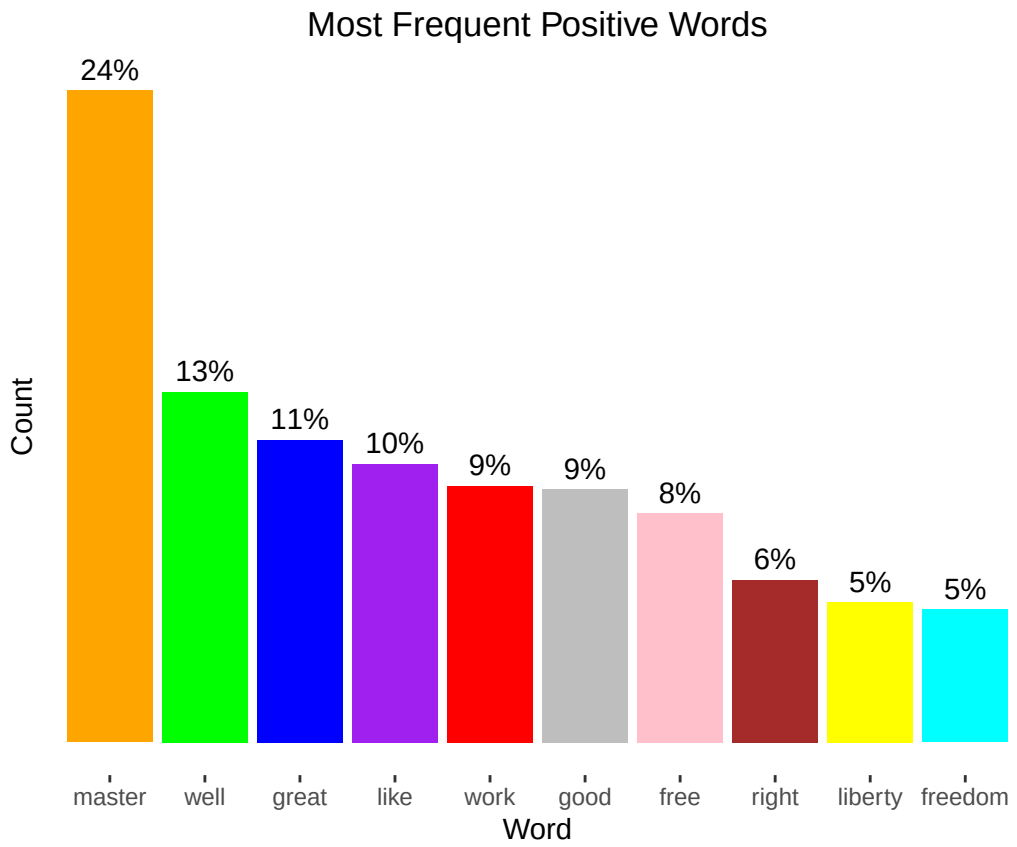
```
#removing the first 763 rows of text which are table of contents
bondage_count <- bondage_count[c(763:nrow(bondage_count)),]
#using unnest_tokens to have each line be broken into indidual rows.
bondage <- bondage_count %>% unnest_tokens(word, text)
bondage
```

```
## # A tibble: 129,096 x 2
##    gutenberg_id word
##           <int> <chr>
##  1          202 chapter
##  2          202 i
##  3          202 _childhood_
##  4          202 place
##  5          202 of
##  6          202 birth
##  7          202 character
##  8          202 of
##  9          202 the
## 10          202 district
## # ... with 129,086 more rows
```

```
bondage_index <- bondage_count %>%
  filter(text != "") %>%
  mutate(linenumber = row_number(),
         chapter = cumsum(str_detect(text, regex("(?<=Chapter )([\\dII]{1,3})", ignore_case =  TRUE))))
bondage_index
```

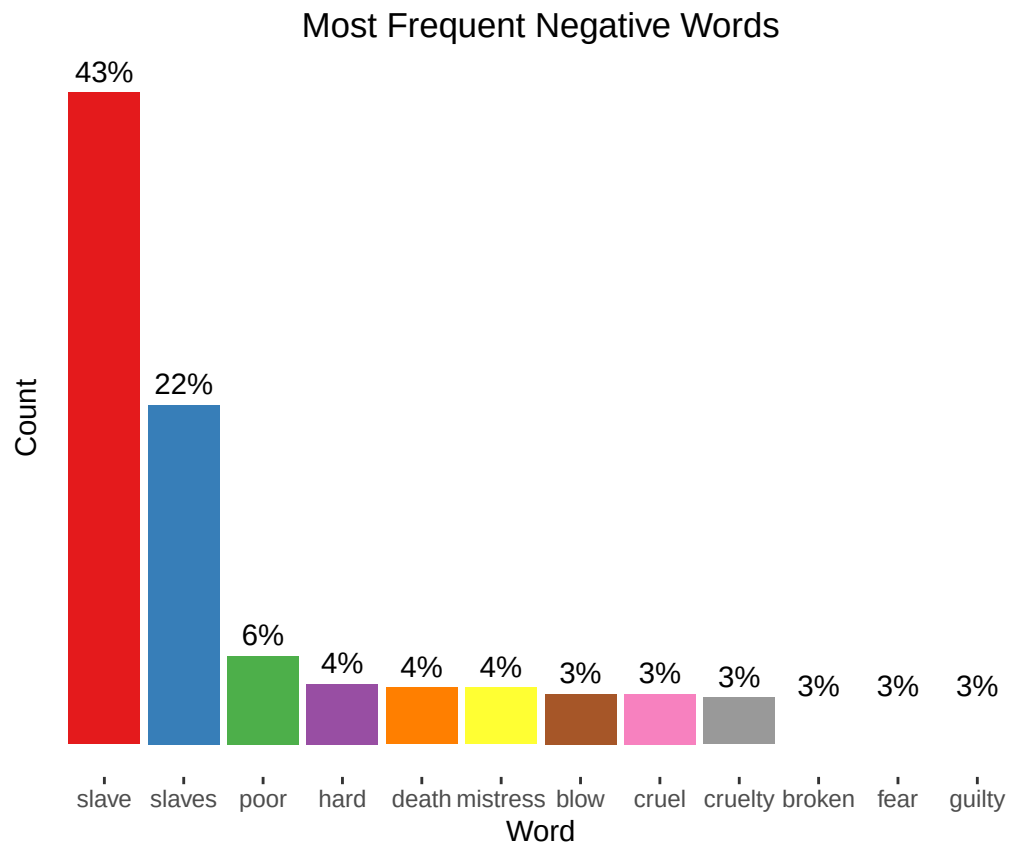
```
## # A tibble: 10,716 x 4
##    gutenberg_id text                                              linen~1 chapter
##           <int> <chr>                                             <int>   <int>
##  1          202 CHAPTER I. _Childhood_                                1       1
##  2          202 PLACE OF BIRTH-CHARACTER OF THE DISTRICT-TUCKAH~      2       1
##  3          202 NAME-CHOPTANK RIVER-TIME OF BIRTH-GENEALOGICAL ~      3       1
##  4          202 TIME-NAMES OF GRANDPARENTS-THEIR POSITION-GRAND~      4       1
##  5          202 ESTEEMED-"BORN TO GOOD LUCK"-SWEET POTATOES-SUP~      5       1
##  6          202 CABIN-ITS CHARMS-SEPARATING CHILDREN-MY AUNTS-T~      6       1
##  7          202 KNOWLEDGE OF BEING A SLAVE-OLD MASTER-GRIEFS AN~      7       1
##  8          202 CHILDHOOD-COMPARATIVE HAPPINESS OF THE SLAVE-BO~      8       1
##  9          202 SLAVEHOLDER.                                          9       1
## 10          202 In Talbot county, Eastern Shore, Maryland, near~     10       1
## # ... with 10,706 more rows, and abbreviated variable name 1: linenumber
```

```
library(ggplot2)

bondage %>%
  inner_join(get_sentiments("bing")) %>%
  filter(sentiment == "positive") %>%
  count(word, sentiment, sort = TRUE) %>%
  top_n(10) %>%
  mutate(word = reorder(word, desc(n))) %>%
  ggplot() +
  aes(x = word, y = n, fill = word) +
  labs(title = "Most Frequent Positive Words") +
  ylab("Count") +
  xlab("Word") +
  geom_col() +
  scale_fill_manual(values = c("orange", "green", "blue", "purple", "red", "gray", "pink", "brown", "yel
  geom_text(aes(label = paste0(round((n/sum(n))*100), "%")), vjust = -0.5) +
  theme(
    panel.background = element_rect(fill = "white", color = NA),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    plot.title = element_text(hjust = 0.5)
  )
```



Most frequent postive words

```r
library(ggplot2)
library(dplyr)
library(tidyr)
library(gutenbergr)
library(tidytext)

# download and tidy data
bondage_count <- gutenberg_download(202)
bondage_count <- bondage_count[c(763:nrow(bondage_count)),]
bondage <- bondage_count %>% unnest_tokens(word, text)
bondage_index <- bondage_count %>%
  filter(text != "") %>%
  mutate(linenumber = row_number(),
         chapter = cumsum(str_detect(text, regex("(?<=Chapter )([\\dII]{1,3})", ignore_case =  TRUE))))

# plot most frequent negative words
bondage %>%
  inner_join(get_sentiments("bing")) %>%
  filter(sentiment == "negative") %>%
  count(word, sentiment, sort = TRUE) %>%
  top_n(10) %>%
  mutate(word = reorder(word, desc(n))) %>%
  ggplot() +
  aes(x = word, y = n, fill = word) +
  labs(title = "Most Frequent Negative Words") +
  ylab("Count") +
  xlab("Word") +
  geom_col() +
  scale_fill_brewer(palette = "Set1") +
  geom_text(aes(label = paste0(round((n/sum(n))*100), "%"), vjust = -0.5)) +
  theme(
    panel.background = element_rect(fill = "white", color = NA),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    plot.title = element_text(hjust = 0.5)
  )
```



Most frequent negative words

```
library(RColorBrewer)

# Color palette for the wordcloud
colors <- brewer.pal(8, "Dark2")

# Wordcloud of non-stopwords
bondage %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100, color = colors))
```

Wordclouds

Loughran Lexicon

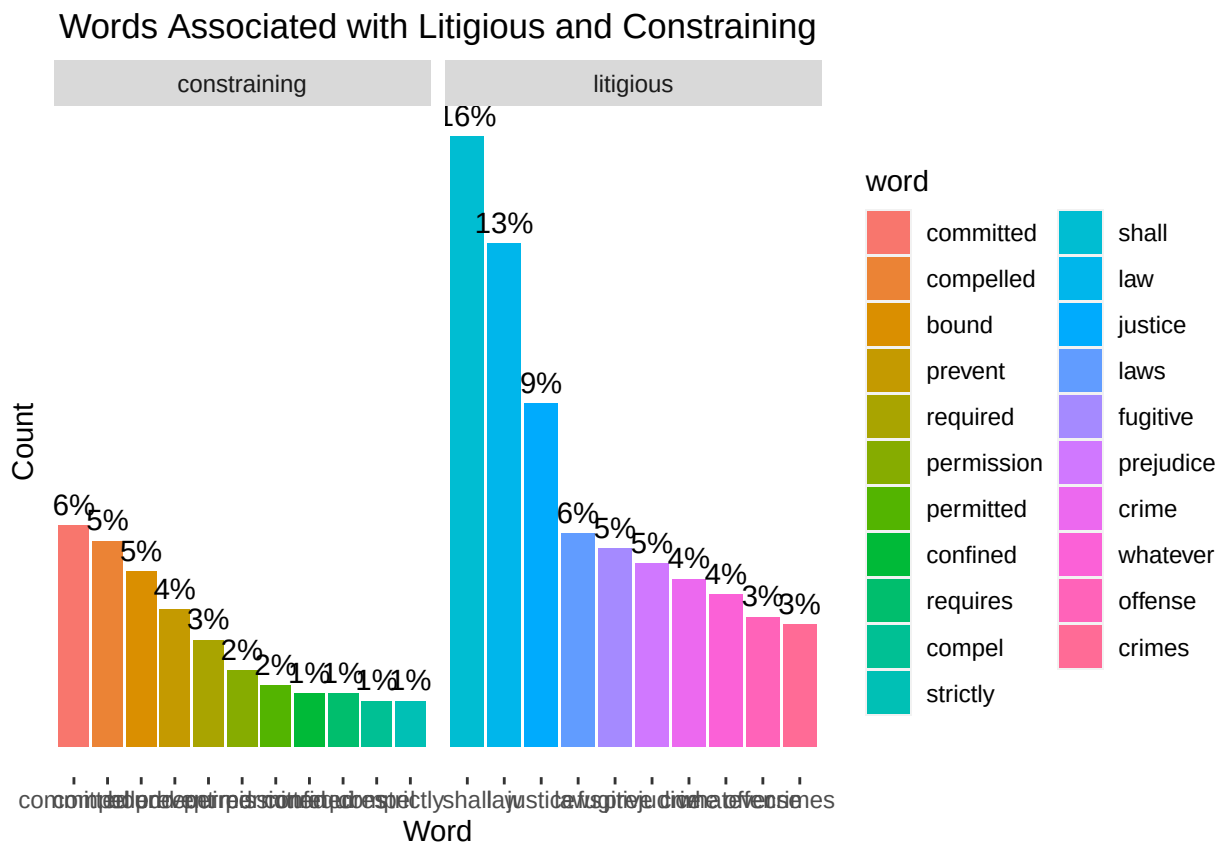
Here , we will use loughran lexicon instead of one of the lexicons used in the sample code.

```
lghrn <- get_sentiments("loughran")
unique(lghrn$sentiment)
```

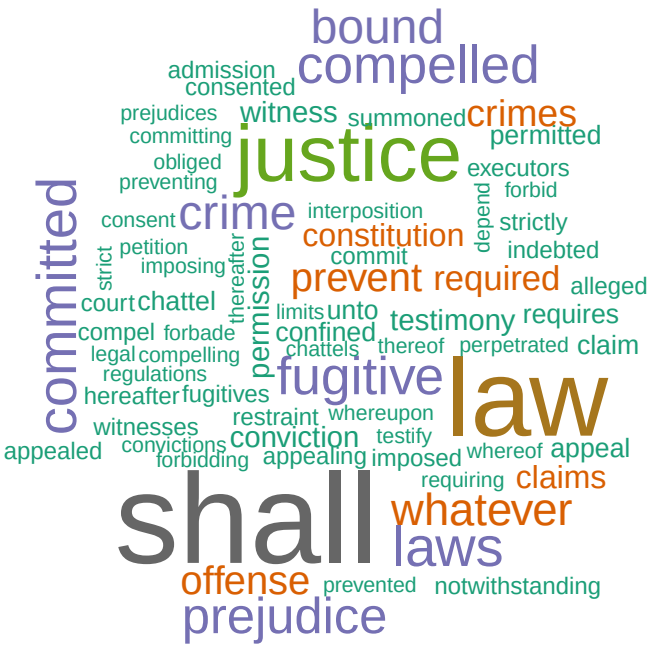
```
## [1] "negative"     "positive"     "uncertainty"  "litigious"    "constraining"
## [6] "superfluous"
```

```
bondage_index %>%
  unnest_tokens(word, text) %>%
  inner_join(get_sentiments("loughran")) %>%
  filter(sentiment %in% c("litigious", "constraining")) %>%
  count(word, sentiment, sort = TRUE) %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  mutate(word = reorder(word, desc(n))) %>%
  ggplot() +
  aes(x = word, y = n, fill = word) +
  labs(title = "Words Associated with Litigious and Constraining") +
  ylab("Count") +
  xlab("Word") +
  geom_col() +
```

```
  geom_text(aes(label = paste0(round(n/sum(n)*100),"%"), vjust = -0.5)) +
  facet_grid(~sentiment, scales = "free_x") +
  theme(
    panel.background = element_rect(fill = "white", color = NA),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    plot.title = element_text(hjust = 0.5)
  )
```




```
library(RColorBrewer)

# Color palette for the wordcloud
colors <- brewer.pal(8, "Dark2")

# Wordcloud of litigious and constraining words
bondage_index %>%
  unnest_tokens(word, text) %>%
  inner_join(get_sentiments("loughran")) %>%
  filter(sentiment %in% c("litigious", "constraining")) %>%
  count(word, sort = TRUE) %>%
  with(wordcloud(word, n, max.words = 100, color = colors))
```



Wordclouds

Conclusion

In conclusion, this assignment has allowed us to explore the topic of sentiment analysis. We have successfully implemented and expanded upon the main example code from chapter 2 of the Text Mining with R book. We have used three different sentiment lexicons - AFINN, Bing, and NRC.to analyze the sentiment of Jane Austen's novels. We have also visualized the sentiment scores over the plot trajectory of each novel and compared the sentiment scores obtained from the different sentiment lexicons. Furthermore, we have identified the most common positive and negative words in the novels using the Bing lexicon. Then we extended using My Bondage and My Freedom" by Frederick Douglass using the gutenbergr library