Sentiment Analysis

Seoyeong Park

2020 01 04

```
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
```

Developing sentiment analysis model in R

Summary

This project is a sentiment analysis with the dataset of Jane Austen's books. I will use a 'bing' lexical analyzer to analyze sentiment score and visualize represented sentiment score with word cloud.

Install tidytext package and other required packages.

```
#install.packages("tidytext")
#install.packages("tidyr")
```

Reading the tidytext package and load the dataset of 'sentiments'.

```
library(tidytext)
sentiments
```

```
## # 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
```

In this project, I will make use of the 'bing' lexicons to extract the sentiments from the data among three general purpose lexicons, which are AFINN, bing, laughran.

```
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
```

Performing sentiment analysis with the inner join

With importing libraries 'janeaustenr', 'stringr' as well as 'tidytext', 'janeaustenr' library will provide the textual data in the form of books written by the novelist Jane Austen. Tidytext will help perform efficient text analysis on data.

```
library(janeaustenr)
library(stringr)

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

  #This line converts the text of books into a tidy format
  unnest_tokens(word, text)
```

I have performed the tidy operation on texts so that each row contains a single word. Now, I will make use of the 'bing' lexicon to and implement filter() over the words. The book I will use here is 'Sense and Sesibility'. I will derive its words to implement out sentiment analysis model.

```
positive_senti <- get_sentiments("bing") %>%
  filter(sentiment == "positive")

tidy_data %>%
  filter(book == "Emma") %>%
  semi_join(positive_senti) %>%
  count(word, sort = TRUE)
```

```
## Joining, by = "word"
```

```
## # A tibble: 668 x 2
##    word         n
##    <chr>    <int>
##  1 well       401
##  2 good       359
##  3 great      264
```

```
##  4 like       200
##  5 better     173
##  6 enough     129
##  7 happy      125
##  8 love       117
##  9 pleasure   115
## 10 right       92
## # ... with 658 more rows
```

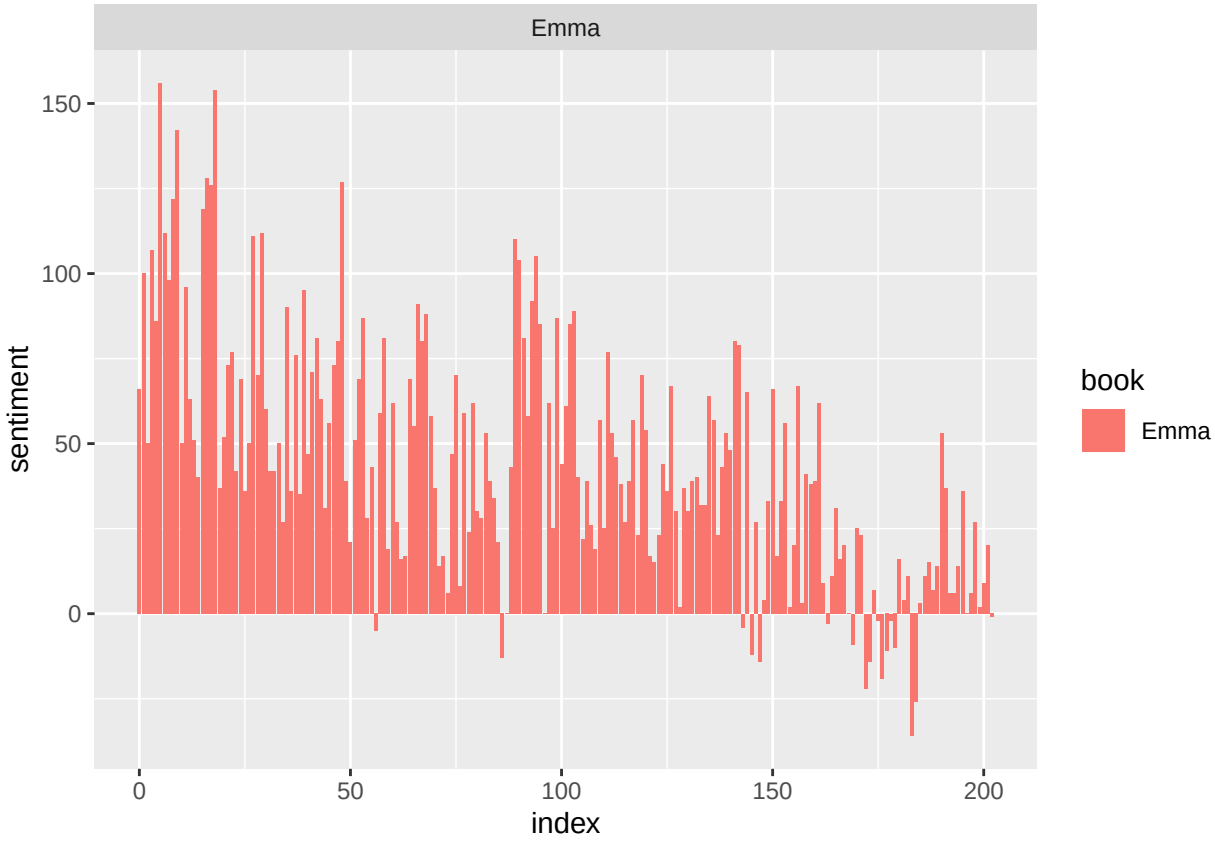
Next, I will segregate the data into separate columns of positive and negative sentiments by using spread(). Then, use mutate() to calculate the total sentiment.

```
library(tidyr)
bing <- get_sentiments("bing")
Emma_sentiment <- tidy_data %>%
  inner_join(bing) %>%
  count(book = "Emma", index=linenumber %/% 80, sentiment) %>%
  spread(sentiment, n, fill=0) %>%
  mutate(sentiment = positive - negative)
```

```
## Joining, by = "word"
```

Visualize the words present in the book 'Emma' based on corresponding positive and negative scores.

```
library(ggplot2)
ggplot(Emma_sentiment, aes(index, sentiment, fill=book)) +
  geom_bar(stat = "identity", show.legend = TRUE) +
  facet_wrap(~book, ncol = 2, scales = "free_x")
```



Count the most common positive and negative words that are present in the novel.

```
counting_words <- tidy_data %>%
  inner_join(bing) %>%
  count(word, sentiment, sort=TRUE)
```

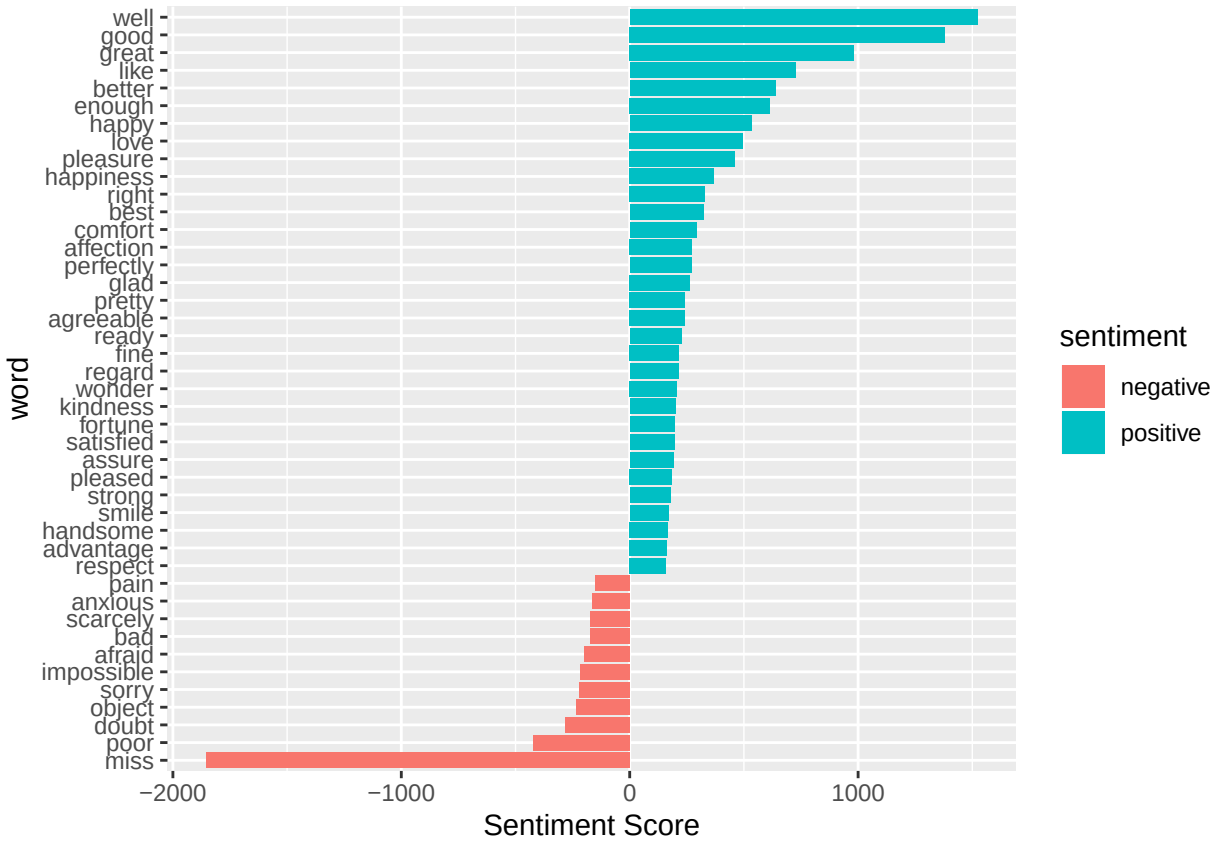
```
## Joining, by = "word"
```

```
head(counting_words)
```

```
## # A tibble: 6 x 3
##   word   sentiment     n
##   <chr>  <chr>     <int>
## 1 miss   negative   1855
## 2 well   positive   1523
## 3 good   positive   1380
## 4 great  positive    981
## 5 like   positive    725
## 6 better positive    639
```

Next, visualize sentiment score. I will plot the scores with the axis labeled with both positive and negative words. Use ggplot() to visualize the data.

```
counting_words %>%
  filter(n>150) %>%
  mutate(n=ifelse(sentiment=="negative", -n, n)) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col() +
  coord_flip() +
  labs(y="Sentiment Score")
```



Finally, I will create a wordcloud that will delineate the most recurring positive and negative words. I will use comparison.cloud() to visualize both negative and positive words in a single wordcloud.

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

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

```
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
tidy_data %>%
  inner_join(bing) %>%
  count(word, sentiment, sort=TRUE) %>%
  acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("red", "blue"),
                   max.words = 100)
```

```
## Joining, by = "word"
```

Above word cloud shows visualization of words group based on negative and positive groups of data.