R Notebook

```
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.2.1
```

```
nrow(sentiments)
```

```
## [1] 6786
```

```
sentiments
```

word <chr>	sentiment <chr>
2-faces	negative
abnormal	negative
abolish	negative
abominable	negative
abominably	negative
abominate	negative
abomination	negative
abort	negative
aborted	negative
aborts	negative

1-10 of 6,786 rows Previous **1** 2 3 4 5 6 … 679 Next

```
get_sentiments("bing")
```

word <chr>	sentiment <chr>
2-faces	negative
abnormal	negative
abolish	negative
abominable	negative
abominably	negative
abominate	negative
abomination	negative

word <chr>	sentiment <chr>
abort	negative
aborted	negative
aborts	negative

1-10 of 6,786 rows Previous **1** 2 3 4 5 6 … 679 Next

janeaustenr: An R Package for Jane Austen's Complete Novels

austen_books: Tidy data frame of Jane Austen's 6 completed, published novels

```
library(janeaustenr)
```

```
## Warning: package 'janeaustenr' was built under R version 4.2.1
```

```
austen_books()
```

	text <chr>	b <c
1	SENSE AND SENSIBILITY	S
2		S
3	by Jane Austen	S
4		S
5	(1811)	S
6		S
7		S
8		S
9		S
10	CHAPTER 1	S

1-10 of 10,000 rows Previous **1** 2 3 4 5 6 … 1000 Next

Calculate Cumulative Sum of a Numeric Object in R Programming – cumsum() Function.

The cumulative sum can be defined as the sum of a set of numbers as the sum value grows with the sequence of numbers.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.1
```

```
##
## 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(stringr)
```

```
## Warning: package 'stringr' was built under R version 4.2.1
```

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

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

word <chr>	sentiment <chr>
abound	positive
abounds	positive
abundance	positive
abundant	positive
accessable	positive
accessible	positive
acclaim	positive
acclaimed	positive
acclamation	positive
accolade	positive

1-10 of 2,005 rows Previous **1** 2 3 4 5 6 … 201 Next

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

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

word <chr>	n <int>
well	401
good	359
great	264
like	200
better	173
enough	129
happy	125
love	117
pleasure	115
right	92

1-10 of 668 rows Previous **1** 2 3 4 5 6 … 67 Next

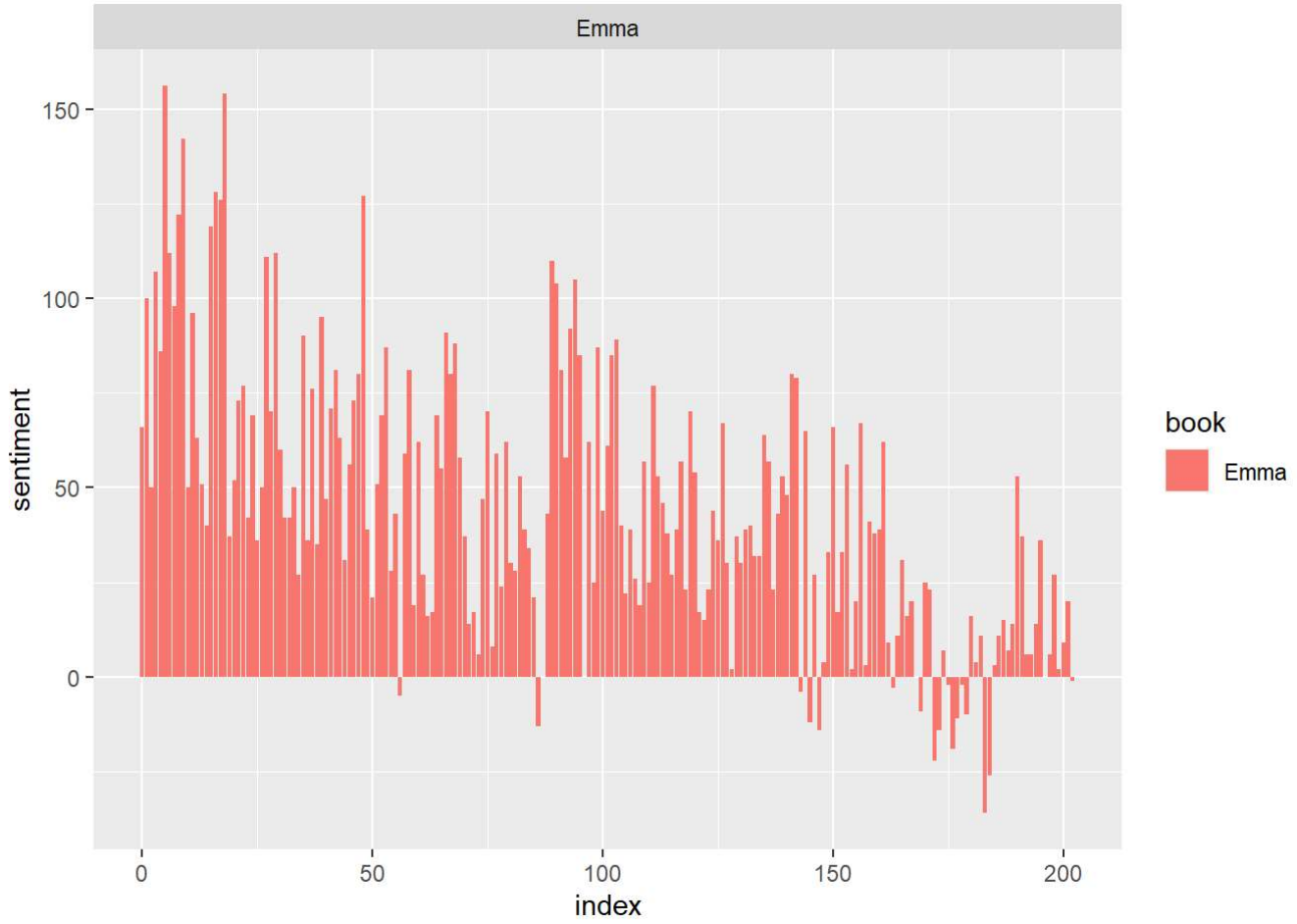
```
library(tidyr)
bing <- get_sentiments("bing")
Emma_sentiment <- tidy_data %>%
 inner_join(bing) %>%
 count(book = "Emma" , index = linenumber %/% 80, sentiment) %>%
 spread(sentiment, n, fill = 0) %>%
 mutate(sentiment = positive - negative)
```

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

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

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




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

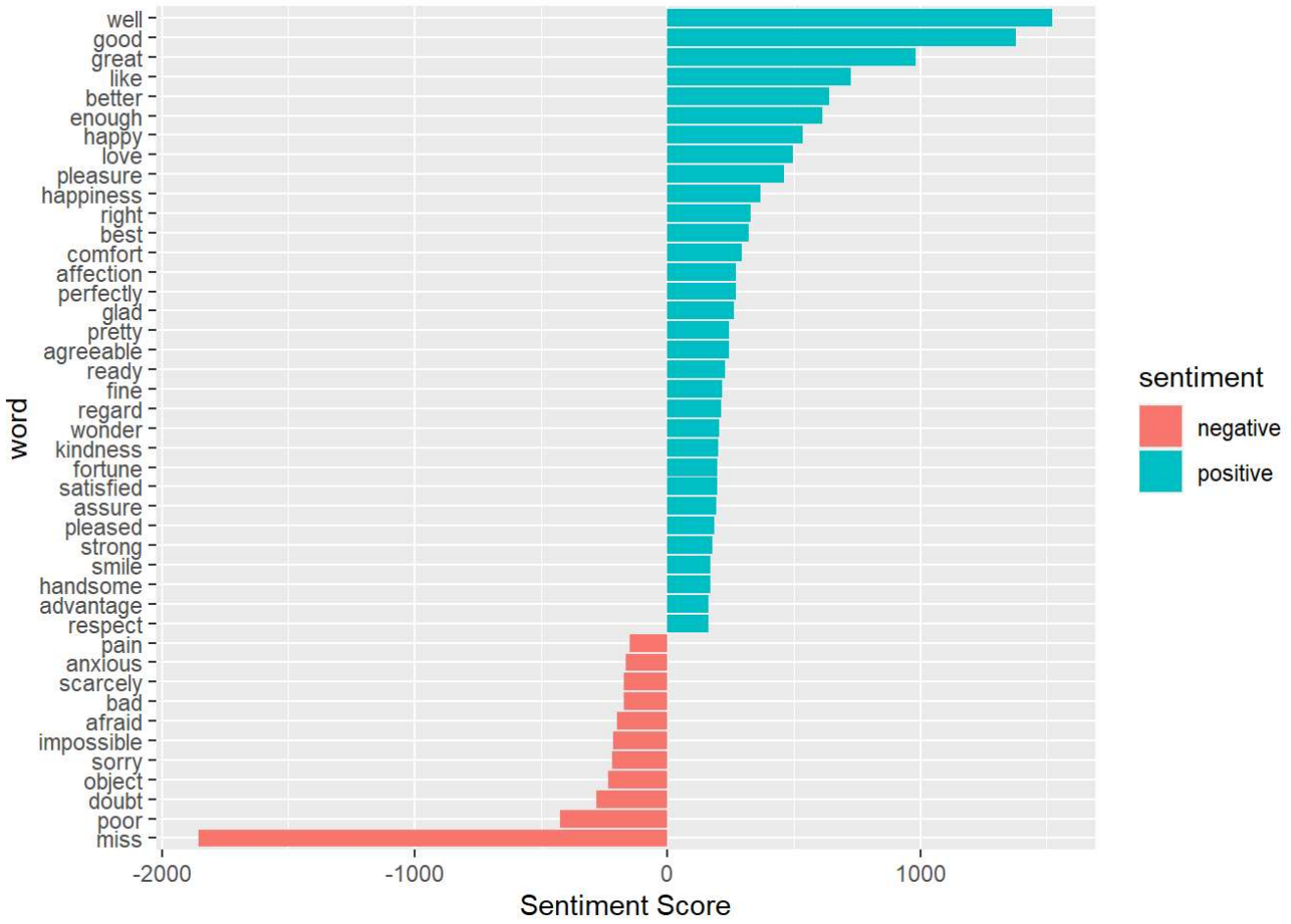
```
## Joining, by = "word"
```

```
head(counting_words)
```

word <chr>	sentiment <chr>	n <int>
miss	negative	1855
well	positive	1523
good	positive	1380
great	positive	981
like	positive	725
better	positive	639

6 rows

```
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")
```




```
#install.packages("wordcloud")
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.2.1
```

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

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

```
library(wordcloud)
```

```
## Warning: package 'wordcloud' was built under R version 4.2.1
```

```
## 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", "dark green"),
          max.words = 100)
```

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

In this project, we went through our project of sentiment analysis in R.

We learnt about the concept of sentiment analysis and implemented it over the dataset of Jane Austen's books.

We were able to delineate it through various visualizations after we performed data wrangling on our data.

We used a lexical analyzer – 'bing' in this instance of our project. Furthermore, we also represented the sentiment score through a plot and also made a visual report of wordcloud.