Assignment 10 - Text Mining

Subhalaxmi Rout

04/05/2020

Assignment Over-view

In Text Mining with R, Chapter 2 looks at Sentiment Analysis. In this assignment, you should start by getting the primary example code from chapter 2 working in an R Markdown document. You should provide a citation to this base code. You're then asked to extend the code in two ways:

- Work with a different corpus of your choosing, and
- Incorporate at least one additional sentiment lexicon (possibly from another R package that you've found through research).

Code from Textbook

The aim of this assignment is to understand sentiment Analysis given in the textbook "Text Mining with R-chapter 2" then add a new corpus and lexicon which is not used in the textbook.

what is corpus?

These types of objects typically contain raw strings annotated with additional metadata and details.

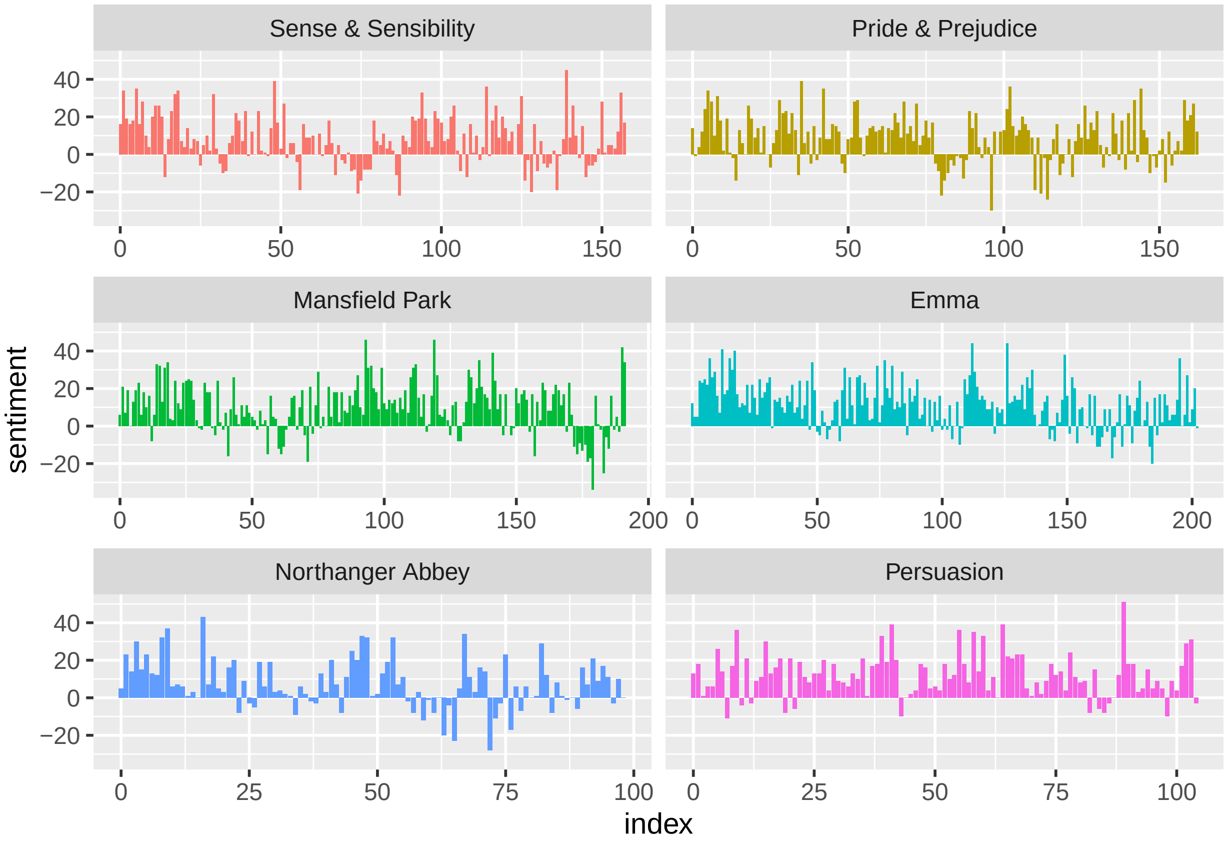
Jane Austen dataset

Using the text of Jane Austen's 6 completed, published novels from the janeaustenr package (Silge 2016), and transform them into a tidy format.

```r
# Load library
library(janeaustenr)
library(dplyr)
library(stringr)
library(tidytext)
library(tidyr)
library(ggplot2)
library(textdata)
library(wordcloud)

# get linenumber and chapter
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)


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

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

```
## # A tibble: 303 x 2
##    word        n
##    <chr>   <int>
##   1 good      359
##   2 young     192
##   3 friend    166
##   4 hope      143
##   5 happy     125
##   6 love      117
##   7 deal       92
##   8 found      92
##   9 present    89
## 10 kind       82
## # ... with 293 more rows
```

```r
jane_austen_sentiment <- tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(book, index = linenumber %/% 80, sentiment) %>%
  spread(sentiment, n, 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")
```



```
# compairing 3 sentiment dictionaries
pride_prejudice <- tidy_books %>%
  filter(book == "Pride & Prejudice")

pride_prejudice
```

```
## # A tibble: 122,204 x 4
##    book              linenumber chapter word
##    <fct>                  <int>   <int> <chr>
##  1 Pride & Prejudice          1       0 pride
##  2 Pride & Prejudice          1       0 and
##  3 Pride & Prejudice          1       0 prejudice
##  4 Pride & Prejudice          3       0 by
##  5 Pride & Prejudice          3       0 jane
##  6 Pride & Prejudice          3       0 austen
##  7 Pride & Prejudice          7       1 chapter
##  8 Pride & Prejudice          7       1 1
##  9 Pride & Prejudice         10       1 it
## 10 Pride & Prejudice         10       1 is
## # ... with 122,194 more rows
```

```
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) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)

get_sentiments("nrc") %>%
     filter(sentiment %in% c("positive",
                             "negative")) %>%
  count(sentiment)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   3324
## 2 positive   2312
```

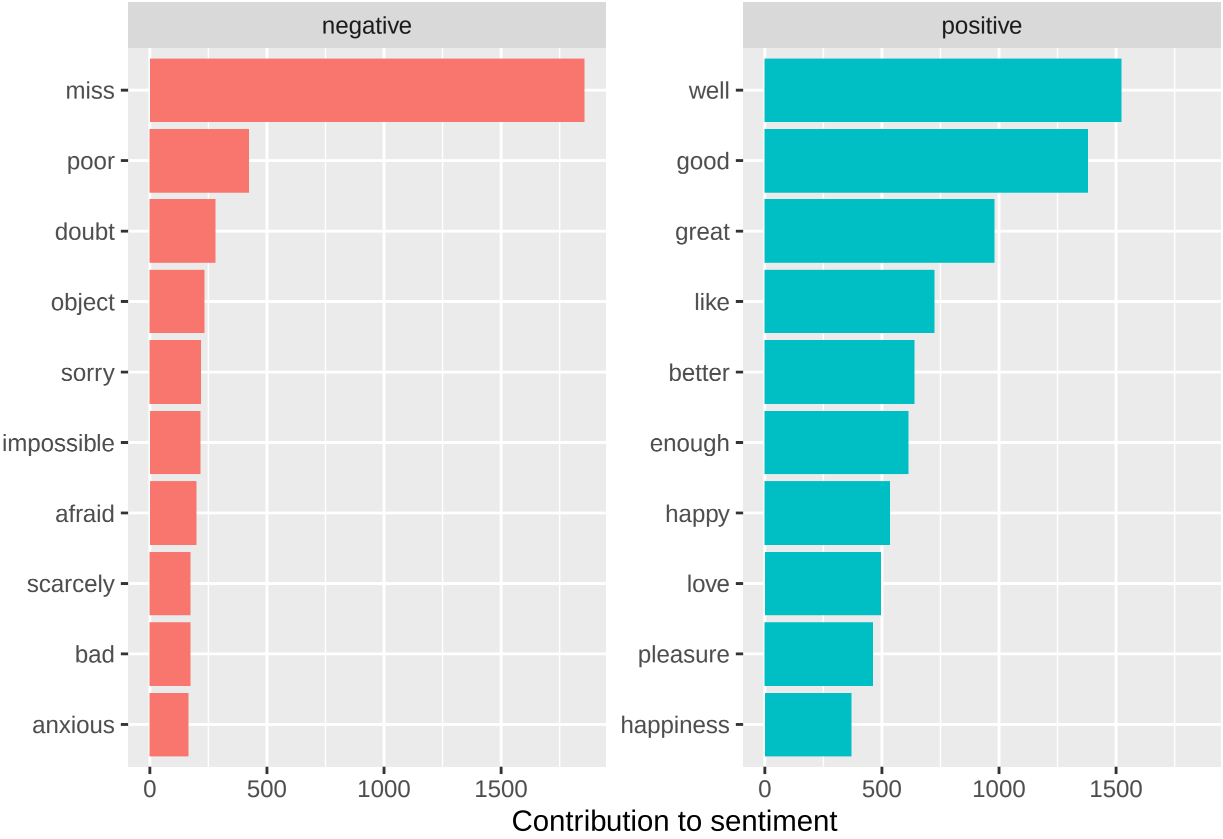
```
get_sentiments("bing") %>%
  count(sentiment)
```

```
## # A tibble: 2 x 2
##   sentiment     n
##   <chr>     <int>
## 1 negative   4781
## 2 positive   2005
```

```
# most common positive and negative words
bing_word_counts <- tidy_books %>%
  inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()

bing_word_counts
```

```
## # A tibble: 2,585 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
##  7 enough   positive    613
##  8 happy    positive    534
##  9 love      positive    495
## 10 pleasure positive    462
## # ... with 2,575 more rows
```

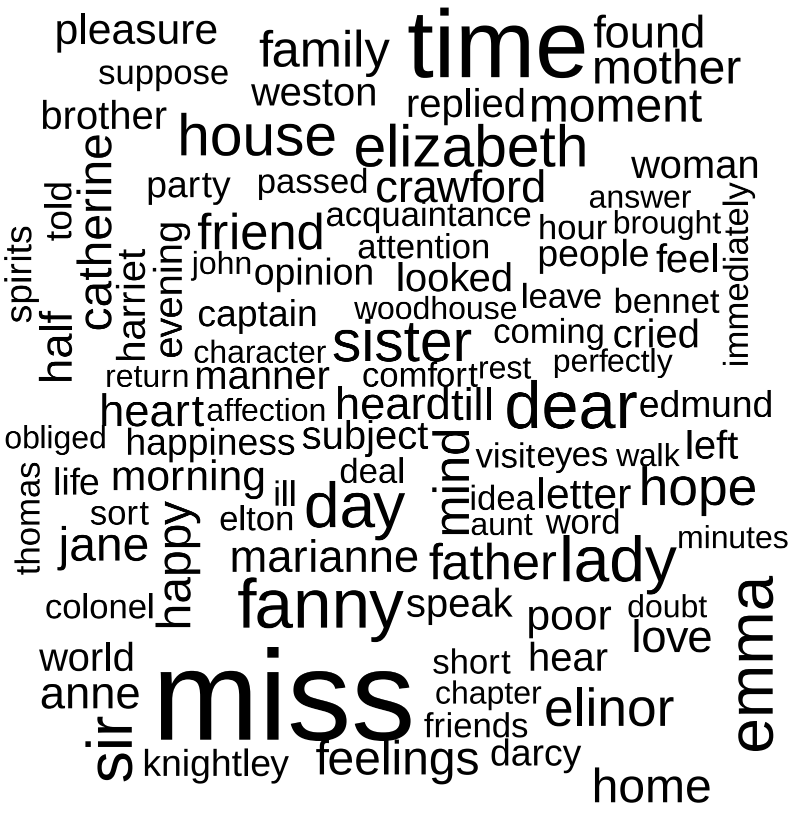
```
bing_word_counts %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to sentiment",
       x = NULL)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



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

custom_stop_words
```

```
## # A tibble: 1,150 x 2
##    word        lexicon
##    <chr>       <chr>
##  1 miss        custom
##   2 a           SMART
##   3 a's         SMART
##   4 able        SMART
##   5 about       SMART
##   6 above       SMART
##   7 according   SMART
##   8 accordingly SMART
##   9 across      SMART
##  10 actually    SMART
## # ... with 1,140 more rows
```

```
# wordclouds
tidy_books %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100))
```



New Corpus

`My Bondage and My Freedom` is an autobiographical slave narrative written by `Frederick Douglass` and published in 1855. Download data using `gutenbergr` package.

Reference: https://docsouth.unc.edu/neh/douglass55/douglass55.html

```
library(gutenbergr)

# get gutenberg_id
```

```
#gutenberg_metadata %>% filter(author == "Douglass, Frederick"
#, title == "My Bondage and My Freedom")

count_of_Bondage_Freedom <- gutenberg_download(202)
```

```
## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest
```

```
## Using mirror http://aleph.gutenberg.org
```

```
count_of_Bondage_Freedom
```

```
## # A tibble: 12,208 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, n~
##  8          202 "excludes the idea of PROPERTY IN THAT BEING."
##  9          202 "--COLERIDGE"
## 10          202 ""
## # ... with 12,198 more rows
```

Convert Data to Tidy

```
count_Bondage_Freedom <- count_of_Bondage_Freedom[c(763:nrow(count_of_Bondage_Freedom)),]

Bondage_Freedom_Chapters <- count_Bondage_Freedom %>%
  filter(text != "") %>%
  mutate(linenumber = row_number(),
         chapter = cumsum(str_detect(text, regex("CHAPTER [\\dIVXLC]", ignore_case =  TRUE))))

Bondage_Freedom_Chapters
```

```
## # A tibble: 10,624 x 4
##    gutenberg_id text                                          linenumber chapter
##           <int> <chr>                                              <int>   <int>
##  1          202 "CHAPTER I. _Childhood_"                               1       1
##  2          202 "PLACE OF BIRTH--CHARACTER OF THE DISTRICT--~          2       1
##  3          202 "NAME--CHOPTANK RIVER--TIME OF BIRTH--GENEAL~          3       1
##  4          202 "COUNTING TIME--NAMES OF GRANDPARENTS--THEIR~          4       1
##  5          202 "ESPECIALLY ESTEEMED--\"BORN TO GOOD LUCK\"-~          5       1
##  6          202 "POTATOES--SUPERSTITION--THE LOG CABIN--ITS ~          6       1
##  7          202 "CHILDREN--MY AUNTS--THEIR NAMES--FIRST KNOW~          7       1
##  8          202 "MASTER--GRIEFS AND JOYS OF CHILDHOOD--COMPA~          8       1
##  9          202 "SLAVE-BOY AND THE SON OF A SLAVEHOLDER."              9       1
## 10          202 "In Talbot county, Eastern Shore, Maryland, ~         10       1
## # ... with 10,614 more rows
```

Lexicon

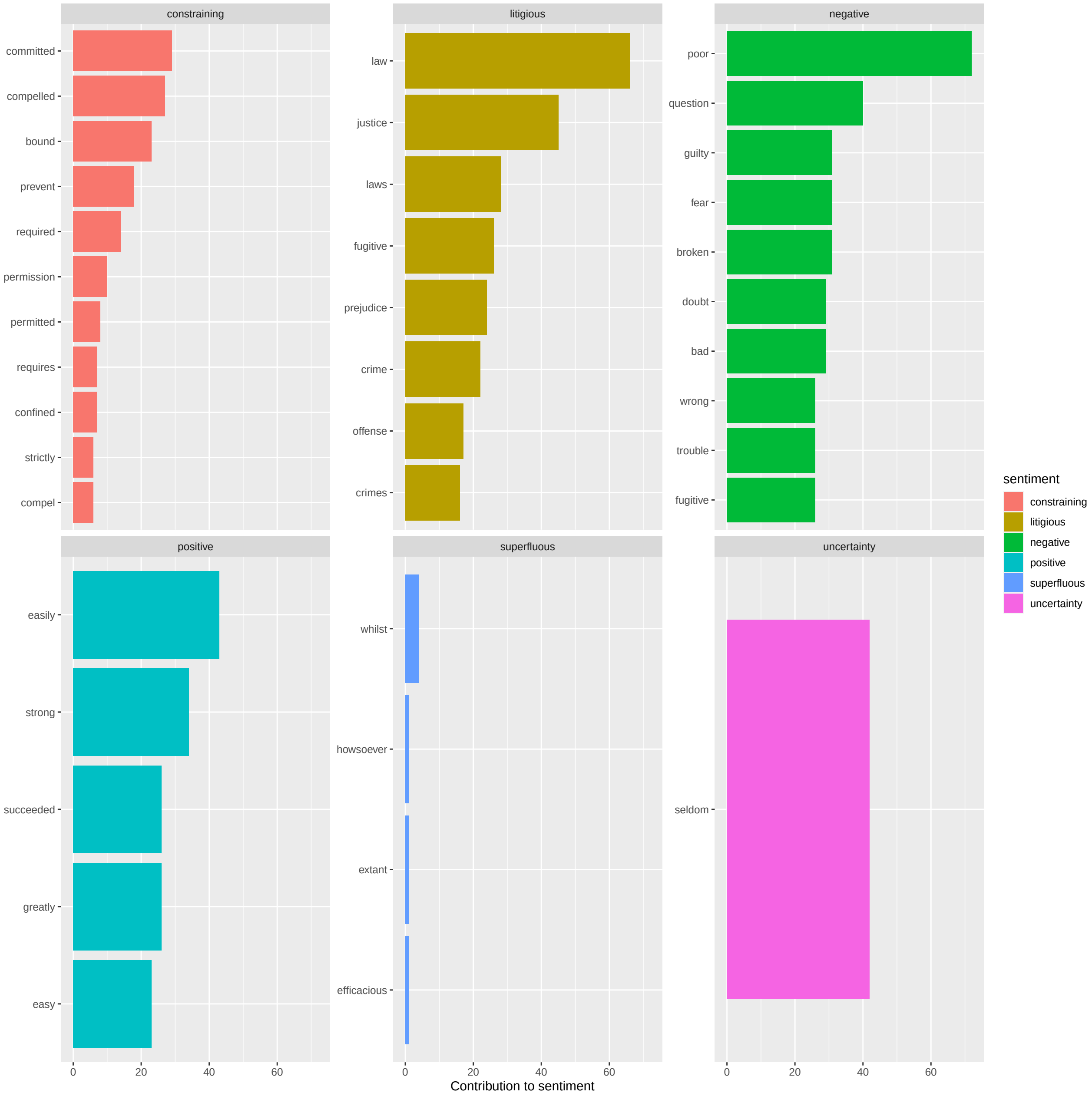
Using Loughran lexicon perform sentiment analysis.

loughran: English sentiment lexicon created for use with financial documents. This lexicon labels words with six possible sentiments important in financial contexts: "negative", "positive", "litigious", "uncertainty", "constraining", or "superfluous".

Reference: https://rdrr.io/cran/textdata/man/lexicon_loughran.html

The two basic arguments to unnest_tokens used here are column names. First we have the output column name that will be created as the text is unnested into it (word, in this case), and then the input column that the text comes from (text, in this case). Remember that text_df above has a column called text that contains the data of interest.

```
Bondage_Freedom_tidy <- Bondage_Freedom_Chapters %>%
  unnest_tokens(word, text) %>%
  inner_join(get_sentiments("loughran")) %>%
  count(word, sentiment, sort = TRUE) %>%
  group_by(sentiment) %>%
  top_n(10) %>% ungroup() %>% mutate(word = reorder(word, n)) %>%
  anti_join(stop_words)

names(Bondage_Freedom_tidy)<-c("word", "sentiment", "Freq")

ggplot(data = Bondage_Freedom_tidy, aes(x = word, y = Freq, fill = sentiment)) +
  geom_bar(stat = "identity") + coord_flip() + facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to sentiment",x = NULL)
```



Analysis

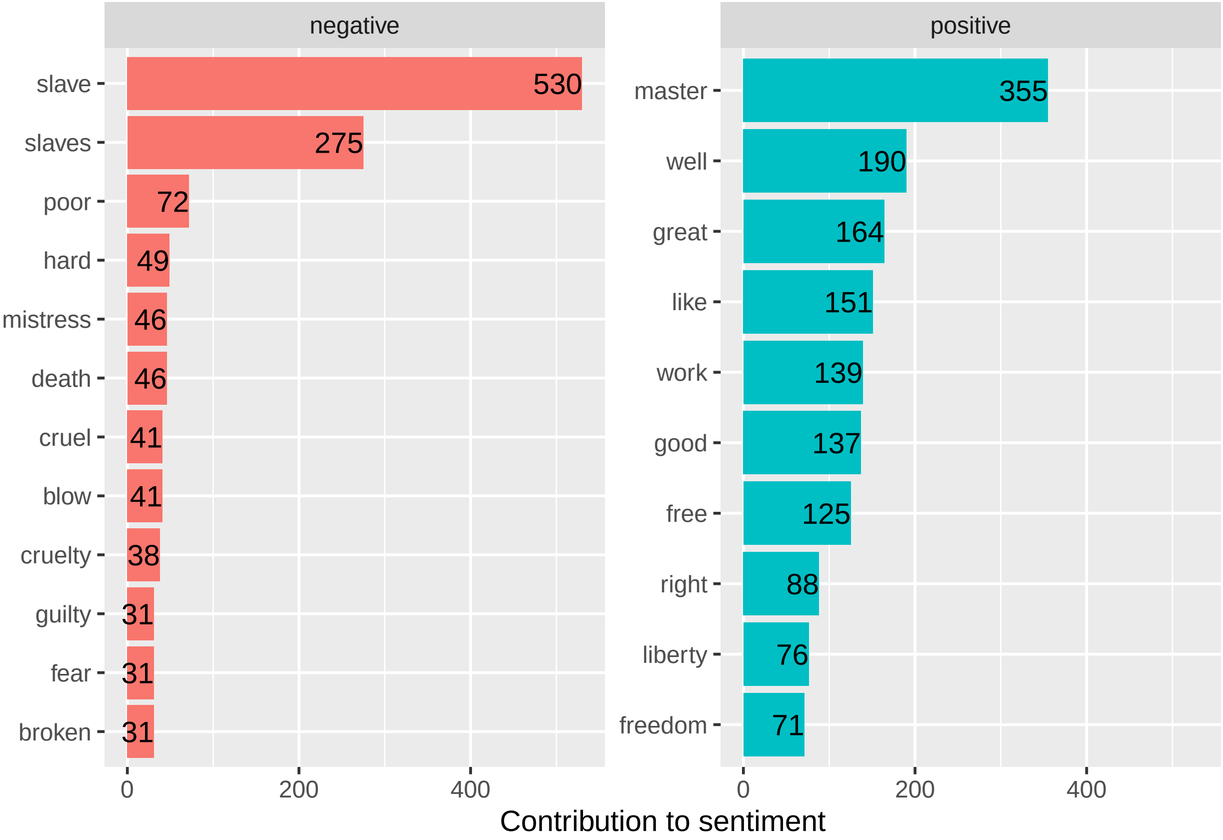
The dataset consist of word, sentiment and Freq.

Frequent used positive and negative words

The most frequent used words for positive sentiments and negative sentiments.

```
Bondage_Freedom_Sentiment_total <- Bondage_Freedom_Chapters %>%
  unnest_tokens(word, text) %>% inner_join(get_sentiments("bing")) %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup()
```

```
Bondage_Freedom_Sentiment_total %>%
  group_by(sentiment) %>%
  top_n(10) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to sentiment",
       x = NULL) +
  coord_flip() +
  geom_text(aes(label = n, hjust = 1.0))
```

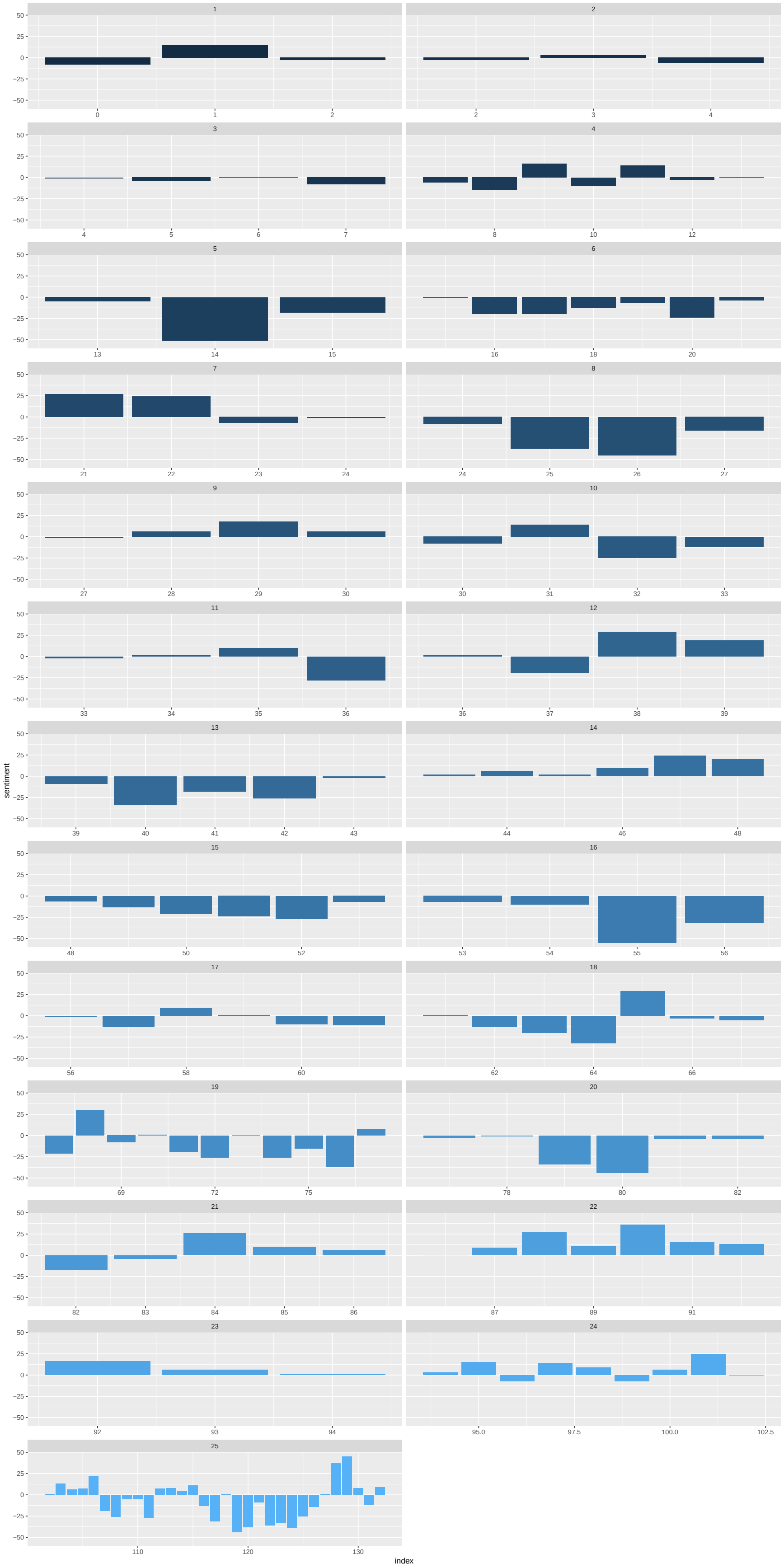


Chapter wise positive and negative words

Apply group by on Chapter so we can get chapter based positive/negative sentiments words. Let's get total number of positive and negative word count using `bing` lexion.

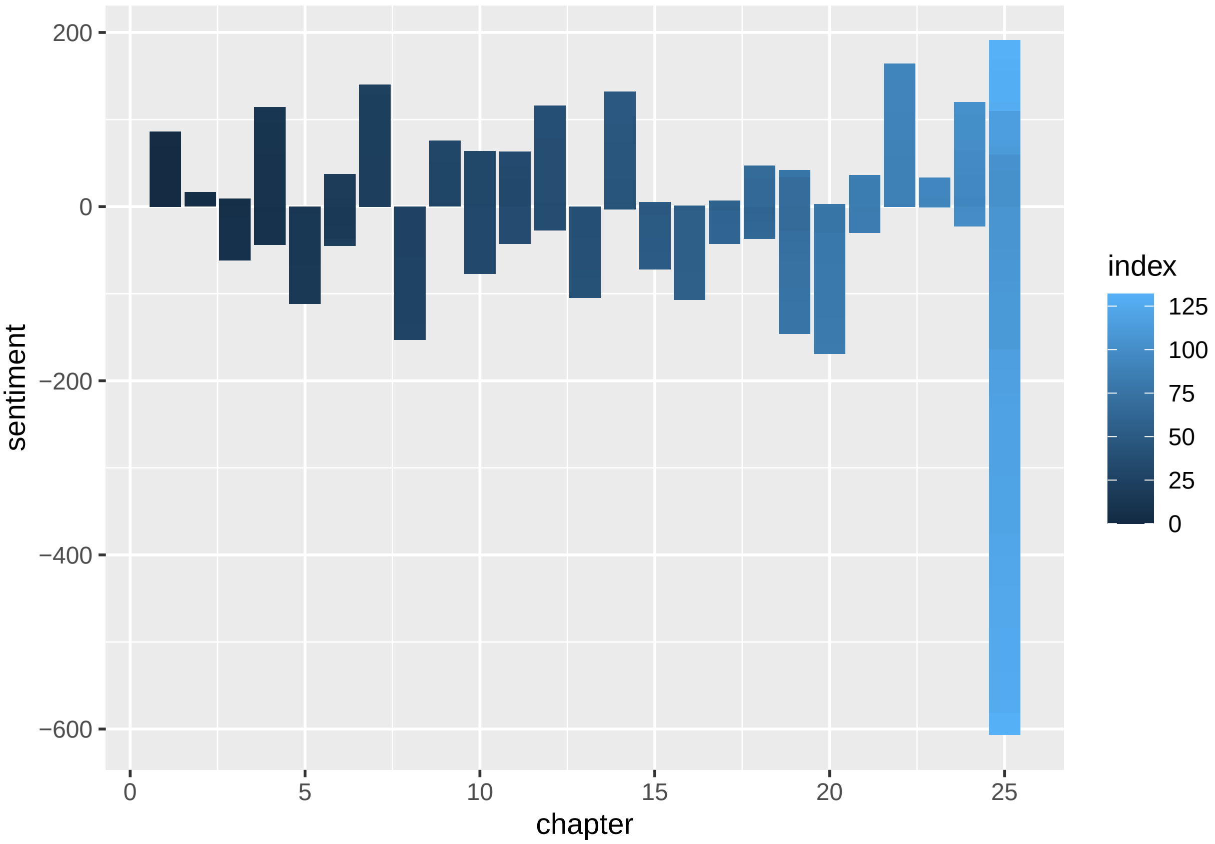
```
Bondage_Freedom_Sentiment <- Bondage_Freedom_Chapters %>%
  unnest_tokens(word, text) %>%
  inner_join(get_sentiments("bing")) %>%
  count(chapter, index = linenumber %/% 80, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(sentiment = positive - negative)
```

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


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
50
25
0
−25
−50
sentiment
index
0 1 2
2 3 4
4 5 6 7
8 10 12
13 14 15
16 18 20
21 22 23 24
24 25 26 27
27 28 29 30
30 31 32 33
33 34 35 36
36 37 38 39
39 40 41 42 43
44 46 48
48 50 52
53 54 55 56
56 58 60
62 64 66
69 72 75
78 80 82
82 83 84 85 86
87 89 91
92 93 94
95.0 97.5 100.0 102.5
110 120 130

The book has 25 chapters, using Finn lexicon we can see which chapter has more positive words and which chapter has more negative words. The suggestion from the book is to use ~ 80 lines of text, and let's try that.

```
Positive_Negative_Count<- Bondage_Freedom_Chapters %>%
   unnest_tokens(word, text) %>%
  inner_join(get_sentiments("afinn")) %>%
  group_by(index = linenumber %/% 80, chapter)%>%
  summarise(sentiment = sum(value))

Positive_Negative_Count%>%
  ggplot(aes(chapter, sentiment, fill=index)) +
  geom_col()
```

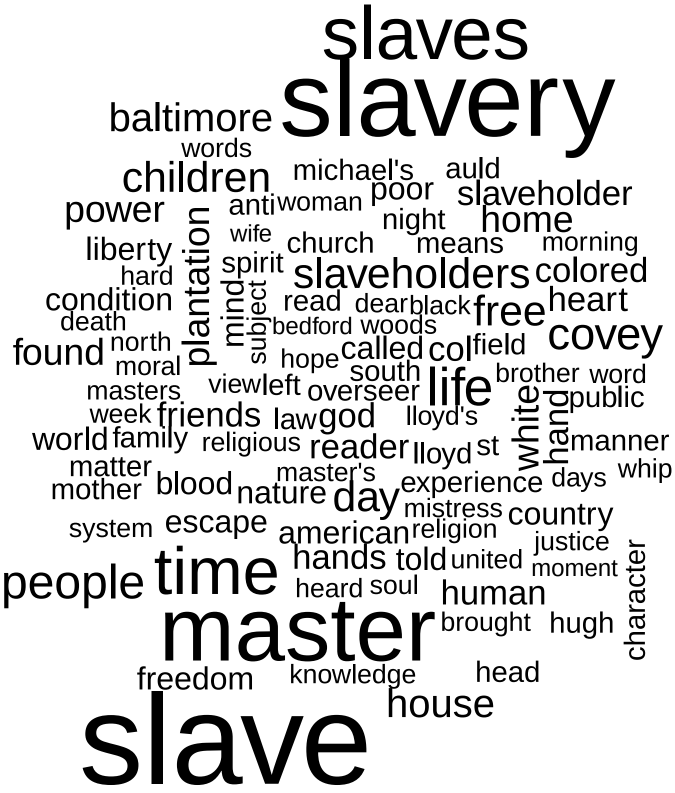


From the above graph we can see Chapter 25 has more negative sentimants among all other chapters.

Wordcloud

Let's look at the most common words in "My Bondage and My Freedom".

```
total_word_count <- Bondage_Freedom_Chapters %>% unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  count(word, sort = TRUE) %>% filter(word != "thomas" )

total_word_count %>% with(wordcloud(word, n, max.words = 100))
```



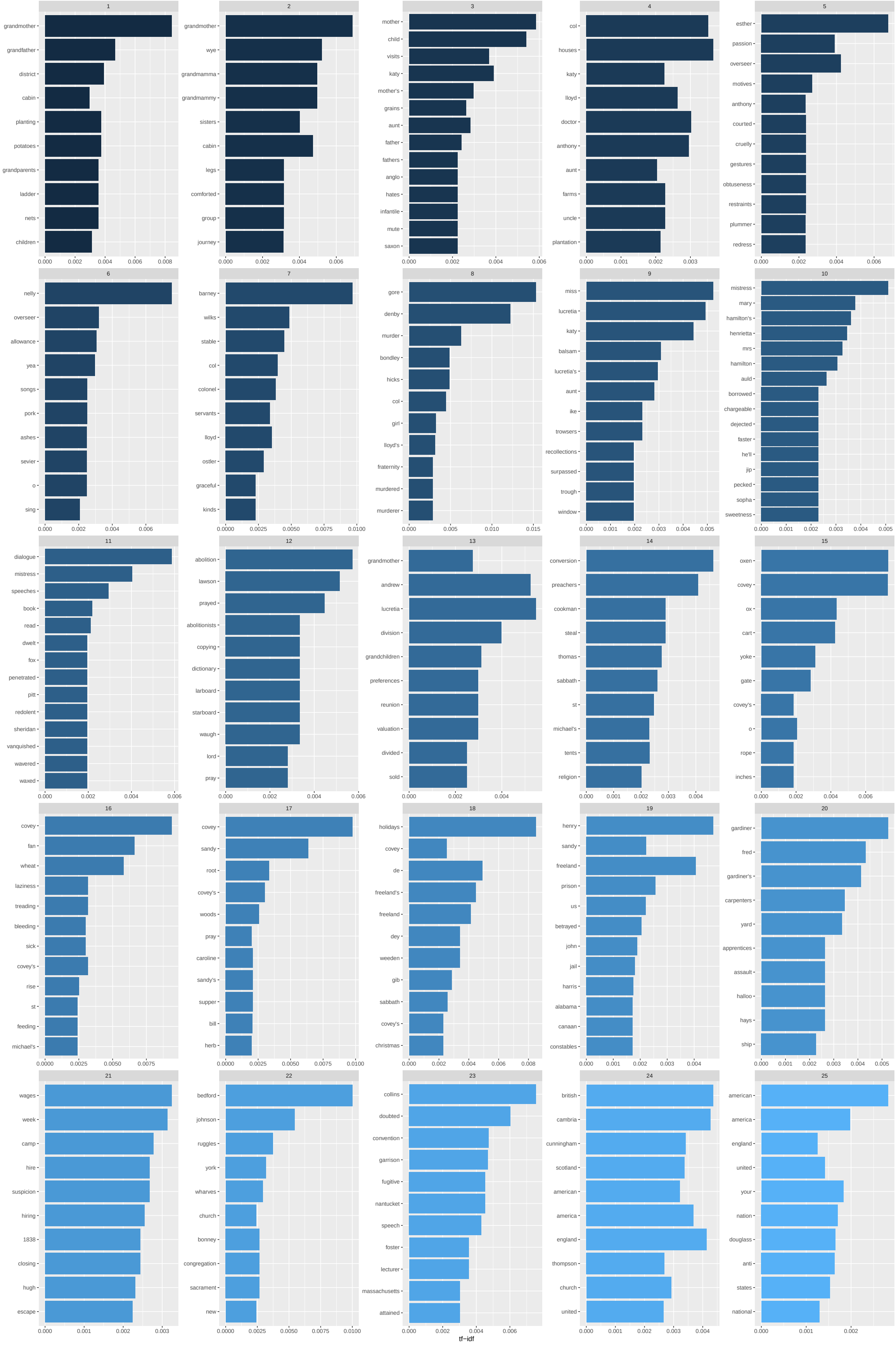
TF-IDF

The statistic tf-idf is intended to measure how important a word is to a document in a collection (or corpus) of documents.

```
book_words <- Bondage_Freedom_Chapters %>%
  unnest_tokens(word, text) %>%
  count(chapter, word, sort = TRUE)

total_words <- book_words %>%
  group_by(chapter) %>%
  dplyr::summarize(total = sum(n))

book_words <- left_join(book_words, total_words)

book_words <- book_words %>%
  bind_tf_idf(word, chapter, n)

book_words %>%
  select(-total) %>%
  arrange(desc(tf_idf))
```

```
## # A tibble: 34,361 x 6
##    chapter word            n      tf   idf  tf_idf
##      <int> <chr>       <int>   <dbl> <dbl>   <dbl>
##  1       8 gore           19 0.00722  2.12 0.0153
##  2       8 denby          10 0.00380  3.22 0.0122
##  3      22 bedford        33 0.00546  1.83 0.0100
##  4      17 covey          46 0.00956  1.02 0.00976
##  5       7 barney         10 0.00300  3.22 0.00967
##  6      16 covey          28 0.00919  1.02 0.00939
##  7      18 holidays       19 0.00336  2.53 0.00850
##  8       1 grandmother    18 0.00664  1.27 0.00845
##  9      23 collins         5 0.00235  3.22 0.00755
## 10       6 nelly          12 0.00234  3.22 0.00755
## # ... with 34,351 more rows
```

```
book_words %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  group_by(chapter) %>%
  top_n(10) %>%
  ungroup() %>%
  ggplot(aes(word, tf_idf, fill = chapter)) +
  geom_col(aes(reorder(word, tf_idf),tf_idf),stat = "identity",show.legend = FALSE) +
  labs(x = NULL, y = "tf-idf") +
  facet_wrap(~chapter, scales = "free") +
  coord_flip()
```


1
grandmother
grandfather
district
cabin
planting
potatoes
grandparents
ladder
nets
children
0.000 0.002 0.004 0.006 0.008
2
grandmother
wye
grandmamma
grandmammy
sisters
cabin
legs
comforted
group
journey
0.000 0.002 0.004 0.006
3
mother
child
visits
katy
mother's
grains
aunt
father
fathers
anglo
hates
infantile
mute
saxon
0.000 0.002 0.004 0.006
4
col
houses
katy
lloyd
doctor
anthony
aunt
farms
uncle
plantation
0.000 0.001 0.002 0.003
5
esther
passion
overseer
motives
anthony
courted
cruelly
gestures
obtuseness
restraints
plummer
redress
0.000 0.002 0.004 0.006
6
nelly
overseer
allowance
yea
songs
pork
ashes
sevier
o
sing
0.000 0.002 0.004 0.006
7
barney
wilks
stable
col
colonel
servants
lloyd
ostler
graceful
kinds
0.0000 0.0025 0.0050 0.0075 0.0100
8
gore
denby
murder
bondley
hicks
col
girl
lloyd's
fraternity
murdered
murderer
0.000 0.005 0.010 0.015
9
miss
lucretia
katy
balsam
lucretia's
aunt
ike
trowsers
recollections
surpassed
trough
window
0.000 0.001 0.002 0.003 0.004 0.005
10
mistress
mary
hamilton's
henrietta
mrs
hamilton
auld
borrowed
chargeable
dejected
faster
he'll
jip
pecked
sopha
sweetness
0.000 0.001 0.002 0.003 0.004 0.005
11
dialogue
mistress
speeches
book
read
dwelt
fox
penetrated
pitt
redolent
sheridan
vanquished
wavered
waxed
0.000 0.002 0.004 0.006
12
abolition
lawson
prayed
abolitionists
copying
dictionary
larboard
starboard
waugh
lord
pray
0.000 0.002 0.004 0.006
13
grandmother
andrew
lucretia
division
grandchildren
preferences
reunion
valuation
divided
sold
0.000 0.002 0.004
14
conversion
preachers
cookman
steal
thomas
sabbath
st
michael's
tents
religion
0.000 0.001 0.002 0.003 0.004
15
oxen
covey
ox
cart
yoke
gate
covey's
o
rope
inches
0.000 0.002 0.004 0.006
16
covey
fan
wheat
laziness
treading
bleeding
sick
covey's
rise
st
feeding
michael's
0.0000 0.0025 0.0050 0.0075
17
covey
sandy
root
covey's
woods
pray
caroline
sandy's
supper
bill
herb
0.0000 0.0025 0.0050 0.0075 0.0100
18
holidays
covey
de
freeland's
freeland
dey
weeden
gib
sabbath
covey's
christmas
0.000 0.002 0.004 0.006 0.008
19
henry
sandy
freeland
prison
us
betrayed
john
jail
harris
alabama
canaan
constables
0.000 0.001 0.002 0.003 0.004
20
gardiner
fred
gardiner's
carpenters
yard
apprentices
assault
halloo
hays
ship
0.000 0.001 0.002 0.003 0.004 0.005
21
wages
week
camp
hire
suspicion
hiring
1838
closing
hugh
escape
0.000 0.001 0.002 0.003
22
bedford
johnson
ruggles
york
wharves
church
bonney
congregation
sacrament
new
0.0000 0.0025 0.0050 0.0075 0.0100
23
collins
doubted
convention
garrison
fugitive
nantucket
speech
foster
lecturer
massachusetts
attained
0.000 0.002 0.004 0.006
tf−idf
24
british
cambria
cunningham
scotland
american
america
england
thompson
church
united
0.000 0.001 0.002 0.003 0.004
25
american
america
england
united
your
nation
douglass
anti
states
national
0.000 0.001 0.002

Conclusion

Sentiment analysis provides a way to understand the attitudes and opinions expressed in texts. We can use sentiment analysis to understand how a narrative arc changes throughout its course or what words with emotional and opinion content are important for a particular text. In this assignment, we added a new corpus from ‘gutenbergr’ package and applied sentiment analysis. From the analysis, we came to know mostly used positive/negative words and chapter wise sentiment analysis. Chapter 25 has more negative sentiments and chapter 7, and chapter 22 have more positive sentiments. We explored TF_IDF analysis also.