Test Mining with R

Smiti Kaul

Feb 9 - present, 2018

Following the article

```
ids <- 1:5
works_sample <- gutenberg_download(gutenberg_id = ids)
glimpse(works_sample)
names(gutenberg_metadata)
works_sample <- gutenberg_download(gutenberg_id = ids, meta_fields = c("title",
    "author"))
glimpse(works_sample)
ids <- filter(gutenberg_subjects, subject_type == "lcc", subject == "PR")
glimpse(ids)
ids_has_text <- filter(gutenberg_metadata, gutenberg_id %in% ids$gutenberg_id,
    has_text == TRUE)
glimpse(ids_has_text)

set.seed(123)
ids_sample <- sample_n(ids_has_text, 10)
glimpse(ids_sample)
works_pr <- gutenberg_download(gutenberg_id = ids_sample$gutenberg_id, meta_fields = c("author",
    "title"))
glimpse(works_pr)
```

Getting Started

```
## [1] "Because I could not stop for Death -"
## [2] "He kindly stopped for me -"
## [3] "The Carriage held but just Ourselves -"
## [4] "and Immortality"
```

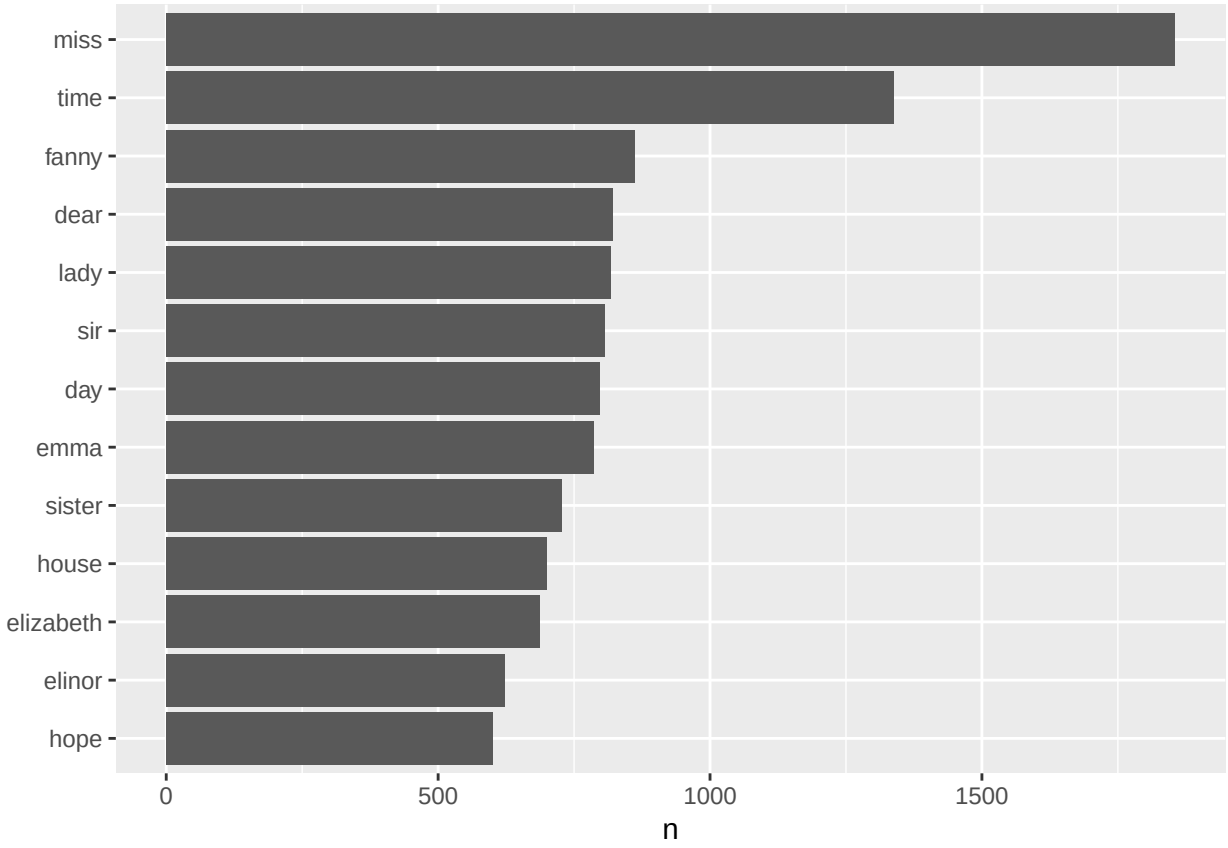
```
## # A tibble: 4 x 2
##    line text
##   <int> <chr>
## 1     1 Because I could not stop for Death -
## 2     2 He kindly stopped for me -
## 3     3 The Carriage held but just Ourselves -
## 4     4 and Immortality
```

```
## # A tibble: 20 x 2
##     line word
##    <int> <chr>
##  1     1 because
##  2     1 i
##  3     1 could
##  4     1 not
##  5     1 stop
##  6     1 for
##  7     1 death
```

```
##  8     2 he
##  9     2 kindly
## 10     2 stopped
## 11     2 for
## 12     2 me
## 13     3 the
## 14     3 carriage
## 15     3 held
## 16     3 but
## 17     3 just
## 18     3 ourselves
## 19     4 and
## 20     4 immortality
```

```
## # A tibble: 73,422 x 4
##    text                  book                linenumber chapter
##    <chr>                 <fct>                    <int>   <int>
##  1 SENSE AND SENSIBILITY Sense & Sensibility          1       0
##  2 ""                    Sense & Sensibility          2       0
##  3 by Jane Austen        Sense & Sensibility          3       0
##  4 ""                    Sense & Sensibility          4       0
##  5 (1811)                Sense & Sensibility          5       0
##  6 ""                    Sense & Sensibility          6       0
##  7 ""                    Sense & Sensibility          7       0
##  8 ""                    Sense & Sensibility          8       0
##  9 ""                    Sense & Sensibility          9       0
## 10 CHAPTER 1             Sense & Sensibility         10       1
## # ... with 73,412 more rows
```

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



Gutenberg: tidy text format

```
hgwells <- gutenberg_download(c(35, 36, 5230, 159))
```

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

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

```
bronte <- gutenberg_download(c(1260, 768, 969, 9182, 767))
```

```
tidy_hgwells <- hgwells %>% unnest_tokens(word, text) %>% anti_join(stop_words)
```

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

```
tidy_hgwells %>% count(word, sort = TRUE)
```

```
## # A tibble: 11,769 x 2
##    word       n
##    <chr>  <int>
##  1 time     454
##  2 people   302
##  3 door     260
##  4 heard    249
##  5 black    232
##  6 stood    229
##  7 white    222
##  8 hand     218
##  9 kemp     213
```

```
## 10 eyes     210
## # ... with 11,759 more rows
```

```
tidy_bronte <- bronte %>% unnest_tokens(word, text) %>% anti_join(stop_words)
```

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

```
tidy_bronte %>% count(word, sort = TRUE)
```

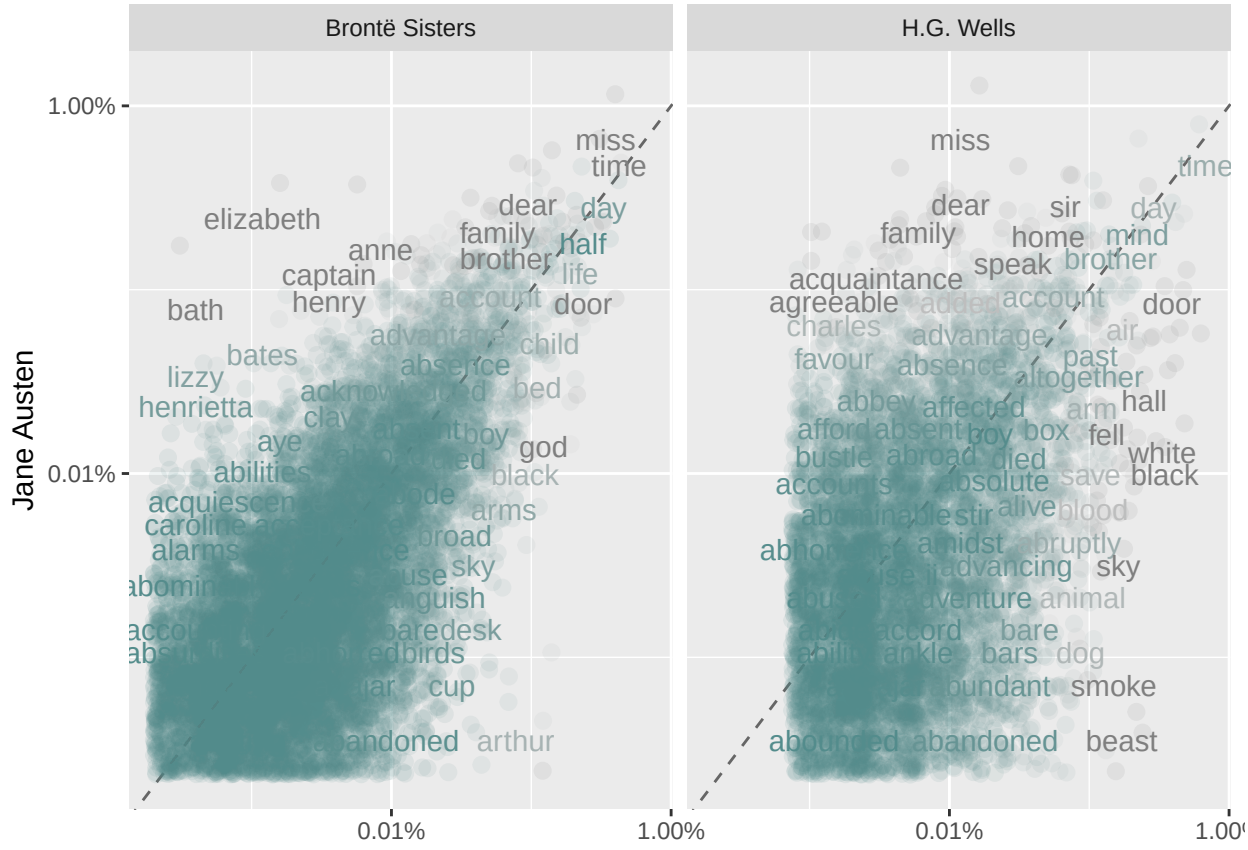
```
## # A tibble: 23,050 x 2
##    word       n
##    <chr>  <int>
##  1 time    1065
##  2 miss     855
##  3 day      827
##  4 hand     768
##  5 eyes     713
##  6 night    647
##  7 heart    638
##  8 looked   601
##  9 door     592
## 10 half     586
## # ... with 23,040 more rows
```

```
frequency <- bind_rows(mutate(tidy_bronte, author = "Brontë Sisters"), mutate(tidy_hgwells,
    author = "H.G. Wells"), mutate(tidy_books, author = "Jane Austen")) %>%
    mutate(word = str_extract(word, "[a-z']+")) %>% count(author, word) %>%
    group_by(author) %>% mutate(proportion = n/sum(n)) %>% select(-n) %>% spread(author,
    proportion) %>% gather(author, proportion, `Brontë Sisters`:`H.G. Wells`)

# expect a warning about rows with missing values being removed
ggplot(frequency, aes(x = proportion, y = `Jane Austen`, color = abs(`Jane Austen` -
    proportion))) + geom_abline(color = "gray40", lty = 2) + geom_jitter(alpha = 0.1,
    size = 2.5, width = 0.3, height = 0.3) + geom_text(aes(label = word), check_overlap = TRUE,
    vjust = 1.5) + scale_x_log10(labels = percent_format()) + scale_y_log10(labels = percent_format()) +
    scale_color_gradient(limits = c(0, 0.001), low = "darkslategray4", high = "gray75") +
    facet_wrap(~author, ncol = 2) + theme(legend.position = "none") + labs(y = "Jane Austen",
    x = NULL)
```

```
## Warning: Removed 41357 rows containing missing values (geom_point).
```

```
## Warning: Removed 41359 rows containing missing values (geom_text).
```




```
cor.test(data = frequency[frequency$author == "Brontë Sisters", ], ~proportion +
    `Jane Austen`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and Jane Austen
## t = 119.65, df = 10404, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7527869 0.7689641
## sample estimates:
##       cor
## 0.7609938
```

```
cor.test(data = frequency[frequency$author == "H.G. Wells", ], ~proportion +
    `Jane Austen`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and Jane Austen
## t = 36.441, df = 6053, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4032800 0.4445987
## sample estimates:
##       cor
```

```
## 0.4241601
```

Sentiment Analysis

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

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

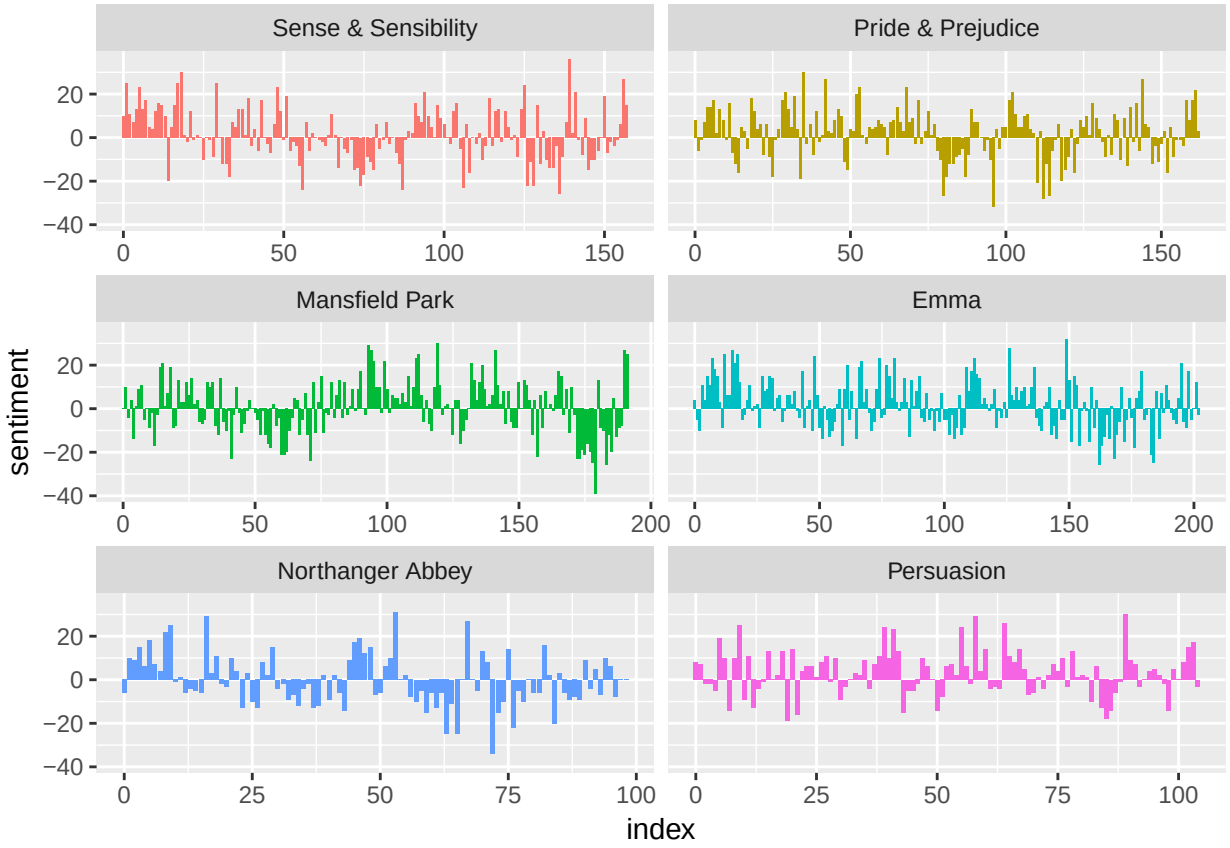
```
## Joining, by = "word"
```

```
## # A tibble: 298 x 2
##    word          n
##    <chr>     <int>
##  1 friend      166
##  2 hope        143
##  3 happy       125
##  4 love        117
##  5 deal         92
##  6 found        92
##  7 happiness    76
##  8 pretty       68
##  9 true         66
## 10 comfort      65
## # ... with 288 more rows
```

```
janeaustensentiment <- tidy_books %>% inner_join(get_sentiments("bing")) %>%
    count(book, index = linenumber%/%80, sentiment) %>% spread(sentiment, n,
    fill = 0) %>% mutate(sentiment = positive - negative)
```

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

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



Most common positive and negative words

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

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

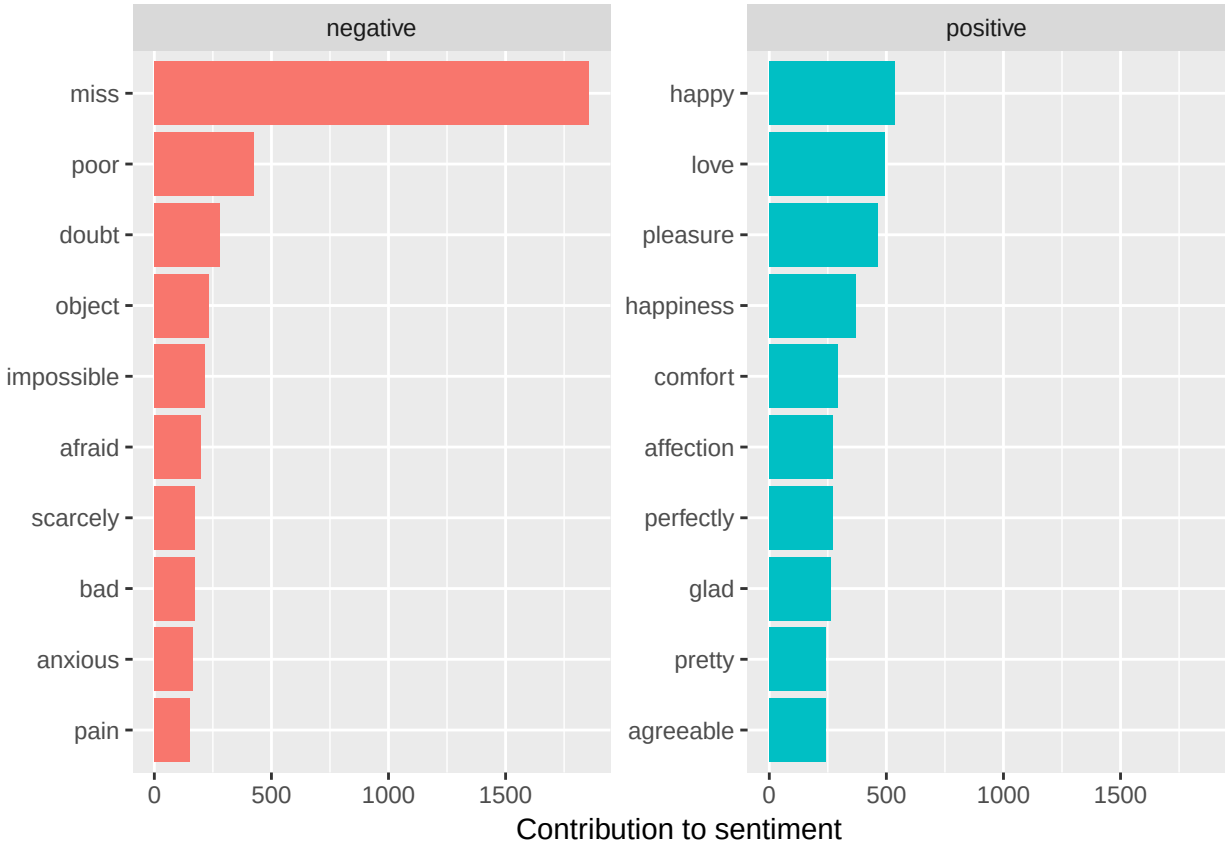
```
bing_word_counts
```

```
## # A tibble: 2,555 x 3
##    word      sentiment     n
##    <chr>     <chr>     <int>
##  1 miss      negative   1855
##  2 happy     positive    534
##  3 love      positive    495
##  4 pleasure  positive    462
##  5 poor      negative    424
##  6 happiness positive    369
##  7 comfort   positive    292
##  8 doubt     negative    281
##  9 affection positive    272
## 10 perfectly positive    271
## # ... with 2,545 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()
```

```
## Selecting by n
```




```
# add 'miss' as a custom stop word
custom_stop_words <- bind_rows(data_frame(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))
```

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


```
tidy_books %>% inner_join(get_sentiments("bing")) %>% count(word, sentiment,
    sort = TRUE) %>% acast(word ~ sentiment, value.var = "n", fill = 0) %>%
    comparison.cloud(colors = c("gray20", "gray80"), max.words = 100)
```

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

Units beyond just words