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"
```

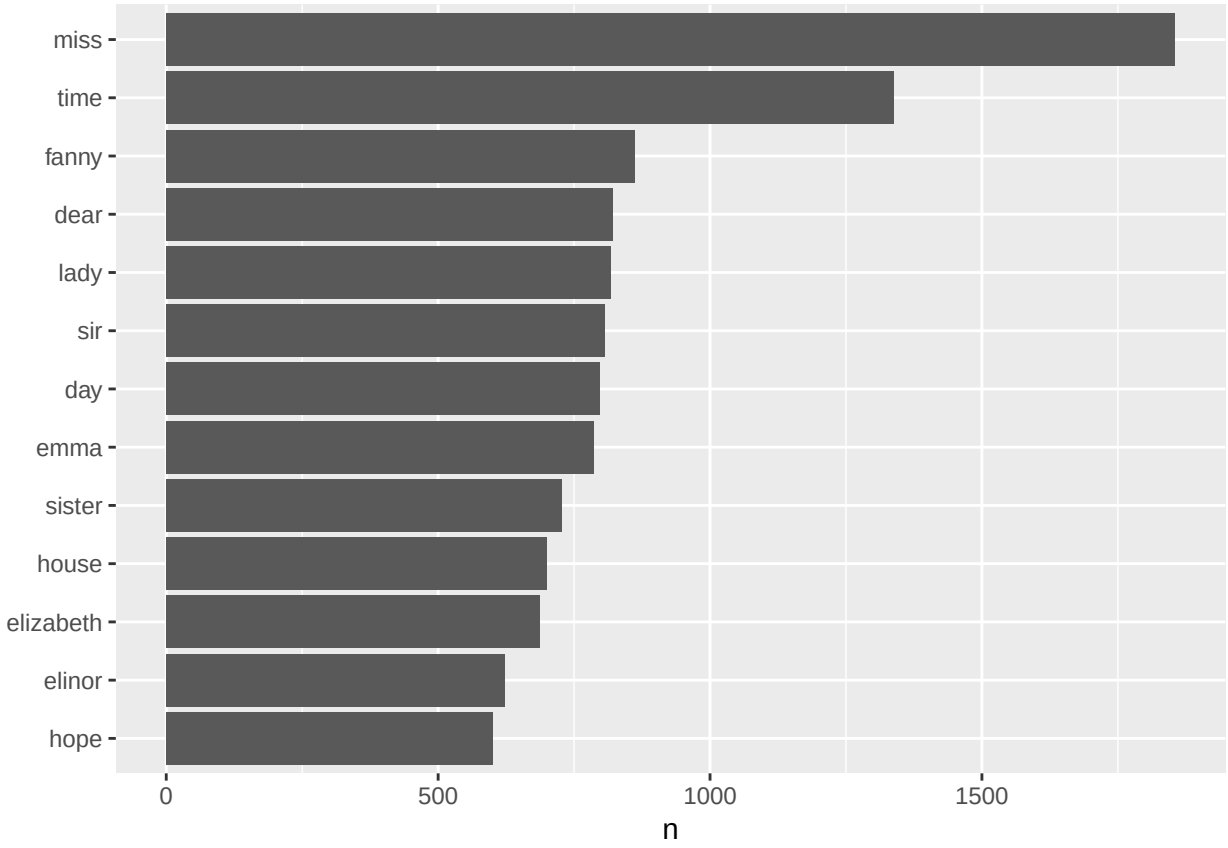
```
## # A tibble: 4 x 2
##    line text
##   <int> <chr>
## 1     1 Because I could not stop for Death -
## 2     2 He kindly stopped for me -
## 3     3 The Carriage held but just Ourselves -
## 4     4 and Immortality
```

```
## # 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)
```

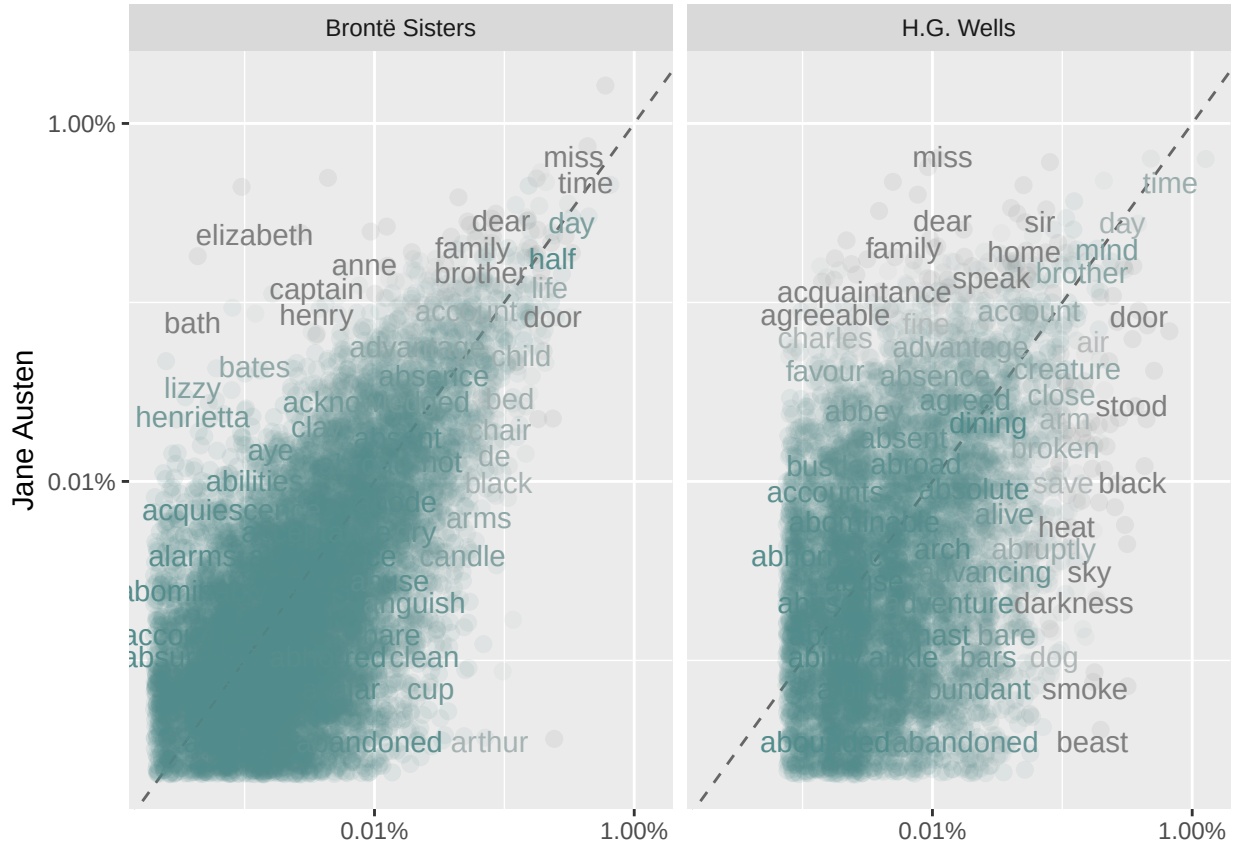
```
## # A tibble: 23,050 x 2
##    word       n
##    <chr>  <int>
##  1 time    1065
##  2 miss     855
##  3 day      827
##  4 hand     768
##  5 eyes     713
##  6 night    647
##  7 heart    638
##  8 looked   601
##  9 door     592
## 10 half     586
## # ... with 23,040 more rows
```

```
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)
```

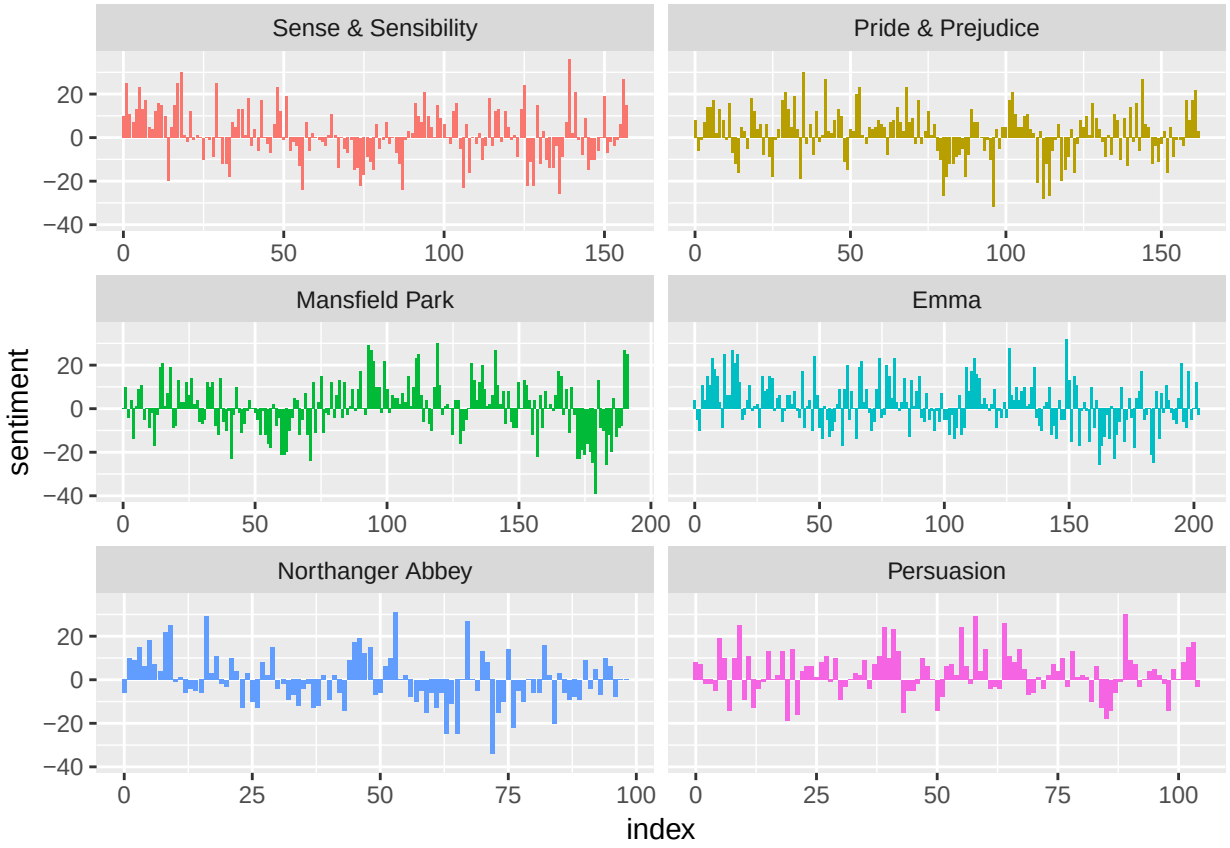
```
## Joining, by = "word"
```

```
## # 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"
```

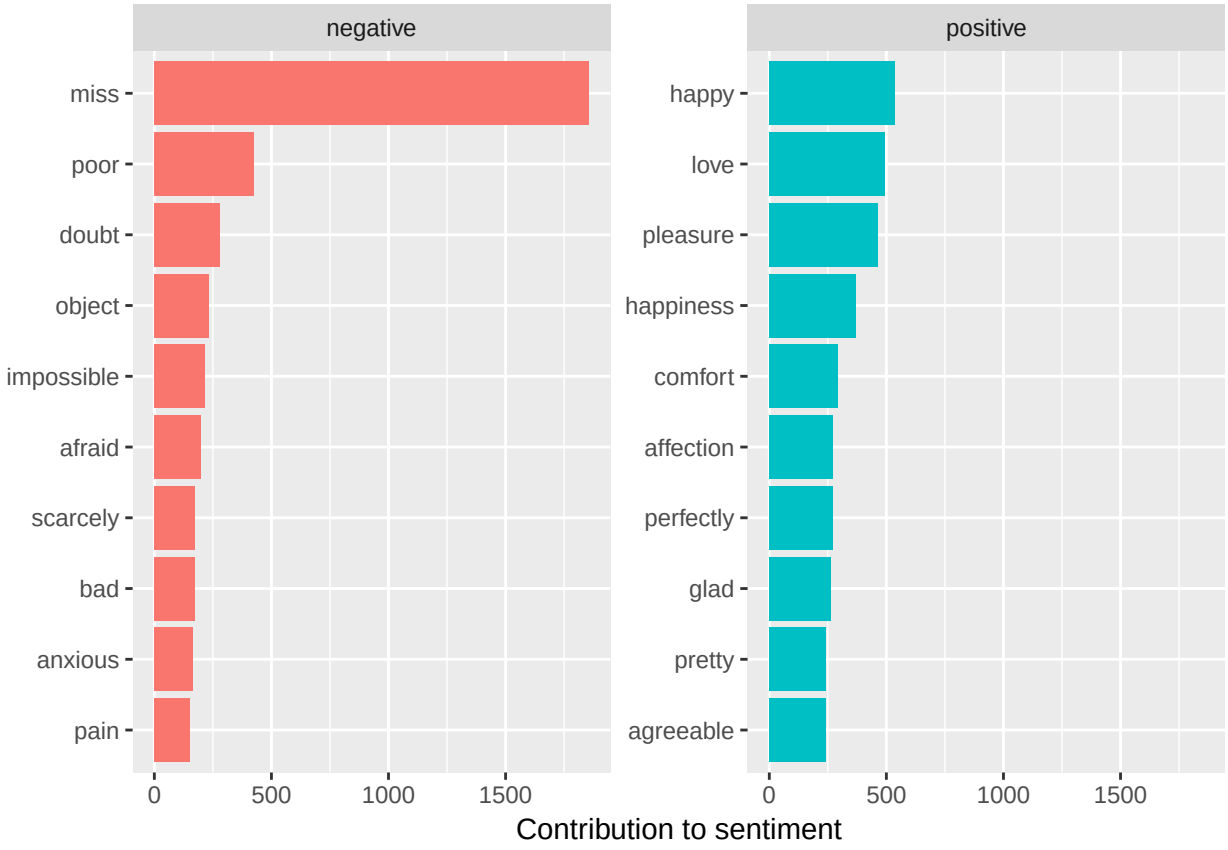
```
bing_word_counts
```

```
## # 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"
```

```
## Warning in wordcloud(word, n, max.words = 100): time could not be fit on
## page. It will not be plotted.
```


```
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

Word and Document Frequency

Term Frequency

```
book_words <- austen_books() %>% unnest_tokens(word, text) %>% count(book, word,
    sort = TRUE) %>% ungroup()

total_words <- book_words %>% group_by(book) %>% summarize(total = sum(n))

book_words <- left_join(book_words, total_words)
```

```
## Joining, by = "book"
```

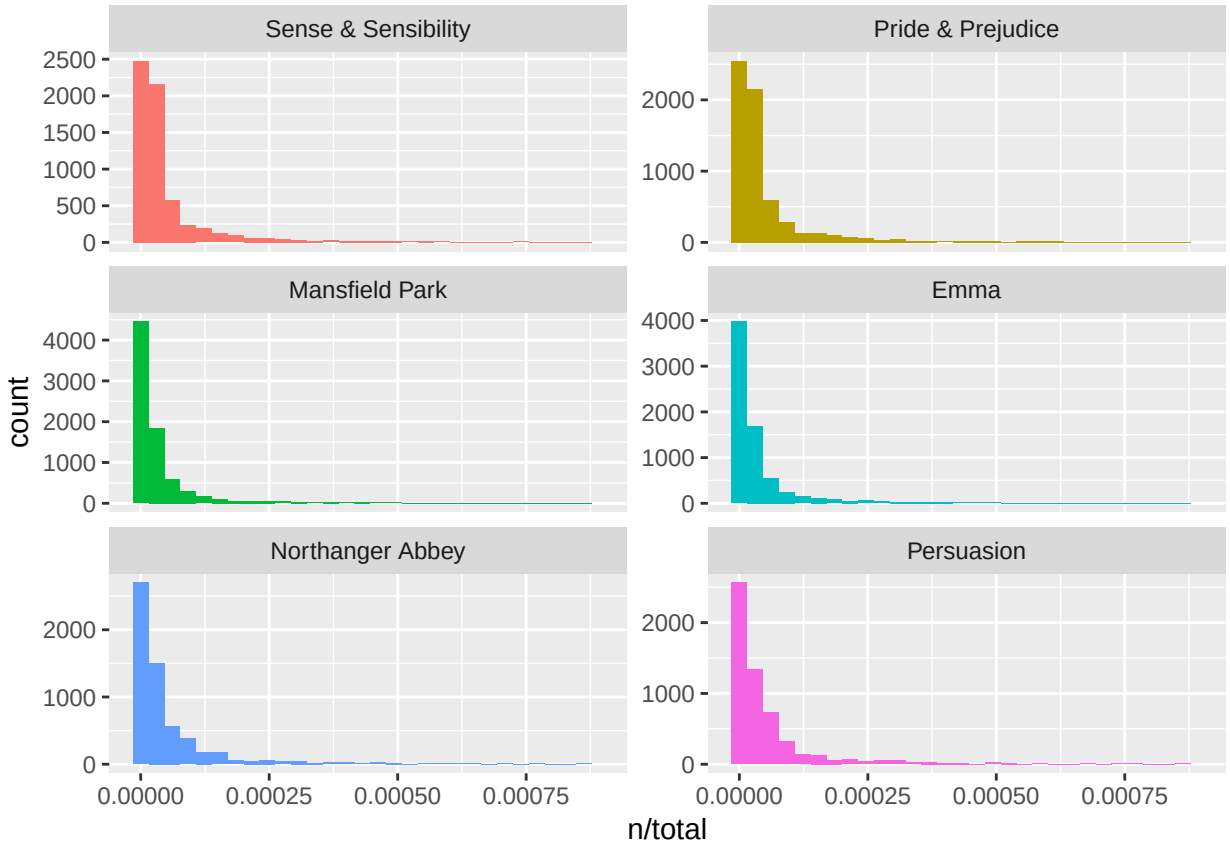
```
head(book_words)
```

```
## # A tibble: 6 x 4
##   book           word      n  total
##   <fct>          <chr> <int>  <int>
## 1 Mansfield Park the    6206 160460
## 2 Mansfield Park to     5475 160460
## 3 Mansfield Park and    5438 160460
## 4 Emma           to     5239 160996
## 5 Emma           the    5201 160996
## 6 Emma           and    4896 160996
```

```
ggplot(book_words, aes(n/total, fill = book)) + geom_histogram(show.legend = FALSE) +
    xlim(NA, 9e-04) + facet_wrap(~book, ncol = 2, scales = "free_y")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 896 rows containing non-finite values (stat_bin).
```



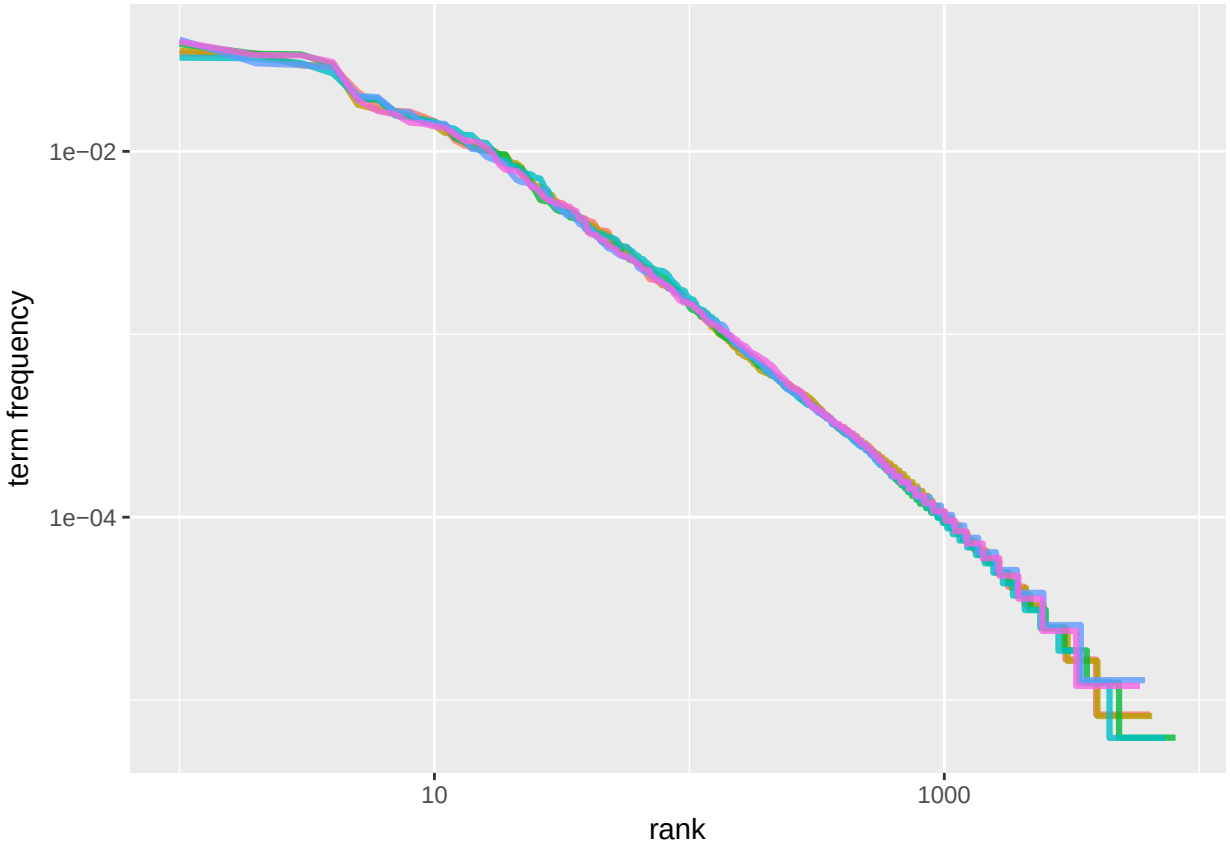
Zipf's Law

```
freq_by_rank <- book_words %>% group_by(book) %>% mutate(rank = row_number(),
    `term frequency` = n/total)
```

```
head(freq_by_rank)
```

```
## # A tibble: 6 x 6
## # Groups:   book [2]
##   book           word      n  total  rank `term frequency`
##   <fct>          <chr> <int>  <int> <int>            <dbl>
## 1 Mansfield Park the     6206 160460     1           0.0387
## 2 Mansfield Park to      5475 160460     2           0.0341
## 3 Mansfield Park and     5438 160460     3           0.0339
## 4 Emma           to      5239 160996     1           0.0325
## 5 Emma           the     5201 160996     2           0.0323
## 6 Emma           and     4896 160996     3           0.0304
```

```
freq_by_rank %>% ggplot(aes(rank, `term frequency`, color = book)) + geom_line(size = 1.1,
    alpha = 0.8, show.legend = FALSE) + scale_x_log10() + scale_y_log10()
```



The bind_tf_idf function

```
book_words <- book_words %>% bind_tf_idf(word, book, n)
head(book_words)
```

```
## # A tibble: 6 x 7
##   book           word      n  total     tf   idf tf_idf
##   <fct>          <chr> <int>  <int>  <dbl> <dbl>  <dbl>
## 1 Mansfield Park the     6206 160460 0.0387     0      0
## 2 Mansfield Park to      5475 160460 0.0341     0      0
## 3 Mansfield Park and     5438 160460 0.0339     0      0
## 4 Emma           to      5239 160996 0.0325     0      0
## 5 Emma           the     5201 160996 0.0323     0      0
## 6 Emma           and     4896 160996 0.0304     0      0
```

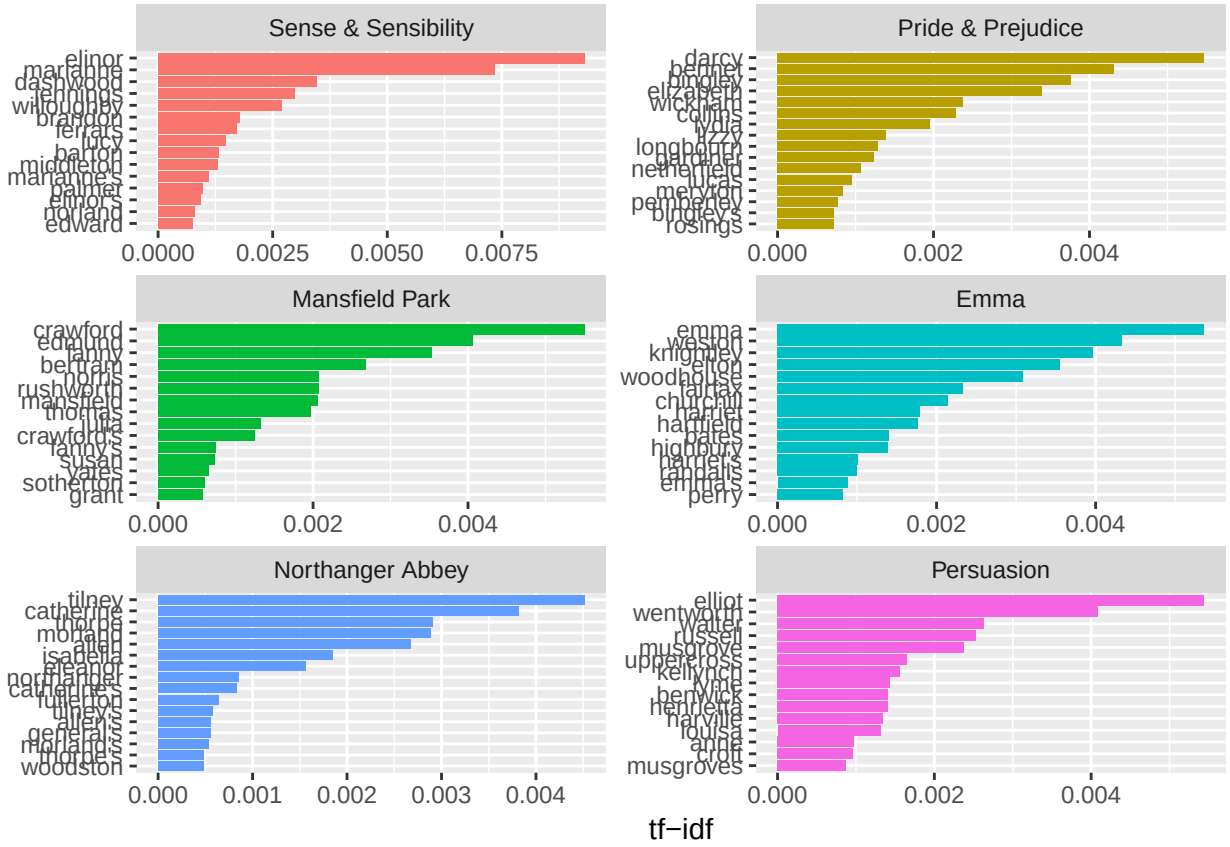
```
book_words %>% select(-total) %>% arrange(desc(tf_idf))
```

```
## # A tibble: 40,379 x 6
##    book                word          n      tf   idf  tf_idf
##    <fct>               <chr>     <int>   <dbl> <dbl>   <dbl>
##  1 Sense & Sensibility elinor      623 0.00519  1.79 0.00931
##  2 Sense & Sensibility marianne    492 0.00410  1.79 0.00735
##  3 Mansfield Park      crawford    493 0.00307  1.79 0.00551
##  4 Pride & Prejudice   darcy       373 0.00305  1.79 0.00547
##  5 Persuasion          elliot      254 0.00304  1.79 0.00544
##  6 Emma                emma        786 0.00488  1.10 0.00536
```

```
##  7 Northanger Abbey    tilney      196 0.00252  1.79 0.00452
##  8 Emma                weston      389 0.00242  1.79 0.00433
##  9 Pride & Prejudice   bennet      294 0.00241  1.79 0.00431
## 10 Persuasion          wentworth   191 0.00228  1.79 0.00409
## # ... with 40,369 more rows
```

```
book_words %>% arrange(desc(tf_idf)) %>% mutate(word = factor(word, levels = rev(unique(word)))) %>%
    group_by(book) %>% top_n(15) %>% ungroup %>% ggplot(aes(word, tf_idf, fill = book)) +
    geom_col(show.legend = FALSE) + labs(x = NULL, y = "tf-idf") + facet_wrap(~book,
    ncol = 2, scales = "free") + coord_flip()
```

```
## Selecting by tf_idf
```



A corpus of physics texts

```
physics <- gutenberg_download(c(37729, 14725, 13476, 5001), meta_fields = "author")
```

```
physics_words <- physics %>% unnest_tokens(word, text) %>% count(author, word,
    sort = TRUE) %>% ungroup()
```

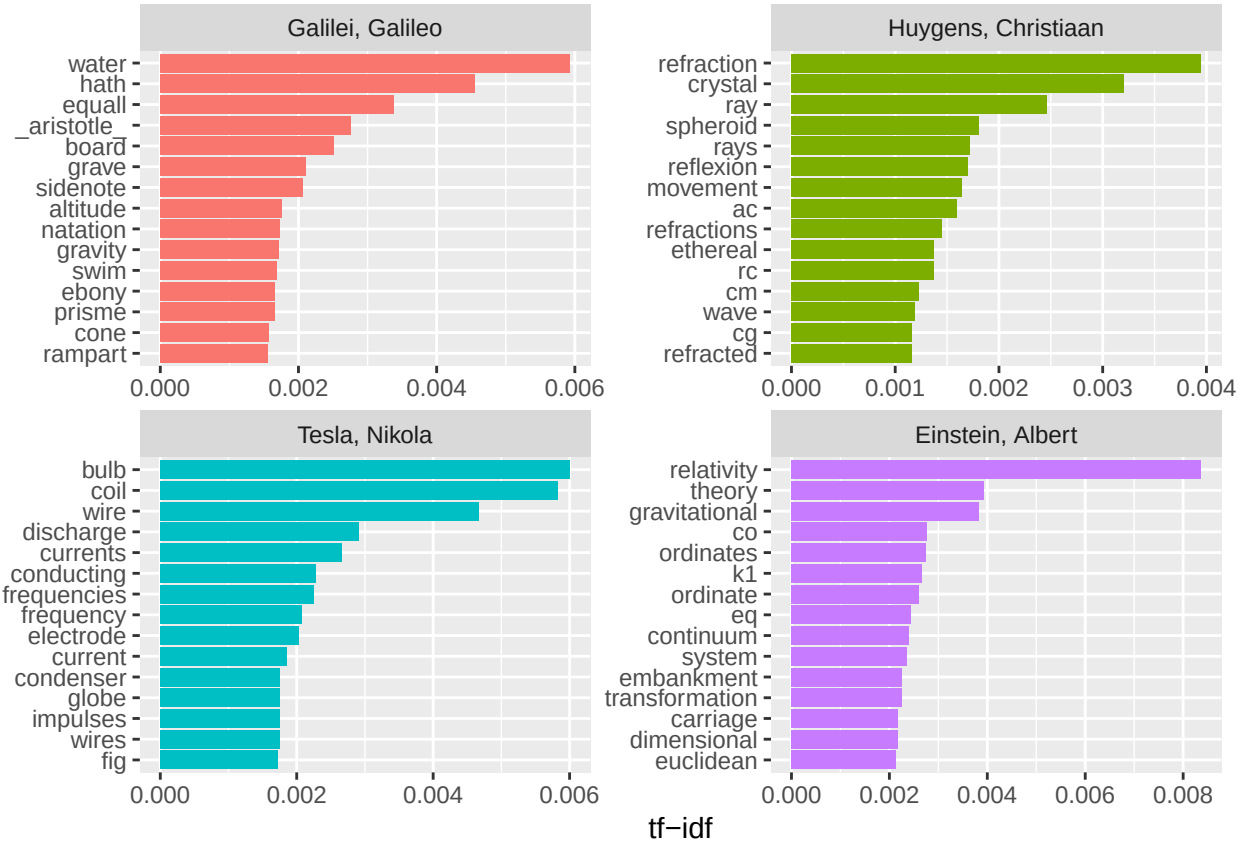
```
head(physics_words)
```

```
## # A tibble: 6 x 3
##   author              word      n
##   <chr>               <chr> <int>
## 1 Galilei, Galileo    the    3760
## 2 Tesla, Nikola       the    3604
```

```
## 3 Huygens, Christiaan the    3553
## 4 Einstein, Albert    the    2994
## 5 Galilei, Galileo    of     2049
## 6 Einstein, Albert    of     2030
```

```
plot_physics <- physics_words %>% bind_tf_idf(word, author, n) %>% arrange(desc(tf_idf)) %>%
    mutate(word = factor(word, levels = rev(unique(word)))) %>% mutate(author = factor(author,
    levels = c("Galilei, Galileo", "Huygens, Christiaan", "Tesla, Nikola", "Einstein, Albert")))

plot_physics %>% group_by(author) %>% top_n(15, tf_idf) %>% ungroup() %>% mutate(word = reorder(word,
    tf_idf)) %>% ggplot(aes(word, tf_idf, fill = author)) + geom_col(show.legend = FALSE) +
    labs(x = NULL, y = "tf-idf") + facet_wrap(~author, ncol = 2, scales = "free") +
    coord_flip()
```



A corpus of _____ texts

```
# gutenberg_metadata %>% filter(title == 'Ramayana, English')
bib <- gutenberg_download(10, meta_fields = "author")
anthem <- gutenberg_download(1249, meta_fields = "author")
sid <- gutenberg_download(2500, meta_fields = "author")
myths_china <- gutenberg_download(15250, meta_fields = "author")
myths_japan <- gutenberg_download(4108, meta_fields = "author")
```

```
anthem_words <- anthem %>% unnest_tokens(word, text) %>% count(author, word,
    sort = TRUE) %>% ungroup()
```

```
anthem_words
```

```
## # A tibble: 2,421 x 3
##    author    word      n
##    <chr>     <chr> <int>
##  1 Rand, Ayn the    1440
##  2 Rand, Ayn we      941
##  3 Rand, Ayn and     883
##  4 Rand, Ayn of      655
##  5 Rand, Ayn to      566
##  6 Rand, Ayn our     409
##  7 Rand, Ayn it      343
##  8 Rand, Ayn in      298
##  9 Rand, Ayn a       293
## 10 Rand, Ayn is      263
## # ... with 2,411 more rows
```

```
bib_words <- bib %>% unnest_tokens(word, text) %>% count(author, word, sort = TRUE) %>%
    ungroup()
```

```
bib_words
```

```
## # A tibble: 12,966 x 3
##    author word      n
##    <chr>  <chr> <int>
##  1 <NA>   the   64023
##  2 <NA>   and   51696
##  3 <NA>   of    34670
##  4 <NA>   to    13580
##  5 <NA>   that  12912
##  6 <NA>   in    12667
##  7 <NA>   he    10419
##  8 <NA>   shall  9838
##  9 <NA>   unto   8997
## 10 <NA>   for    8970
## # ... with 12,956 more rows
```

```
sid_words <- sid %>% unnest_tokens(word, text) %>% count(author, word, sort = TRUE) %>%
    ungroup()
```

```
sid_words
```

```
## # A tibble: 3,606 x 3
##    author         word      n
##    <chr>          <chr> <int>
##  1 Hesse, Hermann the    2045
##  2 Hesse, Hermann and    1365
##  3 Hesse, Hermann to     1145
##  4 Hesse, Hermann of      988
##  5 Hesse, Hermann he      941
##  6 Hesse, Hermann a       911
##  7 Hesse, Hermann his     708
##  8 Hesse, Hermann in      629
##  9 Hesse, Hermann had     524
## 10 Hesse, Hermann was     511
## # ... with 3,596 more rows
```

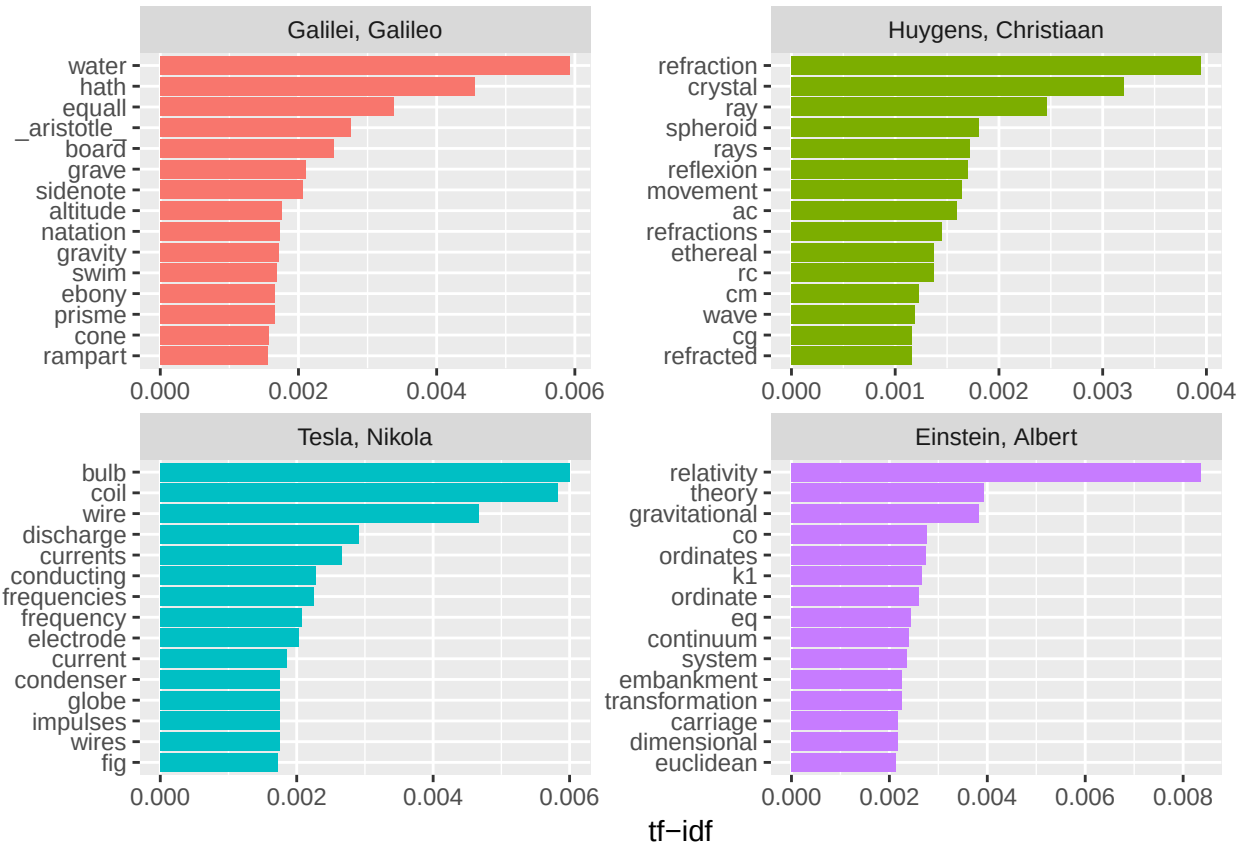
```
myths_china_words <- myths_china %>% unnest_tokens(word, text) %>% count(author,
    word, sort = TRUE) %>% ungroup()

myths_china_words
```

```
## # A tibble: 10,920 x 3
##    author                                      word      n
##    <chr>                                       <chr> <int>
##  1 Werner, E. T. C. (Edward Theodore Chalmers) the   10106
##  2 Werner, E. T. C. (Edward Theodore Chalmers) of     5111
##  3 Werner, E. T. C. (Edward Theodore Chalmers) and    4054
##  4 Werner, E. T. C. (Edward Theodore Chalmers) to     3455
##  5 Werner, E. T. C. (Edward Theodore Chalmers) a      2415
##  6 Werner, E. T. C. (Edward Theodore Chalmers) in     2393
##  7 Werner, E. T. C. (Edward Theodore Chalmers) his    1477
##  8 Werner, E. T. C. (Edward Theodore Chalmers) he     1392
##  9 Werner, E. T. C. (Edward Theodore Chalmers) was    1360
## 10 Werner, E. T. C. (Edward Theodore Chalmers) that    982
## # ... with 10,910 more rows
```

```
plot_bib <- bib_words %>% bind_tf_idf(word, author, n) %>% arrange(desc(tf_idf)) %>%
    mutate(word = factor(word, levels = rev(unique(word)))) %>% mutate(author = factor(author,
    levels = c("Galilei, Galileo", "Huygens, Christiaan", "Tesla, Nikola", "Einstein, Albert")))

plot_physics %>% group_by(author) %>% top_n(15, tf_idf) %>% ungroup() %>% mutate(word = reorder(word,
    tf_idf)) %>% ggplot(aes(word, tf_idf, fill = author)) + geom_col(show.legend = FALSE) +
    labs(x = NULL, y = "tf-idf") + facet_wrap(~author, ncol = 2, scales = "free") +
    coord_flip()
```



Converting to and from non-tidy formats

Tidying a document-term matrix

```
data("AssociatedPress", package = "topicmodels")
AssociatedPress
```

```
## <<DocumentTermMatrix (documents: 2246, terms: 10473)>>
## Non-/sparse entries: 302031/23220327
## Sparsity           : 99%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

```
terms <- Terms(AssociatedPress)
head(terms)
```

```
## [1] "aaron"      "abandon"    "abandoned"  "abandoning" "abbott"
## [6] "abboud"
```

```
ap_td <- tidy(AssociatedPress)
ap_td
```

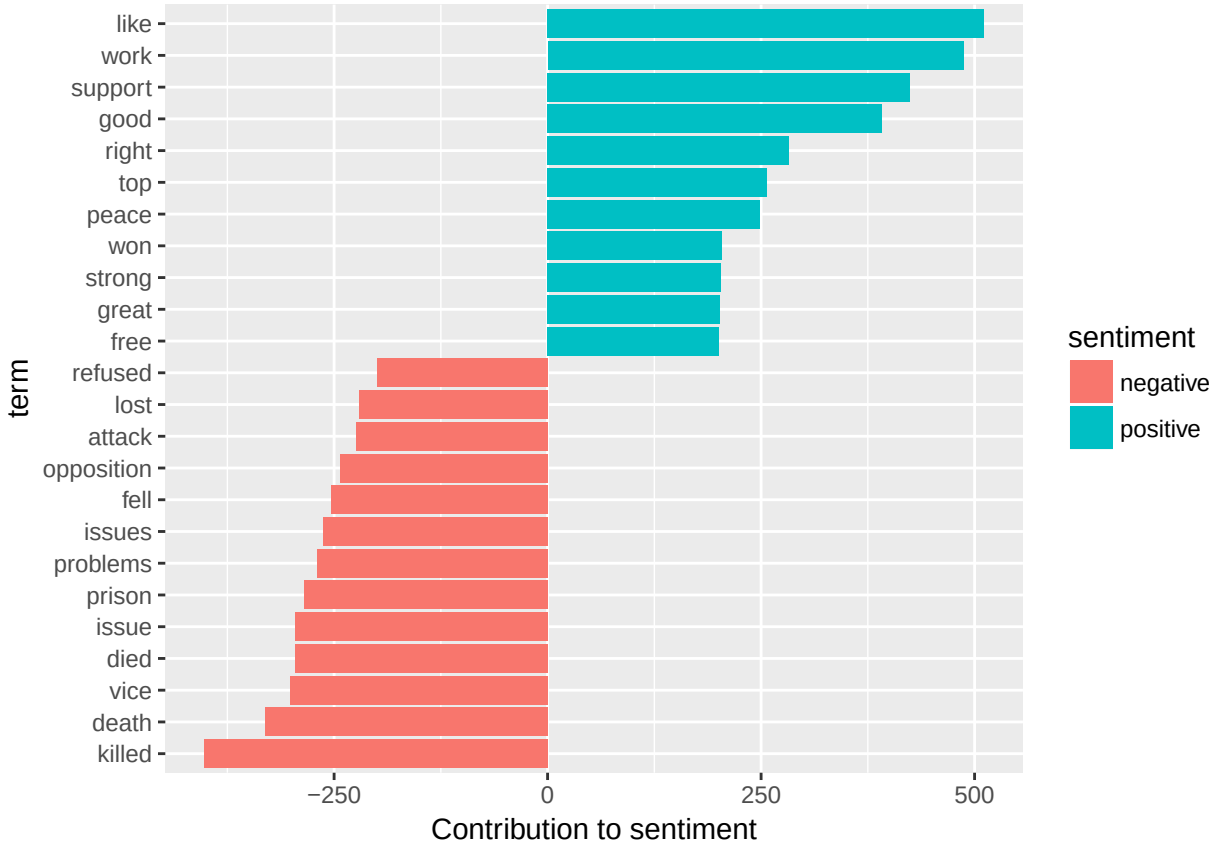
```
## # A tibble: 302,031 x 3
##    document term       count
##       <int> <chr>      <dbl>
##  1        1 adding      1.00
##  2        1 adult       2.00
```

```
##  3        1 ago         1.00
##  4        1 alcohol     1.00
##  5        1 allegedly   1.00
##  6        1 allen       1.00
##  7        1 apparently  2.00
##  8        1 appeared    1.00
##  9        1 arrested    1.00
## 10        1 assault     1.00
## # ... with 302,021 more rows
```

```
ap_sentiments <- ap_td %>% inner_join(get_sentiments("bing"), by = c(term = "word"))
ap_sentiments
```

```
## # A tibble: 30,094 x 4
##    document term    count sentiment
##       <int> <chr>   <dbl> <chr>
##  1        1 assault  1.00 negative
##  2        1 complex  1.00 negative
##  3        1 death    1.00 negative
##  4        1 died     1.00 negative
##  5        1 good     2.00 positive
##  6        1 illness  1.00 negative
##  7        1 killed   2.00 negative
##  8        1 like     2.00 positive
##  9        1 liked    1.00 positive
## 10        1 miracle  1.00 positive
## # ... with 30,084 more rows
```

```
ap_sentiments %>%
  count(sentiment, term, wt = count) %>%
  ungroup() %>%
  filter(n >= 200) %>%
  mutate(n = ifelse(sentiment == "negative", -n, n)) %>%
  mutate(term = reorder(term, n)) %>%
  ggplot(aes(term, n, fill = sentiment)) +
  geom_bar(stat = "identity") +
  ylab("Contribution to sentiment") +
  coord_flip()
```




```
data("data_corpus_inaugural", package = "quanteda")
inaug_dfm <- quanteda::dfm(data_corpus_inaugural, verbose = FALSE)
inaug_dfm
```

```
## Document-feature matrix of: 58 documents, 9,357 features (91.8% sparse).
```

```
inaug_td <- tidy(inaug_dfm)
inaug_td
```

```
## # A tibble: 44,709 x 3
##    document        term            count
##    <chr>           <chr>           <dbl>
##  1 1789-Washington fellow-citizens  1.00
##  2 1797-Adams      fellow-citizens  3.00
##  3 1801-Jefferson  fellow-citizens  2.00
##  4 1809-Madison    fellow-citizens  1.00
##  5 1813-Madison    fellow-citizens  1.00
##  6 1817-Monroe     fellow-citizens  5.00
##  7 1821-Monroe     fellow-citizens  1.00
##  8 1841-Harrison   fellow-citizens 11.0
##  9 1845-Polk       fellow-citizens  1.00
## 10 1849-Taylor     fellow-citizens  1.00
## # ... with 44,699 more rows
```

```
inaug_tf_idf <- inaug_td %>%
  bind_tf_idf(term, document, count) %>%
  arrange(desc(tf_idf))
```

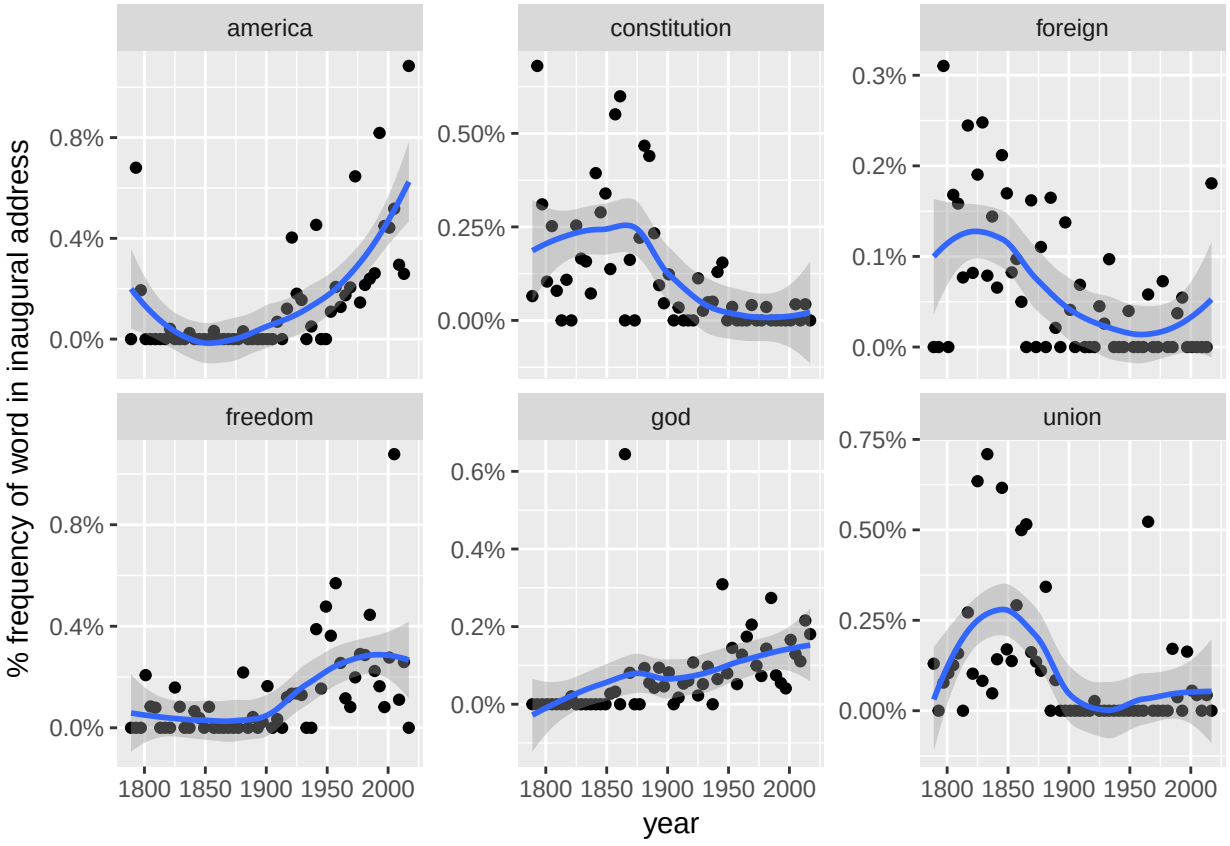
```
inaug_tf_idf
```

```
## # A tibble: 44,709 x 6
##    document        term         count      tf   idf tf_idf
##    <chr>           <chr>        <dbl>   <dbl> <dbl>  <dbl>
##  1 1793-Washington arrive        1.00 0.00680  4.06 0.0276
##  2 1793-Washington upbraidings   1.00 0.00680  4.06 0.0276
##  3 1793-Washington violated      1.00 0.00680  3.37 0.0229
##  4 1793-Washington willingly     1.00 0.00680  3.37 0.0229
##  5 1793-Washington incurring     1.00 0.00680  3.37 0.0229
##  6 1793-Washington previous      1.00 0.00680  2.96 0.0201
##  7 1793-Washington knowingly     1.00 0.00680  2.96 0.0201
##  8 1793-Washington injunctions   1.00 0.00680  2.96 0.0201
##  9 1793-Washington witnesses     1.00 0.00680  2.96 0.0201
## 10 1793-Washington besides       1.00 0.00680  2.67 0.0182
## # ... with 44,699 more rows
```

```
year_term_counts <- inaug_td %>%
  extract(document, "year", "(\\d+)", convert = TRUE) %>%
  complete(year, term, fill = list(count = 0)) %>%
  group_by(year) %>%
  mutate(year_total = sum(count))
```

```
year_term_counts %>%
  filter(term %in% c("god", "america", "foreign", "union", "constitution", "freedom")) %>%
  ggplot(aes(year, count / year_total)) +
  geom_point() +
  geom_smooth() +
  facet_wrap(~ term, scales = "free_y") +
  scale_y_continuous(labels = scales::percent_format()) +
  ylab("% frequency of word in inaugural address")
```

```
## `geom_smooth()` using method = 'loess'
```



Casting tidy text data into a matrix

```
ap_td %>%
  cast_dtm(document, term, count)
```

```
## <<DocumentTermMatrix (documents: 2246, terms: 10473)>>
## Non-/sparse entries: 302031/23220327
## Sparsity           : 99%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

```
ap_td %>%
  cast_dfm(document, term, count)
```

```
## Document-feature matrix of: 2,246 documents, 10,473 features (98.7% sparse).
```

```
library(Matrix)
```

```
##
## Attaching package: 'Matrix'
```

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

```
# cast into a Matrix object
m <- ap_td %>%
  cast_sparse(document, term, count)
```

```
class(m)
```

```
## [1] "dgCMatrix"
## attr(,"package")
## [1] "Matrix"
```

```
dim(m)
```

```
## [1]  2246 10473
```

```
# create a dtm of Jane Austen's books
austen_dtm <- austen_books() %>%
  unnest_tokens(word, text) %>%
  count(book, word) %>%
  cast_dtm(book, word, n)

austen_dtm
```

```
## <<DocumentTermMatrix (documents: 6, terms: 14520)>>
## Non-/sparse entries: 40379/46741
## Sparsity           : 54%
## Maximal term length: 19
## Weighting          : term frequency (tf)
```

Tidying corpus objects with metadata

```
data("acq")
acq
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 50
```

```
# first document
acq[[1]]
```

```
## <<PlainTextDocument>>
## Metadata:  15
## Content:  chars: 1287
```

```
acq_td <- tidy(acq)
acq_td
```

```
## # A tibble: 50 x 16
##    author  datetimestamp       description heading   id    language origin
##    <chr>   <dttm>              <chr>       <chr>     <chr> <chr>    <chr>
##  1 <NA>    1987-02-26 10:18:06 ""          COMPUTER~ 10    en       Reute~
##  2 <NA>    1987-02-26 10:19:15 ""          OHIO MAT~ 12    en       Reute~
##  3 <NA>    1987-02-26 10:49:56 ""          MCLEAN'S~ 44    en       Reute~
##  4 By Cal~ 1987-02-26 10:51:17 ""          CHEMLAWN~ 45    en       Reute~
##  5 <NA>    1987-02-26 11:08:33 ""          <COFAB I~ 68    en       Reute~
##  6 <NA>    1987-02-26 11:32:37 ""          INVESTME~ 96    en       Reute~
##  7 By Pat~ 1987-02-26 11:43:13 ""          AMERICAN~ 110   en       Reute~
##  8 <NA>    1987-02-26 11:59:25 ""          HONG KON~ 125   en       Reute~
##  9 <NA>    1987-02-26 12:01:28 ""          LIEBERT ~ 128   en       Reute~
## 10 <NA>    1987-02-26 12:08:27 ""          GULF APP~ 134   en       Reute~
## # ... with 40 more rows, and 9 more variables: topics <chr>,
```

```
## #   lewissplit <chr>, cgisplit <chr>, oldid <chr>, places <list>,
## #   people <lgl>, orgs <lgl>, exchanges <lgl>, text <chr>
```

```
acq_tokens <- acq_td %>%
  select(-places) %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words, by = "word")

# most common words
acq_tokens %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 1,566 x 2
##    word         n
##    <chr>    <int>
##  1 dlrs       100
##  2 pct         70
##  3 mln         65
##  4 company     63
##  5 shares      52
##  6 reuter      50
##  7 stock       46
##  8 offer       34
##  9 share       34
## 10 american    28
## # ... with 1,556 more rows
```

```
# tf-idf
acq_tokens %>%
  count(id, word) %>%
  bind_tf_idf(word, id, n) %>%
  arrange(desc(tf_idf))
```

```
## # A tibble: 2,853 x 6
##    id    word         n     tf   idf tf_idf
##    <chr> <chr>    <int>  <dbl> <dbl>  <dbl>
##  1 186   groupe       2 0.133   3.91  0.522
##  2 128   liebert      3 0.130   3.91  0.510
##  3 474   esselte      5 0.109   3.91  0.425
##  4 371   burdett      6 0.103   3.91  0.405
##  5 442   hazleton     4 0.103   3.91  0.401
##  6 199   circuit      5 0.102   3.91  0.399
##  7 162   suffield     2 0.100   3.91  0.391
##  8 498   west         3 0.100   3.91  0.391
##  9 441   rmj          8 0.121   3.22  0.390
## 10 467   nursery      3 0.0968  3.91  0.379
## # ... with 2,843 more rows
```

```
pacman::p_load(tm.plugin.webmining, purrr, rJava)
library(tm.plugin.webmining)
library(purrr)

company <- c("Microsoft", "Apple", "Google", "Amazon", "Facebook",
             "Twitter", "IBM", "Yahoo", "Netflix")
symbol <- c("MSFT", "AAPL", "GOOG", "AMZN", "FB", "TWTR", "IBM", "YHOO", "NFLX")
```

```
download_articles <- function(symbol) {
  WebCorpus(GoogleFinanceSource(paste0("NASDAQ:", symbol)))
}

# this doesnt work
#stock_articles <- data_frame(company = company,
#                             symbol = symbol) %>%
#  mutate(corpus = map(symbol, download_articles))
# stock_articles
```

Topic modeling

Latent Dirichlet allocation

```
library(topicmodels)
```

```
## Warning: package 'topicmodels' was built under R version 3.4.4
```

```
data("AssociatedPress")
AssociatedPress
```

```
## <<DocumentTermMatrix (documents: 2246, terms: 10473)>>
## Non-/sparse entries: 302031/23220327
## Sparsity           : 99%
## Maximal term length: 18
## Weighting          : term frequency (tf)
```

```
ap_lda <- LDA(AssociatedPress, k = 2, control = list(seed = 1234))
ap_lda
```

```
## A LDA_VEM topic model with 2 topics.
```

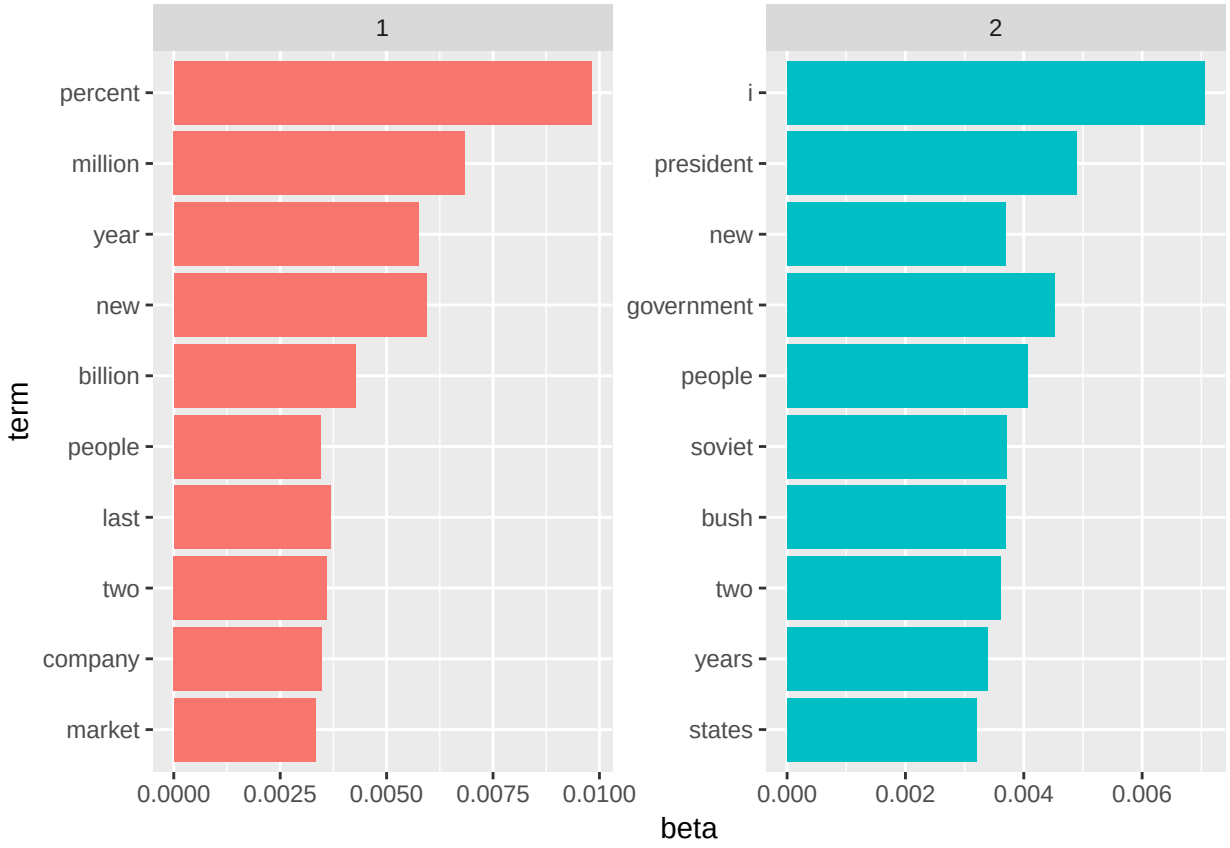
```
library(tidytext)

ap_topics <- tidy(ap_lda, matrix = "beta")
ap_topics
```

```
## # A tibble: 20,946 x 3
##    topic term                                          beta
##    <int> <chr>                                        <dbl>
##  1     1 aaron      0.00000000000169
##  2     2 aaron      0.0000390
##  3     1 abandon    0.0000265
##  4     2 abandon    0.0000399
##  5     1 abandoned  0.000139
##  6     2 abandoned  0.0000588
##  7     1 abandoning 0.00000000000000000000000000000000245
##  8     2 abandoning 0.0000234
##  9     1 abbott     0.00000213
## 10     2 abbott     0.0000297
## # ... with 20,936 more rows
```

```
ap_top_terms <- ap_topics %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
```

```
ap_top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```




```
beta_spread <- ap_topics %>%
  mutate(topic = paste0("topic", topic)) %>%
  spread(topic, beta) %>%
  filter(topic1 > .001 | topic2 > .001) %>%
  mutate(log_ratio = log2(topic2 / topic1))

beta_spread
```

```
## # A tibble: 198 x 4
##    term              topic1      topic2 log_ratio
##    <chr>              <dbl>       <dbl>     <dbl>
##  1 administration 0.000431  0.00138         1.68
##  2 ago            0.00107   0.000842      - 0.339
##  3 agreement      0.000671  0.00104         0.630
##  4 aid            0.0000476 0.00105         4.46
##  5 air            0.00214   0.000297      - 2.85
##  6 american       0.00203   0.00168       - 0.270
##  7 analysts       0.00109   0.000000578   -10.9
##  8 area           0.00137   0.000231      - 2.57
```

```
##  9 army           0.000262  0.00105         2.00
## 10 asked          0.000189  0.00156         3.05
## # ... with 188 more rows
```

```
ap_documents <- tidy(ap_lda, matrix = "gamma")
ap_documents
```

```
## # A tibble: 4,492 x 3
##    document topic    gamma
##       <int> <int>    <dbl>
##  1        1     1 0.248
##  2        2     1 0.362
##  3        3     1 0.527
##  4        4     1 0.357
##  5        5     1 0.181
##  6        6     1 0.000588
##  7        7     1 0.773
##  8        8     1 0.00445
##  9        9     1 0.967
## 10       10     1 0.147
## # ... with 4,482 more rows
```

```
tidy(AssociatedPress) %>%
  filter(document == 6) %>%
  arrange(desc(count))
```

```
## # A tibble: 287 x 3
##    document term           count
##       <int> <chr>          <dbl>
##  1        6 noriega        16.0
##  2        6 panama         12.0
##  3        6 jackson         6.00
##  4        6 powell          6.00
##  5        6 administration  5.00
##  6        6 economic        5.00
##  7        6 general         5.00
##  8        6 i               5.00
##  9        6 panamanian      5.00
## 10        6 american        4.00
## # ... with 277 more rows
```

Example: the great library heist

```
titles <- c("Twenty Thousand Leagues under the Sea", "The War of the Worlds",
            "Pride and Prejudice", "Great Expectations")
books <- gutenberg_works(title %in% titles) %>%
  gutenberg_download(meta_fields = "title")

# divide into documents, each representing one chapter
by_chapter <- books %>%
  group_by(title) %>%
  mutate(chapter = cumsum(str_detect(text, regex("^chapter ", ignore_case = TRUE)))) %>%
  ungroup() %>%
  filter(chapter > 0) %>%
  unite(document, title, chapter)
```

```r
# split into words
by_chapter_word <- by_chapter %>%
  unnest_tokens(word, text)

# find document-word counts
word_counts <- by_chapter_word %>%
  anti_join(stop_words) %>%
  count(document, word, sort = TRUE) %>%
  ungroup()
```

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

```r
word_counts
```

```
## # A tibble: 104,722 x 3
##    document                 word        n
##    <chr>                    <chr>   <int>
##  1 Great Expectations_57    joe        88
##  2 Great Expectations_7     joe        70
##  3 Great Expectations_17    biddy      63
##  4 Great Expectations_27    joe        58
##  5 Great Expectations_38    estella    58
##  6 Great Expectations_2     joe        56
##  7 Great Expectations_23    pocket     53
##  8 Great Expectations_15    joe        50
##  9 Great Expectations_18    joe        50
## 10 The War of the Worlds_16 brother    50
## # ... with 104,712 more rows
```

```r
chapters_dtm <- word_counts %>%
  cast_dtm(document, word, n)

chapters_dtm
```

```
## <<DocumentTermMatrix (documents: 193, terms: 18215)>>
## Non-/sparse entries: 104722/3410773
## Sparsity           : 97%
## Maximal term length: 19
## Weighting          : term frequency (tf)
```

```r
chapters_lda <- LDA(chapters_dtm, k = 4, control = list(seed = 1234))
chapters_lda
```

```
## A LDA_VEM topic model with 4 topics.
```

```r
chapter_topics <- tidy(chapters_lda, matrix = "beta")
chapter_topics
```

```
## # A tibble: 72,860 x 3
##    topic term                                                    beta
##    <int> <chr>                                                  <dbl>
##  1     1 joe                                                 1.44e-17
##  2     2 joe                                                 5.96e-61
##  3     3 joe                                                 9.88e-25
##  4     4 joe                                                 1.45e- 2
##  5     1 biddy                                               5.14e-28
##  6     2 biddy                                               5.02e-73
```

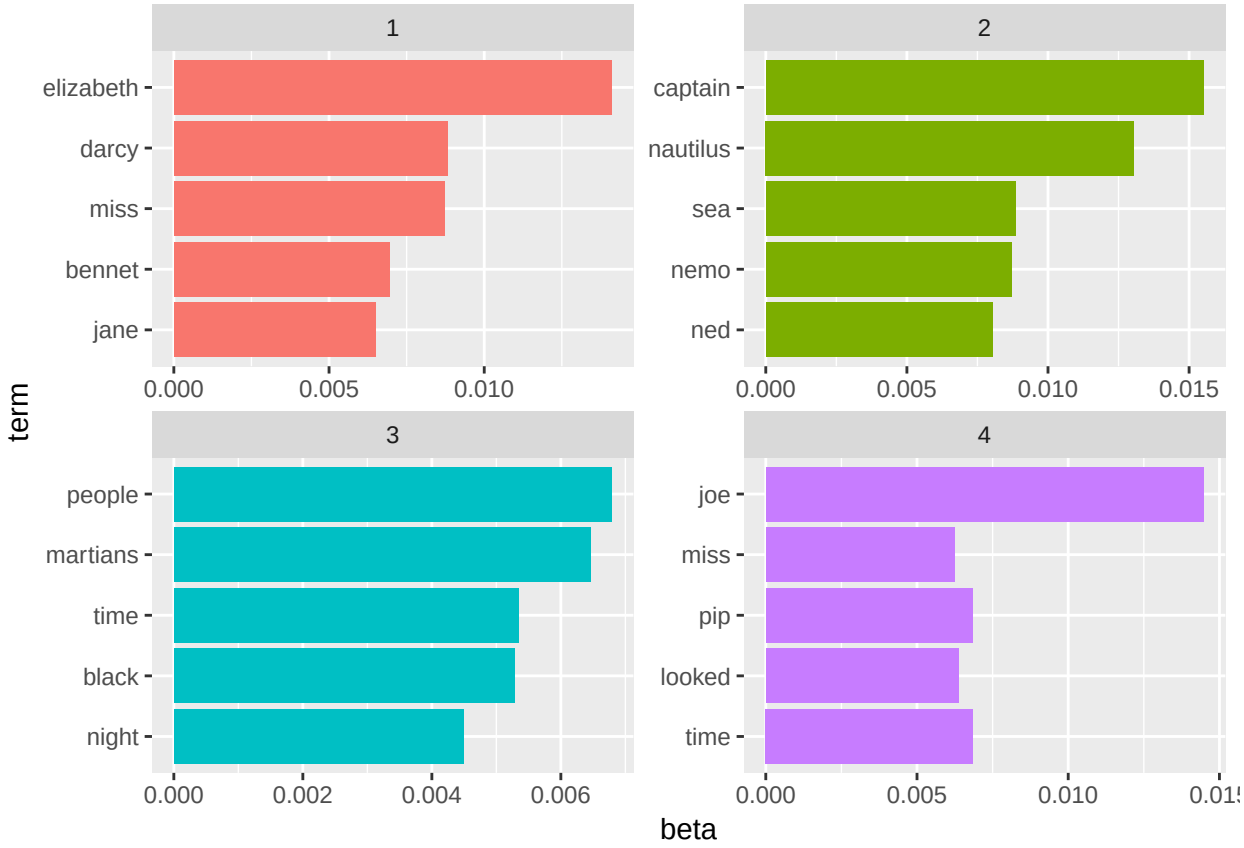
```
##  7     3 biddy                                                  4.31e-48
##  8     4 biddy                                                  4.78e- 3
##  9     1 estella                                                2.43e- 6
## 10     2 estella                                                4.32e-68
## # ... with 72,850 more rows
```

```
top_terms <- chapter_topics %>%
  group_by(topic) %>%
  top_n(5, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
```

```
top_terms
```

```
## # A tibble: 20 x 3
##    topic term         beta
##    <int> <chr>       <dbl>
##  1     1 elizabeth 0.0141
##  2     1 darcy     0.00881
##  3     1 miss      0.00871
##  4     1 bennet    0.00694
##  5     1 jane      0.00649
##  6     2 captain   0.0155
##  7     2 nautilus  0.0131
##  8     2 sea       0.00884
##  9     2 nemo      0.00871
## 10     2 ned       0.00803
## 11     3 people    0.00679
## 12     3 martians  0.00646
## 13     3 time      0.00534
## 14     3 black     0.00528
## 15     3 night     0.00449
## 16     4 joe       0.0145
## 17     4 time      0.00685
## 18     4 pip       0.00683
## 19     4 looked    0.00637
## 20     4 miss      0.00623
```

```
top_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```

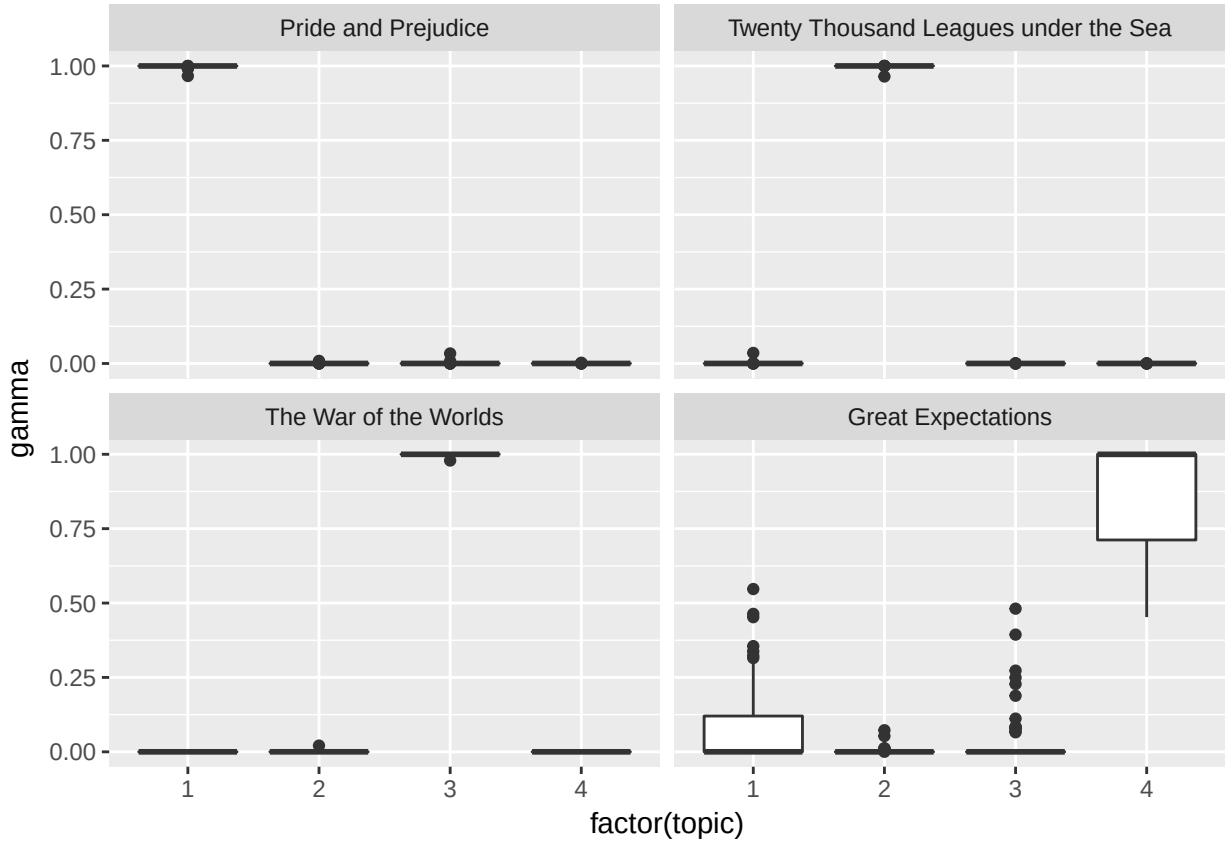



```
chapters_gamma <- tidy(chapters_lda, matrix = "gamma")
chapters_gamma <- chapters_gamma %>%
  separate(document, c("title", "chapter"), sep = "_", convert = TRUE)

chapters_gamma
```

```
## # A tibble: 772 x 4
##    title                 chapter topic     gamma
##    <chr>                   <int> <int>     <dbl>
##  1 Great Expectations         57     1 0.0000134
##  2 Great Expectations          7     1 0.0000146
##  3 Great Expectations         17     1 0.0000210
##  4 Great Expectations         27     1 0.0000190
##  5 Great Expectations         38     1 0.355
##  6 Great Expectations          2     1 0.0000171
##  7 Great Expectations         23     1 0.547
##  8 Great Expectations         15     1 0.0124
##  9 Great Expectations         18     1 0.0000126
## 10 The War of the Worlds      16     1 0.0000107
## # ... with 762 more rows
```

```
chapters_gamma %>%
  mutate(title = reorder(title, gamma * topic)) %>%
  ggplot(aes(factor(topic), gamma)) +
  geom_boxplot() +
  facet_wrap(~ title)
```




```
chapter_classifications <- chapters_gamma %>%
  group_by(title, chapter) %>%
  top_n(1, gamma) %>%
  ungroup()

chapter_classifications
```

```
## # A tibble: 193 x 4
##    title               chapter topic gamma
##    <chr>                 <int> <int> <dbl>
##  1 Great Expectations       23     1 0.547
##  2 Pride and Prejudice      43     1 1.000
##  3 Pride and Prejudice      18     1 1.000
##  4 Pride and Prejudice      45     1 1.000
##  5 Pride and Prejudice      16     1 1.000
##  6 Pride and Prejudice      29     1 1.000
##  7 Pride and Prejudice      10     1 1.000
##  8 Pride and Prejudice       8     1 1.000
##  9 Pride and Prejudice      56     1 1.000
## 10 Pride and Prejudice      47     1 1.000
## # ... with 183 more rows
```

```
book_topics <- chapter_classifications %>%
  count(title, topic) %>%
  group_by(title) %>%
  top_n(1, n) %>%
  ungroup() %>%
```

```
  transmute(consensus = title, topic)

chapter_classifications %>%
  inner_join(book_topics, by = "topic") %>%
  filter(title != consensus)
```

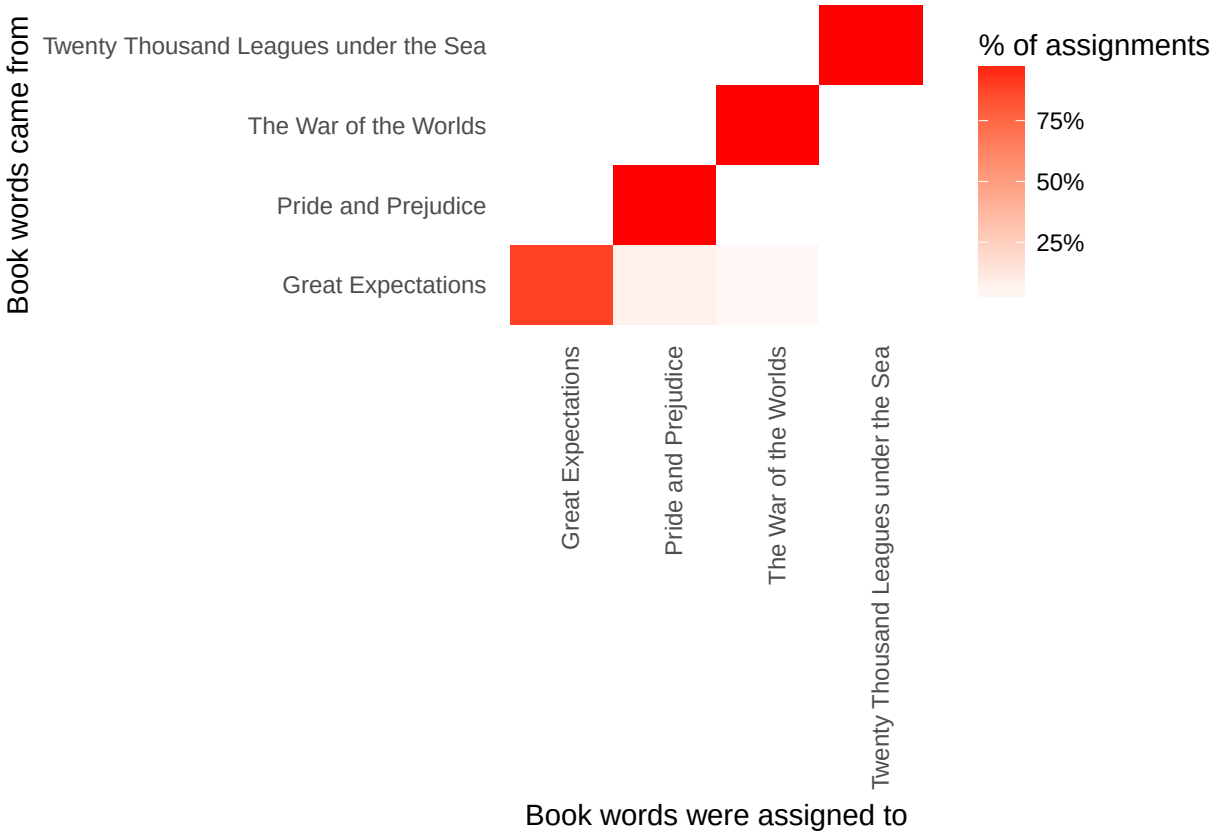
```
## # A tibble: 2 x 5
##   title              chapter topic gamma consensus
##   <chr>                <int> <int> <dbl> <chr>
## 1 Great Expectations      23     1 0.547 Pride and Prejudice
## 2 Great Expectations      54     3 0.481 The War of the Worlds
```

```
assignments <- augment(chapters_lda, data = chapters_dtm)
assignments <- assignments %>%
  separate(document, c("title", "chapter"), sep = "_", convert = TRUE) %>%
  inner_join(book_topics, by = c(".topic" = "topic"))

assignments
```

```
## # A tibble: 104,722 x 6
##    title              chapter term  count .topic consensus
##    <chr>                <int> <chr> <dbl>  <dbl> <chr>
##  1 Great Expectations      57 joe   88.0    4.00 Great Expectations
##  2 Great Expectations       7 joe   70.0    4.00 Great Expectations
##  3 Great Expectations      17 joe    5.00   4.00 Great Expectations
##  4 Great Expectations      27 joe   58.0    4.00 Great Expectations
##  5 Great Expectations       2 joe   56.0    4.00 Great Expectations
##  6 Great Expectations      23 joe    1.00   4.00 Great Expectations
##  7 Great Expectations      15 joe   50.0    4.00 Great Expectations
##  8 Great Expectations      18 joe   50.0    4.00 Great Expectations
##  9 Great Expectations       9 joe   44.0    4.00 Great Expectations
## 10 Great Expectations      13 joe   40.0    4.00 Great Expectations
## # ... with 104,712 more rows
```

```
assignments %>%
  count(title, consensus, wt = count) %>%
  group_by(title) %>%
  mutate(percent = n / sum(n)) %>%
  ggplot(aes(consensus, title, fill = percent)) +
  geom_tile() +
  scale_fill_gradient2(high = "red", label = percent_format()) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1),
        panel.grid = element_blank()) +
  labs(x = "Book words were assigned to",
       y = "Book words came from",
       fill = "% of assignments")
```




```
wrong_words <- assignments %>%
  filter(title != consensus)

wrong_words %>%
  count(title, consensus, term, wt = count) %>%
  ungroup() %>%
  arrange(desc(n))
```

```
## # A tibble: 3,551 x 4
##    title              consensus              term         n
##    <chr>              <chr>                  <chr>    <dbl>
##  1 Great Expectations Pride and Prejudice    love      44.0
##  2 Great Expectations Pride and Prejudice    sergeant  37.0
##  3 Great Expectations Pride and Prejudice    lady      32.0
##  4 Great Expectations Pride and Prejudice    miss      26.0
##  5 Great Expectations The War of the Worlds  boat      25.0
##  6 Great Expectations The War of the Worlds  tide      20.0
##  7 Great Expectations The War of the Worlds  water     20.0
##  8 Great Expectations Pride and Prejudice    father    19.0
##  9 Great Expectations Pride and Prejudice    baby      18.0
## 10 Great Expectations Pride and Prejudice    flopson   18.0
## # ... with 3,541 more rows
```

```
word_counts %>%
  filter(word == "flopson")
```

```
## # A tibble: 3 x 3
##   document             word        n
```

```
##   <chr>                 <chr>   <int>
## 1 Great Expectations_22 flopson    10
## 2 Great Expectations_23 flopson     7
## 3 Great Expectations_33 flopson     1
```

Alternative LDA implementations

```
library(mallet)
```

```
## Warning: package 'mallet' was built under R version 3.4.3
```

```
# create a vector with one string per chapter
collapsed <- by_chapter_word %>%
  anti_join(stop_words, by = "word") %>%
  mutate(word = str_replace(word, "'", "")) %>%
  group_by(document) %>%
  summarize(text = paste(word, collapse = " "))

# create an empty file of "stopwords"
file.create(empty_file <- tempfile())
```

```
## [1] TRUE
```

```
docs <- mallet.import(collapsed$document, collapsed$text, empty_file)

mallet_model <- MalletLDA(num.topics = 4)
mallet_model$loadDocuments(docs)
mallet_model$train(100)

# word-topic pairs
tidy(mallet_model)
```

```
## # A tibble: 71,064 x 3
##    topic term           beta
##    <int> <chr>         <dbl>
##  1     1 chapter 0.000945
##  2     2 chapter 0.000000270
##  3     3 chapter 0.00146
##  4     4 chapter 0.00271
##  5     1 fathers 0.000000230
##  6     2 fathers 0.000000270
##  7     3 fathers 0.000939
##  8     4 fathers 0.000000259
##  9     1 family  0.000000230
## 10     2 family  0.000000270
## # ... with 71,054 more rows
```

```
# document-topic pairs
tidy(mallet_model, matrix = "gamma")
```

```
## # A tibble: 772 x 3
##    document              topic  gamma
##    <chr>                 <int>  <dbl>
##  1 Great Expectations_1      1 0.116
##  2 Great Expectations_10     1 0.0459
##  3 Great Expectations_11     1 0.0689
##  4 Great Expectations_12     1 0.0885
```

```
##  5 Great Expectations_13     1 0.0385
##  6 Great Expectations_14     1 0.0871
##  7 Great Expectations_15     1 0.0942
##  8 Great Expectations_16     1 0.109
##  9 Great Expectations_17     1 0.0317
## 10 Great Expectations_18     1 0.0245
## # ... with 762 more rows
```

```
# column needs to be named "term" for "augment"
term_counts <- rename(word_counts, term = word)
augment(mallet_model, term_counts)
```

```
## # A tibble: 104,722 x 4
##    document                term        n .topic
##    <chr>                   <chr>   <int>  <int>
##  1 Great Expectations_57   joe        88      2
##  2 Great Expectations_7    joe        70      2
##  3 Great Expectations_17   biddy      63      2
##  4 Great Expectations_27   joe        58      2
##  5 Great Expectations_38   estella    58      2
##  6 Great Expectations_2    joe        56      2
##  7 Great Expectations_23   pocket     53      2
##  8 Great Expectations_15   joe        50      2
##  9 Great Expectations_18   joe        50      2
## 10 The War of the Worlds_16 brother    50      4
## # ... with 104,712 more rows
```