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"
```

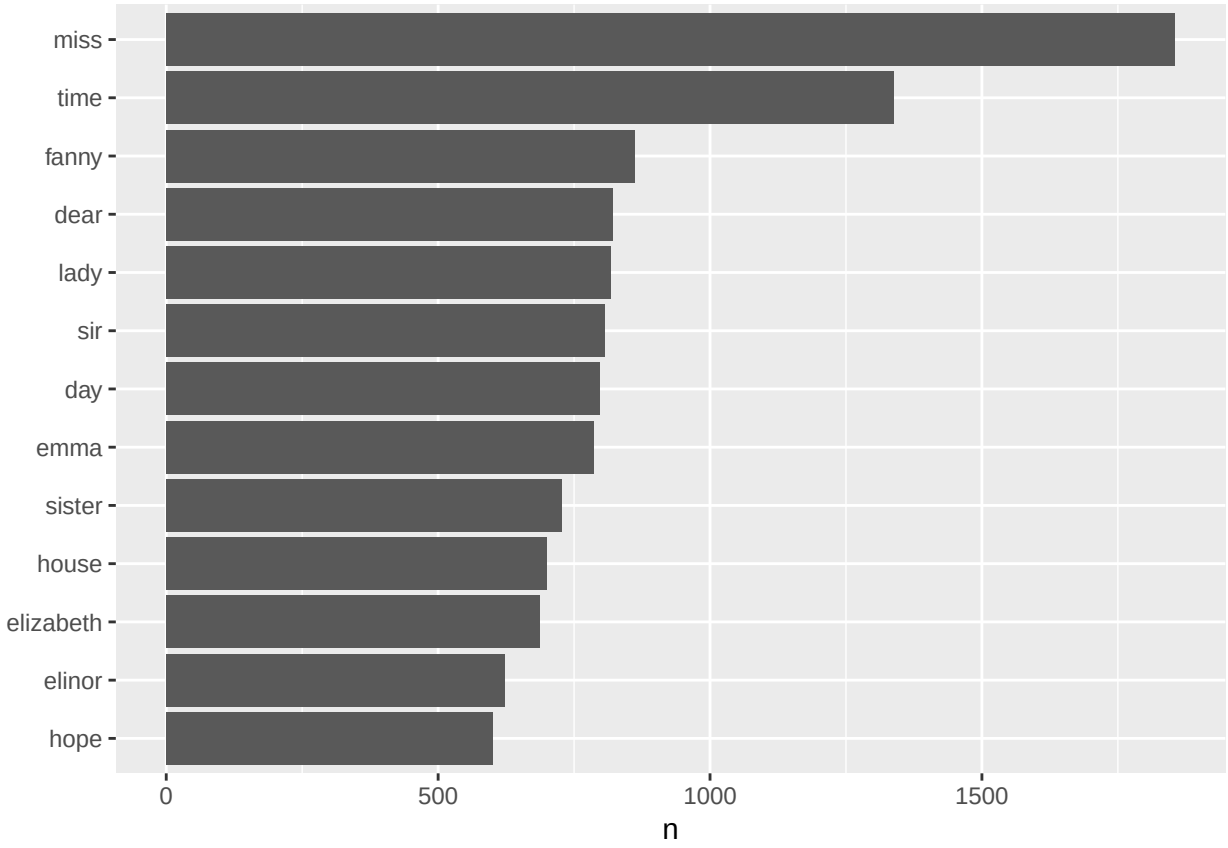
```
## # A tibble: 4 x 2
##    line text
##   <int> <chr>
## 1     1 Because I could not stop for Death -
## 2     2 He kindly stopped for me -
## 3     3 The Carriage held but just Ourselves -
## 4     4 and Immortality
```

```
## # 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)
```

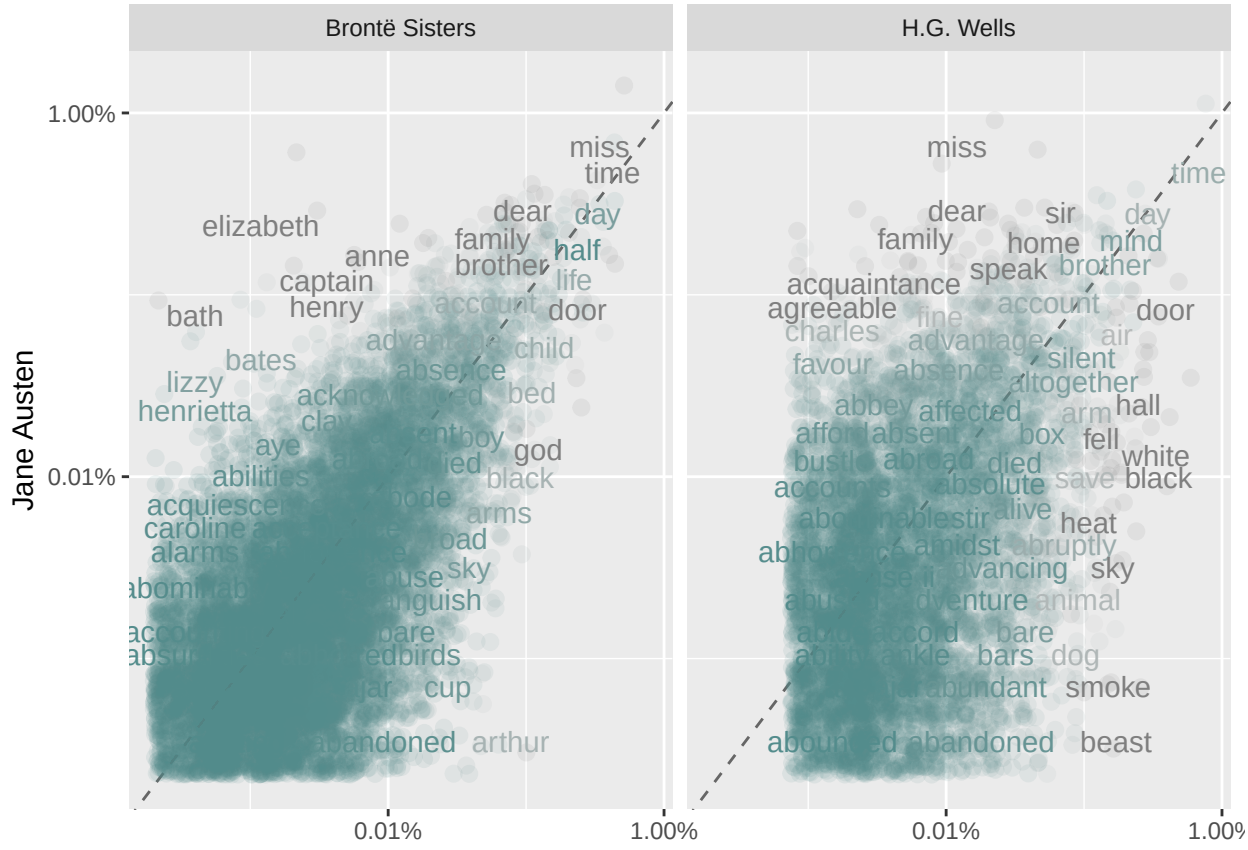
```
## # A tibble: 23,050 x 2
##    word       n
##    <chr>  <int>
##  1 time    1065
##  2 miss     855
##  3 day      827
##  4 hand     768
##  5 eyes     713
##  6 night    647
##  7 heart    638
##  8 looked   601
##  9 door     592
## 10 half     586
## # ... with 23,040 more rows
```

```
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)
```

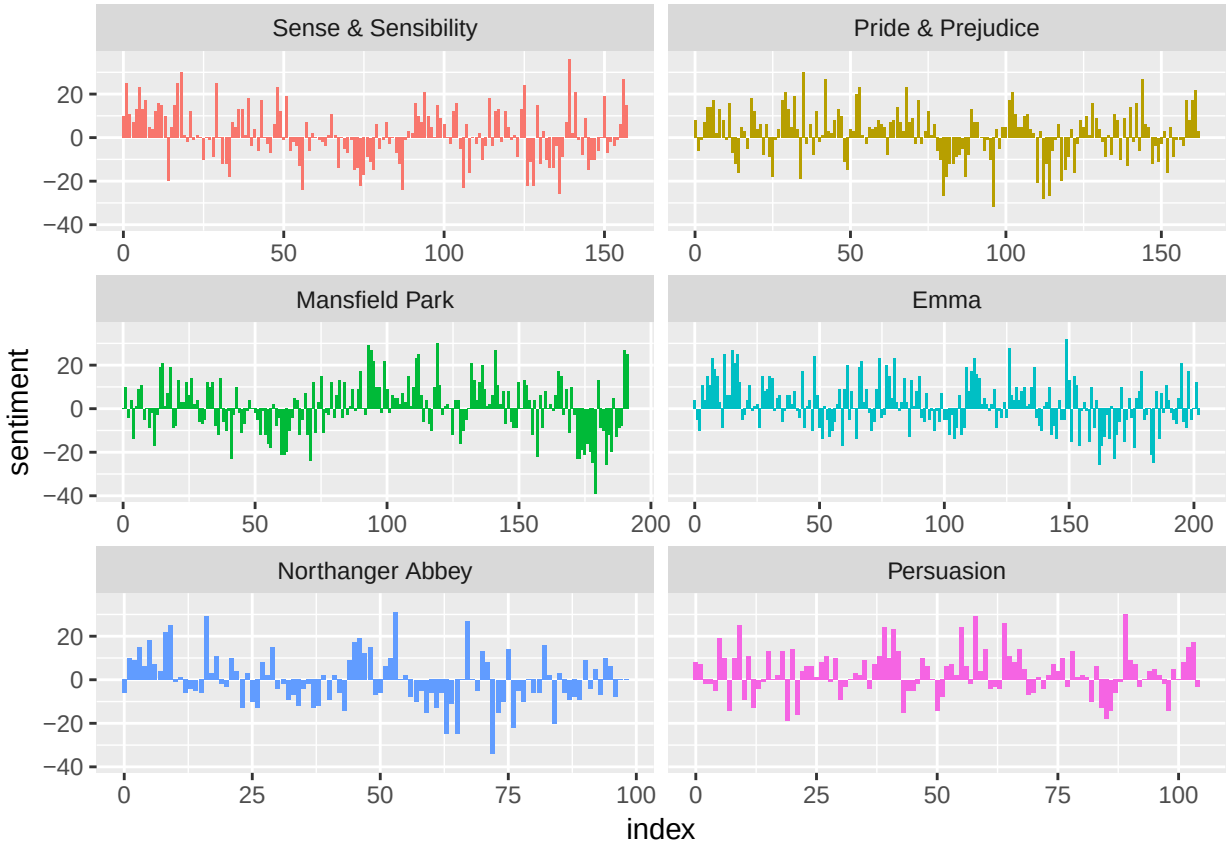
```
## Joining, by = "word"
```

```
## # 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"
```

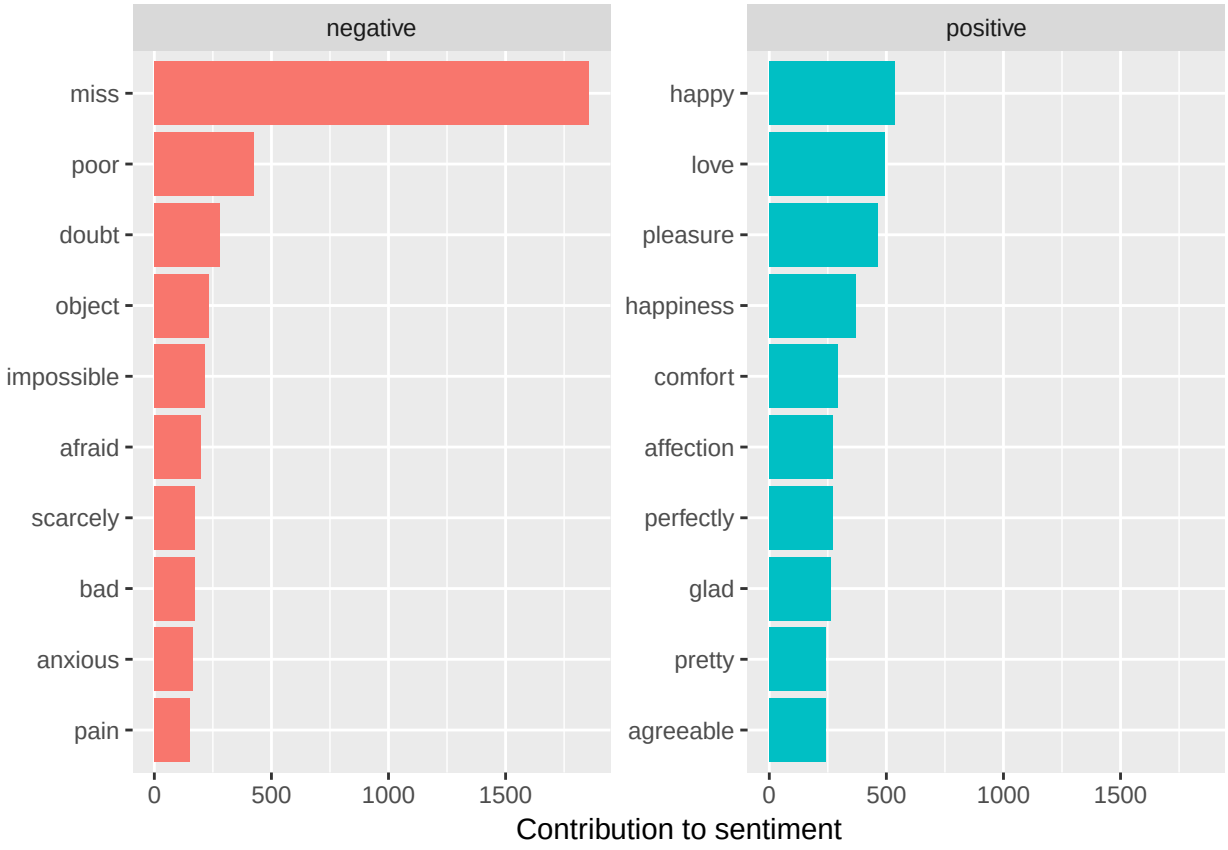
```
bing_word_counts
```

```
## # 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): lady could not be fit on
## page. It will not be plotted.
```

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

```
## Warning in wordcloud(word, n, max.words = 100): woodhouse 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"
```

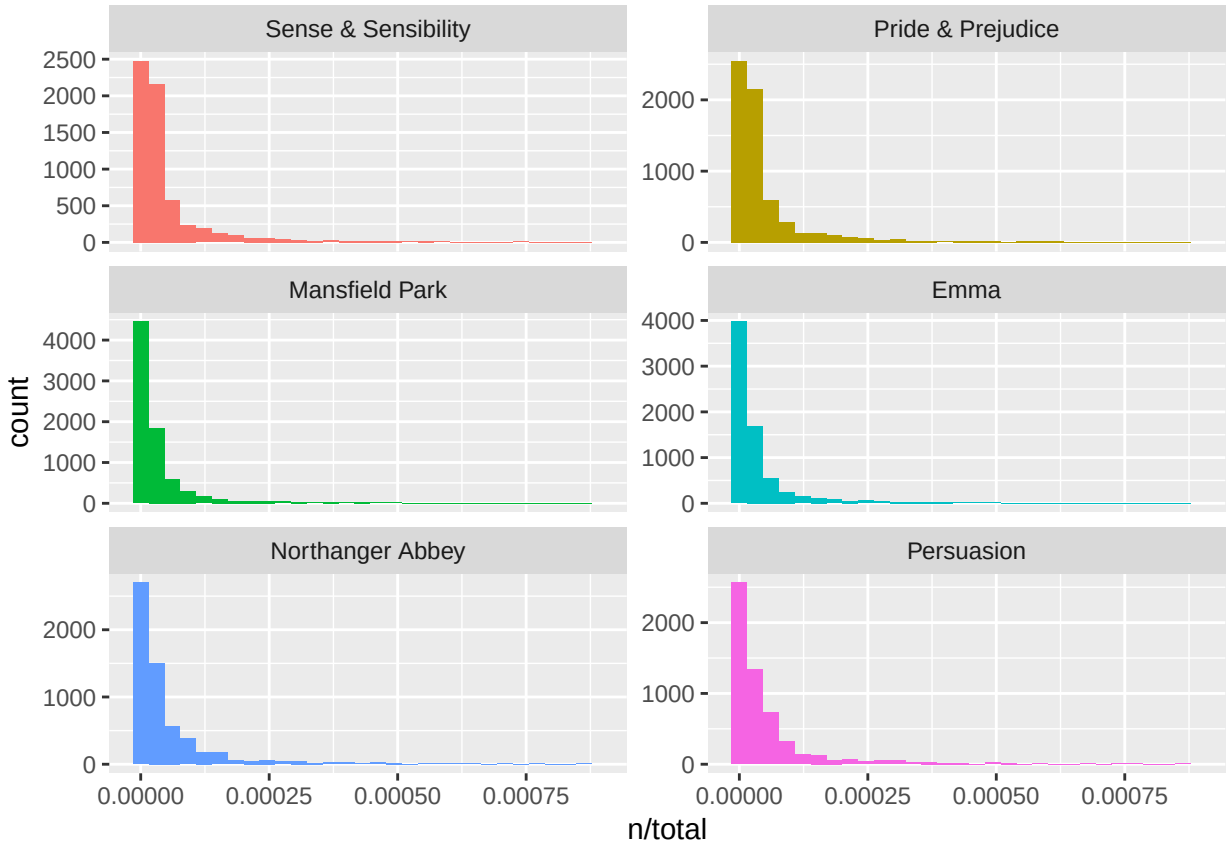
```
head(book_words)
```

```
## # 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).
```



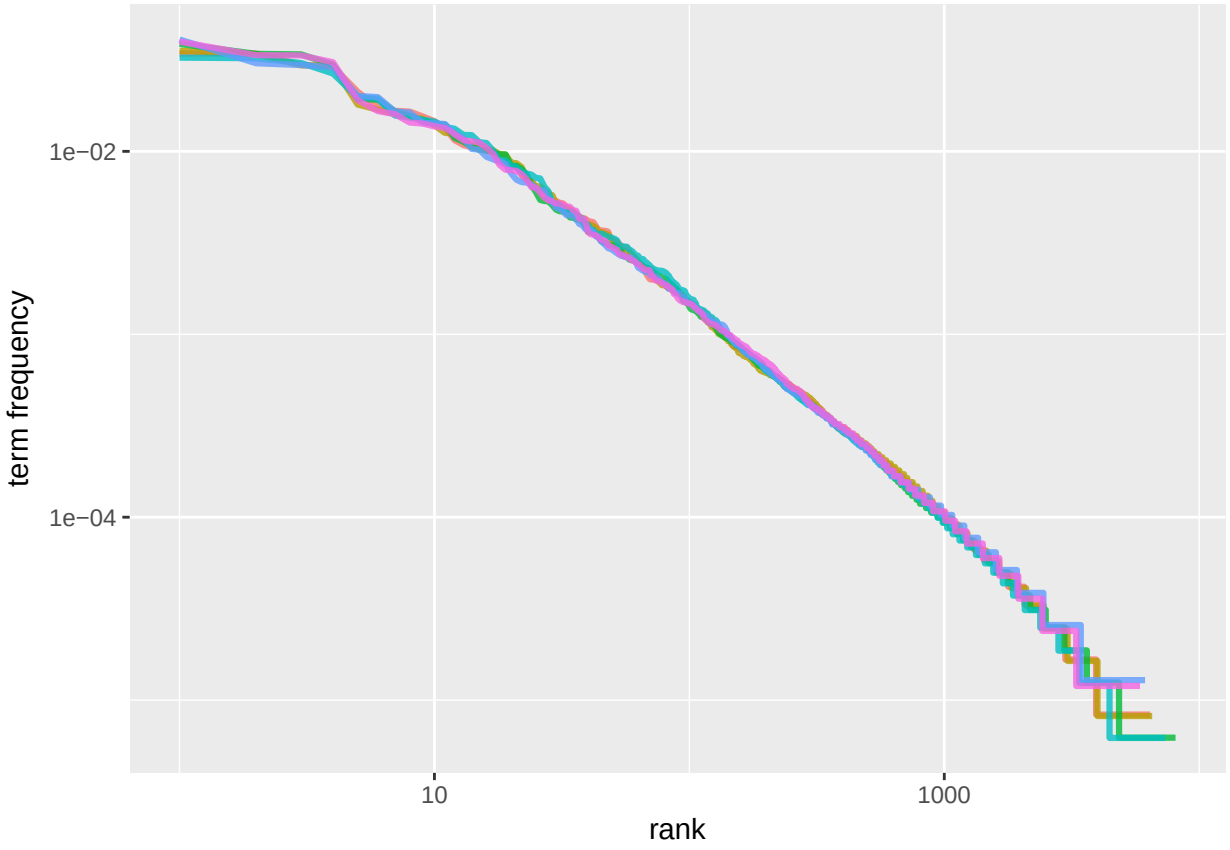
Zipf's Law

```
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
```

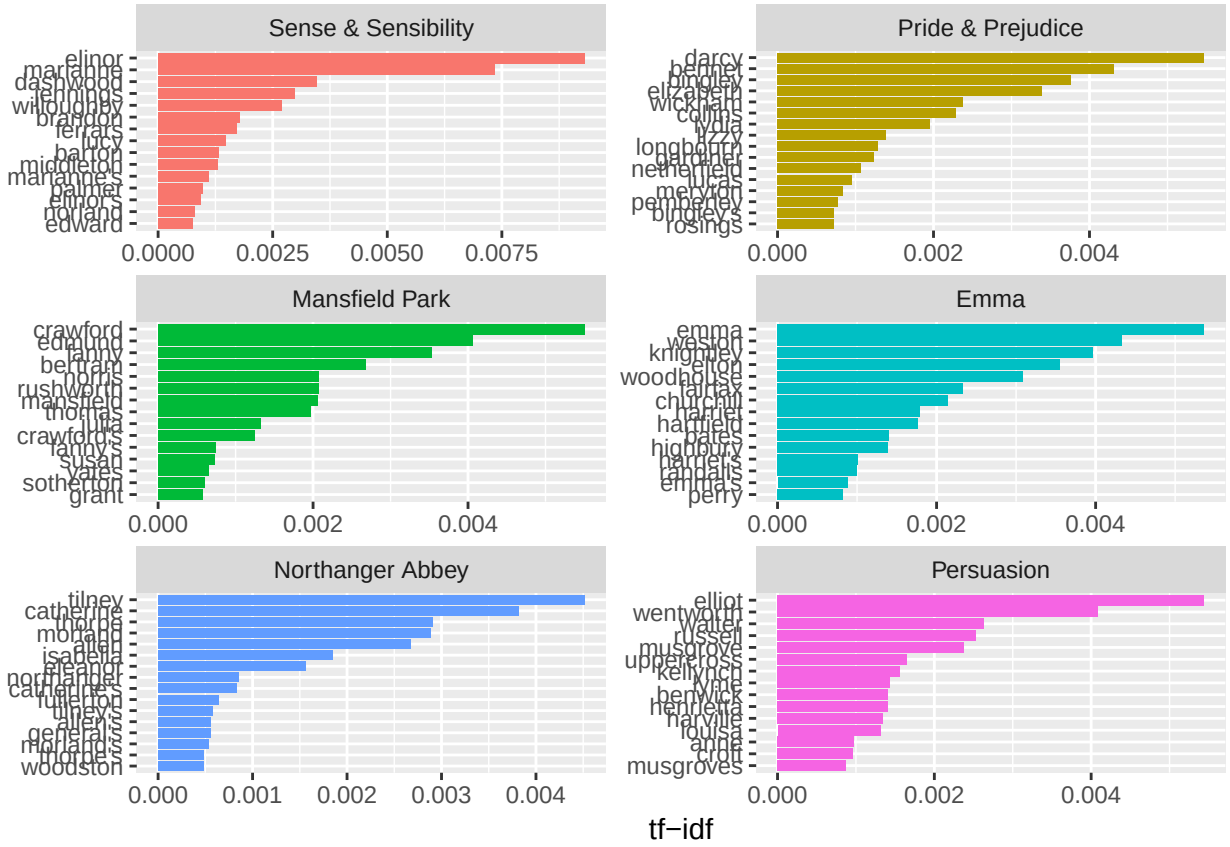
```
book_words %>% select(-total) %>% arrange(desc(tf_idf))
```

```
## # 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()
```

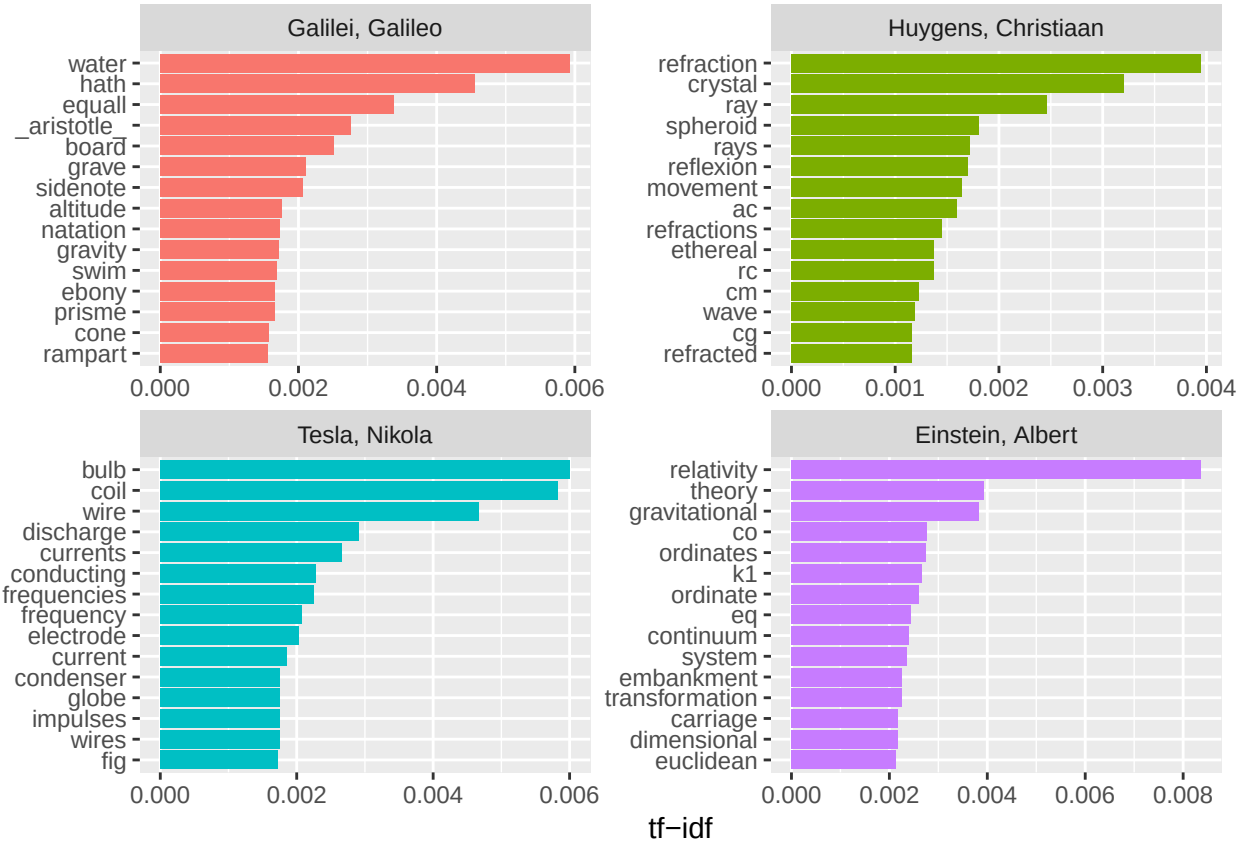
```
head(physics_words)
```

```
## # 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
```

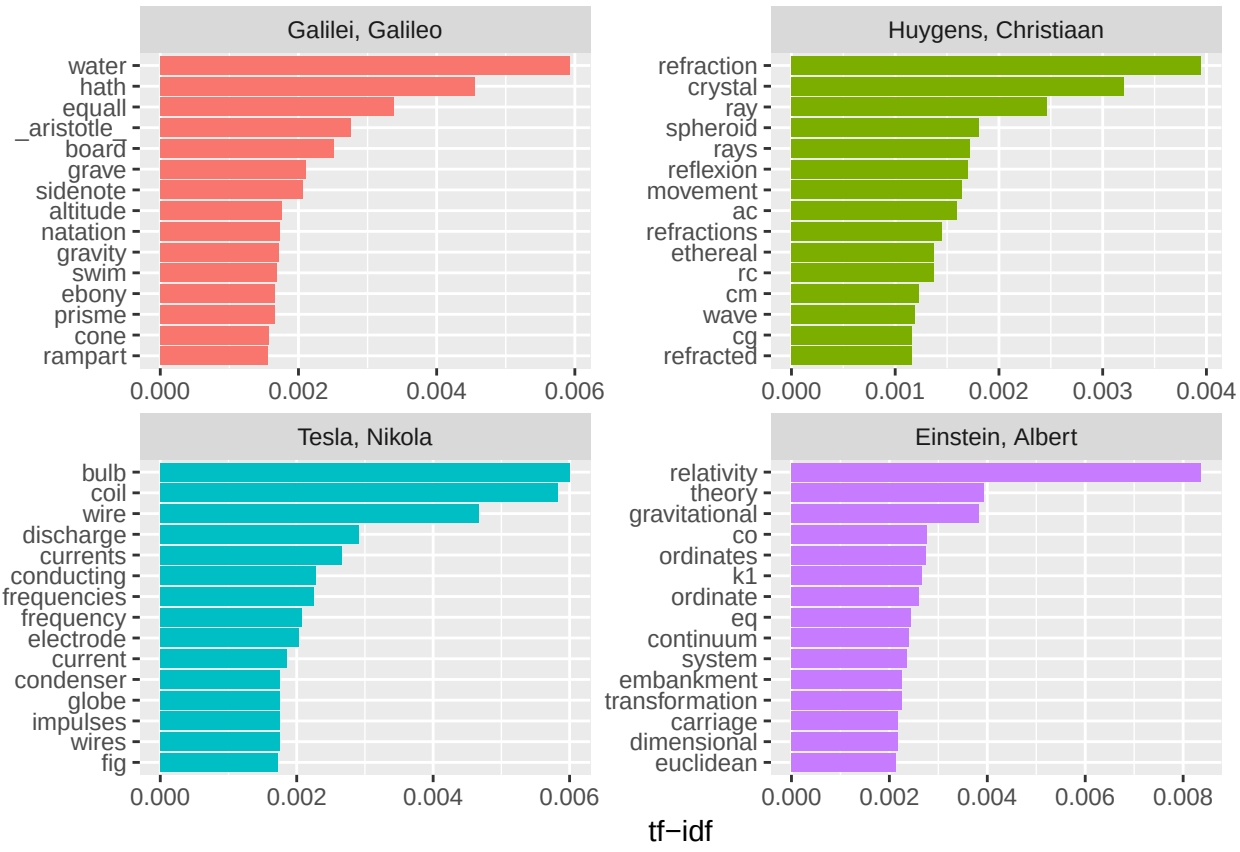
```
myths_china_words <- myths_china %>% unnest_tokens(word, text) %>% count(author,
    word, sort = TRUE) %>% ungroup()

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()
```


Galilei, Galileo
water
hath
equall
aristotle
board
grave
sidenote
altitude
natation
gravity
swim
ebony
prisme
cone
rampart
0.000
0.002
0.004
0.006
Huygens, Christiaan
refraction
crystal
ray
spheroid
rays
reflexion
movement
ac
refractions
ethereal
rc
cm
wave
cg
refracted
0.000
0.001
0.002
0.003
0.004
Tesla, Nikola
bulb
coil
wire
discharge
currents
conducting
frequencies
frequency
electrode
current
condenser
globe
impulses
wires
fig
0.000
0.002
0.004
0.006
Einstein, Albert
relativity
theory
gravitational
co
ordinates
k1
ordinate
eq
continuum
system
embankment
transformation
carriage
dimensional
euclidean
0.000
0.002
0.004
0.006
0.008
tf−idf