

# Test Mining with R

*Smiti Kaul*

*Feb 9 - present, 2018*

## Following the article

```
ids <- 1:5
works_sample <- gutenbergl_download(gutenberg_id = ids)
glimpse(works_sample)
names(gutenberg_metadata)
works_sample <- gutenbergl_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 <- gutenbergl_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
```

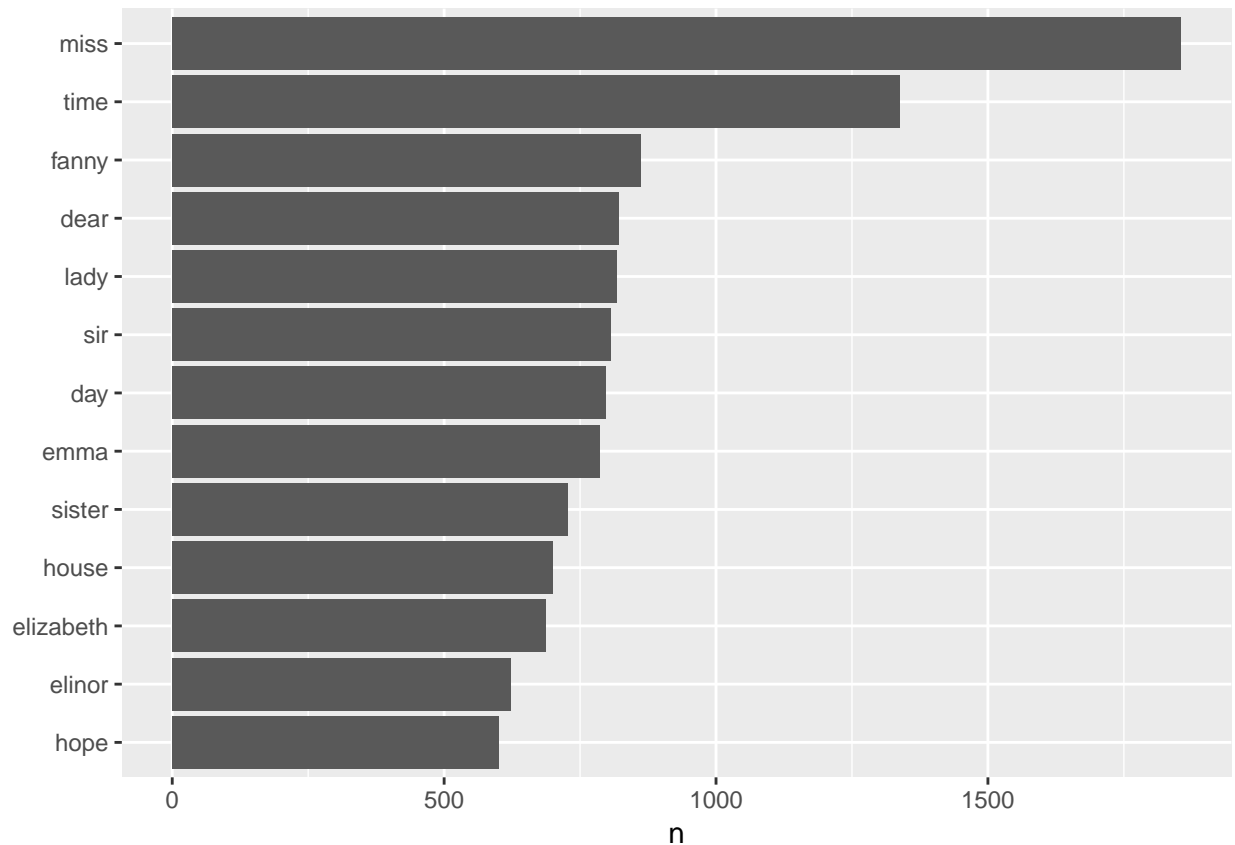
```

## 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 <- gutenberglownload(c(35, 36, 5230, 159))
## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest
## Using mirror http://aleph.gutenberg.org
bronte <- gutenberglownload(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)

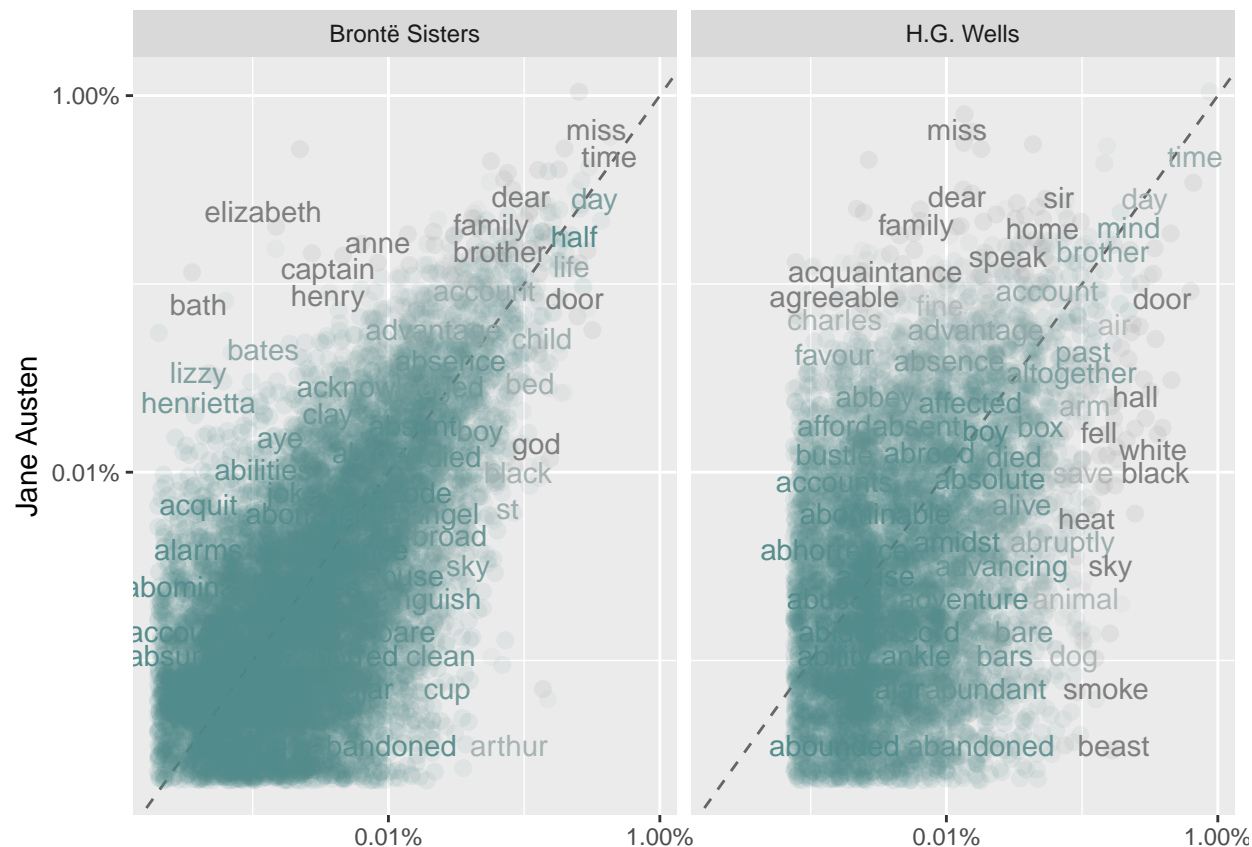
## # A tibble: 23,050 x 2
##   word      n
##   <chr> <int>
## 1 time    1065
## 2 miss     855
## 3 day      827
## 4 hand     768
## 5 eyes     713
## 6 night    647
## 7 heart    638
## 8 looked   601
## 9 door     592
## 10 half    586
## # ... with 23,040 more rows

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")

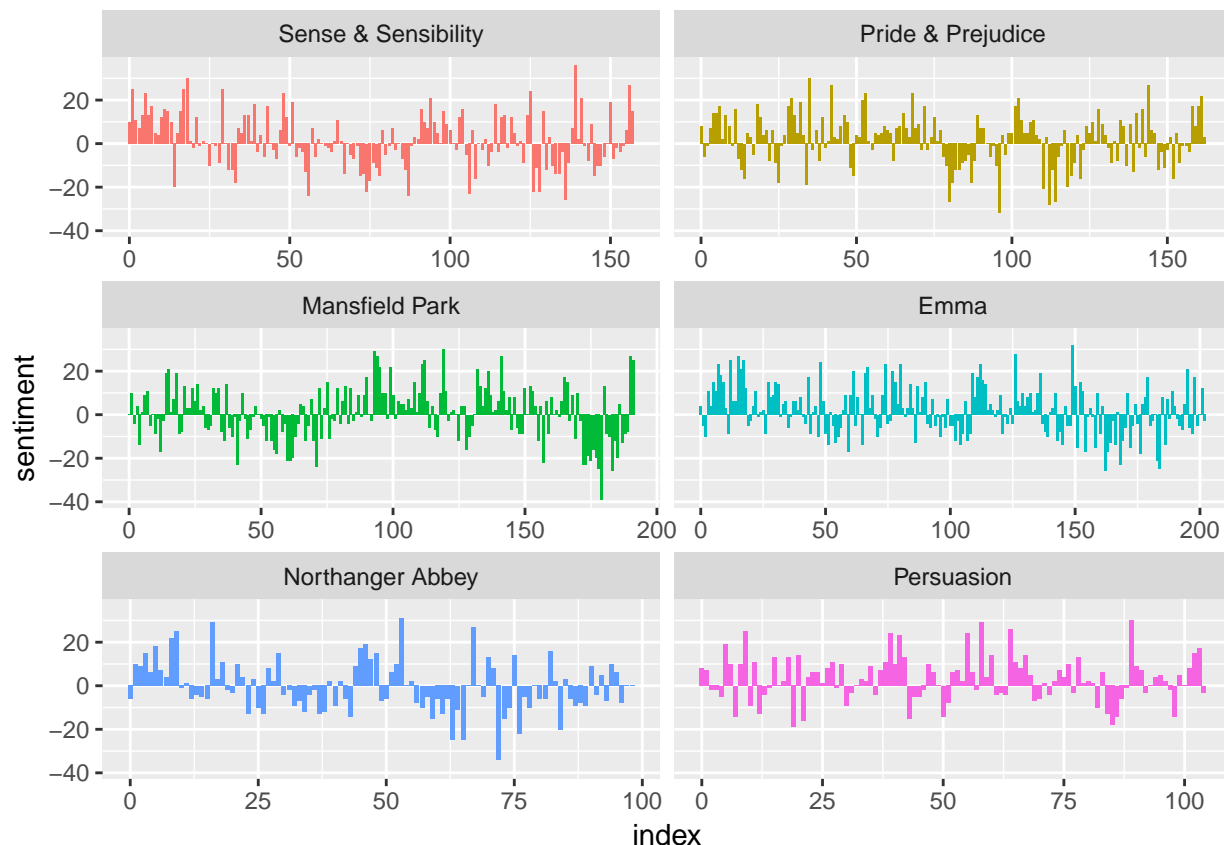
tidy_books %>% filter(book == "Emma") %>% inner_join(nrcjoy) %>% count(word,
  sort = TRUE)

## 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

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

## Joining, by = "word"

ggplot(jane Austensentiment, 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()

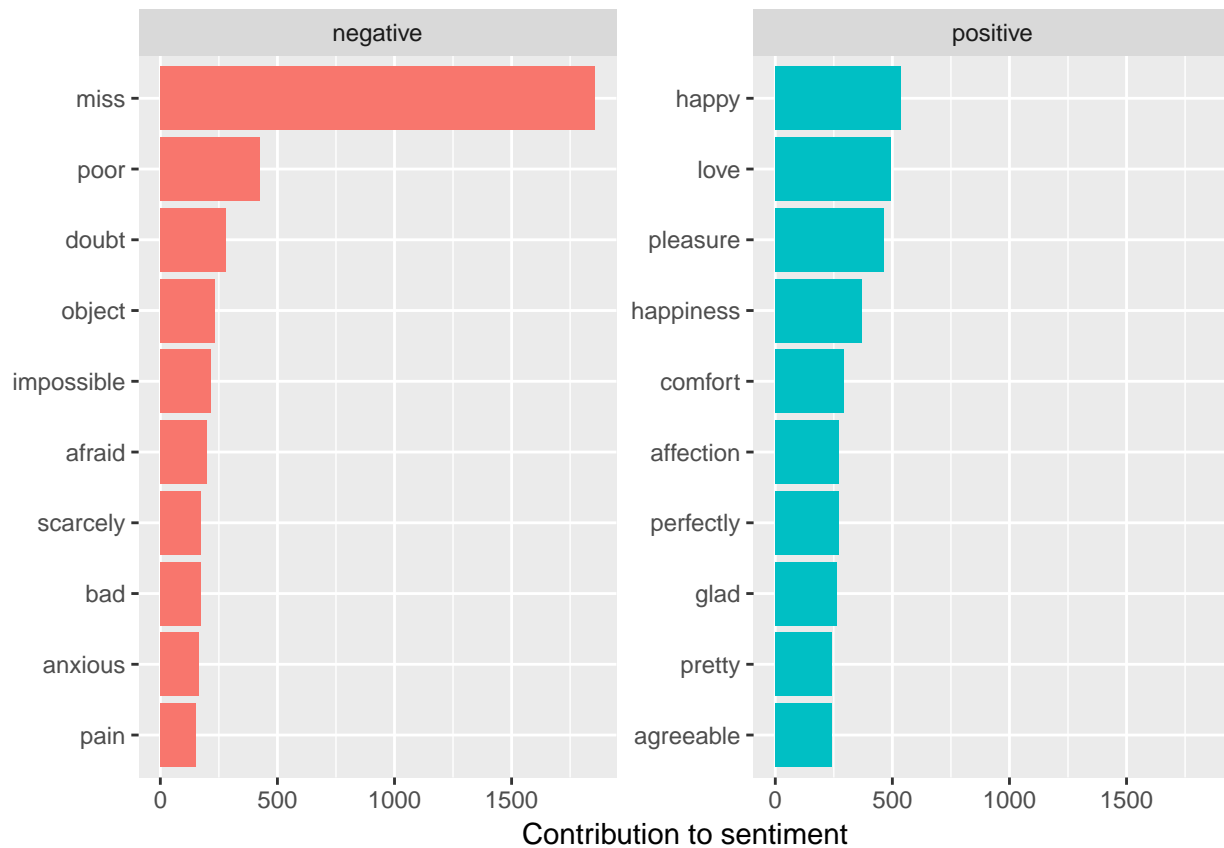
## Joining, by = "word"
bing_word_counts

## # 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

## 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"
```

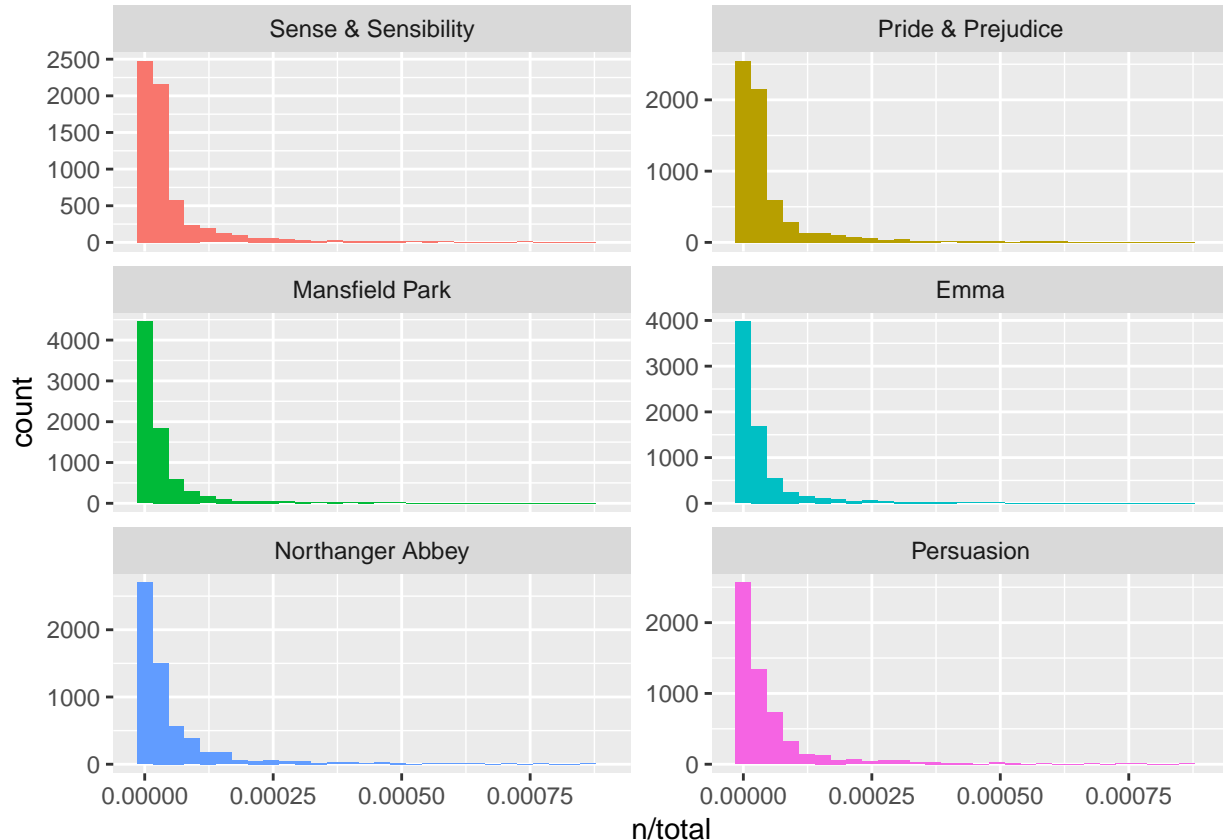
```
head(book_words)
```

```
## # 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).
```



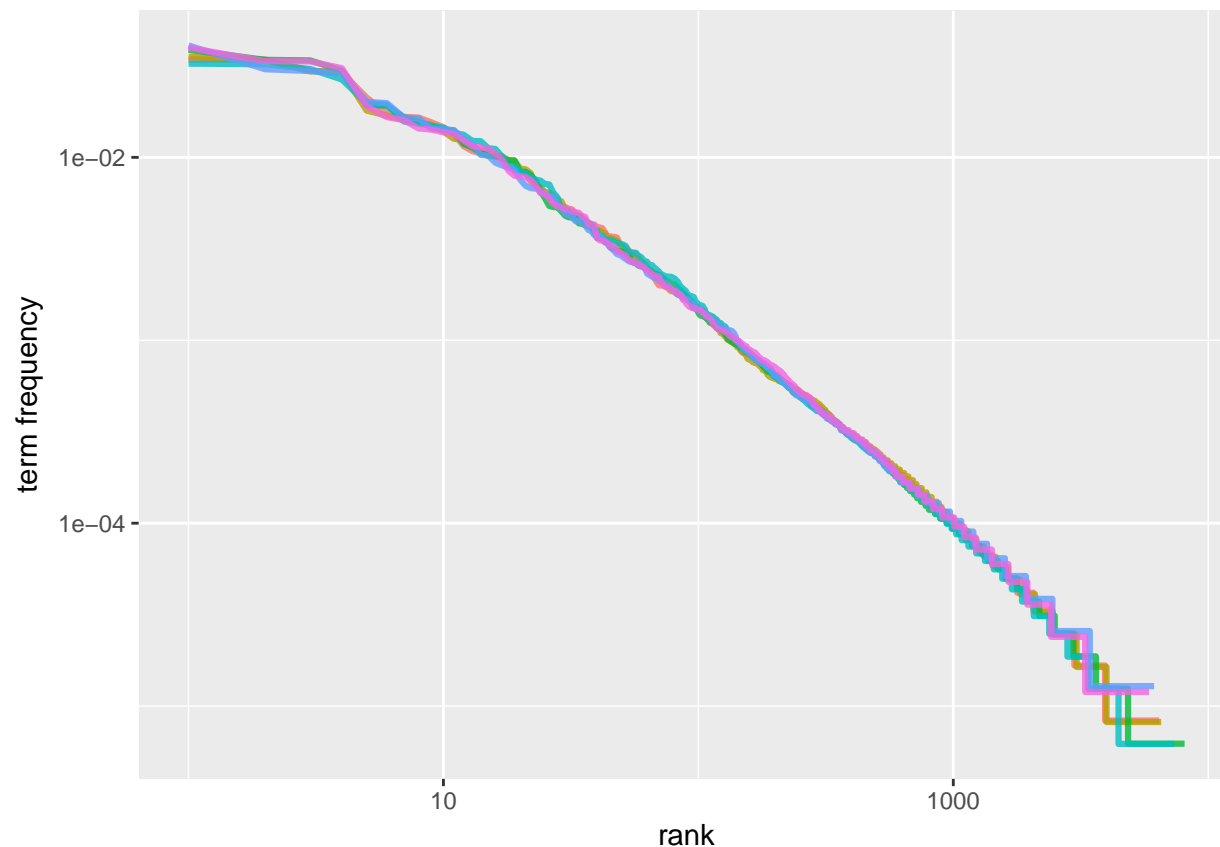
## Zipf's Law

```
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))
```

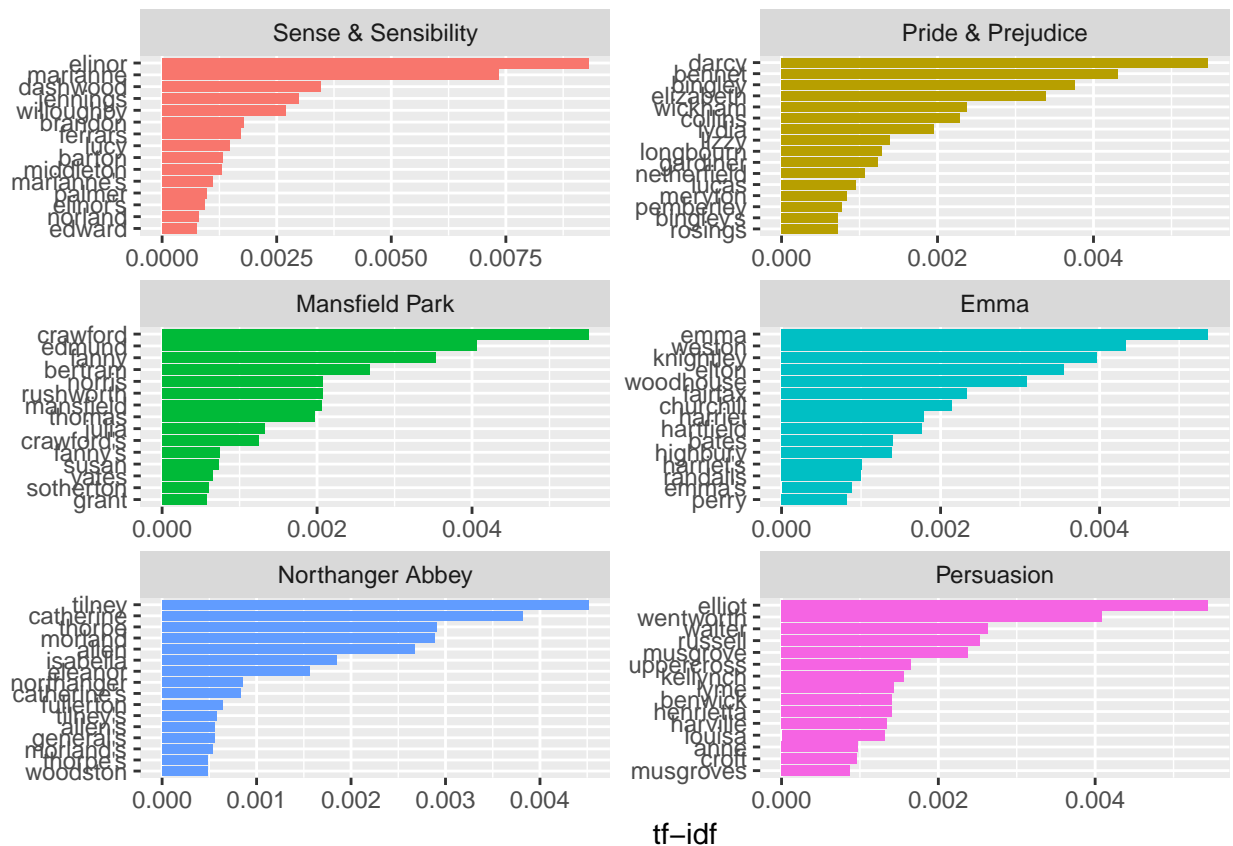
```
## # A tibble: 40,379 x 6
```

	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 <- gutenbergs_download(c(37729, 14725, 13476, 5001), meta_fields = "author")
physics_words <- physics %>% unnest_tokens(word, text) %>% count(author, word,
  sort = TRUE) %>% ungroup()
```

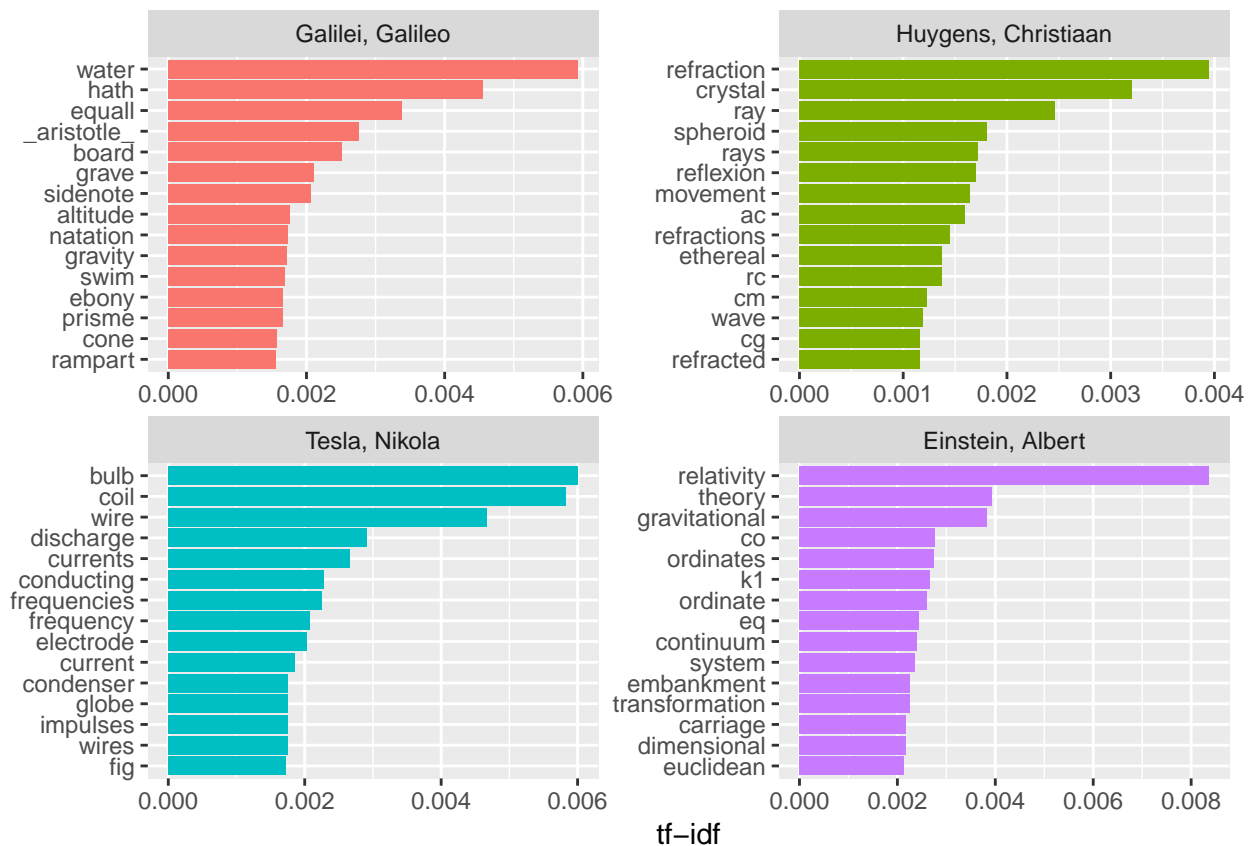
```
head(physics_words)
```

```
## # 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

```
# gutenber metadata %>% filter(title == 'Ramayana, English')
bib <- gutenber_download(10, meta_fields = "author")
anthem <- gutenber_download(1249, meta_fields = "author")
sid <- gutenber_download(2500, meta_fields = "author")
myths_china <- gutenber_download(15250, meta_fields = "author")
myths_japan <- gutenber_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

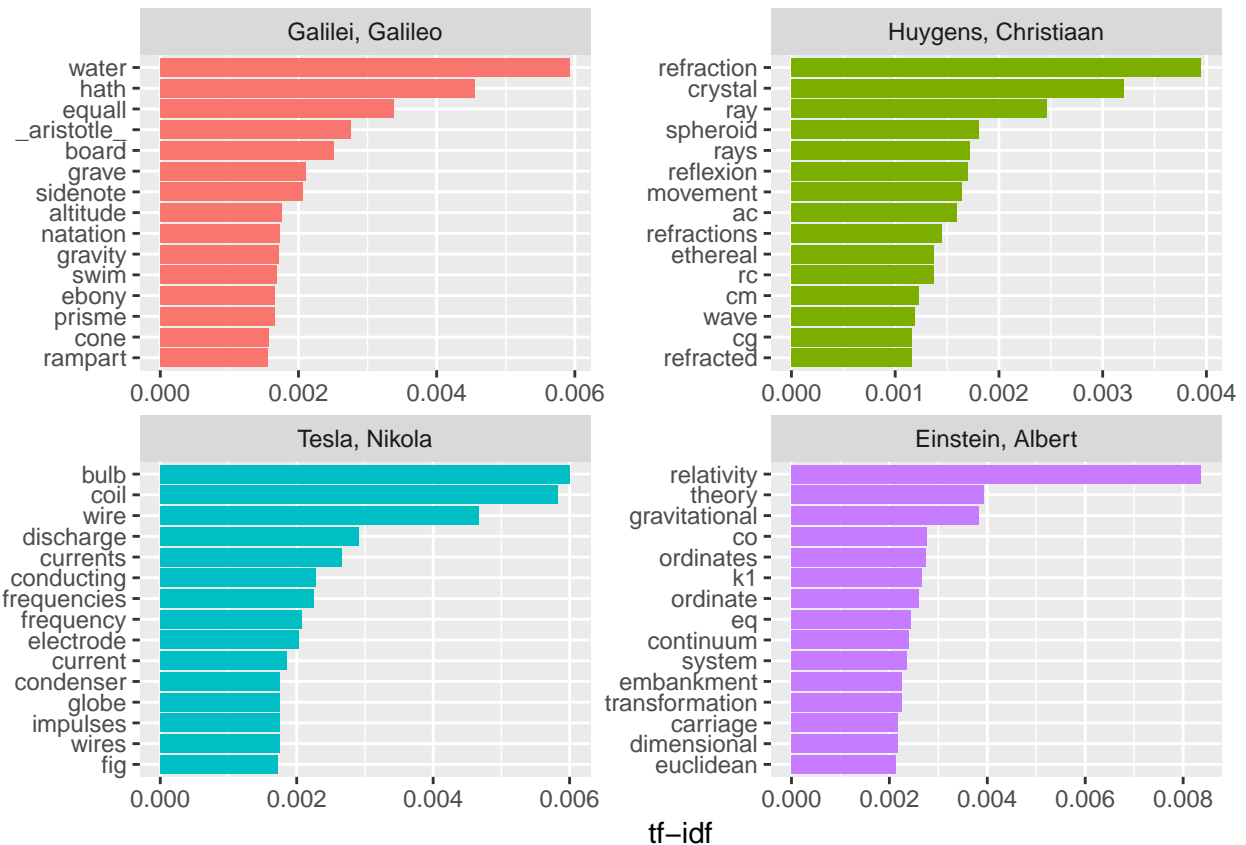
## # A tibble: 10,920 x 3
##   author                                word      n
##   <chr>                                <chr> <int>
## 1 Werner, E. T. C. (Edward Theodore Chalmers) the    10106
## 2 Werner, E. T. C. (Edward Theodore Chalmers) of      5111
## 3 Werner, E. T. C. (Edward Theodore Chalmers) 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
```

```

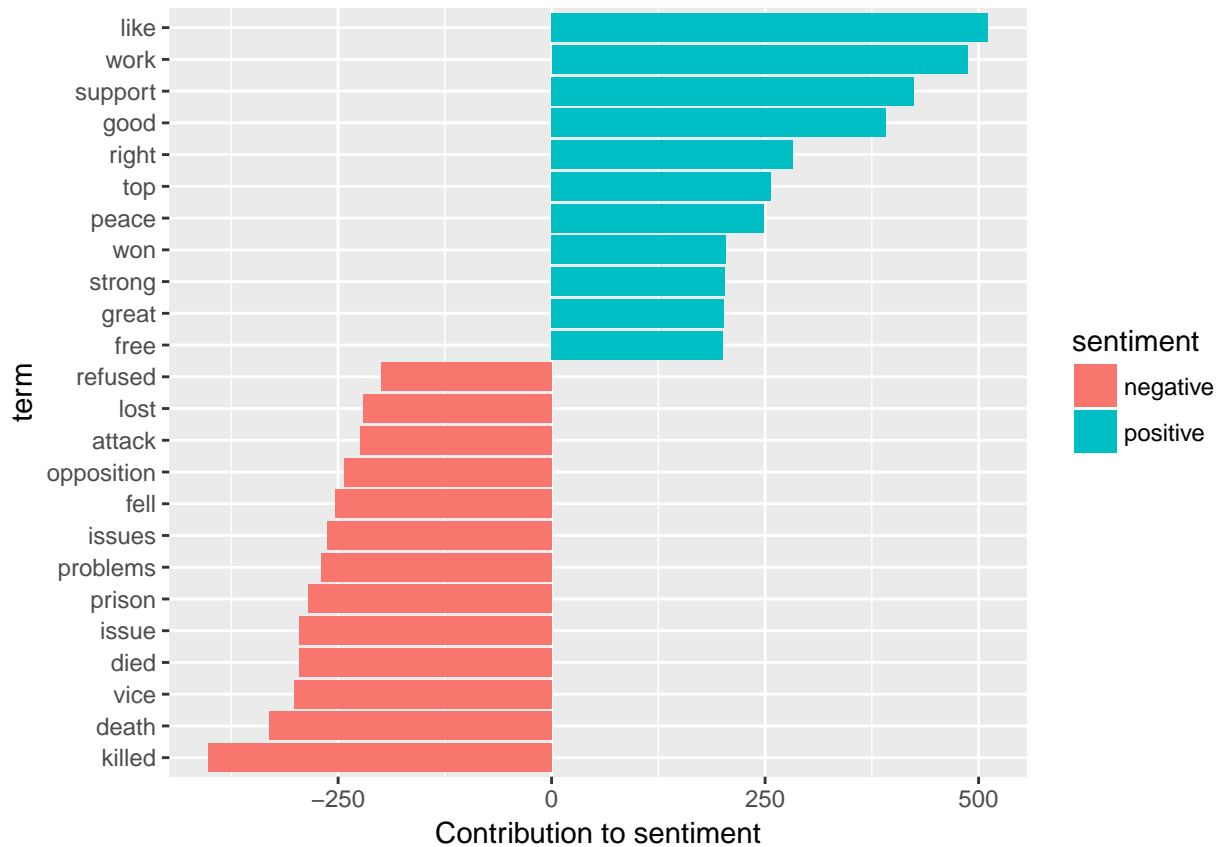
## 3      1 ago      1.00
## 4      1 alcohol  1.00
## 5      1 allegedly 1.00
## 6      1 allen    1.00
## 7      1 apparently 2.00
## 8      1 appeared  1.00
## 9      1 arrested  1.00
## 10     1 assault   1.00
## # ... with 302,021 more rows

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

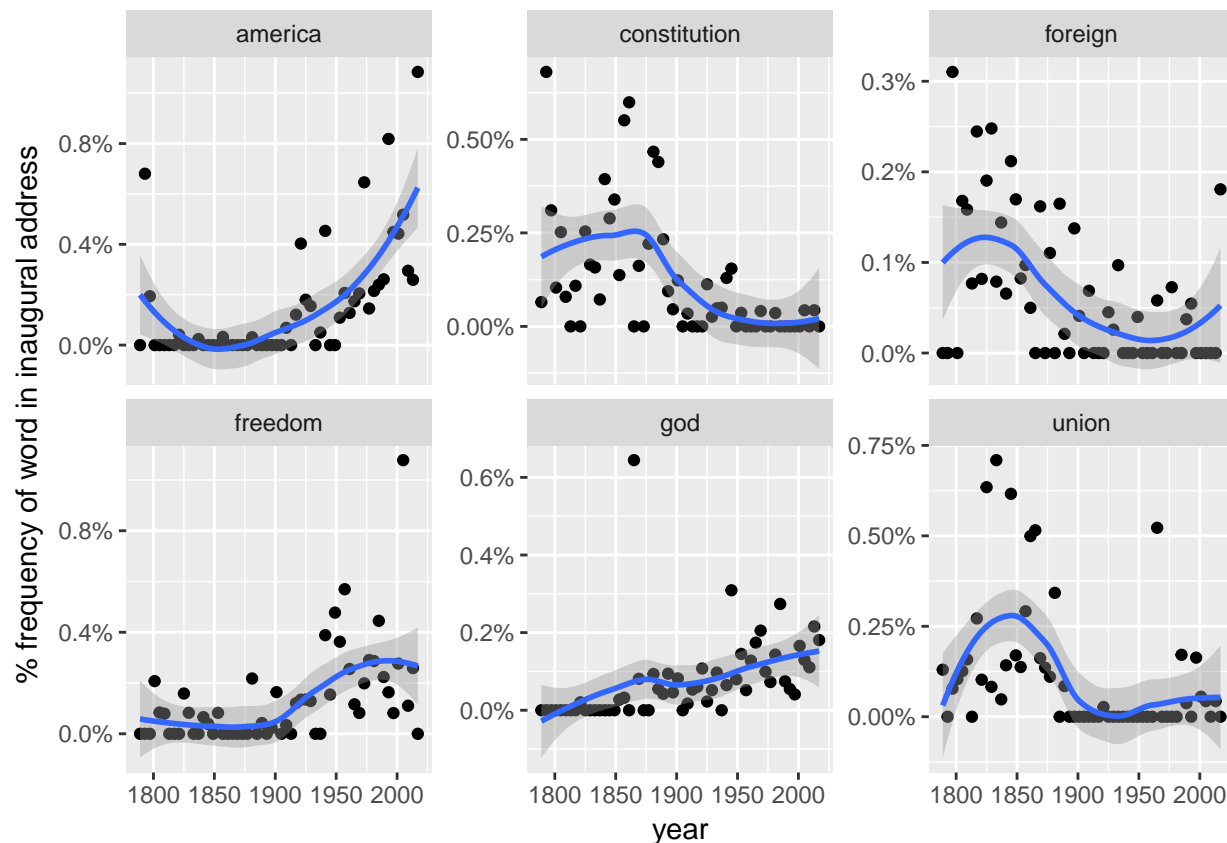
## # A tibble: 44,709 x 6
##   document      term      count      tf      idf tf_idf
##   <chr>         <chr>    <dbl>    <dbl> <dbl> <dbl>
## 1 1793-Washington arrive      1.00 0.00680 4.06 0.0276
## 2 1793-Washington upbraidings 1.00 0.00680 4.06 0.0276
## 3 1793-Washington violated    1.00 0.00680 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      timestamp      description heading    id language origin
##   <chr>      <dtm>      <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")

```

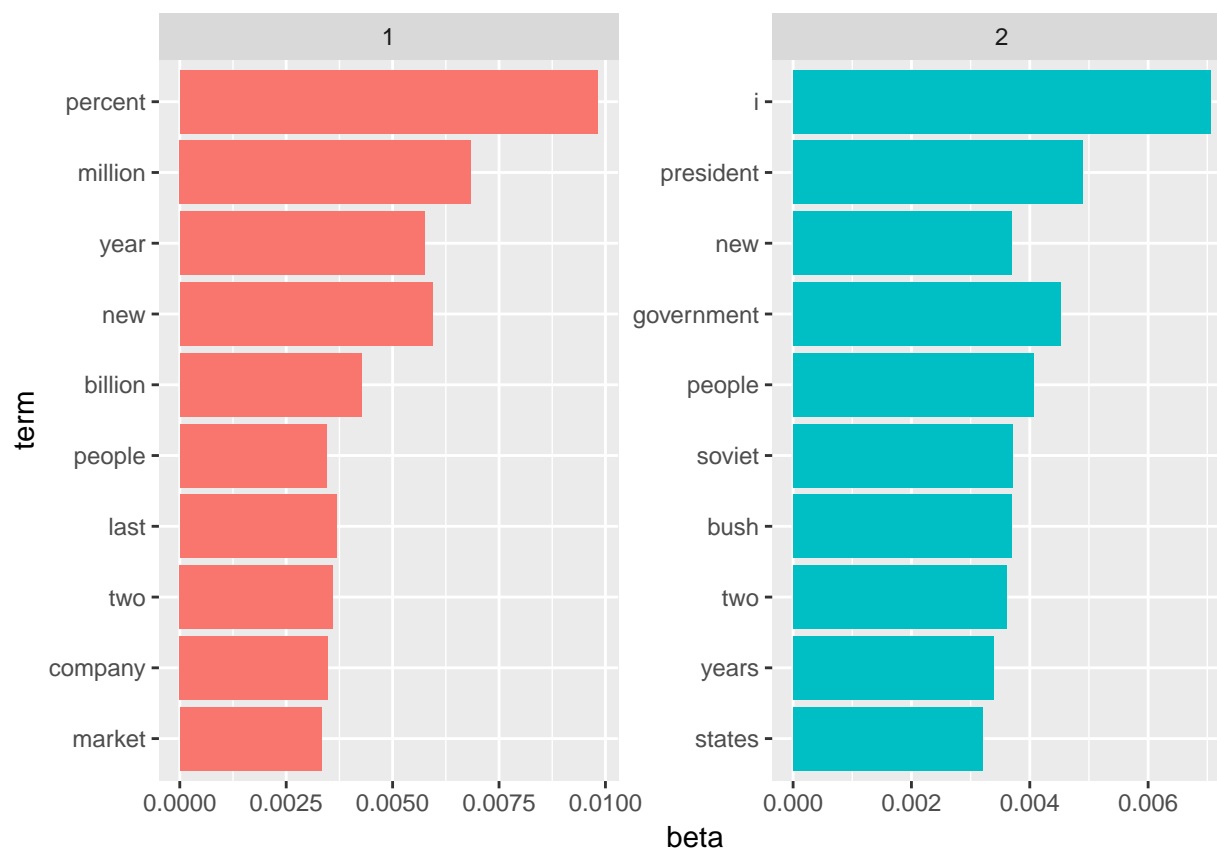




```

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 <- gutenbergs_works(title %in% titles) %>%
  gutenbergs_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)
```

```

# 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"

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

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)

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

## A LDA_VEM topic model with 4 topics.

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     5 1 biddy 5.14e-28
## 6     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

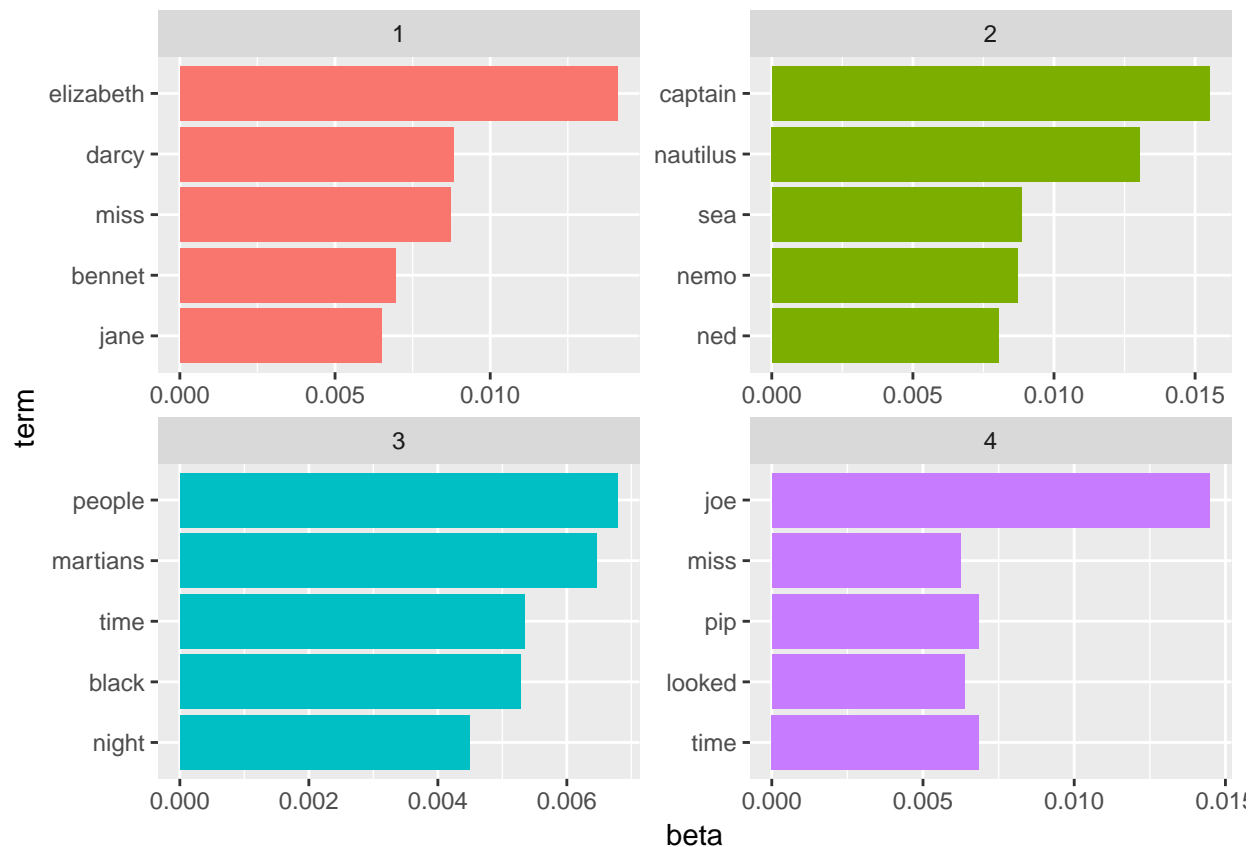
top_terms <- chapter_topics %>%
  group_by(topic) %>%
  top_n(5, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

top_terms

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

```

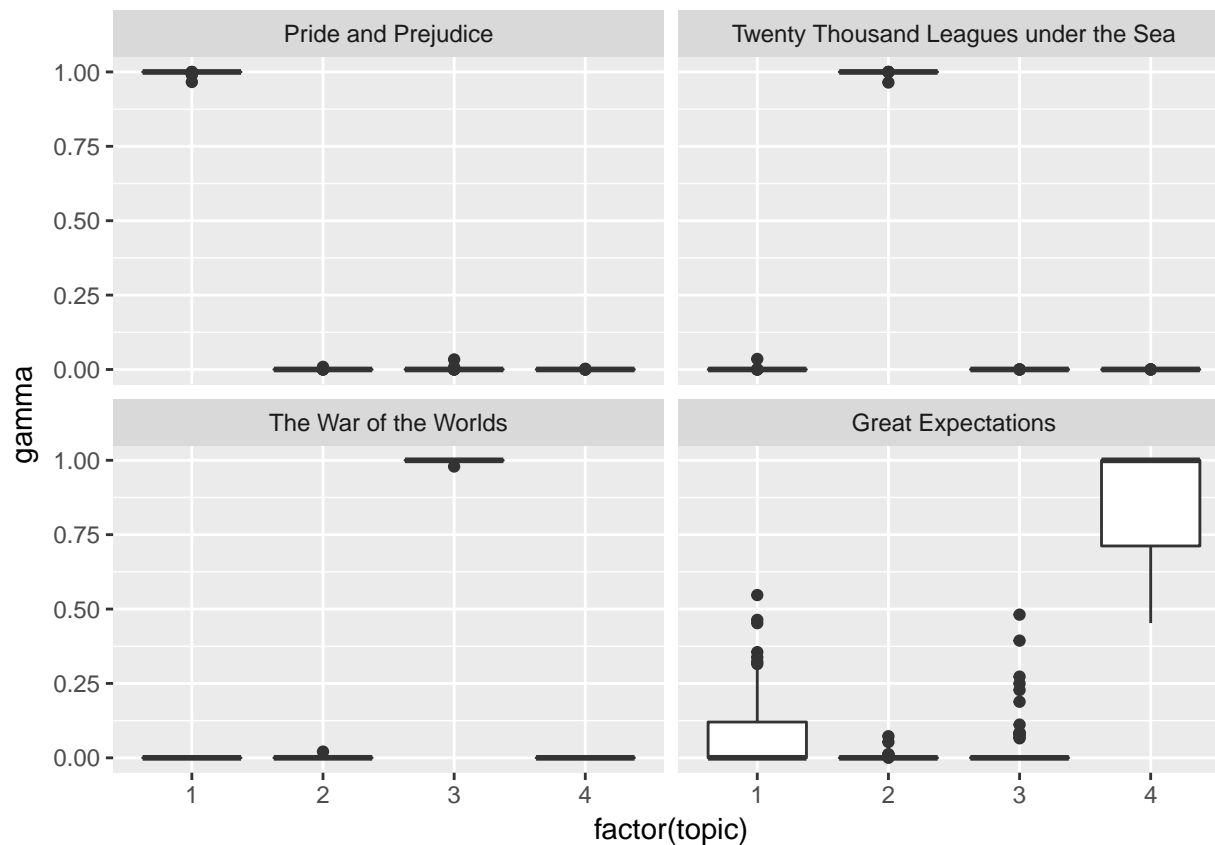
## # A tibble: 772 x 4
##   title                chapter topic    gamma
##   <chr>                <int> <int>    <dbl>
## 1 Great Expectations    57     1 0.0000134
## 2 Great Expectations     7     1 0.0000146
## 3 Great Expectations    17     1 0.0000210
## 4 Great Expectations    27     1 0.0000190
## 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

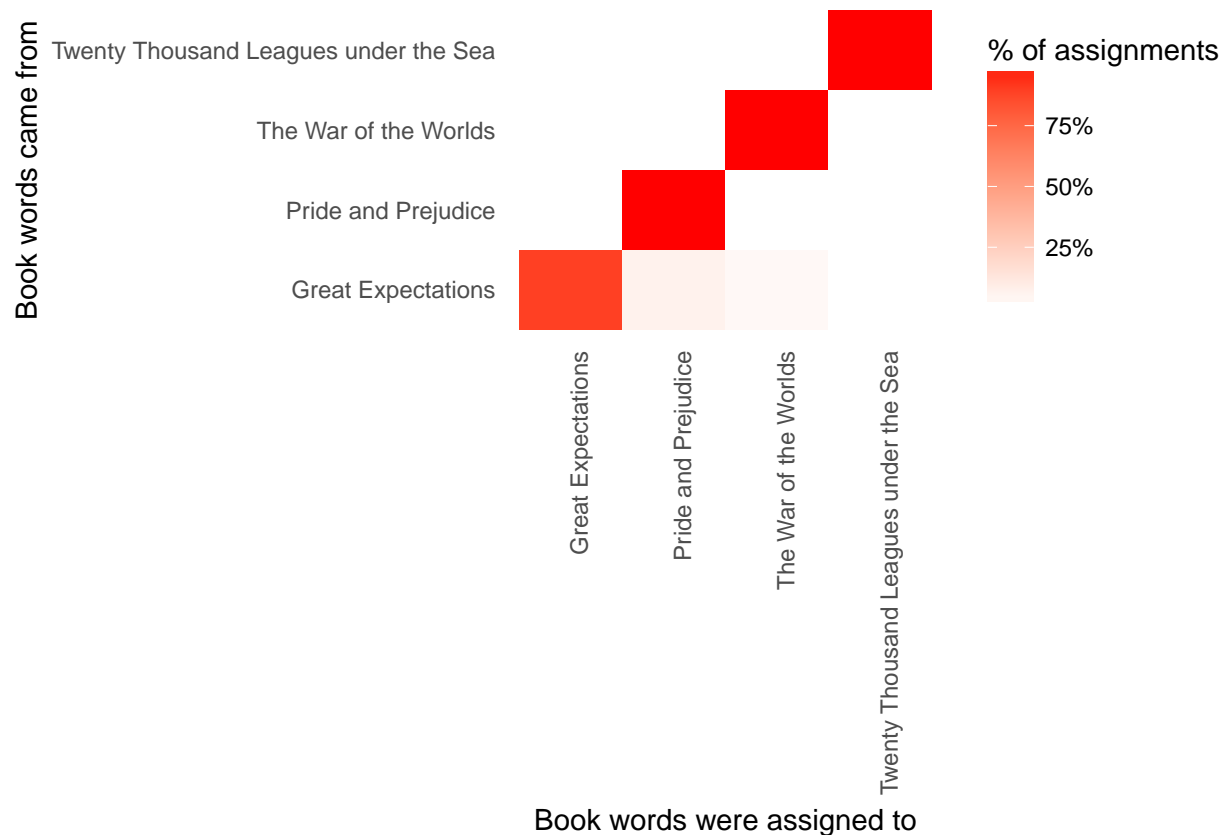
assignments <- augment(chapters_lda, data = chapters_dtm)
assignments <- assignments %>%
  separate(document, c("title", "chapter"), sep = "_", convert = TRUE) %>%
  inner_join(book_topics, by = c(".topic" = "topic"))

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 1 chapter 0.00103
## 2     2 2 chapter 0.00172
## 3     3 3 chapter 0.000000260
## 4     4 4 chapter 0.00227
## 5     1 1 fathers 0.000000304
## 6     2 2 fathers 0.000730
## 7     3 3 fathers 0.000000260
## 8     4 4 fathers 0.000000267
## 9     1 1 family 0.000000304
## 10    2 2 family 0.00425
## # ... 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 0.0782
## 2 Great Expectations_10     1 0.0200
## 3 Great Expectations_11     1 0.0302
## 4 Great Expectations_12     1 0.0558
```

```
## 5 Great Expectations_13      1 0.0160
## 6 Great Expectations_14      1 0.0666
## 7 Great Expectations_15      1 0.0399
## 8 Great Expectations_16      1 0.0461
## 9 Great Expectations_17      1 0.0340
## 10 Great Expectations_18     1 0.0232
## # ... 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     4
## 2 Great Expectations_7     joe      70     4
## 3 Great Expectations_17    biddy     63     4
## 4 Great Expectations_27    joe      58     4
## 5 Great Expectations_38    estella   58     3
## 6 Great Expectations_2     joe      56     4
## 7 Great Expectations_23    pocket    53     3
## 8 Great Expectations_15    joe      50     4
## 9 Great Expectations_18    joe      50     4
## 10 The War of the Worlds_16 brother    50     3
## # ... with 104,712 more rows
```

## Case Study: comparing Twitter archives

### 7.1 Getting the data and distribution of tweets

```
library(lubridate)

## Warning: package 'lubridate' was built under R version 3.4.3
##
## Attaching package: 'lubridate'
##
## The following object is masked from 'package:base':
##
##   date

library(ggplot2)
library(dplyr)
library(readr)

tweets_julia <- read_csv("tweets_julia.csv")

## Parsed with column specification:
## cols(
##   tweet_id = col_double(),
##   in_reply_to_status_id = col_double(),
##   in_reply_to_user_id = col_double(),
##   timestamp = col_character(),
##   source = col_character(),
##   text = col_character(),
```

```

##   retweeted_status_id = col_double(),
##   retweeted_status_user_id = col_double(),
##   retweeted_status_timestamp = col_character(),
##   expanded_urls = col_character()
## )

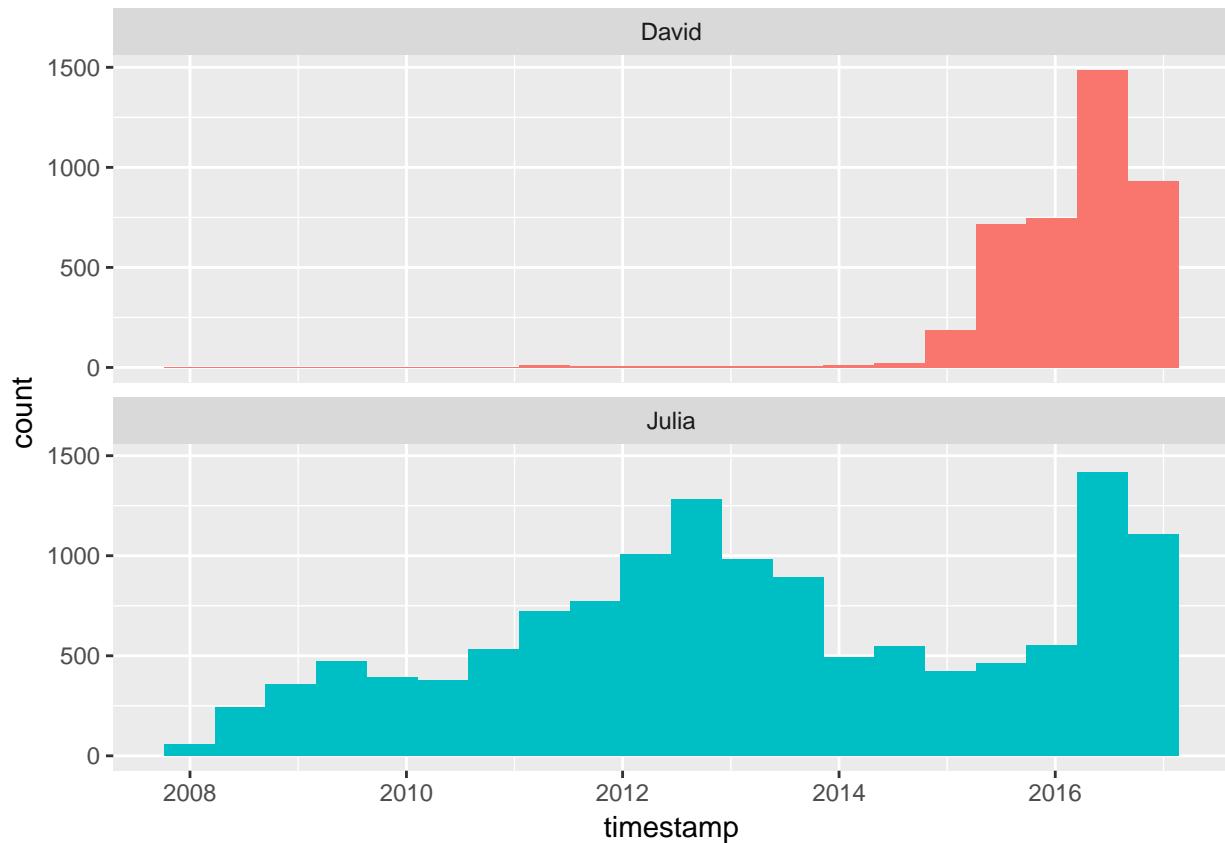
tweets_dave <- read_csv("tweets_dave.csv")

## Parsed with column specification:
## cols(
##   tweet_id = col_double(),
##   in_reply_to_status_id = col_double(),
##   in_reply_to_user_id = col_double(),
##   timestamp = col_character(),
##   source = col_character(),
##   text = col_character(),
##   retweeted_status_id = col_double(),
##   retweeted_status_user_id = col_double(),
##   retweeted_status_timestamp = col_character(),
##   expanded_urls = col_character()
## )

tweets <- bind_rows(tweets_julia %>%
  mutate(person = "Julia"),
  tweets_dave %>%
  mutate(person = "David")) %>%
  mutate(timestamp = ymd_hms(timestamp))

ggplot(tweets, aes(x = timestamp, fill = person)) +
  geom_histogram(position = "identity", bins = 20, show.legend = FALSE) +
  facet_wrap(~person, ncol = 1)

```



## 7.2 Word frequencies

```
library(tidytext)
library(stringr)

replace_reg <- "https://t.co/[A-Za-z\\d]+|http://[A-Za-z\\d]+|&|<|>|RT|https"
unnest_reg <- "([A-Za-z_\\d#@']|'?![A-Za-z_\\d#@'])"
tidy_tweets <- tweets %>%
  filter(!str_detect(text, "^RT")) %>%
  mutate(text = str_replace_all(text, replace_reg, "")) %>%
  unnest_tokens(word, text, token = "regex", pattern = unnest_reg) %>%
  filter(!word %in% stop_words$word,
         str_detect(word, "[a-z]"))

frequency <- tidy_tweets %>%
  group_by(person) %>%
  count(word, sort = TRUE) %>%
  left_join(tidy_tweets %>%
            group_by(person) %>%
            summarise(total = n())) %>%
  mutate(freq = n/total)

## Joining, by = "person"

frequency

## # A tibble: 20,736 x 5
```

```
## # Groups:   person [2]
##   person word          n total   freq
##   <chr>  <chr>        <int> <int>  <dbl>
## 1 Julia  time          584 74572 0.00783
## 2 Julia  @selkie1970      570 74572 0.00764
## 3 Julia  @skedman         531 74572 0.00712
## 4 Julia  day           467 74572 0.00626
## 5 Julia  baby          408 74572 0.00547
## 6 David  @hadleywickham  315 20161 0.0156
## 7 Julia  love           304 74572 0.00408
## 8 Julia  @haleynburke    299 74572 0.00401
## 9 Julia  house          289 74572 0.00388
## 10 Julia morning       278 74572 0.00373
## # ... with 20,726 more rows

library(tidyr)

frequency <- frequency %>%
  select(person, word, freq) %>%
  spread(person, freq) %>%
  arrange(Julia, David)

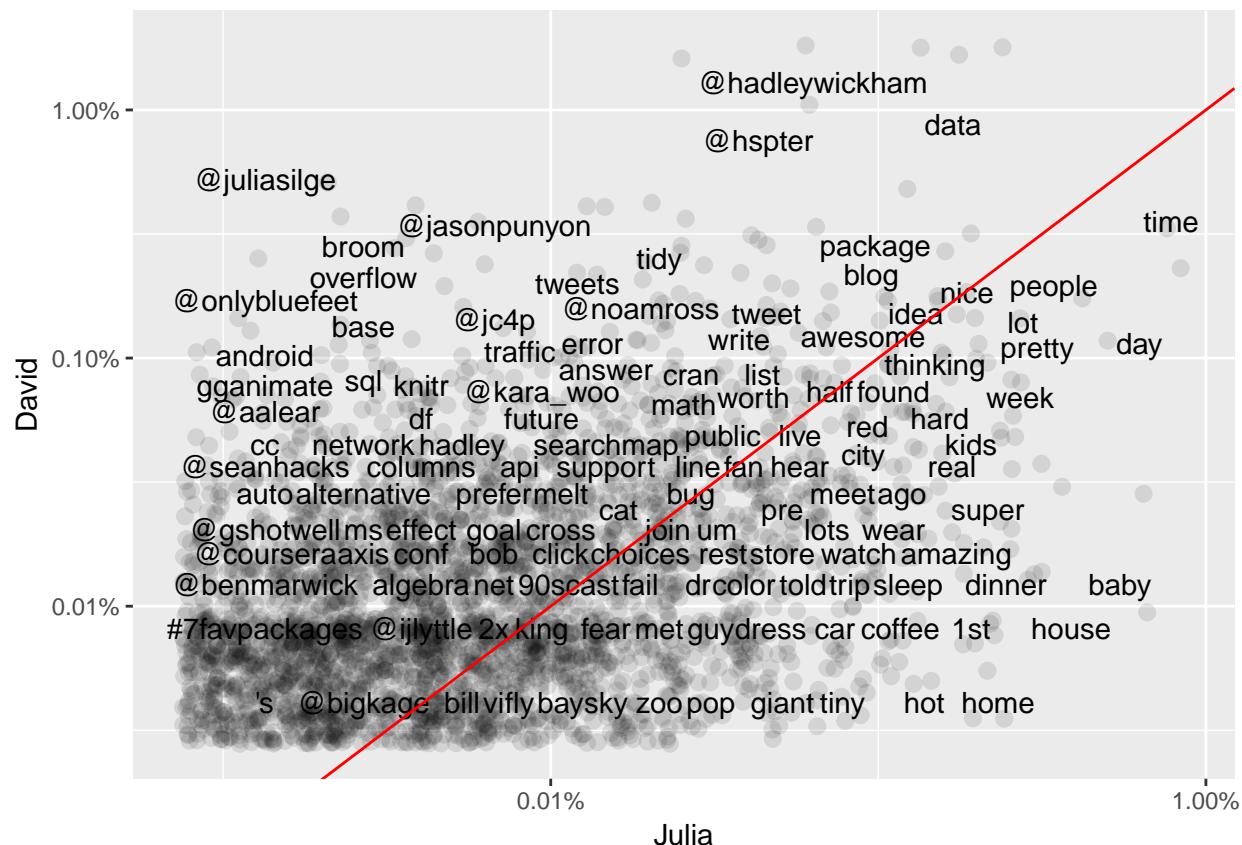
frequency

## # A tibble: 17,640 x 3
##   word          David      Julia
##   <chr>        <dbl>    <dbl>
## 1 's           0.0000496 0.0000134
## 2 @accidental__art 0.0000496 0.0000134
## 3 @alice_data     0.0000496 0.0000134
## 4 @alistiche     0.0000496 0.0000134
## 5 @corynissen     0.0000496 0.0000134
## 6 @jennybryan's   0.0000496 0.0000134
## 7 @jsvine        0.0000496 0.0000134
## 8 @lizasperling   0.0000496 0.0000134
## 9 @ognyanova      0.0000496 0.0000134
## 10 @rbloggers     0.0000496 0.0000134
## # ... with 17,630 more rows

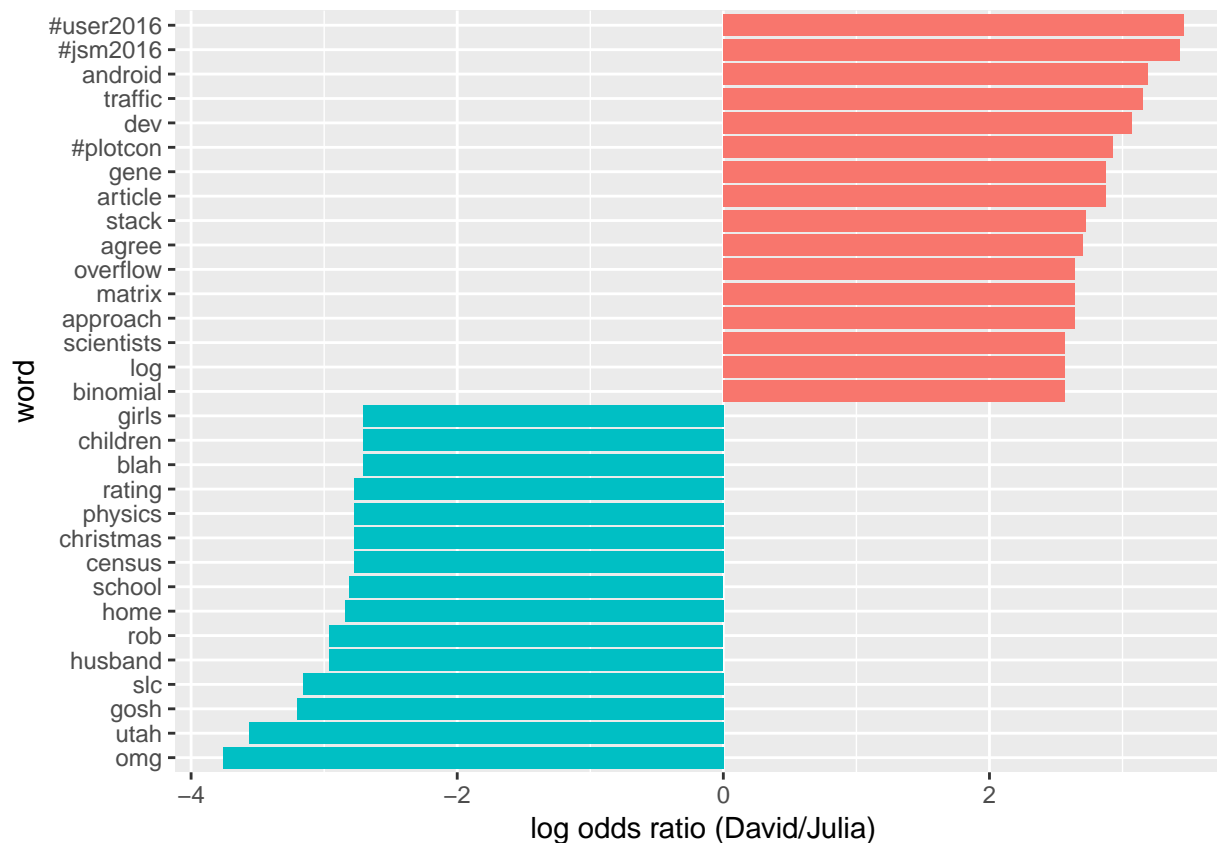
library(scales)

ggplot(frequency, aes(Julia, David)) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.25, height = 0.25) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  geom_abline(color = "red")

## Warning: Removed 14544 rows containing missing values (geom_point).
## Warning: Removed 14544 rows containing missing values (geom_text).
```



### 7.3: Comparing word usage



## 7.4: Changes in word use

```
words_by_time <- tidy_tweets %>%
  filter(!str_detect(word, "^@")) %>%
  mutate(time_floor = floor_date(timestamp, unit = "1 month")) %>%
  count(time_floor, person, word) %>%
  ungroup() %>%
  group_by(person, time_floor) %>%
  mutate(time_total = sum(n)) %>%
  group_by(word) %>%
  mutate(word_total = sum(n)) %>%
  ungroup() %>%
  rename(count = n) %>%
  filter(word_total > 30)
```

words\_by\_time

## # A tibble: 970 x 6

##	time_floor	person	word	count	time_total	word_total
##	<dtm>	<chr>	<chr>	<int>	<int>	<int>
##	1 2016-01-01 00:00:00	David	#rstats	2	307	324
##	2 2016-01-01 00:00:00	David	bad	1	307	33
##	3 2016-01-01 00:00:00	David	bit	2	307	45
##	4 2016-01-01 00:00:00	David	blog	1	307	60
##	5 2016-01-01 00:00:00	David	broom	2	307	41

```

## 6 2016-01-01 00:00:00 David call      2      307      31
## 7 2016-01-01 00:00:00 David check    1      307      42
## 8 2016-01-01 00:00:00 David code     3      307      49
## 9 2016-01-01 00:00:00 David data     2      307     276
## 10 2016-01-01 00:00:00 David day      2      307      65
## # ... with 960 more rows

nested_data <- words_by_time %>%
  nest(-word, -person)

nested_data

## # A tibble: 112 x 3
##   person word    data
##   <chr>  <chr>  <list>
## 1 David  #rstats <tibble [12 x 4]>
## 2 David  bad    <tibble [9 x 4]>
## 3 David  bit    <tibble [10 x 4]>
## 4 David  blog   <tibble [12 x 4]>
## 5 David  broom  <tibble [10 x 4]>
## 6 David  call   <tibble [9 x 4]>
## 7 David  check  <tibble [12 x 4]>
## 8 David  code   <tibble [10 x 4]>
## 9 David  data   <tibble [12 x 4]>
## 10 David day    <tibble [8 x 4]>
## # ... with 102 more rows

library(purrr)

nested_models <- nested_data %>%
  mutate(models = map(data, ~ glm(cbind(count, time_total) ~ time_floor, .,
                                family = "binomial")))

nested_models

## # A tibble: 112 x 4
##   person word    data      models
##   <chr>  <chr>  <list>    <list>
## 1 David  #rstats <tibble [12 x 4]> <S3: glm>
## 2 David  bad    <tibble [9 x 4]> <S3: glm>
## 3 David  bit    <tibble [10 x 4]> <S3: glm>
## 4 David  blog   <tibble [12 x 4]> <S3: glm>
## 5 David  broom  <tibble [10 x 4]> <S3: glm>
## 6 David  call   <tibble [9 x 4]> <S3: glm>
## 7 David  check  <tibble [12 x 4]> <S3: glm>
## 8 David  code   <tibble [10 x 4]> <S3: glm>
## 9 David  data   <tibble [12 x 4]> <S3: glm>
## 10 David day    <tibble [8 x 4]> <S3: glm>
## # ... with 102 more rows

library(broom)

## Warning: package 'broom' was built under R version 3.4.3

slopes <- nested_models %>%
  unnest(map(models, tidy)) %>%
  filter(term == "time_floor") %>%

```



```

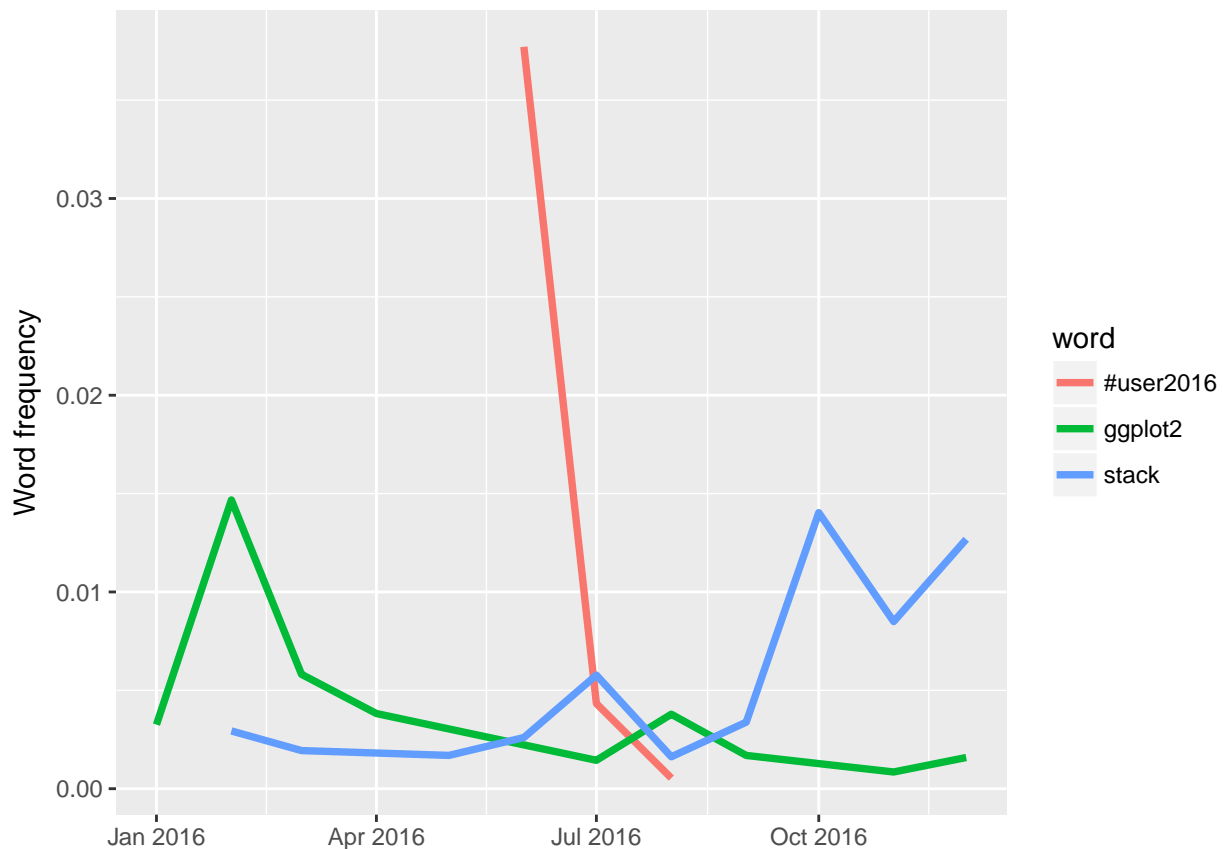
mutate(adjusted.p.value = p.adjust(p.value))
top_slopes <- slopes %>%
  filter(adjusted.p.value < 0.1)

top_slopes

## # A tibble: 6 x 8
##   person word  term estimate std.error statistic p.value adjusted.p.value
##   <chr> <chr> <chr>    <dbl>    <dbl>    <dbl> <dbl>    <dbl>
## 1 David  ggpl~ time~ -8.26e-8  1.97e-8   -4.20 2.72e-5    0.00300
## 2 Julia  #rst~ time~ -4.50e-8  1.12e-8   -4.02 5.93e-5    0.00647
## 3 Julia  post~ time~ -4.82e-8  1.45e-8   -3.31 9.23e-4    0.0978
## 4 Julia  read~ time~ -9.33e-8  2.54e-8   -3.67 2.44e-4    0.0263
## 5 David  stack~ time~  8.04e-8  2.19e-8    3.67 2.46e-4    0.0263
## 6 David  #use~ time~ -8.18e-7  1.55e-7   -5.27 1.33e-7    0.0000148

words_by_time %>%
  inner_join(top_slopes, by = c("word", "person")) %>%
  filter(person == "David") %>%
  ggplot(aes(time_floor, count/time_total, color = word)) +
  geom_line(size = 1.3) +
  labs(x = NULL, y = "Word frequency")

```

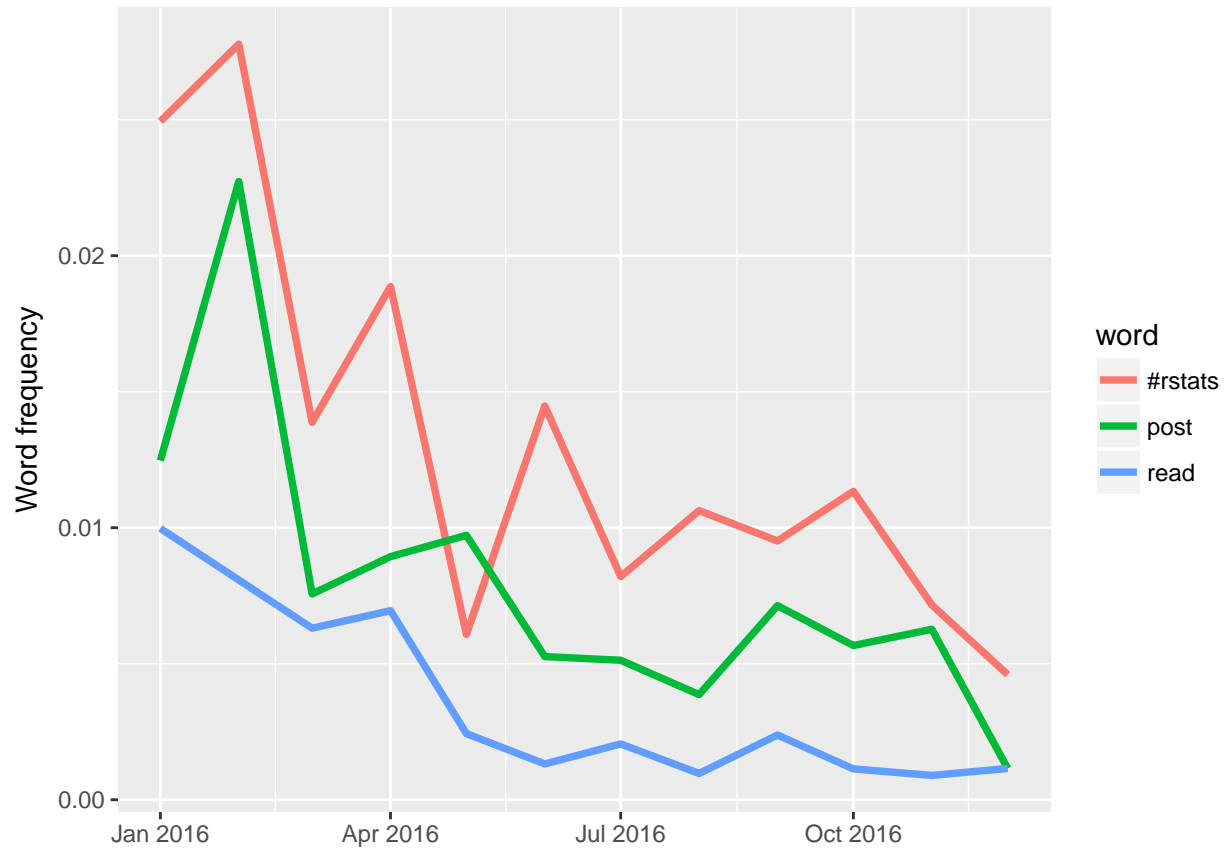


```

words_by_time %>%
  inner_join(top_slopes, by = c("word", "person")) %>%
  filter(person == "Julia") %>%
  ggplot(aes(time_floor, count/time_total, color = word)) +

```

```
geom_line(size = 1.3) +
labs(x = NULL, y = "Word frequency")
```



## 7.5: Favorites and retweets

```
tweets_julia <- read_csv("juliasilge_tweets.csv")

## Parsed with column specification:
## cols(
##   id = col_double(),
##   created_at = col_datetime(format = ""),
##   source = col_character(),
##   retweets = col_integer(),
##   favorites = col_integer(),
##   text = col_character()
## )

tweets_dave <- read_csv("drob_tweets.csv")

## Parsed with column specification:
## cols(
##   id = col_double(),
##   created_at = col_datetime(format = ""),
##   source = col_character(),
##   retweets = col_integer(),
##   favorites = col_integer(),
##   text = col_character()
```

```
## )
tweets <- bind_rows(tweets_julia %>%
  mutate(person = "Julia"),
  tweets_dave %>%
  mutate(person = "David")) %>%
  mutate(created_at = ymd_hms(created_at))

tidy_tweets <- tweets %>%
  filter(!str_detect(text, "^ (RT|@)")) %>%
  mutate(text = str_replace_all(text, replace_reg, "")) %>%
  unnest_tokens(word, text, token = "regex", pattern = unnest_reg) %>%
  anti_join(stop_words)

## Joining, by = "word"

tidy_tweets

## # A tibble: 11,078 x 7
##       id created_at      source retweets favorites person word
##   <dbl> <dtm>      <chr>    <int>    <int> <chr> <chr>
## 1  5.94e17 2015-05-01 01:57:38 Instagr~      0        0 Julia violet
## 2  5.94e17 2015-05-01 01:57:38 Instagr~      0        0 Julia hubble
## 3  5.94e17 2015-05-01 01:57:38 Instagr~      0        0 Julia space
## 4  5.94e17 2015-05-01 01:57:38 Instagr~      0        0 Julia teles~
## 5  5.94e17 2015-05-01 01:57:38 Instagr~      0        0 Julia kinde~
## 6  5.94e17 2015-05-01 01:57:38 Instagr~      0        0 Julia week
## 7  5.94e17 2015-05-01 01:57:38 Instagr~      0        0 Julia pretty
## 8  5.94e17 2015-05-01 01:57:38 Instagr~      0        0 Julia happy
## 9  5.94e17 2015-05-01 02:49:02 Twitter~      0        0 Julia life
## 10 5.94e17 2015-05-01 02:49:02 Twitter~      0        0 Julia pee
## # ... with 11,068 more rows

totals <- tidy_tweets %>%
  group_by(person, id) %>%
  summarise(rts = sum(retweets)) %>%
  group_by(person) %>%
  summarise(total_rts = sum(rts))

totals

## # A tibble: 2 x 2
##   person total_rts
##   <chr>    <int>
## 1 David    110171
## 2 Julia    12701

word_by_rts <- tidy_tweets %>%
  group_by(id, word, person) %>%
  summarise(rts = first(retweets)) %>%
  group_by(person, word) %>%
  summarise(retweets = median(rts), uses = n()) %>%
  left_join(totals) %>%
  filter(retweets != 0) %>%
  ungroup()

## Joining, by = "person"
```

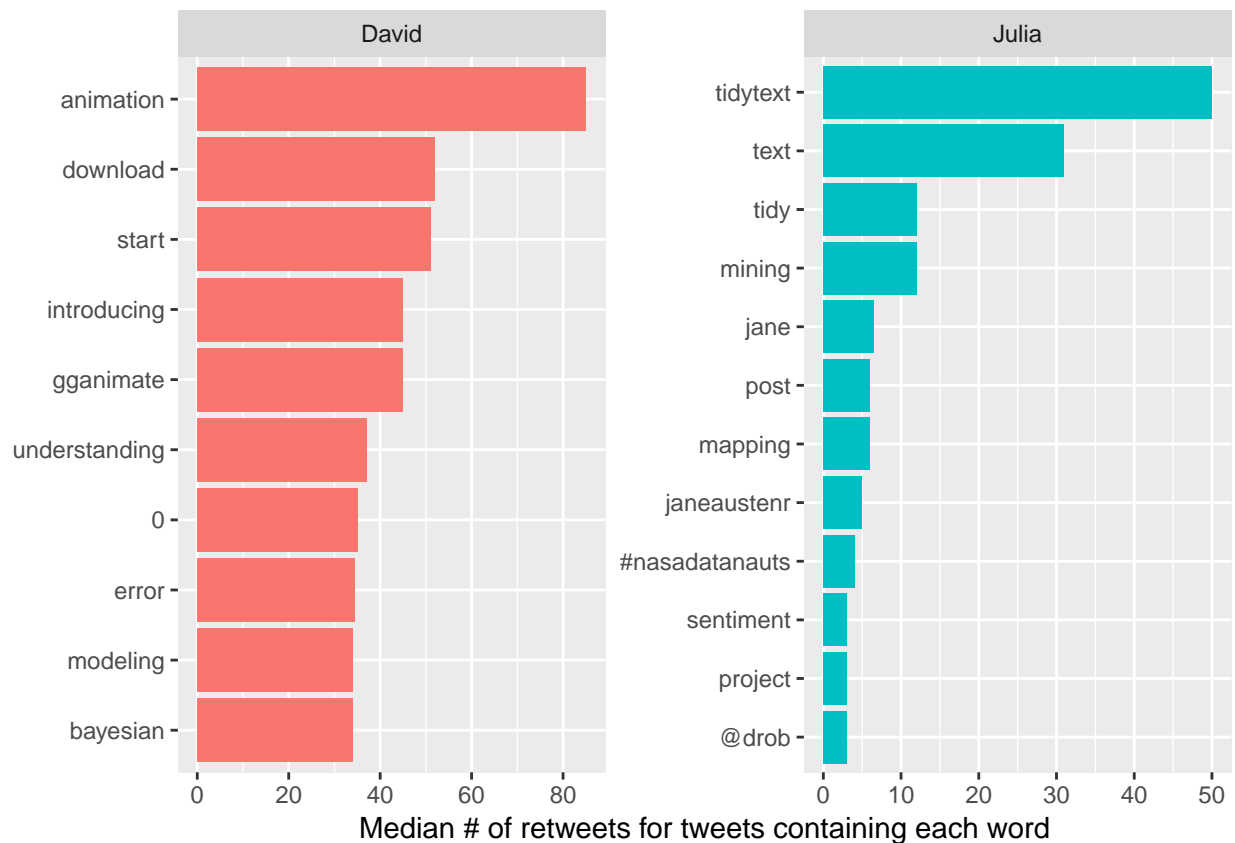
```

word_by_rts %>%
  filter(uses >= 5) %>%
  arrange(desc(retweets))

## # A tibble: 178 x 5
##   person word      retweets  uses total_rts
##   <chr> <chr>      <dbl> <int>    <int>
## 1 David animation    85.0     5    110171
## 2 David download    52.0     5    110171
## 3 David start      51.0     7    110171
## 4 Julia tidytext    50.0     7     12701
## 5 David gganimate    45.0     8    110171
## 6 David introducing  45.0     6    110171
## 7 David understanding 37.0     6    110171
## 8 David 0           35.0     7    110171
## 9 David error       34.5     8    110171
## 10 David bayesian   34.0     7    110171
## # ... with 168 more rows

word_by_rts %>%
  filter(uses >= 5) %>%
  group_by(person) %>%
  top_n(10, retweets) %>%
  arrange(retweets) %>%
  ungroup() %>%
  mutate(word = factor(word, unique(word))) %>%
  ungroup() %>%
  ggplot(aes(word, retweets, fill = person)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ person, scales = "free", ncol = 2) +
  coord_flip() +
  labs(x = NULL,
       y = "Median # of retweets for tweets containing each word")

```



```

totals <- tidy_tweets %>%
  group_by(person, id) %>%
  summarise(favs = sum(favorites)) %>%
  group_by(person) %>%
  summarise(total_favs = sum(favs))

word_by_favs <- tidy_tweets %>%
  group_by(id, word, person) %>%
  summarise(favs = first(favorites)) %>%
  group_by(person, word) %>%
  summarise(favorites = median(favs), uses = n()) %>%
  left_join(totals) %>%
  filter(favorites != 0) %>%
  ungroup()

## Joining, by = "person"

word_by_favs %>%
  filter(uses >= 5) %>%
  group_by(person) %>%
  top_n(10, favorites) %>%
  arrange(favorites) %>%
  ungroup() %>%
  mutate(word = factor(word, unique(word))) %>%
  ungroup() %>%
  ggplot(aes(word, favorites, fill = person)) +
  geom_col(show.legend = FALSE) +

```

```
facet_wrap(~ person, scales = "free", ncol = 2) +
coord_flip() +
labs(x = NULL,
     y = "Median # of favorites for tweets containing each word")
```

