

01-tidy-text.R

mojmir

2020-11-04

```
text <- c(
  "Because I could not stop for Death -",
  "He kindly stopped for me -",
  "The Carriage held but just Ourselves -",
  "and Immortality"
)

text

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

```
library(dplyr)
text_df <- tibble(line = 1:4, text = text)

text_df
```

```
## # 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 token is a meaningful unit of text, most often a word, that we are interested in using for further

```
library(tidytext)

text_df %>%
  unnest_tokens(word, text)
```

```
## # 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
## # ... with 10 more rows
```

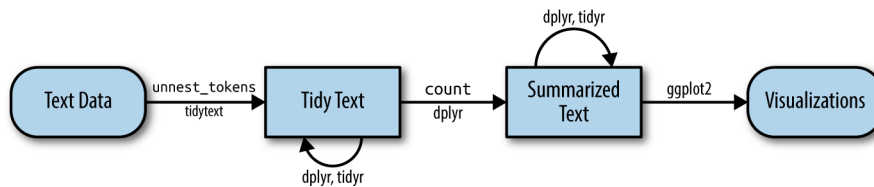


Figure 1: A flowchart of a typical text analysis using tidy data principles. This chapter shows how to summarize and visualize text using these tools.

```
library(janeaustrnr)
library(dplyr)
library(stringr)

original_books <- austen_books() %>%
  group_by(book) %>%
  mutate(
    linenumber = row_number(),
    chapter = cumsum(str_detect(text, regex("^chapter [\\divxlc]",
      ignore_case = TRUE
    )))
  ) %>%
  ungroup()

original_books
```

```
## # 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
```

```
library(tidytext)
tidy_books <- original_books %>%
  unnest_tokens(word, text)

tidy_books
```

```
## # A tibble: 725,055 x 4
##   book          linenumber chapter word
##   <fct>          <int>    <int> <chr>
## 1 Sense & Sensibility      1        0 sense
## 2 Sense & Sensibility      1        0 and
```

```
## 3 Sense & Sensibility      1      0 sensibility
## 4 Sense & Sensibility      3      0 by
## 5 Sense & Sensibility      3      0 jane
## 6 Sense & Sensibility      3      0 austen
## 7 Sense & Sensibility      5      0 1811
## 8 Sense & Sensibility     10      1 chapter
## 9 Sense & Sensibility     10      1 1
## 10 Sense & Sensibility     13      1 the
## # ... with 725,045 more rows
```

```
data(stop_words)
```

```
tidy_books <- tidy_books %>%
  anti_join(stop_words)
```

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

```
## # A tibble: 13,914 x 2
##   word      n
##   <chr> <int>
## 1 miss    1855
## 2 time    1337
## 3 fanny    862
## 4 dear     822
## 5 lady     817
## 6 sir      806
## 7 day      797
## 8 emma     787
## 9 sister   727
## 10 house   699
## # ... with 13,904 more rows
```

```
library(ggplot2)
```

```
tidy_books %>%
  count(word, sort = TRUE) %>%
  filter(n > 600) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```

To learn more about gutenbergr, check out the [package's tutorial at rOpenSci](<https://ropensci.org/>)

```
library(gutenbergr)
```

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

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

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

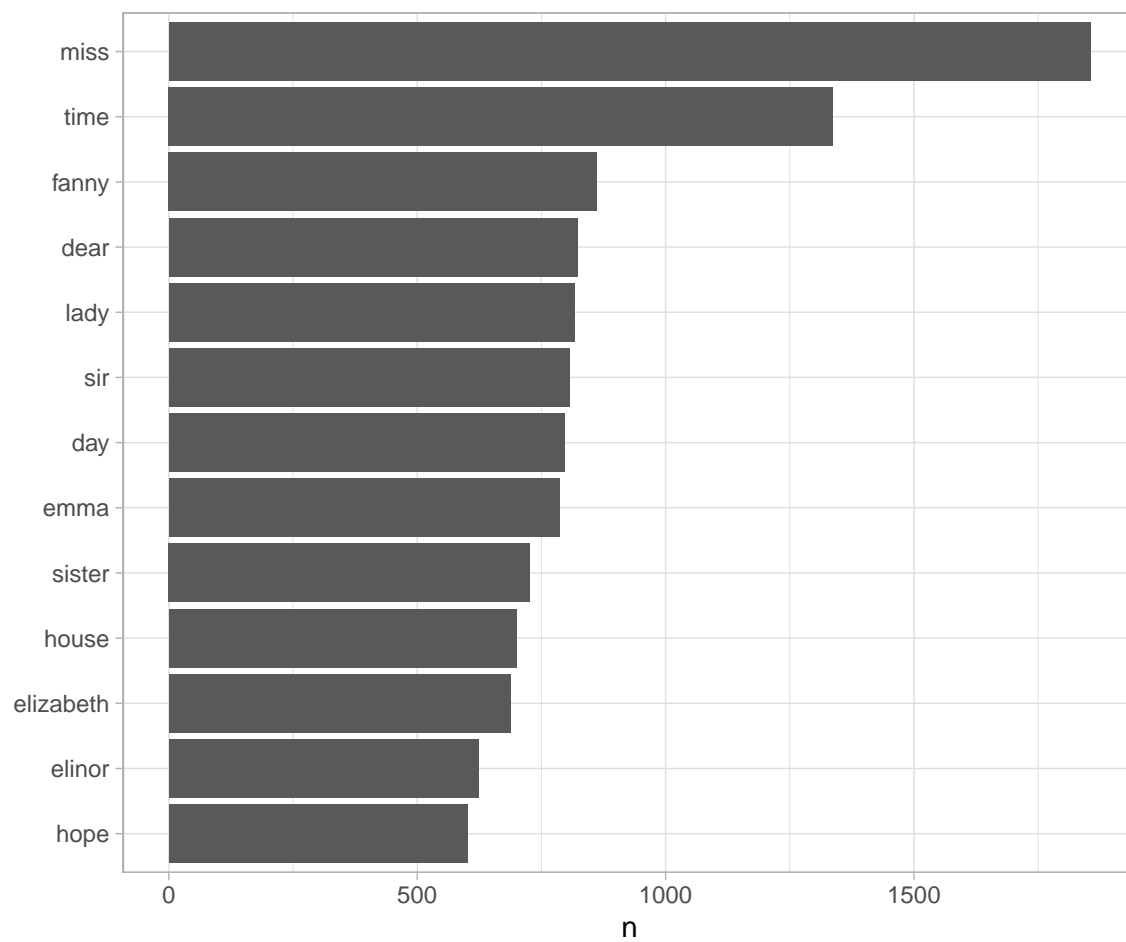


Figure 2: The most common words in Jane Austen's novels

```
## # 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
```

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

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

```
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
```

```
library(tidyr)
```

```
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`)
```

```
library(scales)
```

```
# 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)
```

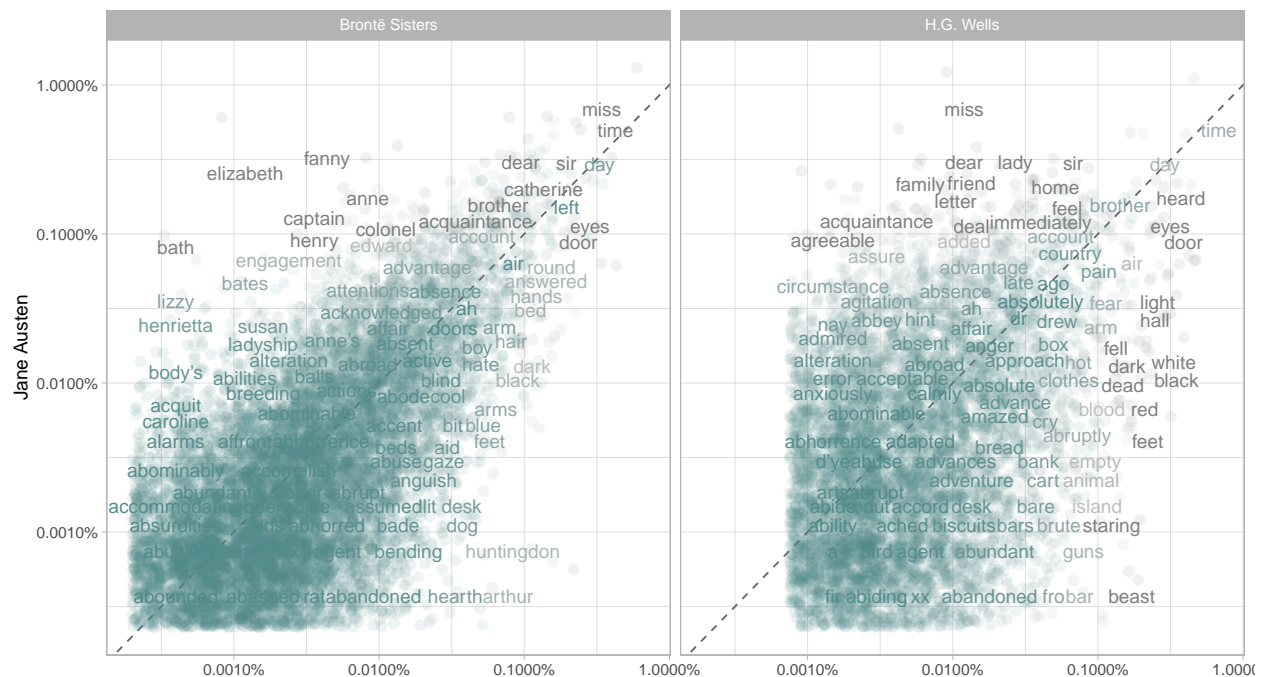


Figure 3: Comparing the word frequencies of Jane Austen, the Brontë sisters, and H.G. Wells

```
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 0.7689642
## 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

```