ST720 Data Science

Tidy Textmining

Seung Jun Shin (sjshin@krea.ac.kr)

Department of Statistics, Korea University

Tidy Text Data

- The tidy text format as being a table with **one-token-per-row**.
- Text data often stored as:
  - **String**: character vectors, within `R`, and often text data is first read into memory in this form.
  - **Corpus**: raw strings annotated with additional metadata and details.
  - **Document-term matrix**: A sparse matrix describing a collection (i.e., a corpus) of documents with one row for each document and one column for each term. The value in the matrix is typically word count or *tf-idf*.

The unnest_tokens function

```
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"
```

The unnest_tokens function

- This is a typical character vector that we might want to analyze. First need to put it into a data frame.

```
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
```

- Not yet compatible with tidy text analysis since each row is made up of multiple combined words.

The unnest_tokens function

- Need to convert to **one-token-per-document-per-row**.
- Need to both break the text into individual tokens (a process called tokenization) and transform it to a tidy data structure.
- use `unnest_tokens()` function.

```
library(tidytext)
text_df %>%
  unnest_tokens(word, text) %>%
  print(n = 5)
```

```
## # A tibble: 20 x 2
##    line word
##   <int> <chr>
## 1     1 because
## 2     1 i
## 3     1 could
## 4     1 not
## 5     1 stop
## # ... with 15 more rows
```

Tidy Textmining

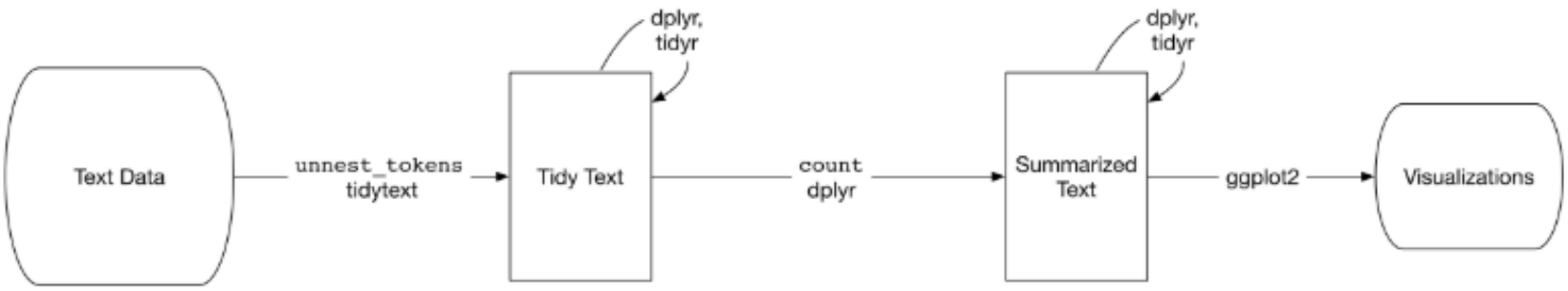


Figure 1: A flowchart of a typical text analysis using tidy data principles.

Tidying the works of Jane Austen

- Jane Austen's 6 completed, published novels from the `janeaustenr` package.

```
library(janeaustenr)
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()
```

Tidying the works of Jane Austen

```
## # 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
```

Tidying the works of Jane Austen

- Restructure it in the **one-token-per-row** format.

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

print(tidy_books, n = 5)
```

```
## # 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
## # ... with 7.250e+05 more rows
```

Tidying the works of Jane Austen

- The default tokenizing is for words, but other options include characters, n-grams, sentences, lines, paragraphs, or separation around a regex pattern.
- Often in text analysis, we will want to remove stop words

```
data(stop_words)

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

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

Tidying the works of Jane Austen

- We can also use dplyr's `count()` to find the most common words.

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

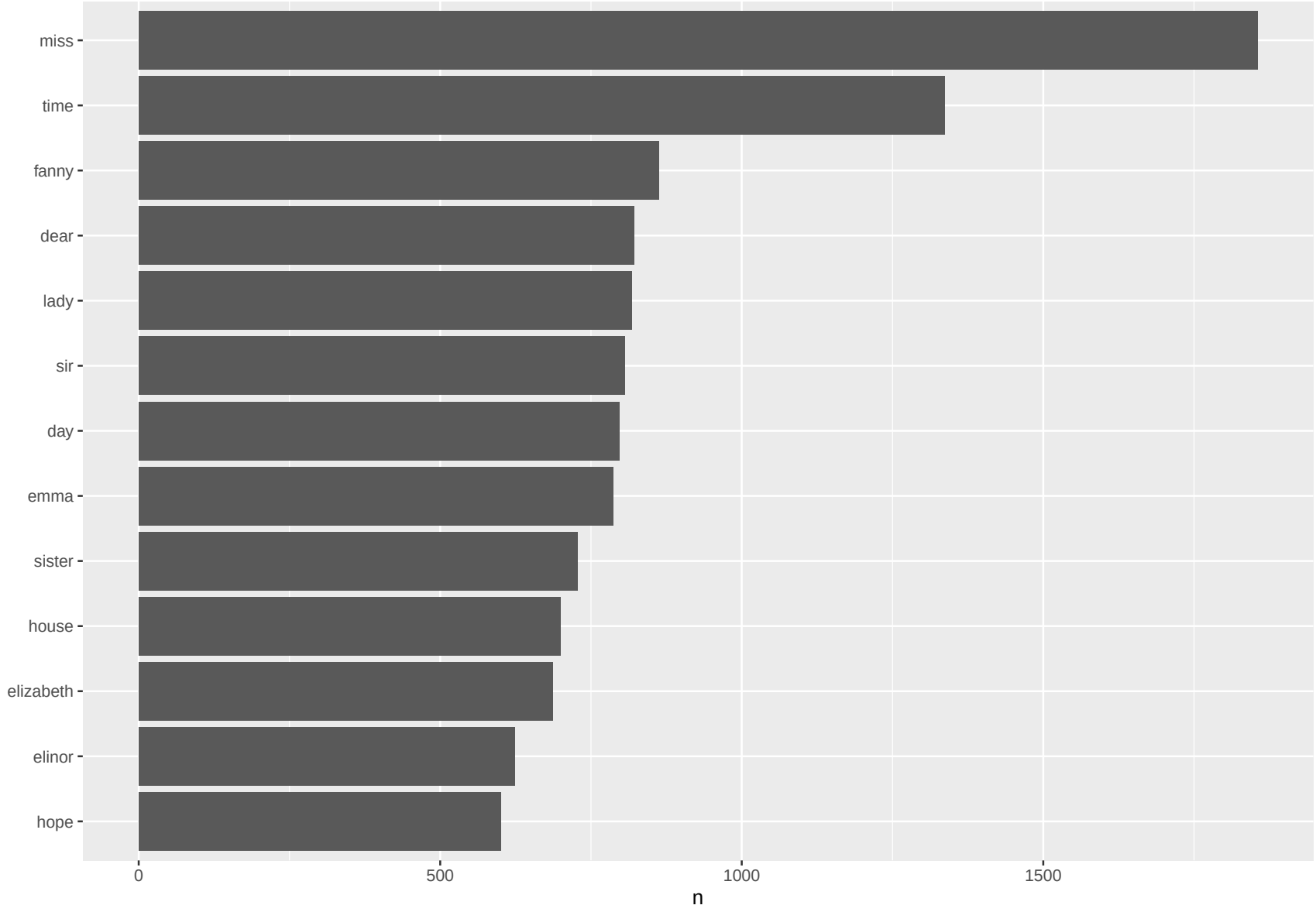
```
## # A tibble: 13,914 x 2
##    word       n
##    <chr>  <int>
##  1 miss    1855
##  2 time    1337
##  3 fanny    862
##  4 dear     822
##  5 lady     817
##  6 sir      806
##  7 day      797
##  8 emma     787
##  9 sister   727
## 10 house    699
## # ... with 13,904 more rows
```

Tidying the works of Jane Austen

- Let's visualize it.

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

Tidying the works of Jane Austen



The gutenbergr package

- Provides access to the public domain works from the Project Gutenberg collection.
- Includes tools both for downloading books (stripping out the unhelpful header/footer information), and a complete dataset of Project Gutenberg metadata that can be used to find works of interest.
- mostly use the function `gutenberg_download()` that downloads one or more works from Project Gutenberg by ID, but you can also use other functions to explore metadata, pair Gutenberg ID with title, author, language, etc., or gather information about authors.

Word frequencies

- Some science fiction and fantasy novels by H.G. Wells
  - The Time Machine,
  - The War of the Worlds,
  - The Invisible Man,
  - The Island of Doctor Moreau.

```
library(gutenbergr)

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

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

Word frequencies

- What are the most common words in these novels?

```
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
```

Word frequencies

- Do the same thing for Brontë sisters.

```
bronte <- gutenberg_download(c(1260, 768, 969, 9182, 767))
tidy_bronte <- bronte %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words)
```

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

Word frequencies

- Let's calculate the frequency for each word for the works of Jane Austen, the Brontë sisters, and H.G. Wells.

```
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`)
```

Word frequencies

```
frequency
```

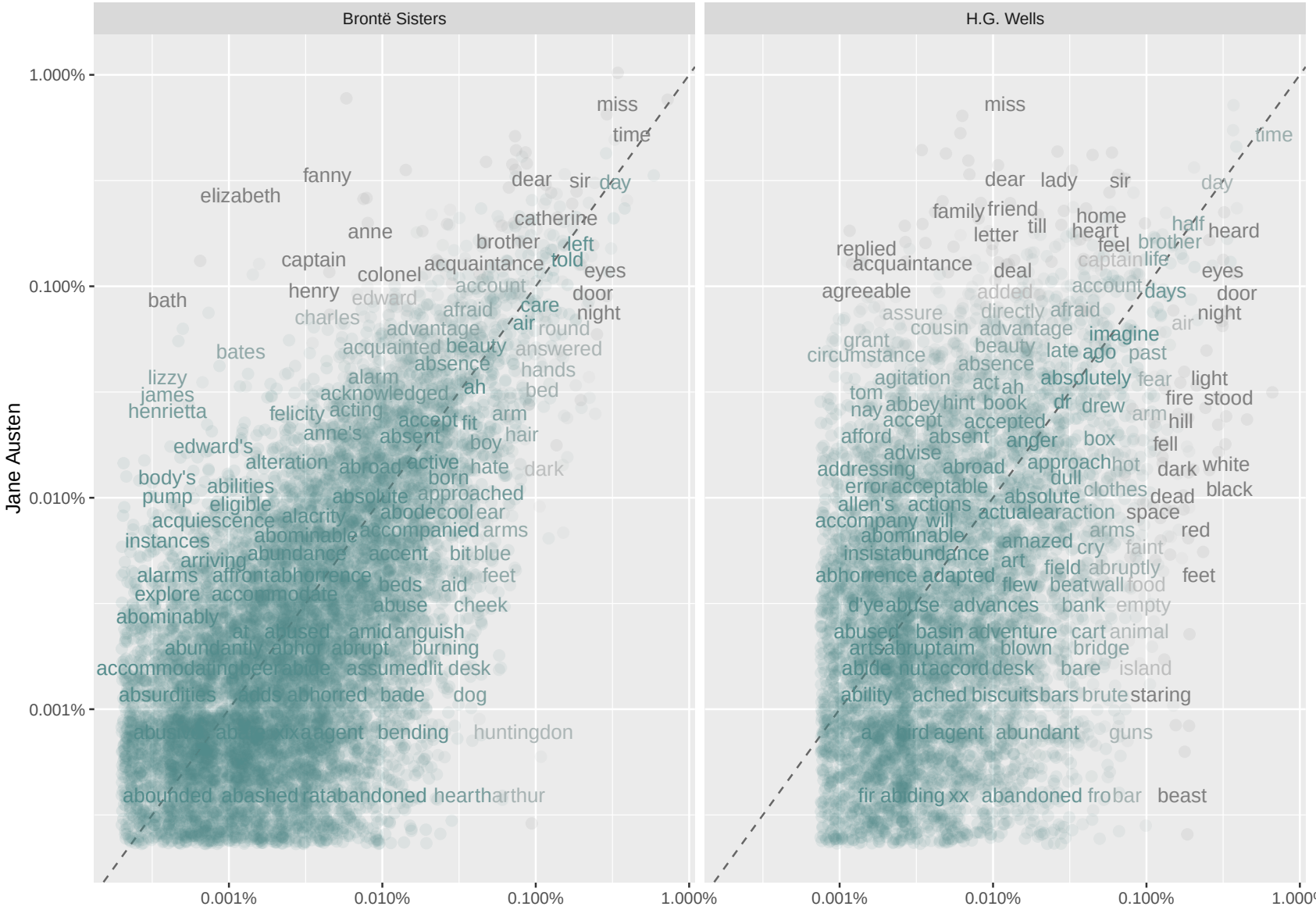
```
## # A tibble: 57,818 x 4
##    word        `Jane Austen` author           proportion
##    <chr>               <dbl> <chr>                 <dbl>
##  1 a              0.00000919 Brontë Sisters  0.0000319
##  2 a'most        NA          Brontë Sisters  0.0000159
##  3 a'n't          0.00000460 Brontë Sisters NA
##  4 aback         NA          Brontë Sisters  0.00000398
##  5 abaht         NA          Brontë Sisters  0.00000398
##  6 abandon       NA          Brontë Sisters  0.0000319
##  7 abandoned      0.00000460 Brontë Sisters  0.0000916
##  8 abandoning    NA          Brontë Sisters  0.00000398
##  9 abandonment   NA          Brontë Sisters  0.0000199
## 10 abart         NA          Brontë Sisters NA
## # ... with 57,808 more rows
```

Word frequencies

```
library(scales)

# expect a warning about rows with missing values being removed
ggplot(frequency, aes(x = proportion, y = `Jane Austen`, color = abs(`Jane Aust
  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 = "gra
  facet_wrap(~author, ncol = 2) +
  theme(legend.position="none") +
  labs(y = "Jane Austen", x = NULL)
```

Word frequencies



Word frequencies

- Words that are close to the line in these plots have similar frequencies in both sets of texts.
  - Austen and Brontë: “miss”, “time”, “day” at the upper frequency end,
  - Austen and Wells: “time”, “day”, “brother” at the high frequency end.
- Words that are far from the line are words that are found more in one set of texts than another.
  - “elizabeth”, “emma”, and “fanny” are found in Austen’s texts but not much in the Brontë texts
  - “arthur” and “dog” are found in the Brontë texts but not the Austen texts.
  - Wells uses words like “beast”, “guns”, “feet”, and “black” that Austen does not
  - Austen uses words like “family”, “friend”, “letter”, and “dear” that Wells does not.

Word frequencies

- Austen and the Brontë sisters use more similar words than Austen and H.G. Wells.
- Not all the words are found in all three sets of texts and there are fewer data points in the panel for Austen and H.G. Wells.

Correlation

- Quantify how similar and different these sets of word frequencies are using a **correlation test**.

```
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
```

Correlation

```
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
```

- Word frequencies are more correlated between the Austen and Brontë novels than between Austen and H.G. Wells.

Summary

- We explored what we mean by tidy data when it comes to text, and how tidy data principles can be applied to natural language processing.
- When text is organized in a format with one token per row, tasks like removing stop words or calculating word frequencies are natural applications of familiar operations within the tidy tool ecosystem.
- The one-token-per-row framework can be extended from single words to n-grams and other meaningful units of text, as well as to many other analysis priorities that we will consider later.