01 The Tidy Text Format

In this project, I want to explore in depth what is behind the tidy text format and I want to analyze how this format can be used to approach questions about word frequency. This will allow me to analyze which words are used most frequently in documents and to compare documents.

01_01 The unnest_tokes Function

As an example, some of Emily Dickinson's texts will be used.

```
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"
```

This is a typical character vector that I might want to analyze. In order to turn it into a tidy text dataset, I first need to put it into a data frame.

```
text_df <- data_frame(line = 1:4, text = text)

text_df
```

```
## # A tibble: 4 × 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
```

In the first step, I have the poem as one document, but I want to explore examples with multiple documents. So within my tidy text framework, I need to both break the text into indicidual tokens (a process called tokenization) and transform it to a tidy data structure.

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

```
## # A tibble: 20 × 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
```

First I have the output column name that will be created as the text is unnested into it (**word**, in this case), and then the input column that the text comes from (**text**, in this case). So I could split the poem into its words.

01_02 Tidying the Works of Jane Austen

Further, I use the text of Jane Austen's 6 completed, published novels from the janeaustenr package (Silge 2016), and transform them into a tidy format.

```
## # A tibble: 73,422 × 4
##                      text                book linenumber chapter
##                     <chr>              <fctr>      <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
```

To work with this as a tidy dataset, I need to restructure it in the **one-token-per-row format**. A token is a meaningful unit of text, most often a word, that we are interested in using for further analysis, and tokenization is the process of splitting text into tokens.

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

tidy_books
```

```
## # A tibble: 725,054 × 4
##                   book linenumber chapter        word
##                 <fctr>      <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 725,044 more rows
```

Now that the data is in one-word-per-row format, I can manipulate it with tidy tools like dplyr. So in the next step I want to remove stop words, which words that are not useful for an analysis, typically extremely common words such as “the”, “of”, “to”, and so forth in English.

```
data(stop_words)

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

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

After I remove the stop words, I want to finde the most common words in all the books as a whole.

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

```
## # A tibble: 13,914 × 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
```

The word counts are stored in a tidy data frame, because I have been using tidy tools. So this allows me for example to create a visualization of the most common words.

```
tidy_books %>%
  count(word, sort = TRUE) %>%
  filter(n > 600) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill=word)) +
  geom_col() +
  xlab(NULL) +
  ylab("frequency n") +
  coord_flip() +
  ggtitle("The most common Words of Jane Austen's 6 completed, published Novels") +
     theme(plot.title = element_text(lineheight=.8, face="bold"))
```

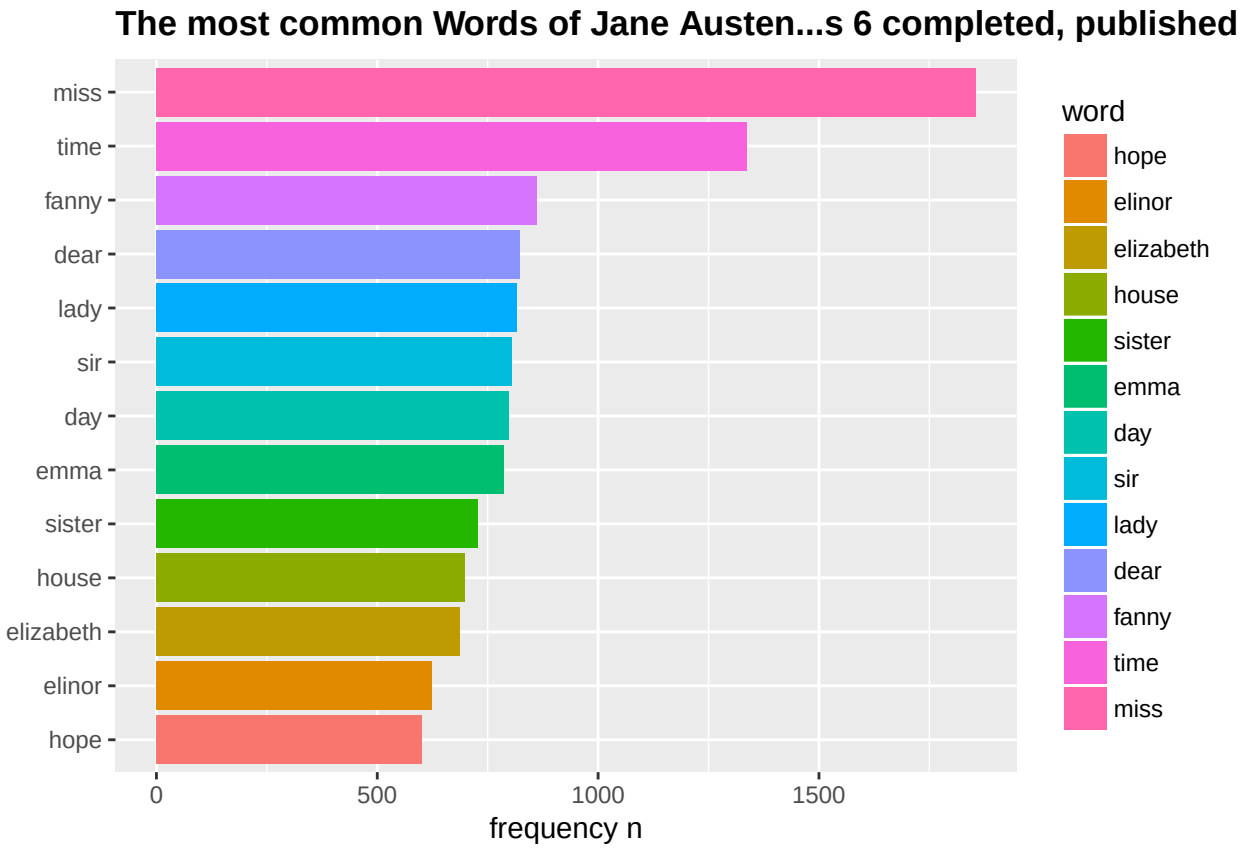
```
## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)): Fontmetrik
## für das Zeichen 0x1e unbekannt
```

```
## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)): Fontmetrik
## für das Zeichen 0x80 unbekannt
```

```
## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)):
```

```
## Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <e2>

## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)):
## Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <80>

## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)):
## Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <99>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <e2>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <80>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <99>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <e2>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <80>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <99>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <e2>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <80>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <99>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <e2>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <80>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <99>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen’s 6
```

```
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <e2>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen's 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <80>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen's 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <99>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen's 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <e2>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen's 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <80>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen's 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <99>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen's 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <e2>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen's 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <80>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen's 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <99>

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen's 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <e2>

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen's 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <80>

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : Konvertierungsfehler für 'The most common Words of Jane Austen's 6
## completed, published Novels' in 'mbcsToSbcs': Punkt ersetzt <99>
```



01_03 The Gutenbergr Package

After I have used the janeaustenr package to explore tidying text, I want to introduce the gutenbergr package (Robinson 2016). The gutenbergr package provides access to the public domain works from the Project Gutenbergcollection.

```
gutenberg_metadata %>%
  filter(title == "Wuthering Heights")
```

```
## # A tibble: 1 × 8
##   gutenberg_id             title        author gutenberg_author_id
##          <int>             <chr>         <chr>               <int>
## 1          768 Wuthering Heights Brontë, Emily                 405
## # ... with 4 more variables: language <chr>, gutenberg_bookshelf <chr>,
## #   rights <chr>, has_text <lgl>
```

01_04 Word Frequencies

A common task in text mining is to look at word frequencies, just like I have done above for Jane Austen's novels, and to compare frequencies across different texts. I can do this intuitively and smoothly using tidy data principles. I already have Jane Austen's works and now I will get two more sets of texts to compare to. First, let's look at some science fiction and fantasy novels by H.G. Wells, who lived in the late 19th and early 20th centuries.

I choose following novels and in brackets are there Project Gutenberg ID number:

- The Time Machine (35)
- The War of the Worlds (36)
- The Invisible Man (5230)
- The Island of Doctor Moreau (159)

```
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
```

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

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

And I analyze the most common words in these novels of H.G. Wells.

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

```
## # A tibble: 11,769 × 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
```

I took as second author, some well-known works of the Brontë sisters, whose lives overlapped with Jane Austen's somewhat but who wrote in a rather different style.

I choose following novels and in brackets are there Project Gutenberg ID number:

- Jane Eyre
- Wuthering Heights
- The Tenant of Wildfell Hall
- Villette
- Agnes Grey

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

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

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

And I analyze the most common words in these novels of the Brontë sisters.

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

```
## # A tibble: 23,050 × 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   602
## 9    door   592
## 10   half   586
## # ... with 23,040 more rows
```

Interesting that "time", "eyes", and "hand" are in the top 10 for both H.G. Wells and the Brontë sisters.

Now, I calculate the frequency for each word for the works of Jane Austen, the Brontë sisters, and H.G. Wells by binding the data frames together.

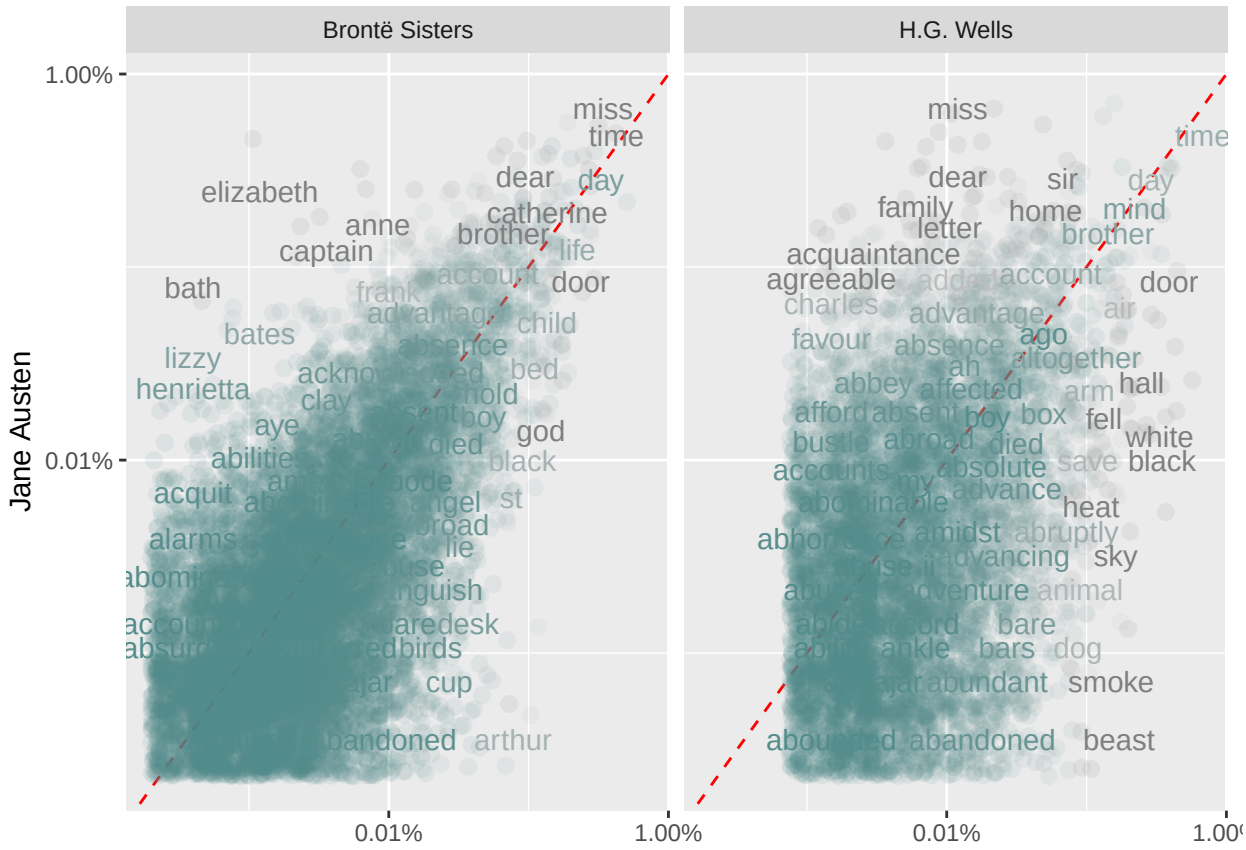
```
frequency <- bind_rows(mutate(tidy_bronte, author = "Brontë Sisters"),
                       mutate(tidy_hgwells, author = "H.G. Wells"),
                       mutate(tidy_books, author = "Jane Austen")) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  count(author, word) %>%
  group_by(author) %>%
  mutate(proportion = n / sum(n)) %>%
  select(-n) %>%
  spread(author, proportion) %>%
  gather(author, proportion, `Brontë Sisters`:`H.G. Wells`)
```

I use str_extract() here because the UTF-8 encoded texts from Project Gutenberg have some examples of words with underscores around them to indicate emphasis (like italics). The tokenizer treated these as words, but I don't want to count "*any*" separately from "any".

```
ggplot(frequency, aes(x = proportion, y = `Jane Austen`, color = abs(`Jane Austen` - proportion))) +
  geom_abline(color = "red", 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).
```



Words that are close to the line in these plots have similar frequencies in both sets of texts, for example, in both Austen and Brontë texts ("miss", "time", "day" at the upper frequency end) or in both Austen and Wells texts ("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. For example, in the Austen-Brontë panel, words like "elizabeth", "emma", and "fanny" (all proper nouns) are found in Austen's texts but not much in the Brontë texts, while words like "arthur" and "dog" are found in the Brontë texts but not the Austen texts. In comparing H.G. Wells with Jane Austen, Wells uses words like "beast", "guns", "feet", and "black" that Austen does not, while Austen uses words like "family", "friend", "letter", and "dear" that Wells does not.

Overall, I notice that the words in the Austen-Brontë panel are closer to the zero-slope line than in the Austen-Wells panel. Also notice that the words extend to lower frequencies in the Austen-Brontë panel; there is empty space in the Austen-Wells panel at low frequency. These characteristics indicate that Austen and the Brontë sisters use more similar words than Austen and H.G. Wells. Also, we see that 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.

Let's quantify how similar and different these sets of word frequencies are using a correlation test. How correlated are the word frequencies between Austen and the Brontë sisters, and between Austen and Wells?

```
cor.test(data = frequency[frequency$author == "Brontë Sisters",],
         ~ proportion + `Jane Austen`)
```

```
##
##  Pearson's product-moment correlation
##
## data:  proportion and Jane Austen
## t = 119.64, df = 10404, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7527837 0.7689611
```

```
## sample estimates:
##       cor
## 0.7609907
```

```
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:
##  0.4032820 0.4446006
## sample estimates:
##      cor
## 0.424162
```

Just as we saw in the plots, the word frequencies are more correlated between the Austen and Brontë novels than between Austen and H.G. Wells.

Summary

I explored what is 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.