Solution slides Part 3 Introduction to R & Data for Humanities

Afternoon session

Text-mining with Tidyverse

Exercise 10

10a.

Note that the usual suspects are here with the highest n, “the”, “and”, “to”, and so forth.

```r
# Exercise 10a.Let's start by looking at the novels of Austen and examine first term frequency, then tf-idf. We can start just by using dplyr verbs such as group_by() and join(). Can you fill in the blanks in the code below based on what you have learned so far and determine the most commonly used words in the novels? (Let's also calculate the total words in each novel here, for later use)

library(dplyr)
library(janeaustenr)
library(tidytext)

book_words <- austen_books() %>%
  unnest_tokens(word, text) %>%
  count(book, word, sort = TRUE)

total_words <- book_words %>%
  group_by(book) %>%
  summarize(total = sum(n))

book_words <- left_join(book_words, total_words)

book_words
```

R Console

tbl_df
40379 x 4

book <fctr>	word <chr>	n <int>	total <int>
Mansfield Park	the	6206	160460
Mansfield Park	to	5475	160460
Mansfield Park	and	5438	160460
Emma	to	5239	160996
Emma	the	5201	160996
Emma	and	4896	160996
Mansfield Park	of	4778	160460
Pride & Prejudice	the	4331	122204
Emma	of	4291	160996
Pride & Prejudice	to	4162	122204

1-10 of 40,379 rows

Previous 1 2 3 4 5 6 ... 100 Next

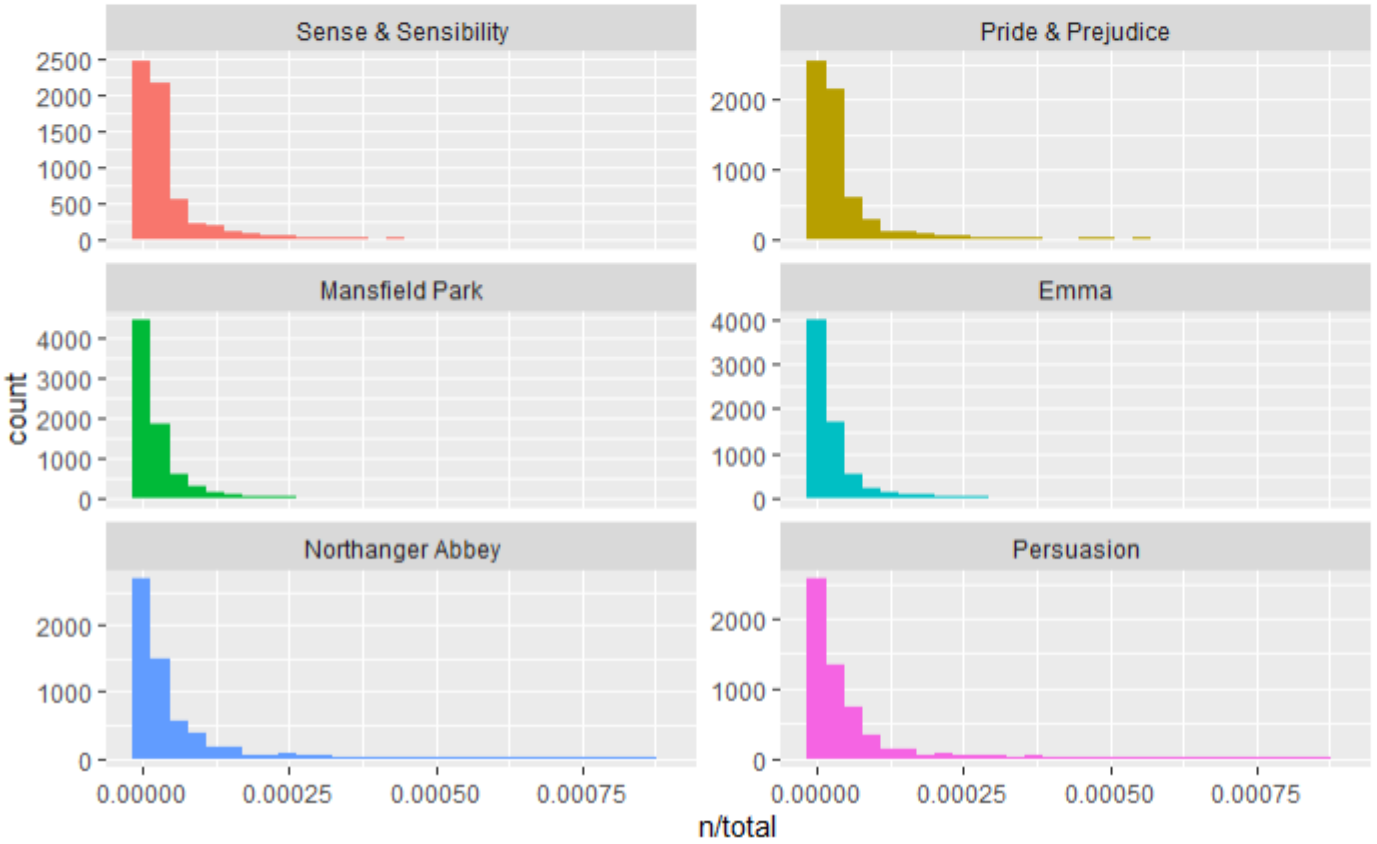
10b.

```r
# Exercise 10b. Now let's plot the distribution of n/total = the number of times a word is used in a book/the total words in that book. Do you remember what package to call on to plot this distribution?

library(ggplot2)

ggplot(book_words, aes(n/total, fill = book)) +
  geom_histogram(show.legend = FALSE) +
  xlim(NA, 0.0009) +
  facet_wrap(~book, ncol = 2, scales = "free_y")
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
Removed 896 rows containing non-finite values (stat_bin).
Removed 6 rows containing missing values (geom_bar).
```



← These plots exhibit similar distributions for all the novels, with many words that occur rarely and fewer words that occur frequently.

10c.

```r
# Exercise 10c.Based on the column headers on the slide, can you fill in the code below and calculate tf-idf?


book_tf_idf <- book_words %>%
  bind_tf_idf(word, book, n)


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

book <fctr>	word <chr>	n <int>	tf <dbl>	idf <dbl>	tf_idf <dbl>
Sense & Sensibility	elinor	623	5.193528e-03	1.7917595	9.305552e-03
Sense & Sensibility	marianne	492	4.101470e-03	1.7917595	7.348847e-03
Mansfield Park	crawford	493	3.072417e-03	1.7917595	5.505032e-03
Pride & Prejudice	darcy	373	3.052273e-03	1.7917595	5.468939e-03
Persuasion	elliot	254	3.036171e-03	1.7917595	5.440088e-03
Emma	emma	786	4.882109e-03	1.0986123	5.363545e-03
Northanger Abbey	tilney	196	2.519928e-03	1.7917595	4.515105e-03
Emma	weston	389	2.416209e-03	1.7917595	4.329266e-03
Pride & Prejudice	bennet	294	2.405813e-03	1.7917595	4.310639e-03
Persuasion	wentworth	191	2.283105e-03	1.7917595	4.090775e-03

1-10 of 40,379 rows

Previous 1 2 3 4 5 6 ... 100 Next

Here we see all proper nouns, names that are in fact important in these novels. None of them occur in all of the novels, and they are important, characteristic words for each text within the corpus of Jane Austen's novels.

10d.

```r
# Excercise 10d.Run the code below to plot the highest tf-idf words in each of Austen's novels. Can you make it so that you plot the scores per novel? And can you make sure that we see tf-idf for the tokens/terms we have been analyzing?

library(forcats)

book_tf_idf %>%
  group_by(book) %>%
  slice_max(tf_idf, n = 8) %>%
  ungroup() %>%
  ggplot(aes(tf_idf, fct_reorder(word, tf_idf), fill = book)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~book, ncol = 2, scales = "free") +
  labs(x = "tf-idf", y = NULL)
```