

Sense and Sensibility Wordcloud

Charles Redmond

October 30, 2017

Abstract

In this article we construct a wordcloud, using the tidytext R package, for Jane Austen's *Sense and Sensibility*.

Sense and Sensibility is a novel by Jane Austen, published in 1811¹. Below we construct a wordcloud for the most common words appearing in the novel.

1 The Jane Austen Package

There is a relatively new package for R, `janeaustenr`, that gives one access to all of the novels written by Jane Austen (Silge and Robinson, 2017). One first has to install this package and bring it in with `library`. You may then call the following function and store the result. The result will be a data frame.

```
library(janeaustenr)
sns<-austen_books()
```

This dataframe has two columns, one for each line in Austen's novels, and one indicating which book the line is from. Let's first filter, using `dplyr`, so that we have only the lines from *Sense and Sensibility*:

```
library(dplyr)
sns<-sns%>%
  filter(book == 'Sense & Sensibility')
head(sns)

## # A tibble: 6 x 2
##           text          book
##           <chr>      <fctr>
## 1 SENSE AND SENSIBILITY Sense & Sensibility
## 2                      Sense & Sensibility
## 3      by Jane Austen Sense & Sensibility
## 4                      Sense & Sensibility
## 5              (1811) Sense & Sensibility
## 6                      Sense & Sensibility
```

¹The novel was published anonymously.

Now we are ready for some data cleaning.

2 Some Data Cleaning

We would like to remove all of the ‘Chapter’ lines. We can use dplyr again, along with the package stringr.

```
library(stringr)
sns<-sns%>%
  filter(!str_detect(sns$text, '^CHAPTER'))
```

Next, we would like to remove the front matter. By inspection, we have determined that the front matter ends on line 11. Therefore we can redefine sns to begin on line 12:

```
sns<-sns[12:12574,]
```

3 The Wordcloud

To make the wordcloud, we first have to break up the lines into words. We can use a function from the tidytext package for this:

```
library(tidytext)
words_df<-sns%>%
  unnest_tokens(word, text)

words_df

## # A tibble: 119,850 x 2
##       book      word
##       <fctr>   <chr>
## 1 Sense & Sensibility the
## 2 Sense & Sensibility family
## 3 Sense & Sensibility of
## 4 Sense & Sensibility dashwood
## 5 Sense & Sensibility had
## 6 Sense & Sensibility long
## 7 Sense & Sensibility been
## 8 Sense & Sensibility settled
## 9 Sense & Sensibility in
## 10 Sense & Sensibility sussex
## # ... with 119,840 more rows
```

We can remove common, unimportant words with the stop_words data frame and some dplyr:

```
words_df<-words_df%>%
  filter(!(word %in% stop_words$word))

words_df

## # A tibble: 36,225 x 2
##       book      word
##   <fctr>   <chr>
## 1 Sense & Sensibility family
## 2 Sense & Sensibility dashwood
## 3 Sense & Sensibility settled
## 4 Sense & Sensibility sussex
## 5 Sense & Sensibility estate
## 6 Sense & Sensibility residence
## 7 Sense & Sensibility norland
## 8 Sense & Sensibility park
## 9 Sense & Sensibility centre
## 10 Sense & Sensibility property
## # ... with 36,215 more rows
```

Now, we need to calculate the frequencies of the words in the novel. Again, we can use standard dplyr techniques for this:

```
word_freq<-words_df%>%
  group_by(word)%>%
  summarize(count=n())

word_freq

## # A tibble: 5,844 x 2
##       word count
##   <chr>   <int>
## 1      1      1
## 2     200      1
## 3    70001      1
## 4 abandoned      1
## 5 abatement      1
## 6 abbeyland      1
## 7    abhor      1
## 8  abhorred      2
## 9 abhorrence      4
## 10 abilities      9
## # ... with 5,834 more rows
```

Finally, it's time to generate the wordcloud:

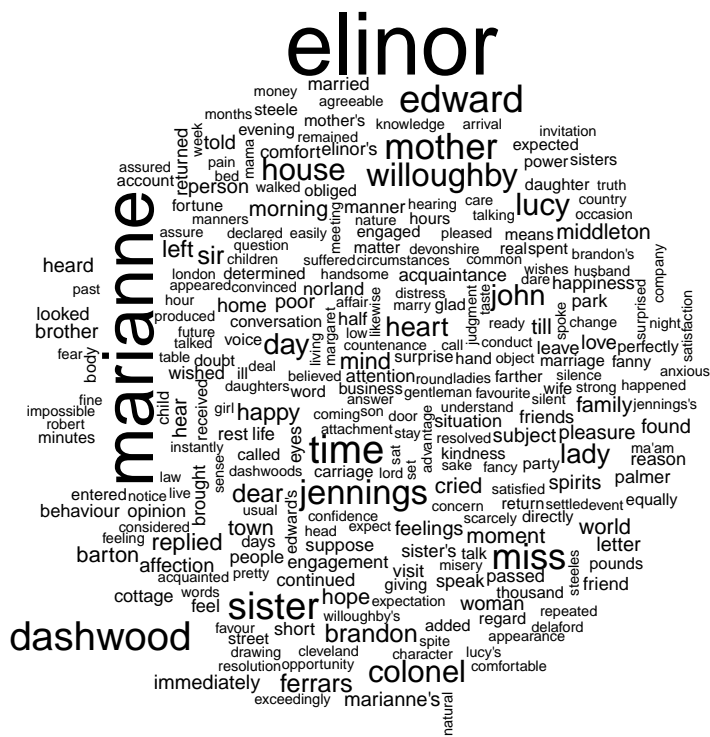
```
library(wordcloud)

## Loading required package: RColorBrewer

library(tm)

## Loading required package: NLP

wordcloud(word_freq$word,word_freq$count,min.freq=25)
```



References

- Silge, J. and Robinson, D. (2017). *Text Mining with R: A Tidy Approach*. O'Reilly Media.
- Wickham, H. and Golemund, G. (2017). *R for Data Science: Import, Tidy, Transform, and Model Data*. O'Reilly Media.