Sense and Sensibility Wordcloud

Bill Fisher

October 27, 2017

Abstract

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

Sense and Sensibility was written and published in 1811 by Jane Austen. Below we construct a word cloud 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 acess to all of the novels written by Jane Austen. One first has to install this package and bring it in with the library function. You may then call the following function and store the result. The result will be a dataframe.

```
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
```

Now we are ready to clean the data.

2 Data Cleaning

We would like to remove all of the 'Chapter' lines. We can use dplyr again, along with 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 dataframe 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
```