Frankenstein's Wordcloud

Bill Fisher

November 1, 2017

Abstract

In this article, we construct a wordcloud using the tidytext R package for Mary Wollstonecraft (Godwin) Shelley's Frankenstein, or the Modern Prometheus.

Frankenstein, or the Modern Prometheus (or simply, Frankenstein for short), is a novel written by English author Mary Shelley (1797-1851) that tells the story of Victor Frankenstein, a young scientist who creates a grotesque but sapient creature in an unorthodox scientific experiment. Shelley started writing the story when she was 18, and the first edition of the novel was published in London in 1818, when she was 20.[1] The book was later published around the world.(contributors)

1 The GutenbergR package

There is a package called GutenbergR whichholds the text of many novels to be deciphered using R. One first has to install this package and bring it in with the library function. Once loaded, we will find our book and download it.[2] Once downloaded, we can store it in our dataframe.

```
library(gutenbergr)
frankenstein<-gutenberg_download(84)
frankenstein

## # A tibble: 7,244 x 2
##    gutenberg_id                                text
##           <int>                               <chr>
##  1           84                          Frankenstein,
##  2           84
##  3           84             or the Modern Prometheus
##  4           84
```

[1] The novel was originally published anonymously. Her name first appeared on the second edition, published in France in 1823.

[2] You can search the database for the title of Frankenstein using the following codeline: gutenberg_works(str_detect(title,'Frankenstein'))

```
##  5           84
##  6           84                                    by
##  7           84
##  8           84 Mary Wollstonecraft (Godwin) Shelley
##  9           84
## 10           84
## # ... with 7,234 more rows
```

The dataframe, frankenstein, now contains our book but we still need to clean up the data. Breaking it down into two major steps, we need to clear the clutter at the beginning of the book and we need to erase the word "chapter" from the beginning of each chapter. To accomplish this, we run the following code:

```
library(stringr)
library(dplyr)

frankenstein<-frankenstein[12:7244,]
frankenstein
```

```
## # A tibble: 7,233 x 2
##    gutenberg_id
##           <int>
##  1           84
##  2           84
##  3           84
##  4           84
##  5           84
##  6           84
##  7           84
##  8           84
##  9           84
## 10           84
## # ... with 7,223 more rows, and 1 more variables: text <chr>
```

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

```
## # A tibble: 7,233 x 2
##    gutenberg_id
##           <int>
##  1           84
##  2           84
##  3           84
##  4           84
##  5           84
```

```
##  6           84
##  7           84
##  8           84
##  9           84
## 10           84
## # ... with 7,223 more rows, and 1 more variables: text <chr>
```

Above, we change the start of our dataframe at the 12th line and include that and the last line. We also use dplyr and the string detect function to elimintae all of the "chapters."
Next, we need to seperate our words. We also need to discard the stop words. To do this we run the following code:

```
library(tidytext)

words_df<-frankenstein%>%
  unnest_tokens(word,text)
words_df

## # A tibble: 75,165 x 2
##    gutenberg_id        word
##           <int>       <chr>
##  1           84      letter
##  2           84           1
##  3           84          st
##  4           84 petersburgh
##  5           84         dec
##  6           84        11th
##  7           84          17
##  8           84          to
##  9           84         mrs
## 10           84     saville
## # ... with 75,155 more rows

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

## # A tibble: 27,274 x 2
##    gutenberg_id        word
##           <int>       <chr>
##  1           84      letter
##  2           84           1
##  3           84          st
##  4           84 petersburgh
##  5           84         dec
```

```
##  6           84        11th
##  7           84          17
##  8           84     saville
##  9           84     england
## 10           84     rejoice
## # ... with 27,264 more rows
```

Now that we have our words seperated, we need to get a count of each word so we can finish constructing our wordcloud. To do this, we will once again use dplyr.

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

word_freq
```

```
## # A tibble: 6,561 x 2
##     word count
##    <chr> <int>
##  1   _i_     1
##  2     1     2
##  3    10     1
##  4    11     1
##  5  11th     2
##  6    12     1
##  7  12th     2
##  8    13     1
##  9  13th     1
## 10    14     1
## # ... with 6,551 more rows
```

Finally, we can construct our wordcloud.

```
library(wordcloud)
wordcloud(word_freq$word,word_freq$count,min.freq=27)
```

If we want, we can look at a certain sentiment's wordcloud. For instance, using tidytext to define each word's sentiment, we can make a wordcloud out of each of Frankenstein's "fear" words with the following piece of code:

```
nrc<-get_sentiments('nrc')

nrc_fear<-nrc%>%
  filter(sentiment=='fear')

frankenstein_words_df<-inner_join(nrc_fear,words_df)

frank_fear_words<-frankenstein_words_df%>%
  group_by(word)%>%
  summarize(count=n())
frank_fear_words

## # A tibble: 493 x 2
```

```
##         word count
##        <chr> <int>
##  1   abandon     2
##  2 abandoned     3
##  3     abhor     5
##  4 abhorrent     1
##  5  abortion     1
##  6   absence     8
##  7     abyss     1
##  8  accident     6
##  9  accursed     4
## 10   accused     7
## # ... with 483 more rows
```

```
wordcloud(frank_fear_words$word,frank_fear_words$count,min.freq=5)
```

References

contributors, W. Frankenstein.

Feinerer, I. and Hornik, K. (2017). *tm: Text Mining Package*. R package version 0.7-1.

Fellows, I. (2014). *wordcloud: Word Clouds*. R package version 2.5.

Robinson, D. (2017). *gutenbergr: Download and Process Public Domain Works from Project Gutenberg*. R package version 0.1.3.

Robinson, D. and Silge, J. (2017). *tidytext: Text Mining using 'dplyr', 'ggplot2', and Other Tidy Tools*. R package version 0.1.4.

silge, J. and Robinson, D. (2017). *text Mining with R: A Tidy Approach*. O'Reilly Media.

Wickham, H. (2017). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.2.0.

Wickham, H., Francois, R., Henry, L., and Mller, K. (2017). *dplyr: A Grammar of Data Manipulation*. R package version 0.7.4.

Wickham, H. and Grolemund, G. (2017). *R for Data Science: Import, Tidy, Transform and Model Data*. O'Reilly Media.