Dracula Wordcloud

Andrew Mayo

November 8, 2017

Abstract

This document will give instructions on how to create a wordcloud from the classic novel Dracula. We will be using the R program along with packages such as tidytext, dplyr, wordcloud.

*Dracula*is a novel written by Bram Stoker published in 1897[1]. This novel is the first introduction of Count Dracula and created many conventions of vampires written about in further literature. Below I will show how to create create a wordcloud based on the most common used words throughout the novel

1 The Gutenberg Package

There is a package in R which gives us access to almost all books located within the public domain. In order to find Dracula we will use the string detect function from the stringr package.

```
library(stringr)
library(gutenbergr)

gutenberg_works(str_detect(title, "Dracula"))
```

```
## # A tibble: 2 x 8
##   gutenberg_id           title       author gutenberg_author_id language
##          <int>           <chr>        <chr>               <int>    <chr>
## 1          345         Dracula Stoker, Bram                 190       en
## 2        10150 Dracula's Guest Stoker, Bram                 190       en
## # ... with 3 more variables: gutenberg_bookshelf <chr>, rights <chr>,
## #   has_text <lgl>
```

As we can see from the output, Dracula is labeled with an ID number of 345. We will now download the book and place it in the variable Dracula.

[1]This is an example of a footnote

```
Dracula <- gutenberg_download(345)

## Determining mirror for Project Gutenberg from http://www.gutenberg.org/robot/harvest
## Using mirror http://aleph.gutenberg.org
```

Now that we have Dracula downloaded we have to pull out the text and remove stop words[2]. In order to do this we will use the tidytext package and dplyr package.

```
library(dplyr)
library(tidytext)
words_df <- Dracula%>%
              unnest_tokens(words,text)

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

Once this is completed we have every single individual word as its own row within a dataframe. Now that all of the words are seperated we want to add up how many times each word appears throughout the novel. In order to do this we will use the dplyr package and group by each word.

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

Now that we have all of the data that we need let's create the wordcloud.

2 The Wordcloud

In order to generate the wordcloud we need the R package wordcloud. In the function we put in the words we want, the count of each word, and the minimum frequency which must be present in order to view the word.

```
library(wordcloud)

## Loading required package:  RColorBrewer

wordcloud(word_freq$words, word_freq$count, min.freq = 60)
```

[2]Stopwords are unimportant words such as the and or

Looks great! Now you can try it with a different book to see the main aspects of the novel.

References

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, Visualilze, and Model Data*. O'Reilly Media.