Peter Pan Wordcloud

Tiffany Gonzalez

November 3, 2017

Abstract

Sir James Matthew Barrie is a Scottish novelist and playwright. He is best known for creating the play, in which later became a famously known Disney movie, Peter Pan. Peter Pan is about an ageless boy and Tinkerbell, who is a fairy and companion to Peter Pan. These two have many adventures in the fantasy place, Neverland. This is the perfect playwright for Halloween because it is the time to dress up as fictional or non-fictional characters (J.M.Barrie, 2016). In this article, we will be using Sir J.M. Barrie's famous playwright 'Peter Pan or The Boy Who Wouldn't Grow Up' to create a wordcloud.[1] Let's bring in all of the packages we will be working with in this article.

```
library(dplyr)
library(tm)
library(tidytext)
library(wordcloud)
library(stringr)
library(gutenbergr)
library(knitr)
```

1 The Gutenbergr Package

There is a package within R, gutenbergr, that gives access to domain works in the Project Gutenberg collection. We have this package installed above. As mentioned previously, we will be focusing on just the book of Peter Pan. To find the book, including Peter Pan, we can run the following code: To just work with one work, out of all the works, you may call the following function and store the result as follows:

```
library(gutenbergr)
library(stringr)
gutenberg_works(str_detect(title,'Peter Pan'))
```

[1] A wordcloud will be later defined as the article continues

```
## # A tibble: 3 x 8
##   gutenberg_id
##          <int>
## 1         1332
## 2        24012
## 3        39755
## # ... with 7 more variables: title <chr>, author <chr>,
## #   gutenberg_author_id <int>, language <chr>, gutenberg_bookshelf <chr>,
## #   rights <chr>, has_text <lgl>
```

When this code is run, the output gives more than one Peter Pan book. However we know the author is J.M. Barrie, so we can use his gutenberg_ID number, and store the result:

```
library(gutenbergr)
peter_pan<-gutenberg_download(39755)

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

2 The Wordcloud

To make the wordcloud, we first have to break up the lines in the book into words. We can use a function from the tidytext package for this. We can run the following code, and store it into words_df:

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

words_df

## # A tibble: 9,479 x 2
##    gutenberg_id         word
##           <int>        <chr>
##  1        39755 illustration
##  2        39755         with
##  3        39755          the
##  4        39755       spring
##  5        39755        comes
##  6        39755        wendy
##  7        39755          the
##  8        39755        story
##  9        39755           of
```

```
## 10        39755        peter
## # ... with 9,469 more rows
```

But within the column of words, we have common, unimportant words such as 'the' 'a' 'was'... These words are referred to as stop words. We can remove these, with the stop_words data frame and dplyr:

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

words_df
```

```
## # A tibble: 3,571 x 2
##    gutenberg_id         word
##           <int>        <chr>
##  1        39755 illustration
##  2        39755       spring
##  3        39755        wendy
##  4        39755        story
##  5        39755        peter
##  6        39755          pan
##  7        39755       retold
##  8        39755        fairy
##  9        39755         play
## 10        39755          sir
## # ... with 3,561 more rows
```

A wordcloud is a picture composed of words, in which the size of each word is based off of its frequency.Therefore, we need to calculate the frequencies of the the words in our dataframe. Again, we can use standard dplyr techniques for this:

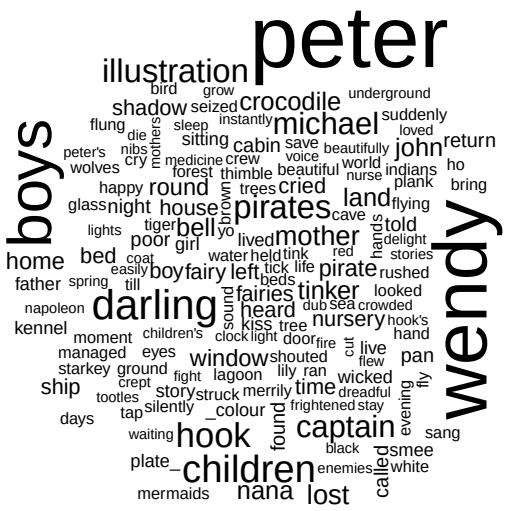
```
library(dplyr)
word_freq<-words_df%>%
  group_by(word)%>%
  summarize(count=n())

word_freq
```

```
## # A tibble: 1,541 x 2
##          word count
##         <chr> <int>
##  1    _colour     8
##  2    _first_     1
##  3     _hear_     1
##  4      _his_     1
```

```
##  5        _i_     1
##  6     _kiss_     1
##  7     _like_     1
##  8       _me_     1
##  9 _thimbles_     1
## 10          1     2
## # ... with 1,531 more rows
```

It's time to create the final product... The Wordcloud!

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



References

Disney, R. (2007). *Walt Disney's Peter Pan.* GOlden/Disney.

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.

J.M.Barrie (2016). *Peter Pan*. Kingman Books.

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.

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 0.7.2.