A Look at Poe's Work

Justin David Minsk

November 6, 2017

Abstract

Edgar Allen Poe is a well known short story and poem writer that tended to write poems in the horror genre such as "The Raven", "The Cask of Amontillado", "The Tell-Tale Heart", and many other horror fan's favorites. The purpose of this paper is to look at the book, "The Complete Poetical Works Of Edgar Allen Poe" and find the most used words in his poetical works, just in time for Halloween.

1 "The Complete Poetical Works Of Edgar Allen Poe"

The first task is to get "The Complete Poetical Works Of Edgar Allen Poe" into R to be able to mine the text. First we need the libraries that we will use to mine the text.

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

Then we use gutenbergr to extract the data into a data frame.

```
ids <- gutenberg_works(author == str_extract(author, "Poe, Edgar Allan"))
#get Poe work's IDs

head(ids$title)

## [1] "The Fall of the House of Usher"

## [2] "First Project Gutenberg Collection of Edgar Allan Poe"

## [3] "The Cask of Amontillado"

## [4] "The Masque of the Red Death"</pre>
```

```
## [5] "The Raven"
## [6] "The Works of Edgar Allan Poe <U+0097> Volume 1"
#see Poe work's IDs

df <- gutenberg_download(10031)
#get the complete poetical works of Edgar Allan Poe</pre>
```

The data frame now looks like this:

```
words_df <- df%>%
 unnest_tokens(word, text)
#split the lines into words
head(words_df)
## # A tibble: 6 x 2
##
     gutenberg_id
                       word
##
            <int>
                      <chr>>
## 1
            10031
                        the
## 2
            10031 complete
## 3
            10031 poetical
## 4
            10031
                      works
## 5
            10031
                         of
            10031
                      edgar
```

Then we need to get rid of all the common words that do not give us anything unique in the text compared to other texts. We do this by using a list of words that are commonly used. This text is unique since it is written in an older English and has some extra common words that need to be filtered out past the normal list.

```
words_df <- words_df%>%
  filter(!(word %in% stop_words$word))
#get rid of common words that are not unique (the, a, etc.)

words_df <- words_df%>%
  filter(!word == "thy" & !word == "thou" & !word == "thee")
#some older words not in out stop_word list that are not useful
```

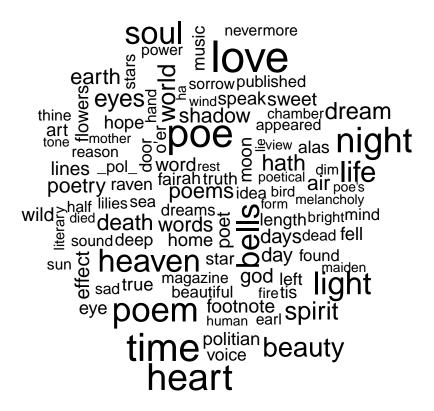
Now to create a count of each instance of a word. Using simple dplyr functions this can easily be done.

```
words_free <- words_df%>%
 group_by(word)%>%
 summarise(count = n())%>%
 arrange(-count)
#make a count of the word
head(words_free)
## # A tibble: 6 x 2
##
     word count
##
    <chr> <int>
## 1 love 116
## 2
      poe
           104
## 3 time
           101
           96
## 4 heart
## 5 night
           90
## 6 poem
             89
```

3 Wordcloud

Now we can create a wordcloud, a good visual representation of the most common words. The bigger the word the higher the frequency. We will use the library wordcloud to generate this image.

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



#make a word cloud

References

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.

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

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

Robinson, J. S. . 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.