# Using R to create a wordcloud from text in an ebook

Judy Minichelli

November 6, 2017

**Abstract**

The article gives instructions on how to download text from an ebook and create a wordcloud in R. A wordclouds is a data visualization that shows the most commonly used words in a large text dataset where size is proportional to frequency; words with the highest count appear larger and in bold. The ebook used in this example is "The Wonderful Wizard of Oz"

*The Wonderful Wizard of Oz* was written by Frank Baum and published in 1900. It was the basis for the 1902 Broadway musical and the classic 1939 movie starring Judy Garland.

## 1 R Packages

The following packages were installed and brought in with library: dplyr, gutenbergr, stringr, tidytext, tm, wordcloud.

The function below stores the result of the book text download in the dataframe "wizard".

```
library(gutenbergr)
wizard<-gutenberg_download(55)
```

This dataframe has two columns, one for each line in the book.
For this exercise, it is not necessary to exclude chapter headings and the first few pages of text; however, the procedures below will accomplish the task.

## 2 How to Clean Data

```
library(stringr)

## Warning:  package 'stringr' was built under R version 3.4.2
```

```
library(dplyr)
```

```
## Warning:  package 'dplyr' was built under R version 3.4.2
```

```
##
## Attaching package:  'dplyr'
## The following objects are masked from 'package:stats':
##
##    filter, lag
## The following objects are masked from 'package:base':
##
##    intersect, setdiff, setequal, union
```

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

The actual text begins on line 10 and ends on 4721 so the wizard data frame can be redefined to exclude the first 9 lines.

```
wizard<-wizard[10:4721,]
```

# 3 The Wordcloud

To make the wordcloud, break the text lines into words.

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

words_df
```

```
## # A tibble: 39,695 x 2
##    gutenberg_id         word
##           <int>        <chr>
##  1           55     contents
##  2           55 introduction
##  3           55            1
##  4           55          the
##  5           55      cyclone
##  6           55            2
##  7           55          the
##  8           55      council
##  9           55         with
## 10           55          the
## # ... with 39,685 more rows
```

Remove commonly used "generic" words from the data frame, for example articles, prepositions, and pronouns like "the", "after", and "you" from the data frame using the stop_words command.

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

words_df
```

```
## # A tibble: 12,304 x 2
##    gutenberg_id         word
##           <int>        <chr>
##  1           55     contents
##  2           55 introduction
##  3           55            1
##  4           55      cyclone
##  5           55            2
##  6           55      council
##  7           55    munchkins
##  8           55            3
##  9           55      dorothy
## 10           55        saved
## # ... with 12,294 more rows
```

Calculate the frequencies of the remaining unique words.

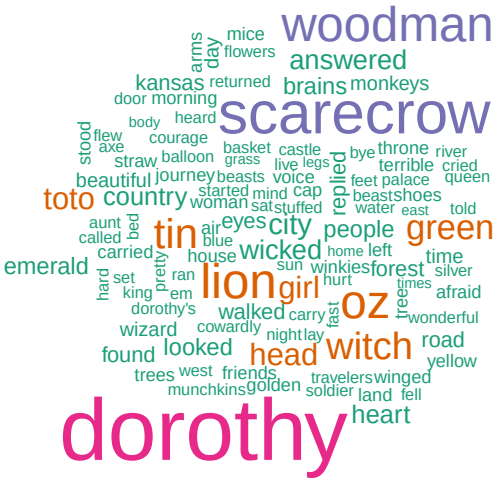
```
word_freq<-words_df%>%
  group_by(word)%>%
  summarize(count=n())

word_freq
```

```
## # A tibble: 2,507 x 2
##     word count
##    <chr> <int>
##  1     1     2
##  2    10     2
##  3    11     2
##  4    12     2
##  5    13     2
##  6    14     2
##  7    15     2
##  8    16     2
##  9    17     2
## 10    18     2
## # ... with 2,497 more rows
```

Generate the wordcloud. If there are too many or too few words, adjust "n" in "min.freq=n" to change the minimum number of occurances required for the word to appear in the wordcloud. The number of colors used in the graphic can be changed by adjusting "n" in "colors=brewer.pal(n,'Dark2'). This helps further differentiate the words used with higher frequency.

```
library(wordcloud)
library(tm)
wordcloud(word_freq$word,word_freq$count,min.freq=20,colors=brewer.pal(4,'Dark2'))
```




## References


Baum, F. (1900). *The Wonderful Wizard of Oz*. George M. Hill Company.

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.

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.