# H.P. Lovecraft Word cloud

Kimberly Staudt

November 6, 2017

**Abstract**

In this article, we create a word cloud using tidytext and gutenbergr,for H.P. Lovecraft's novel, *The Dunwich Horror*.

This 1929 horror classic follows the peculiar Wilbur Whateley,and his wizard grandfather who create a monster.This world cloud is constructed of the most frequently used words in the novel.

## 1 Packages and Filtering

To begin install the following packages: tidytext, tm, wordcloud, gutenbergr, and stringr. Dplyr comes with tidytext, so pull this from the library as seen below. To find the novel's ID, filter the title. This can apply to any novel within gutenbergr.Once the ID number is found, download and store into the data frame, Horror.

```
library(tidytext)
library(dplyr)
library(gutenbergr)

Dunwich<-gutenberg_works() %>%
  filter(title == "The Dunwich Horror")

Horror<-gutenberg_download(50133)
```

Next, us the string detect function to filter and remove text with the word CHAPTER.

```
library(stringr)
Horror<-Horror%>%
  filter(!str_detect(Horror$text,'CHAPTER'))
```

## 2 Create a New Column

Unnest the text to create a new column, word. Stop words are common language, such as articles or pronouns, that would be overly present in text. We are trying to isolate unique language that separates the novel from others by eliminating these with the ”!” character. Store this data into a new data frame.

```
words_df<-Horror%>%
  unnest_tokens(word,text)

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

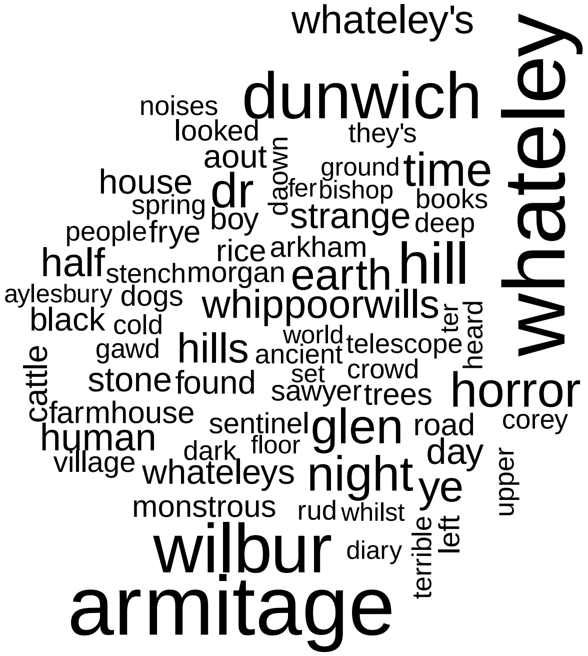
To find the frequency of the words, group the data by word and use count to summarize how many times this word appears. Store into a new data frame.

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

## 3 Visualization

Now we can create our word cloud by using the word frequency data frame. The freq function allows you to adjust the size of the cloud. By using a frequency of 10, words appear 10 times in the text before they are displayed in the cloud. The higher the number, the smaller the word cloud. RColorBrewer allows you to choose the color pallete of your desire, but in this example our word cloud will be black.

```
library(wordcloud)
wordcloud(word_freq$word,word_freq$count,min.freq=10)
```


whateley's
whateley
dunwich
noises
looked
they's
aout
daown
ground
time
house
dr
fer
bishop
spring
books
boy
strange
people
frye
deep
rice
arkham
half
stench
morgan
earth
hill
aylesbury
dogs
whippoorwills
black
cold
ter
heard
world
gawd
hills
ancient
telescope
cattle
stone
found
set
crowd
sawyer
trees
horror
farmhouse
sentinel
glen
road
corey
human
dark
floor
day
village
whateleys
night
ye
upper
monstrous
rud
whilst
left
terrible
wilbur
diary
armitage