Some Simple SPOOKY Data Analysis

Cindy Rush

January 22, 2018

Introduction

This files contains some simple analysis of the SPOOKY data. The goal is to remind ourselves of some of our basic tools for working with text data in `R` and also to practice reproducibility. You should be able to put this file in the `doc` folder of your `Project 1` repository and it should just run (provided you have `multiplot.R` in the `libs` folder and `spooky.csv` in the `data` folder). If you open to file from a forked `Week1-GitHub` repo, you should have no trouble running the code directly.

Setup the libraries

First we want to install and load libraries we need along the way. Note that the following code is completely reproducible – you don't need to add any code on your own to make it run.

```
packages.used <- c("ggplot2", "dplyr", "tidytext", "wordcloud", "stringr", "ggridges")

# check packages that need to be installed.
packages.needed <- setdiff(packages.used, intersect(installed.packages()[,1], packages.used))

# install additional packages
if(length(packages.needed) > 0) {
  install.packages(packages.needed, dependencies = TRUE, repos = 'http://cran.us.r-project.org')
}

library(ggplot2)
library(dplyr)
library(tidytext)
library(wordcloud)
library(stringr)
library(ggridges)

source("../libs/multiplot.R")
```

Read in the data

The following code assumes that the dataset `spooky.csv` lives in a `data` folder (and that we are inside a `docs` folder).

```
spooky <- read.csv('../data/spooky.csv', as.is = TRUE)
```

An overview of the data structure and content

Let's first remind ourselves of the structure of the data.

```r
head(spooky)
```

```
##        id
## 1 id26305
## 2 id17569
## 3 id11008
## 4 id27763
## 5 id12958
## 6 id22965
##
## 1
## 2
## 3
## 4
## 5
## 6 A youth passed in solitude, my best years spent under your gentle and feminine fosterage, has so r
##   author
## 1    EAP
## 2    HPL
## 3    EAP
## 4    MWS
## 5    HPL
## 6    MWS
```

```r
summary(spooky)
```

```
##       id                 text              author
##  Length:19579       Length:19579       Length:19579
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
```

We see from the above that each row of our data contains a unique ID, a single sentence text excerpt, and an abbreviated author name. `HPL` is Lovecraft, `MWS` is Shelly, and `EAP` is Poe. Here are a few example sentences:

```r
spooky$text[1]
```

```
## [1] "This process, however, afforded me no means of ascertaining the dimensions of my dungeon; as I m
```

```r
spooky$text[13494]
```

```
## [1] "While my companion contemplated with a serious and satisfied spirit the magnificent appearances
```

```r
spooky$text[666]
```

```
## [1] "What was it I paused to think what was it that so unnerved me in the contemplation of the House
```

We finally note that there are no missing values, and we change author name to be a factor variable, which will help us later on.

```r
sum(is.na(spooky))
```

```
## [1] 0
```

```r
spooky$author <- as.factor(spooky$author)
```

An intro to tidytext

For my tutorials on Project 1, I will be using tidytext. If this is new to you, here's a textbook that can help: *Text Mining with R; A Tidy Approach.* It teaches the basic handling of natural language data in R using tools from the "tidyverse". The tidy text format is a table with one token per row, where a token is a word.

Data Cleaning

We first use the unnest_tokens() function to drop all punctuation and transform all words into lower case. At least for now, the punctuation isn't really important to our analysis – we want to study the words. In addition, tidytext contains a dictionary of stop words, like "and" or "next", that we will get rid of for our analysis, the idea being that the non-common words (. . . maybe the SPOOKY words) that the authors use will be more interesting.

```
spooky_wrd <- unnest_tokens(spooky, word, text)
head(spooky_wrd)
```

```
##          id author     word
## 1   id26305    EAP     this
## 1.1 id26305    EAP  process
## 1.2 id26305    EAP  however
## 1.3 id26305    EAP afforded
## 1.4 id26305    EAP       me
## 1.5 id26305    EAP       no
```

```
head(stop_words)
```

```
## # A tibble: 6 x 2
##        word lexicon
##       <chr>   <chr>
## 1         a   SMART
## 2       a's   SMART
## 3      able   SMART
## 4     about   SMART
## 5     above   SMART
## 6 according   SMART
```

```
tail(stop_words)
```

```
## # A tibble: 6 x 2
##       word lexicon
##      <chr>   <chr>
## 1      you    onix
## 2    young    onix
## 3  younger    onix
## 4 youngest    onix
## 5     your    onix
## 6    yours    onix
```

```
spooky_wrd <- anti_join(spooky_wrd, stop_words, by = "word")
head(spooky_wrd)
```

```
##        id author         word
## 1 id26305    EAP      process
## 2 id26305    EAP     afforded
## 3 id26305    EAP        means
## 4 id26305    EAP ascertaining
## 5 id26305    EAP   dimensions
## 6 id26305    EAP      dungeon
```

Data Visualization

First we'll do some simple numerical summaries of the data to provide some nice visualizations.

```
p1 <- ggplot(spooky) +
      geom_bar(aes(author, fill = author)) +
      theme(legend.position = "none")

spooky$sen_length <- str_length(spooky$text)
head(spooky$sen_length)
```

```
## [1] 231  71 200 206 174 468
```

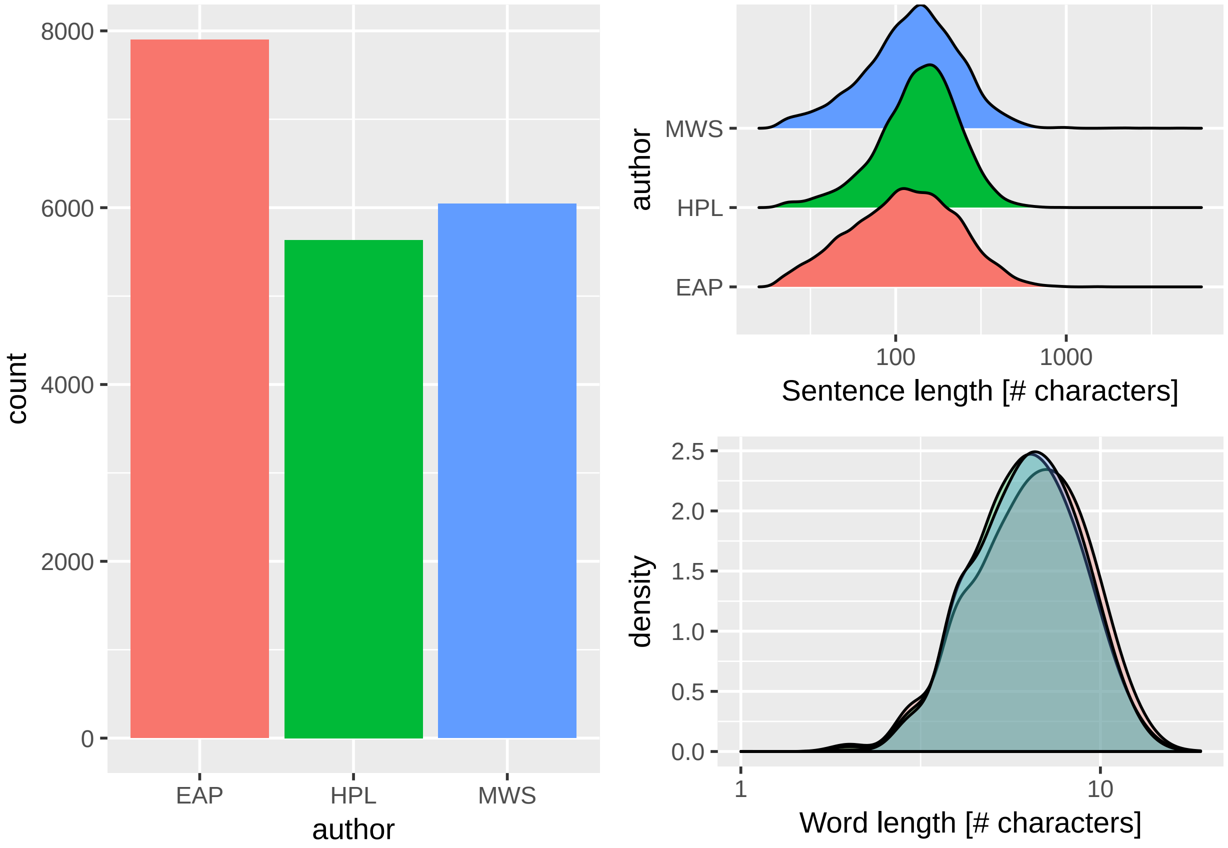
```
p2 <- ggplot(spooky) +
      geom_density_ridges(aes(sen_length, author, fill = author)) +
      scale_x_log10() +
      theme(legend.position = "none") +
      labs(x = "Sentence length [# characters]")

spooky_wrd$word_length <- str_length(spooky_wrd$word)
head(spooky_wrd$word_length)
```

```
## [1]  7  8  5 12 10  7
```

```
p3 <- ggplot(spooky_wrd) +
      geom_density(aes(word_length, fill = author), bw = 0.05, alpha = 0.3) +
      scale_x_log10() +
      theme(legend.position = "none") +
      labs(x = "Word length [# characters]")

layout <- matrix(c(1, 2, 1, 3), 2, 2, byrow = TRUE)
multiplot(p1, p2, p3, layout = layout)
```



From the above plots we find:

-
-
-

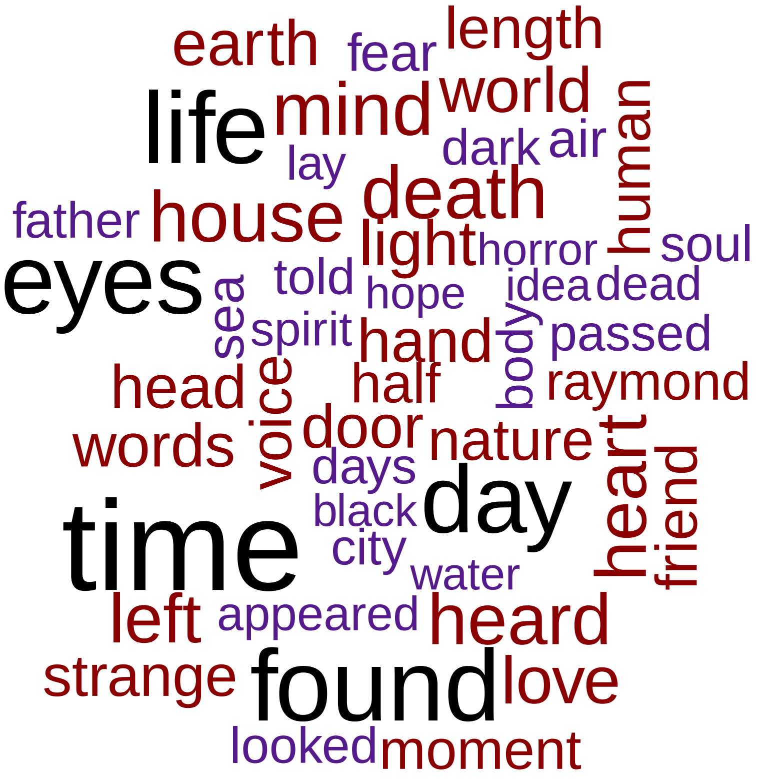
Now we study some of the most common words in the entire data set. With the below code we plot the fifty most common words in the entire datset. We see that “time”, “life”, and “night” all appear frequently.

```
words <- names(table(spooky_wrd$word))
freqs <- table(spooky_wrd$word)
head(sort(freqs, decreasing = TRUE))
```

```
##
##  time  life found night  eyes   day
##   729   563   559   559   540   516
```

```
wordcloud(words, freqs, max.words = 50, color = c("purple4", "red4", "black"))
```

```
## Warning in wordcloud(words, freqs, max.words = 50, color = c("purple4", :
## night could not be fit on page. It will not be plotted.
```


length
earth
fear
world
life
mind
human
dark
air
lay
death
father
house
light
horror
soul
eyes
told
hope
idea
dead
sea
spirit
hand
body
passed
head
voice
half
raymond
door
nature
words
days
heart
friend
day
time
black
city
water
left
appeared
heard
strange
found
love
looked
moment