Project1_SPOOKY_Textual_Analysis

Xiangyu Liu-xl2703

January 24, 2018

This notebook was written for course STAT 4243 at Columbia University so I did my best to go through intensive detail and include the theory used for data cleaning, text mining and natural language processing such as sentiment analysis, topic modeling, etc.

My goal is to identify and illustrate similarities and differences identified by my analysis. Since it is the spooky text dataset, I pay more attention on the comparision between fear words and joy words. If you are also curious about it, please wait and see!

Part0: Setup the libraries and functions

Let the fun begin!

```
library(tidytext) # for text analysis
library(wordcloud) # for text visualization
```

```
## Loading required package: RColorBrewer
```

```
library(ggridges) # for visualization
```

```
## Loading required package: ggplot2
```

```
library(ggplot2) # for visualization
library(ggfortify) # for visualization
library(Rmisc) # for data analysis
```

```
## Loading required package: lattice
```

```
## Loading required package: plyr
```

```
library(tibble) # for data wrangling
library(dplyr) # for data manipulation
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr) # for string manipulation
library(tidyr) # for data wrangling
library(scales) # for visualization
library(SnowballC) # for text analysis
library(topicmodels) # for text analysis
library(tidyverse) # for data analysis
```

```
## ── Attaching packages ──────────────────────────── tidyverse 1.2.1 ──
```

```
## ✔ readr   1.1.1     ✔ purrr   0.2.4
## ✔ readr   1.1.1     ✔ forcats 0.2.0
```

```
## ── Conflicts ─────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::arrange()    masks plyr::arrange()
## ✖ readr::col_factor() masks scales::col_factor()
## ✖ purrr::compact()    masks plyr::compact()
## ✖ dplyr::count()      masks plyr::count()
## ✖ purrr::discard()    masks scales::discard()
## ✖ dplyr::failwith()   masks plyr::failwith()
## ✖ dplyr::filter()     masks stats::filter()
## ✖ dplyr::id()         masks plyr::id()
## ✖ dplyr::lag()        masks stats::lag()
## ✖ dplyr::mutate()     masks plyr::mutate()
## ✖ dplyr::rename()     masks plyr::rename()
## ✖ dplyr::summarise()  masks plyr::summarise()
## ✖ dplyr::summarize()  masks plyr::summarize()
```

```
library(treemapify) # for visualization
library(topicmodels) # for topic modeling
```

I also want to use acast function from the reshape2 in a package. However, it causes future errors from my package. So I use reshape2::acast to prevent the conflict. The reshape2 package is for data wrangling.

Then I manually define a function ‘multiplot’, which can also be called from the existing package. I use this function to render multiple ggplots in one image.

```
multiplot <- function(..., plotlist = NULL, file, cols = 1, layout = NULL) {
  require(grid)

  plots <- c(list(...), plotlist)

  numPlots = length(plots)

  if (is.null(layout)) {
    layout <- matrix(seq(1, cols * ceiling(numPlots/cols)),
                     ncol = cols, nrow = ceiling(numPlots/cols))
}

if (numPlots == 1) {
print(plots[[1]])

} else {
grid.newpage()
pushViewport(viewport(layout = grid.layout(nrow(layout), ncol(layout))))

for (i in 1:numPlots) {
  matchidx <- as.data.frame(which(layout == i, arr.ind = TRUE))

  print(plots[[i]], vp = viewport(layout.pos.row = matchidx$row,
                                  layout.pos.col = matchidx$col))
 }
}
 }
```

Part 1: Data Preparation

The first part of our journey is data preparation. Data preparation is a necessary, but often tedious activity that is a critical first step in data analytics projects. In this part, I complete 5 important tasks, inluding reading the data, making an overview of the data structure and content, checking missing values, reformating features and data cleaning.

Read the data

The first thing I' ll do is load in the data. I load the data from the local file.

```
spooky <- read.csv("/Users/xiangyu/Documents/GitHub/Spring2018/Project_Starter_Codes/
data/spooky.csv", as.is = TRUE)
```

An overview of the data structure and content

Then let's get a quick overview of the data structure and content.

```
summary(spooky)
```

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

```
glimpse(spooky)
```

```
## Observations: 19,579
## Variables: 3
## $ id     <chr> "id26305", "id17569", "id11008", "id27763", "id12958", ...
## $ text   <chr> "This process, however, afforded me no means of ascerta...
## $ author <chr> "EAP", "HPL", "EAP", "MWS", "HPL", "MWS", "EAP", "EAP",...
```

I find the data set consists of 19579 rows and 3 columns. Each row contains a single scary sentence, such as: 'This process, however, afforded me no means of ascertaining the dimensions of my dungeon; as I might make its circuit, and return to the point whence I set out, without being aware of the fact; so perfectly uniform seemed the wall.' There are 3 authors in the whole dataset. 'HPL' refers to HP Lovecraft, 'MWS' refers to Mary Wollstonecraft Shelley and 'EAP' refers to Edgar Allan Poe.

Missing values

Another important thing for us to do is to check the missing values. The concept of missing values is important to understand in order to successfully manage data. If the missing values are not handled properly, then I may end up drawing an inaccurate inference about the data. Due to improper handling, the result obtained by me will differ from ones where the missing values are present.

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

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

Great! There are no missing values. It absolutely saves me a lot of time.

Reformating features

Then I change the author name to a factor, which will help me a lot later on.

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

Data Cleaning

Then I split a column into tokens using the tokenizers package. 'unnest_tokens' from tidytext package help us drop punctuation and transforms all words into lower case. In addition, I use anti_join to omit all the stop words, like 'and', 'or', because these words are not related to the text in our analysis. However, in order to make sure I don't miss any information given by these stop words, I may analyze them specifically later on.

```
spooky_wrd <- spooky %>% unnest_tokens(word, text)

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

write.csv(spooky_wrd, file='spooky_wrd.csv')
```

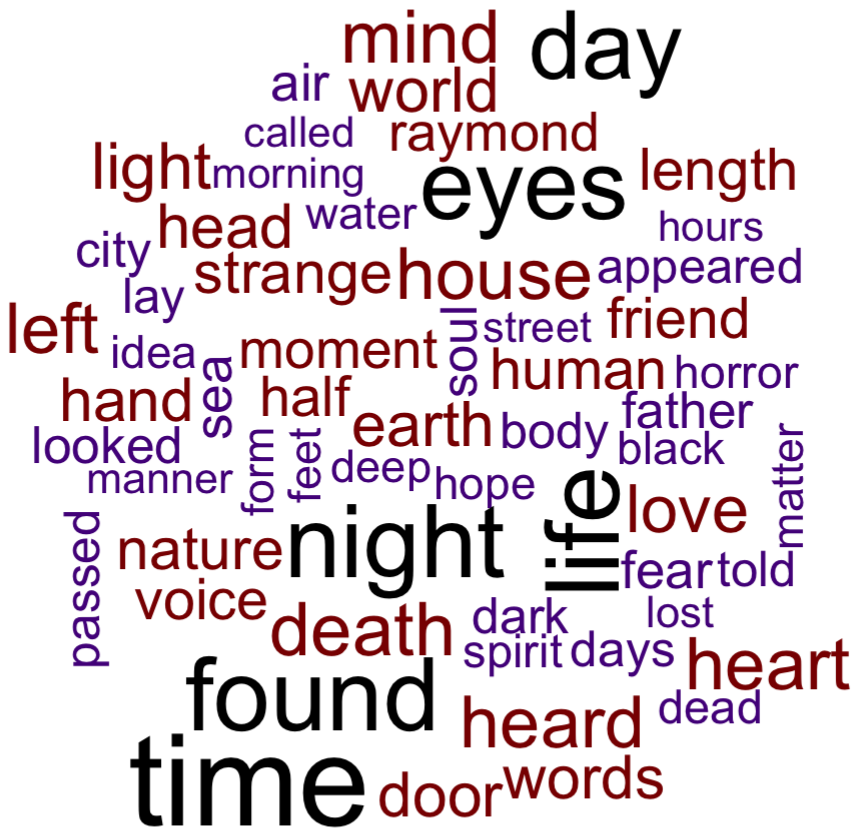
Part 2: Overview of data visualization

Congratulations! We have arrived at the second part of our project, which is the overview of data visualization. In this part, I want to do some basic visualizations in order to get a general picture of our text characteristics, including word clouds, word count, sentence length and word length.

Author-dependent Word Clouds

First, I want to get the overview of the data visualization. I am curious about the frequent words used by these authors. Are these frequent words are all scary words like ‘death’, ‘fear’, ‘blood’? Let’s go and have a look!

```
spooky_wrd %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 60, color = c("purple4", "red4", "black")))
```

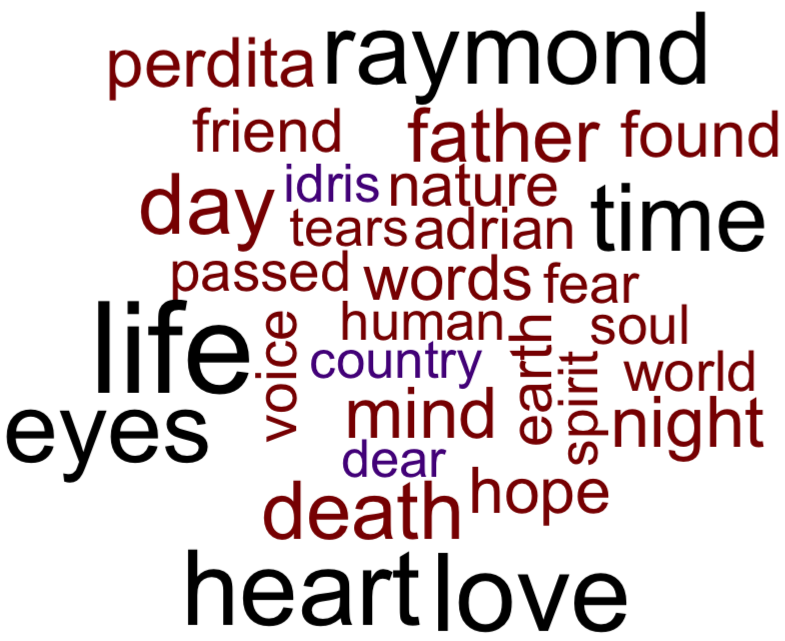


Those are the 60 most common words in my entire `spooky` set. The differene colours represent different frequencies. The most frequency words are labelled as black, which are ‘eyes’, ‘day’, ‘time’, ‘life’ and ‘found’. Among them I can find some scary ones which I guessed before, like ‘death’, ‘life’, ‘dark’, ‘black’ and ‘horror’. But to my surprise, I also find some joy words here, such as ‘love’, ‘hope’ and ‘friend’.

In order to compare the similarities and differences among three authors, I try to do the same for each author.

First, let’s see Mary’s wordcloud:

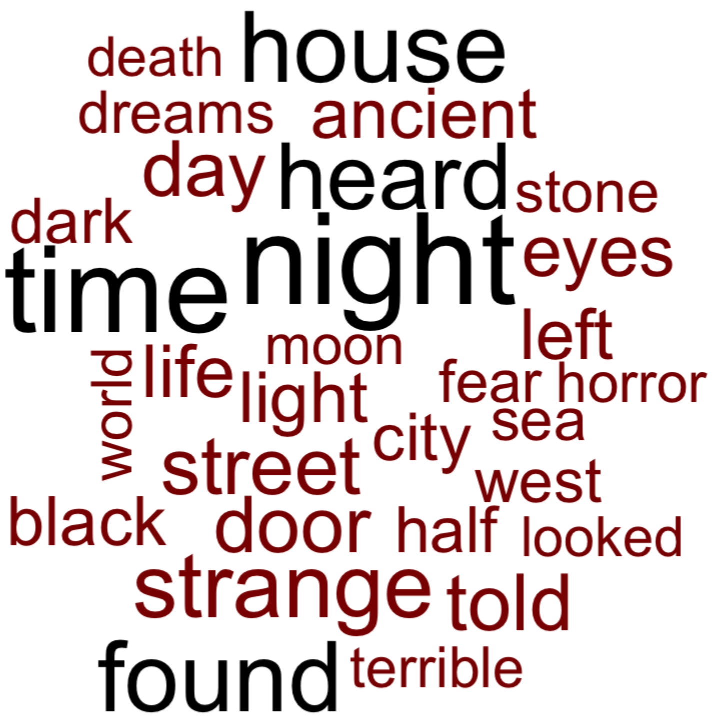
```
spooky_wrd %>%
  filter(author == "MWS") %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 30, color = c("purple4", "red4", "black")))
```



Those are the 30 most common words in the entire Mary's sentences. I notice that the most frequent words of Mary are 'heart', 'time', 'life', 'eys', 'raymond' and 'love'! It seems that Mary use both joy words and scary words frequently.

Let's also take a look at HP's word cloud:

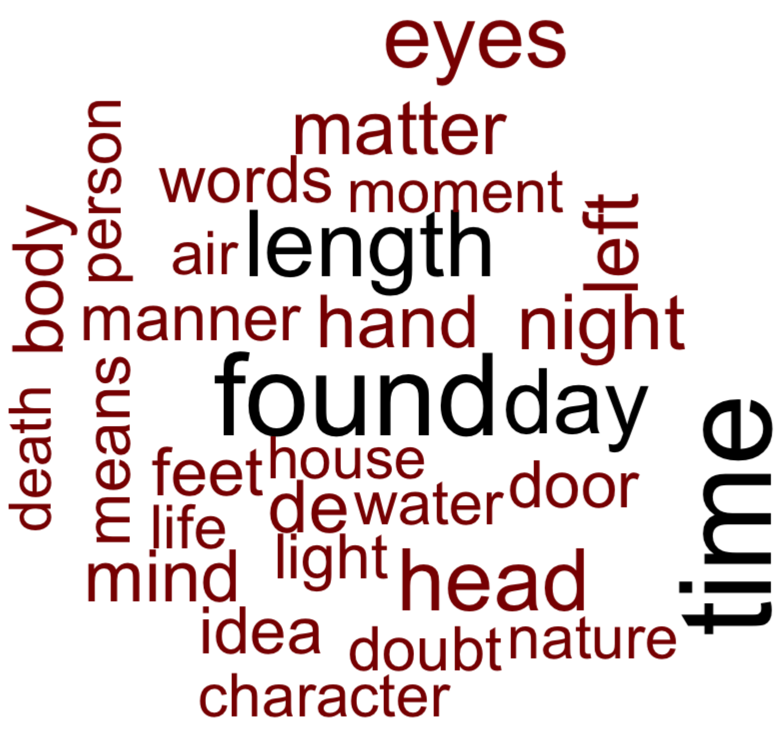
```
spooky_wrd %>%
  filter(author == "HPL") %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 30, color = c("purple4", "red4", "black")))
```



Those are the 30 most common words in the entire HP's sentences. I notice that the most frequent words of HP are 'night', 'time', 'heard', 'house','house'. Up to now, it's hard for me to draw any conclusion regarding HP's preference in using scary words and joy words. Thus, in the next part, I decide to compare the frequency of some strong scary words and joy words among authors.

Finally, let's see Edgar Allan Poe's wordcloud:

```
spooky_wrd %>%
  filter(author == "EAP") %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 30, color = c("purple4", "red4", "black")))
```



Those are the 30 most common words in the entire Edgar's sentences. I notice that the most frequent words of Mary are 'length', 'time', 'day', 'found'. It seems that Edgar's wordcloud is most similar with the wordcloud of the entire dataset. Is it because Edgar use more words than others? Thus, I decide to explore the author count and sentence and word length in the next step.

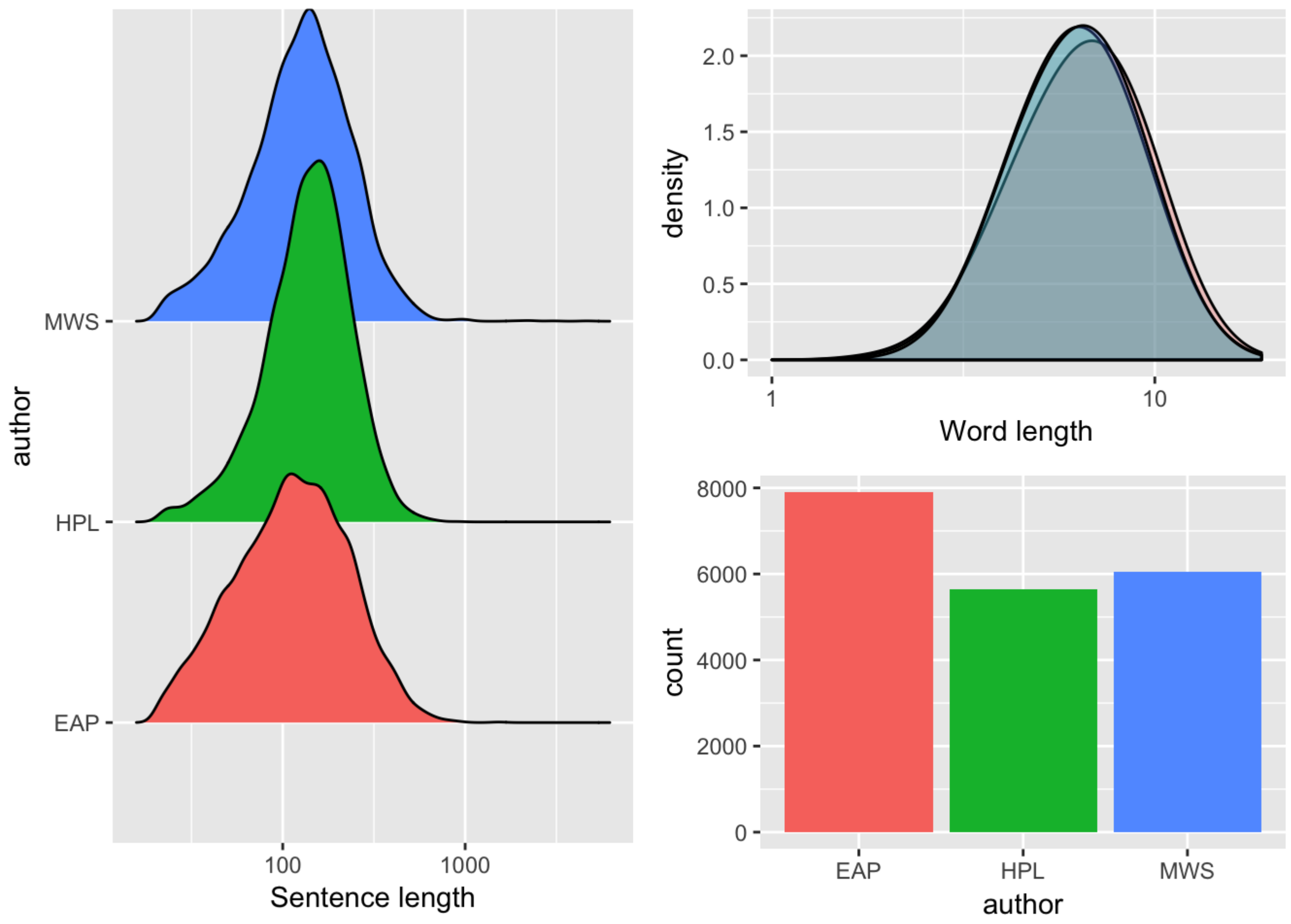
Author count and sentence/word length

As I have mentioned before, to get a clearer picture and confirm my guess, I count the words of each author as well as their word leangths and sentence lengths.

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

spooky$sen_length <- str_length(spooky$text) #count the length of each sentence

sentence_length <- ggplot(spooky) +
      geom_density_ridges(aes(sen_length, author, fill = author)) +
      scale_x_log10() +
      theme(legend.position = "none") +
      labs(x = "Sentence length")


spooky_wrd$word_length <- str_length(spooky_wrd$word) #count the length of each word

word_length <- ggplot(spooky_wrd) +
      geom_density(aes(word_length, fill = author), bw = 0.1, alpha = 0.3) +
      scale_x_log10() +
      theme(legend.position = "none") +
      labs(x = "Word length")


multiplot( sentence_length, word_length,word_count, layout = matrix(c(1, 2, 1, 3), 2,
 2, byrow = TRUE))
```

```
## Loading required package: grid
```

```
## Picking joint bandwidth of 0.0414
```



It is so exciting that Edgar’s word count is bigger than others. It means that Edgar has more words than others, which confirms my guess in 2.1! In addition, Edgar’s word length is also a bit longer than others. HP and Mary seem to have very similar word lengths. However, HP’s sentence length is a bit longer than others. It seems that he may like to use shorter sentences.

Since shorter sentences are probably more easier to understand, does it mean it is more easy to read HP’s novels than others ?

To conclude, the overview of data visualization part confirms my many conjectures as well as causes more thinkings. In the next part, I decide to explore deeper and get more concise conclusions.

Part 3: Characteristics of words

Word is the smallest unit of the whole text. So let’s first compare the characteristics of words among three authors!

解释 Frequency of words by authors是wordcloud by author的进一步延伸，我们发现 “love” is clearly more used by Shelley than by Lovecraft. The word “half” is only found in Poe and Lovecraft, but not in Shelley’s work at a notable frequency.

Relative word frequencies

In order to explore deeper based on author dependent word clouds, here I want to compare the relative word frequencies between each author. Since we have 3 authors altogether, we have 3 pairs. If the word occurs more than 20 times, then it can be included to the analysis.

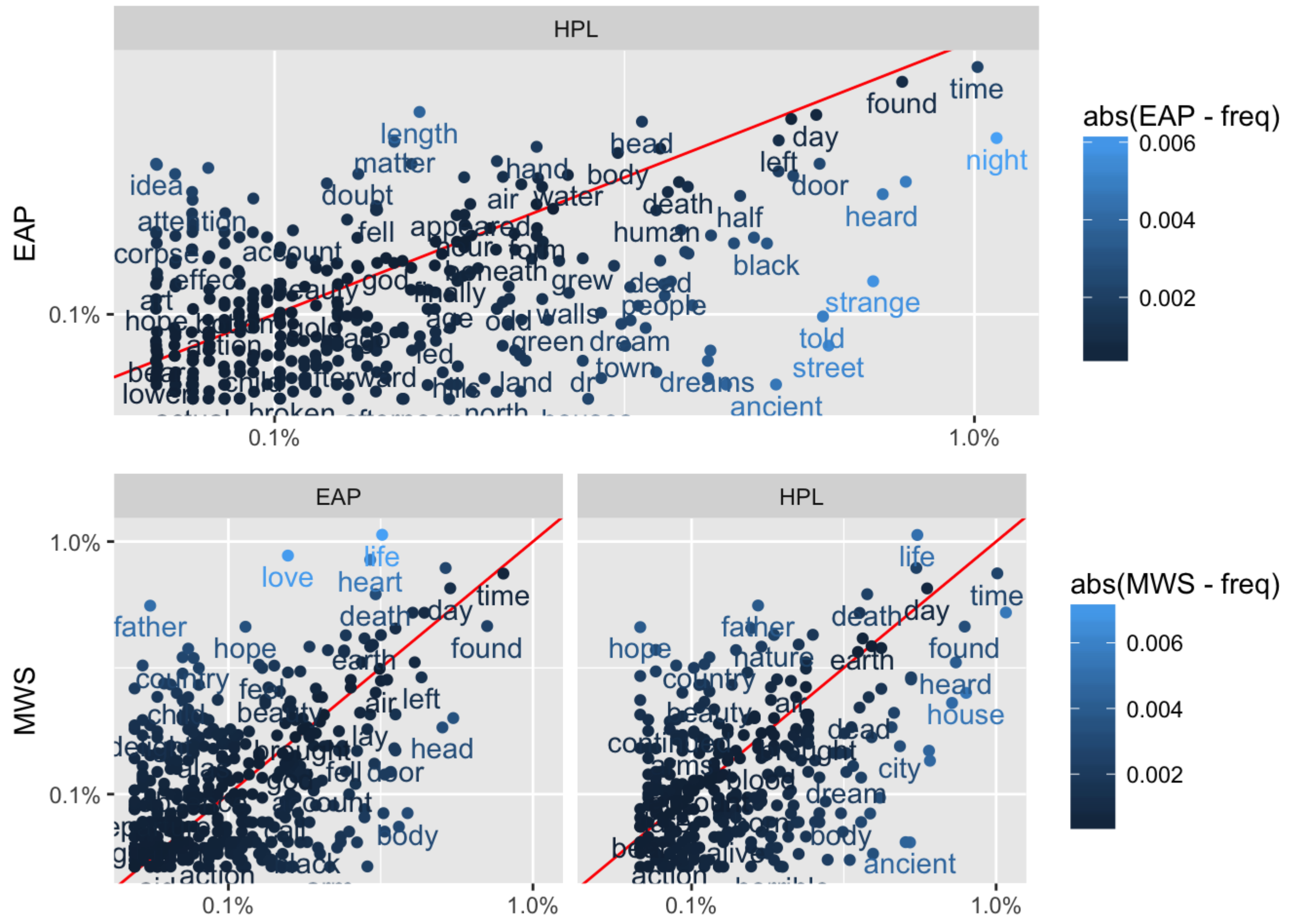
```
frequency1 <-spooky_wrd %>%
  count(author, word) %>%
  filter(n > 15) %>%
  group_by(author) %>%
  mutate(freq = n / sum(n)) %>%
  select(-n) %>%
  spread(author, freq) %>%
  gather(author, freq, HPL:MWS) %>%
  filter(!is.na(freq) & !is.na(author) & author == "HPL") # This is the HPL:MWS frequ
ency.



f1<-ggplot(frequency1, aes(freq, EAP , color = abs(EAP - freq))) +
  geom_abline(color = "red", lty = 1) +
  geom_jitter(alpha = 1) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  facet_wrap(~author, ncol = 5) +
  labs(y = "EAP", x = NULL) # This plot compare Edgar's word frequency with HP's.

frequency2 <-spooky_wrd %>%
  count(author, word) %>%
  filter(n > 15) %>%
  group_by(author) %>%
  mutate(freq = n / sum(n)) %>%
  select(-n) %>%
  spread(author, freq) %>%
  gather(author, freq, EAP:HPL) %>%
  filter(!is.na(freq) & !is.na(author)) # This is the EAP:HPL frequency.

f2<- ggplot(frequency2, aes(freq, MWS , color = abs(MWS - freq))) +
  geom_abline(color = "red", lty = 1) +
  geom_jitter(alpha = 1) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  facet_wrap(~author, ncol = 5) +
  labs(y = "MWS", x = NULL) # This plot compare Mary's word frequency with Edgar's an
d HP's.

multiplot(f1, f2, cols=1)
```

```
## Warning: Removed 358 rows containing missing values (geom_point).
```

```
## Warning: Removed 358 rows containing missing values (geom_text).
```

```
## Warning: Removed 948 rows containing missing values (geom_point).
```

```
## Warning: Removed 948 rows containing missing values (geom_text).
```



Let me first give a brief explanation of the above picture. If the word is at the top left corner, it means this word is highly used by the anthor on the left side beacuse the relative word frequency of this author is much higher. If the word is at the bottom right corner, it means this word is highly used by the anthor on the top beacuse the relative word frequency of this author is much higher.

Therefore, I find that Edgar uses ‘idea’, ‘body’ more often than others. Mary uses ‘father’, ‘hope’ and ‘love’ more than others. HP uses ‘ancient’, ‘street’ more frequently.

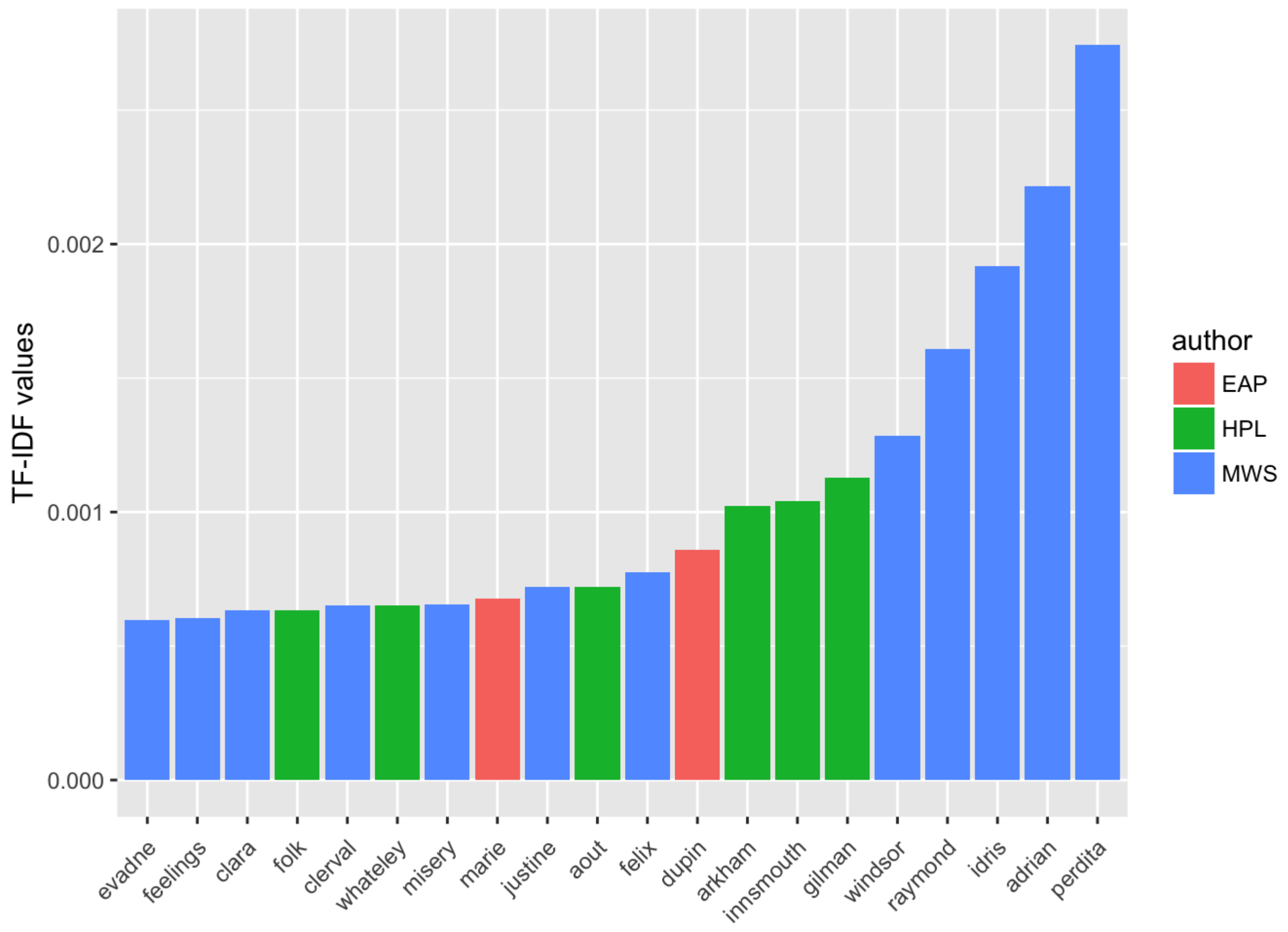
I conclude that Mary are more likely to use joy words than other two authors. Does it mean that Mary’s novel is not as scary as others? If yes, then maybe somebody like me can take a try!

TF-IDF

Next, I want to choose one of the most popular term-weighting schemes, which is called TF-IDF, not only because it is popular, but also because it is a a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The tf-idf value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus. I use TF-IDF to adjust for the fact that some words appear more frequently in general. By the way, TF-IDF is short for term frequency–inverse document frequency.

First, I pick the top twenty tf_idf scores in all the text.

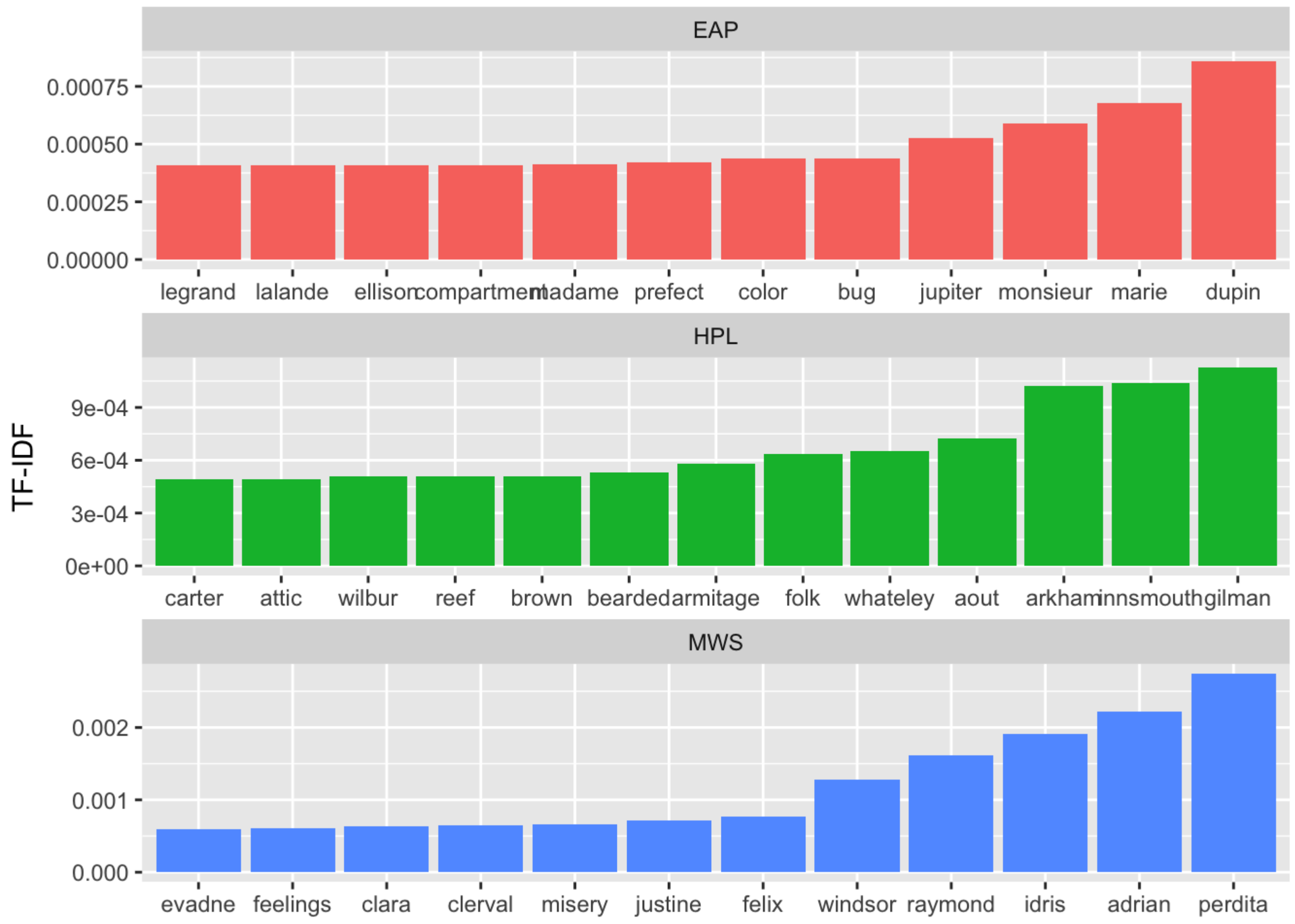
```
tf_idf <- spooky_wrd %>%    #caiculate the tf-idf value per word per author
  count(author, word) %>%
  bind_tf_idf(word, author, n)


spooky_wrd %>%
  count(author, word) %>%
  bind_tf_idf(word, author, n) %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  top_n(20, tf_idf) %>%
  ggplot(aes(word, tf_idf, fill = author)) +
  geom_col() +
  labs(x = NULL, y = "TF-IDF values") +
  theme(legend.position = "right", axis.text.x  = element_text(angle=45, hjust=1, vju
st=0.9))
```



Wow! It is surprising that there are so many names here! I can see ‘Perdita’, ‘Adrian’, ‘Justine’, etc. These names appear more frequently than other words.

As always, I pay the closest attention to scary words and joy words. However, because of so many names existing, I can not find enough emotional words here. Thus, I decide to do the same TF-IDF analysis on each author in order to explore deeper.

```
tf_idf %>%
  arrange(desc(tf_idf)) %>%
  mutate(word = factor(word, levels = rev(unique(word)))) %>%
  group_by(author) %>%
  top_n(12, tf_idf) %>%
  ungroup() %>%
  ggplot(aes(word, tf_idf, fill = author),bin=2) +
  geom_col() +
  labs(x = NULL, y = "TF-IDF") +
  theme(legend.position = "none") +
  facet_wrap(~ author, ncol = 1, scales = "free")
```



Let’s go to find the emotional words here! I can see ‘perfect’ from Edgar, ‘reef’ from HP, and ‘feelings’, ‘misery’ from Mary. I find that relatively, Mary tends to use more emotional words. However, there are still not so many emotional words here. And I am also confused about what these words are used for. So, in the next part, I would like to do the bigrams and trigrams. And maybe sentiment analysis later…

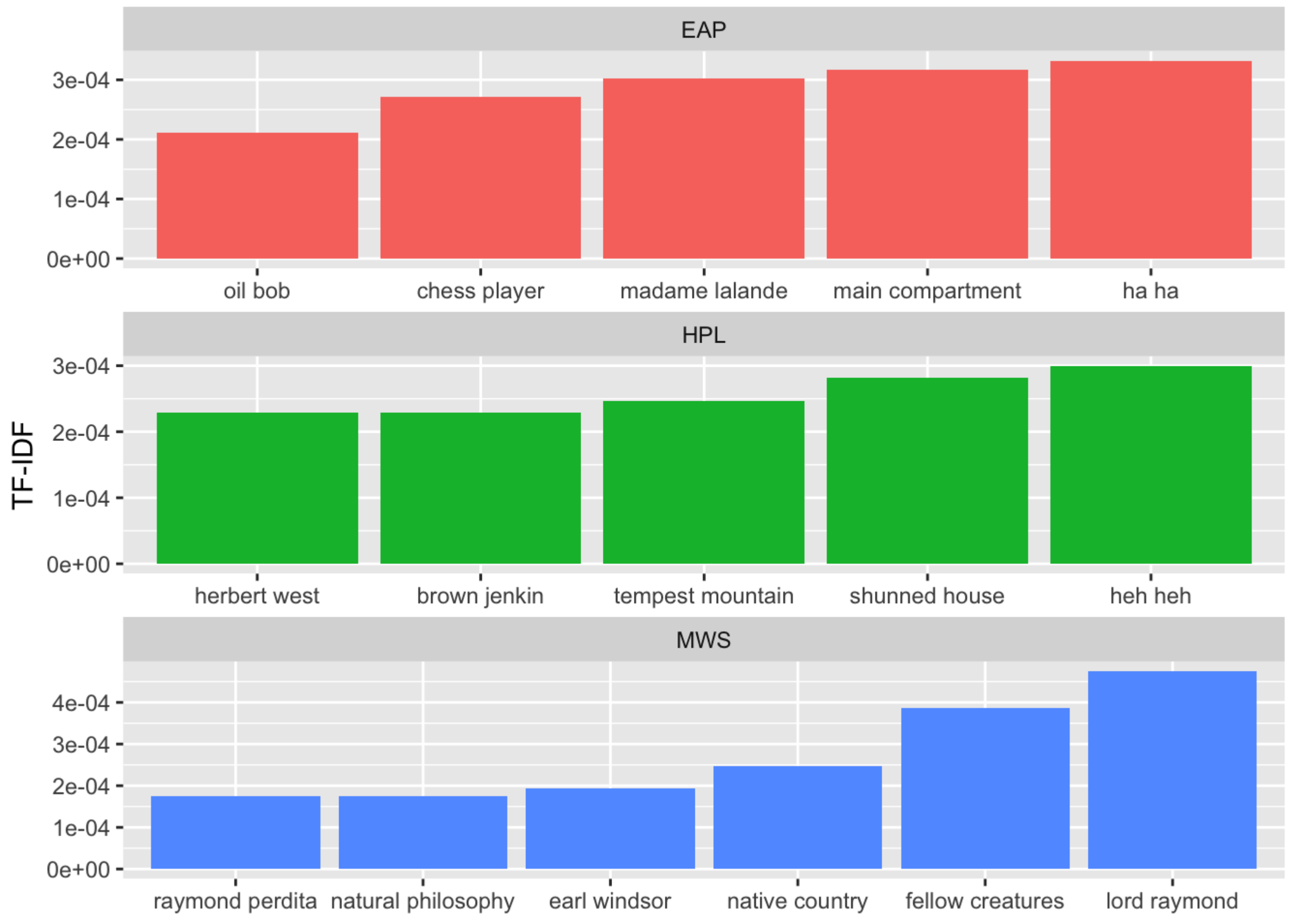
Part 4: Bigrams and trigrams

In this part, I want to use bigrams and trigrams to get the meaning of the sentence by separating it into small pieces. Bigrams and trigrams would help me to filter out most name words.

Bigrams

Bigram can regard words two at a time. Each two adjacent words create a bigram, such as “I read”, “read a”, “a book”, “book about”, “about the”, “the history”, “history of”, “of America”.

```
tfidf_bigram <- spooky_wrd %>%
  select(author, word) %>%
  unnest_tokens(bigram, word, token = "ngrams", n = 2) %>%
  separate(bigram, c("w1", "w2"), sep = " ") %>%
  filter(!w1 %in% stop_words$word) %>%
  filter(!w2 %in% stop_words$word) %>%
  unite(bigram, w1, w2, sep = " ") %>%
  count(author, bigram) %>%
  bind_tf_idf(bigram, author, n) %>%
  arrange(desc(tf_idf))


tfidf_bigram %>%
  arrange(desc(tf_idf)) %>%
  mutate(bigram = factor(bigram, levels = rev(unique(bigram)))) %>%
  group_by(author) %>%
  top_n(5, tf_idf) %>%
  ungroup() %>%
  ggplot(aes(bigram, tf_idf, fill = author)) +
  geom_col() +
  labs(x = NULL, y = "TF-IDF") +
  facet_wrap(~ author, ncol = 1, scales = "free")+
  theme(legend.position = "none")
```

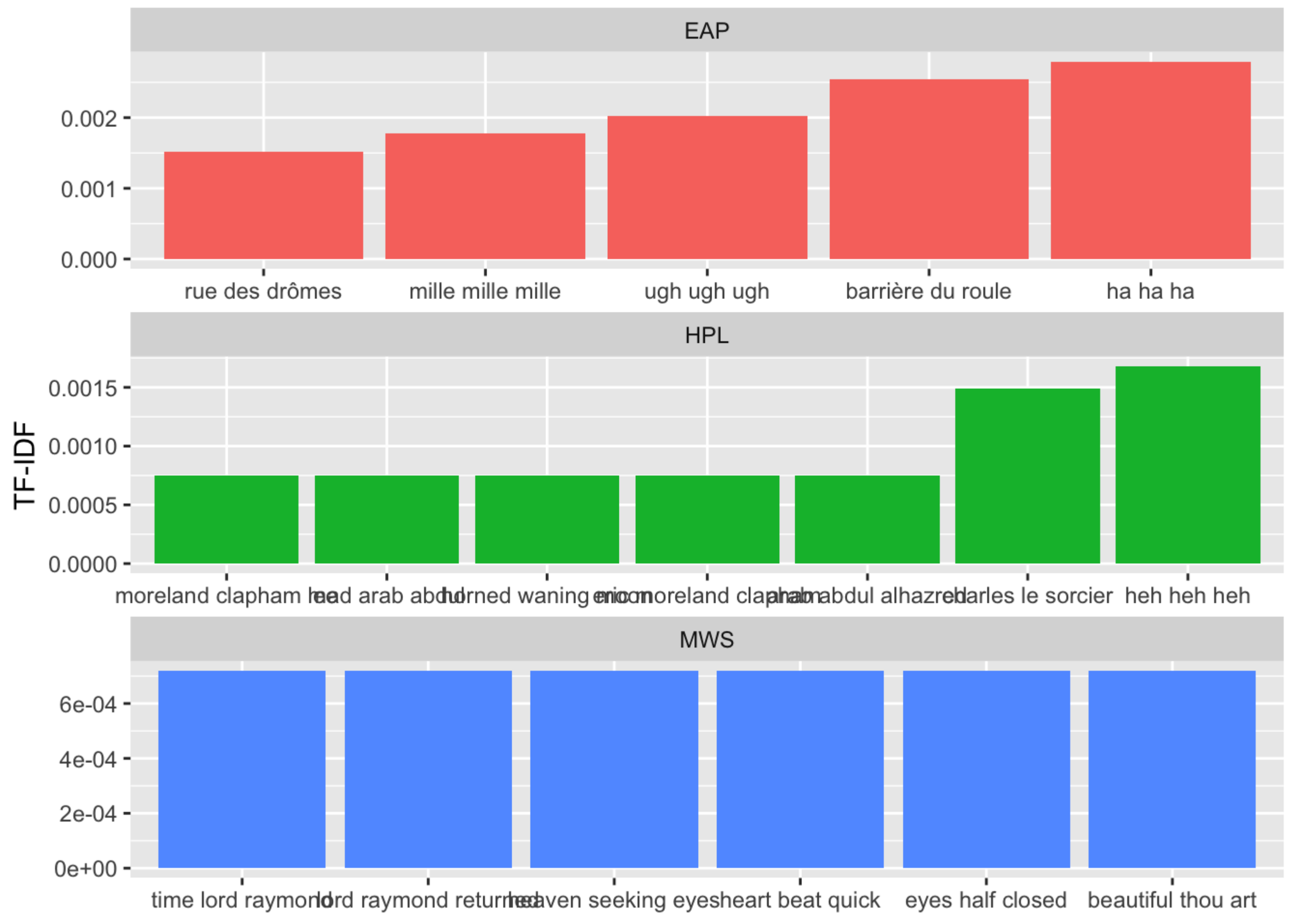


Now, we filter out all the name words here. That’s great! I find that Edgar and HP use more joy words than Mary, because they use ‘ha ha’, ‘heh heh’ more frequently. For Mary, it seems that she really likes ‘lord raymond’!

Trigrams

Trigrams can regard words three at a time. Each three adjacent words create a trigram, such as “I read a”, “read a book”, “a book about”, “book about the”, “about the history”, “the history of”, “history of America”.

```
tfidf_trigram <- spooky %>%
  select(author, text) %>%
  unnest_tokens(trigram, text, token = "ngrams", n = 3) %>%
  separate(trigram, c("w1", "w2", "w3"), sep = " ") %>%
  filter(!w1 %in% stop_words$word) %>%
  filter(!w2 %in% stop_words$word) %>%
  filter(!w3 %in% stop_words$word) %>%
  unite(trigram, w1, w2, w3, sep = " ") %>%
  count(author, trigram) %>%
  bind_tf_idf(trigram, author, n) %>%
  arrange(desc(tf_idf))




tfidf_trigram %>%
  arrange(desc(tf_idf)) %>%
  mutate(trigram = factor(trigram, levels = rev(unique(trigram)))) %>%
  group_by(author) %>%
  top_n(5, tf_idf) %>%
  ungroup() %>%
  ggplot(aes(trigram, tf_idf, fill = author)) +
  geom_col() +
  labs(x = NULL, y = "TF-IDF") +
  theme(legend.position = "none") +
  facet_wrap(~ author, ncol = 1, scales = "free")
```



I find that Edgar tends to use more reduplicated words, such as ‘mile mile mile’, ‘ugh ugh ugh’ and ‘ha ha ha’. It is a unique writing style of Edgar. Mary focuses more on the mental activities of the characters because she talks more about the eyes and hearts. Maybe it is because Mary is the only female author here, so her work has more detailed descriptions than others.

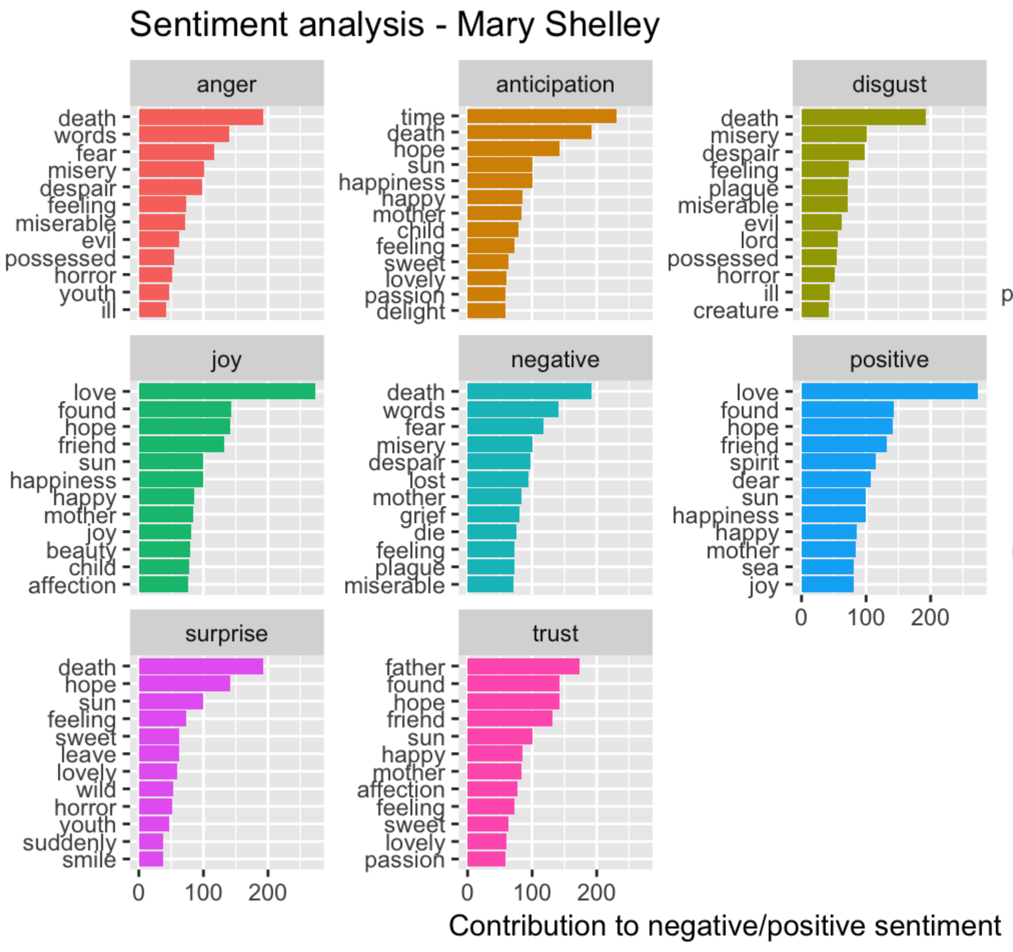
Part 5: Sentiment analysis

What is sentiment analysis? Sentiment Analysis is the process of determining whether a piece of writing is positive, negative or neutral. It’s also known as opinion mining, deriving the opinion or attitude of a speaker. I use sentiment analysis to explorer deeper into the fear words and joy words of each author.

Sentiment analysis of each author (fear words VS joy words)

In the following pictures, I get specific sentiment lexicons of each author. Although there are so many categories here, I just care about the fear words and joy words. First, let’s look at Mary’s!

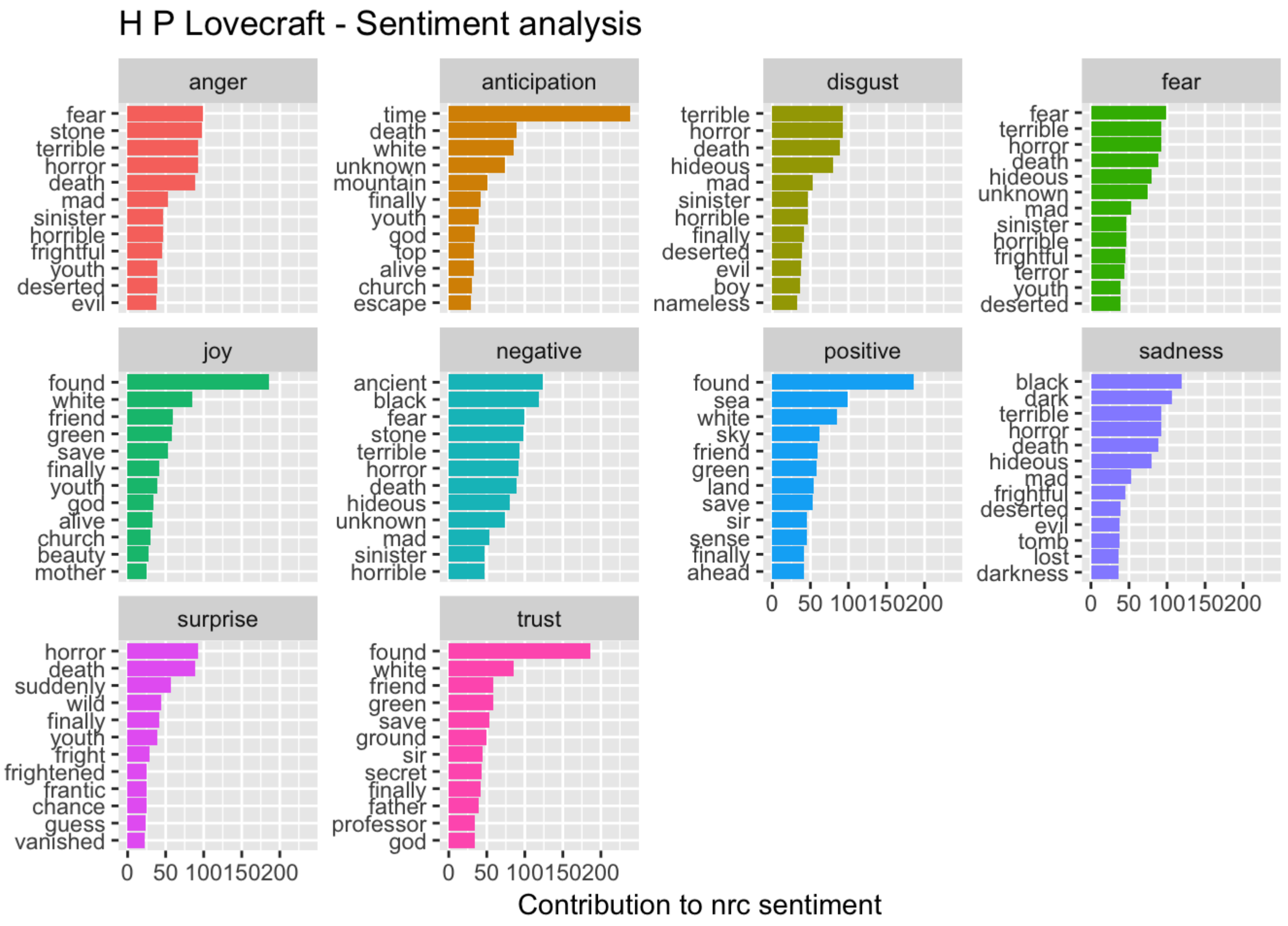
```r
spooky_wrd %>%
  filter(author == "MWS") %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup() %>%
  group_by(sentiment) %>%
  top_n(12, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to negative/positive sentiment", x = NULL) +
  coord_flip() +
  ggtitle("Sentiment analysis - Mary Shelley")
```



For Mary, the fear words are less than joy words. It just confirms my guess at 3.1 that Mary are more likely to use joy words than other two authors. She uses fear words like ‘death’, ‘fear’, ‘misery’, etc. She uses joy words like ‘love’, ‘found’, ‘hope’, etc.

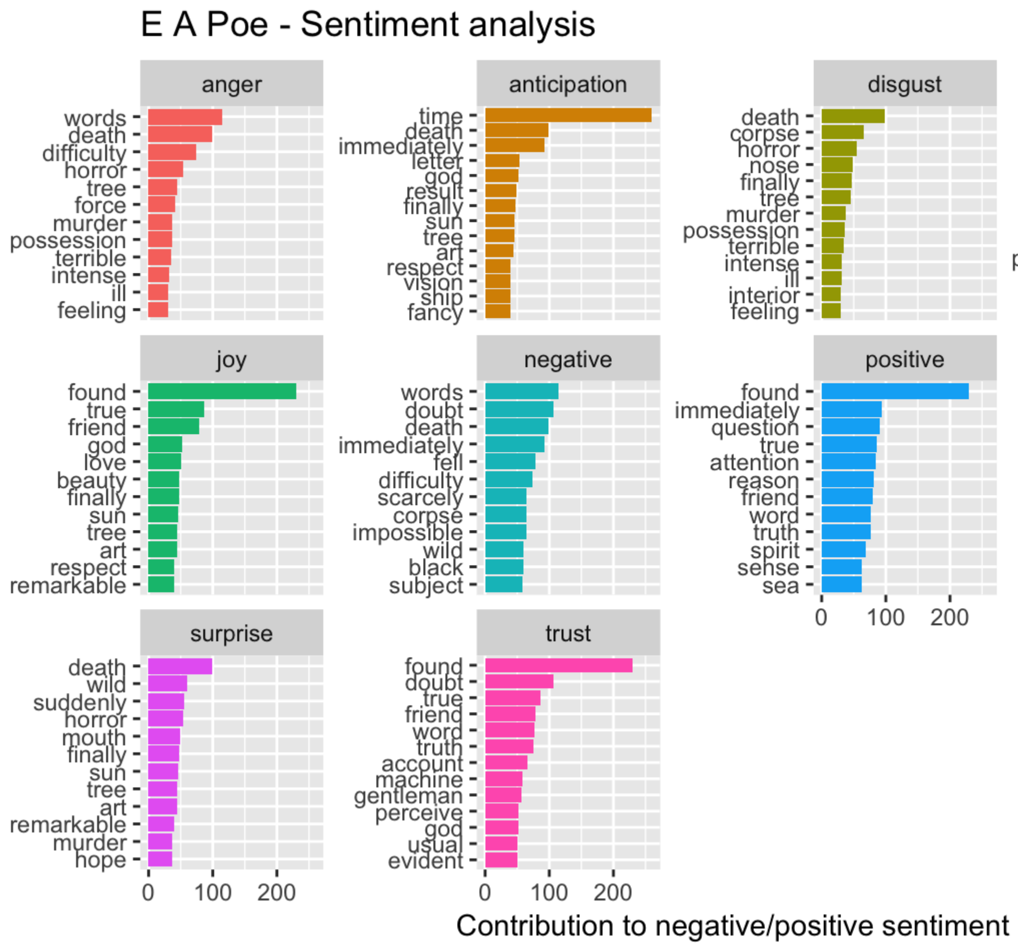
Then let’s look at HP’s!

```
spooky_wrd %>%
  filter(author == "HPL") %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup() %>%
  group_by(sentiment) %>%
  top_n(12, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to nrc sentiment", x = NULL) +
  coord_flip() +
  ggtitle("H P Lovecraft - Sentiment analysis")
```



In general, HP uses fear words more often than joy words! He uses fear words like ‘fear’, ‘terrible’, ‘horror’, etc. He uses joy words like ‘found’, ‘white’, ‘friend’, etc. In addtion, it is obvious that he uses ‘found’ much more frequently than other words. So I conclude that HP’s novel also includes suspenseful stories.

```
spooky_wrd %>%
  filter(author == "EAP") %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  ungroup() %>%
  group_by(sentiment) %>%
  top_n(12, n) %>%
  ungroup() %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~sentiment, scales = "free_y") +
  labs(y = "Contribution to negative/positive sentiment", x = NULL) +
  coord_flip() +
  ggtitle("E A Poe - Sentiment analysis")
```



Wow! Edgar uses even more ‘frequent’s than HP. It seems that they both love suspential plots and their writing styles may be similar. Edagr uses fear words like ’doubt’, ‘death’, ‘difficulty’, etc. He uses joy words like ‘found’, ‘true’, ‘friend’, etc. The joy words used by Edgar are also similar to those used by HP.

Comparison wordcloud

A word cloud is a graphical representation of word frequency. Instead of making a simple wordcloud, I want to make a comparison one, that is, to compare all these different emotional words in a single word cloud.

```
spooky_wrd %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  count(word, sentiment, sort = TRUE) %>%
  reshape2::acast(word ~ sentiment, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("black", "red"), max.words = 100)
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): black could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): finally could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): misery could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): excited could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): possessed could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): feeling could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): curiosity could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): sudden could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): happy could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): plague could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): horrible could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): green could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): mystery could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): save could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): miserable could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): late could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): remarkable could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): mother could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): creature could not be fit on page. It will not be plotted.
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): impossible could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): scarcely could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): nose could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): spirits could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): delight could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): fancy could not be fit on page. It will not be plotted.
```

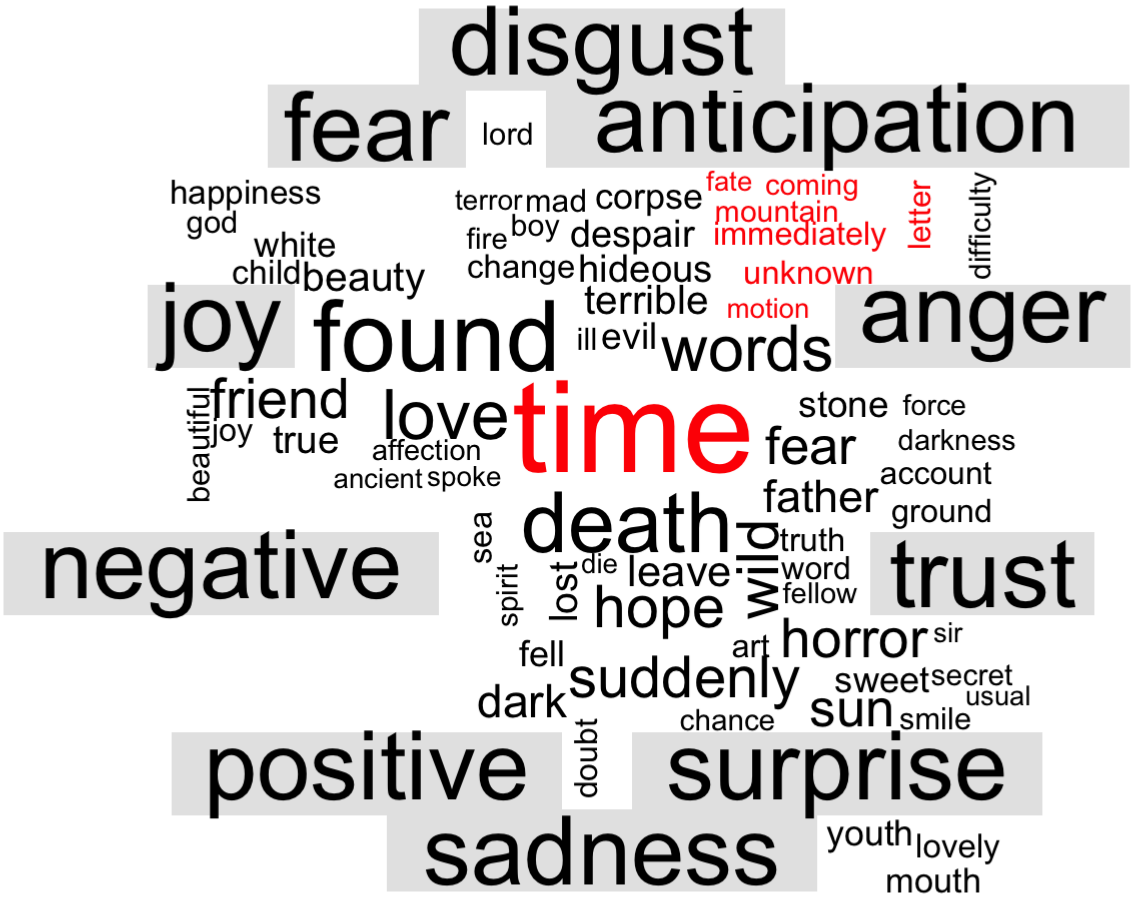
```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): tree could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): deserted could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): guess could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): grief could not be fit on page. It will not be plotted.
```

```
## Warning in comparison.cloud(., colors = c("black", "red"), max.words =
## 100): companion could not be fit on page. It will not be plotted.
```

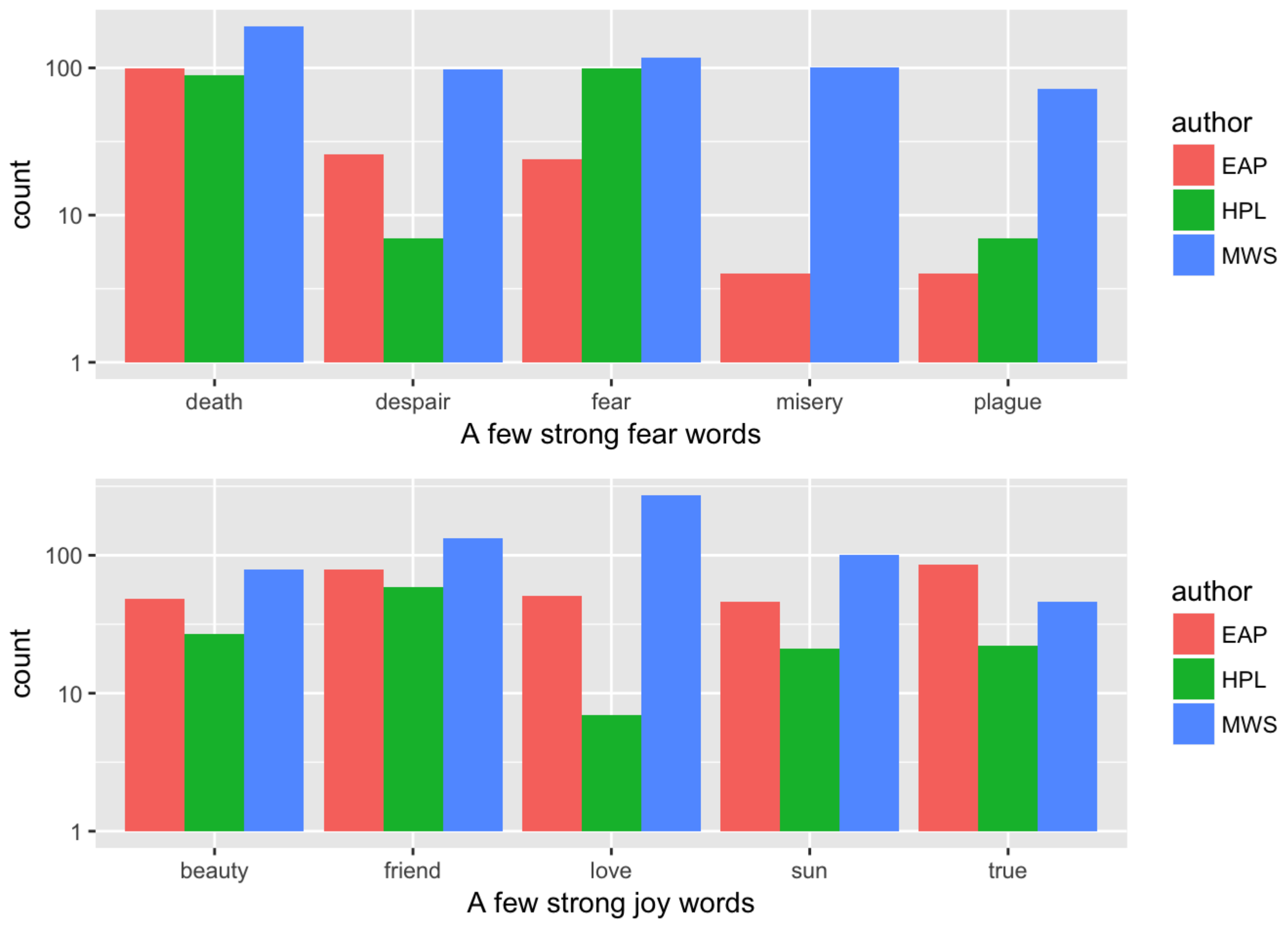


From this picture, we can find that although we do analysis on the spooky text, fear words are only a small fraction of the whole text. So I decide to pick some strong fear words and joy words randomly and compare the frequency of these words among three authors.

A few strong fear words VS joy words

I choose these fear words and joy words purely randomly (not influenced by the outcome above) and want to see the frequency of words within the same category.

```
#choose five fear words randomly
fear_wrd <- c("death", "fear", "misery", "despair", "plague")


fear<-spooky_wrd %>%
  filter(word %in% fear_wrd)%>%
  ggplot(aes(word, fill = author)) +
  geom_bar(position = "dodge") +
  scale_y_log10() +
  labs(x = "A few strong fear words")

#choose five joy words randomly
joy_wrd <- c("true", "friend", "love", "beauty", "sun")

joy<-spooky_wrd %>%
  filter(word %in% joy_wrd)%>%
  ggplot(aes(word, fill = author)) +
  geom_bar(position = "dodge") +
  scale_y_log10() +
  labs(x = "A few strong joy words")

multiplot(fear, joy, cols=1)
```



For the fear words, I find that Edgar uses ‘death’, ‘deapair’ and ‘fear’ more frequently. HP doesn’t like the word ‘misery’. The frequency of these fear words used by Mary are very similar.

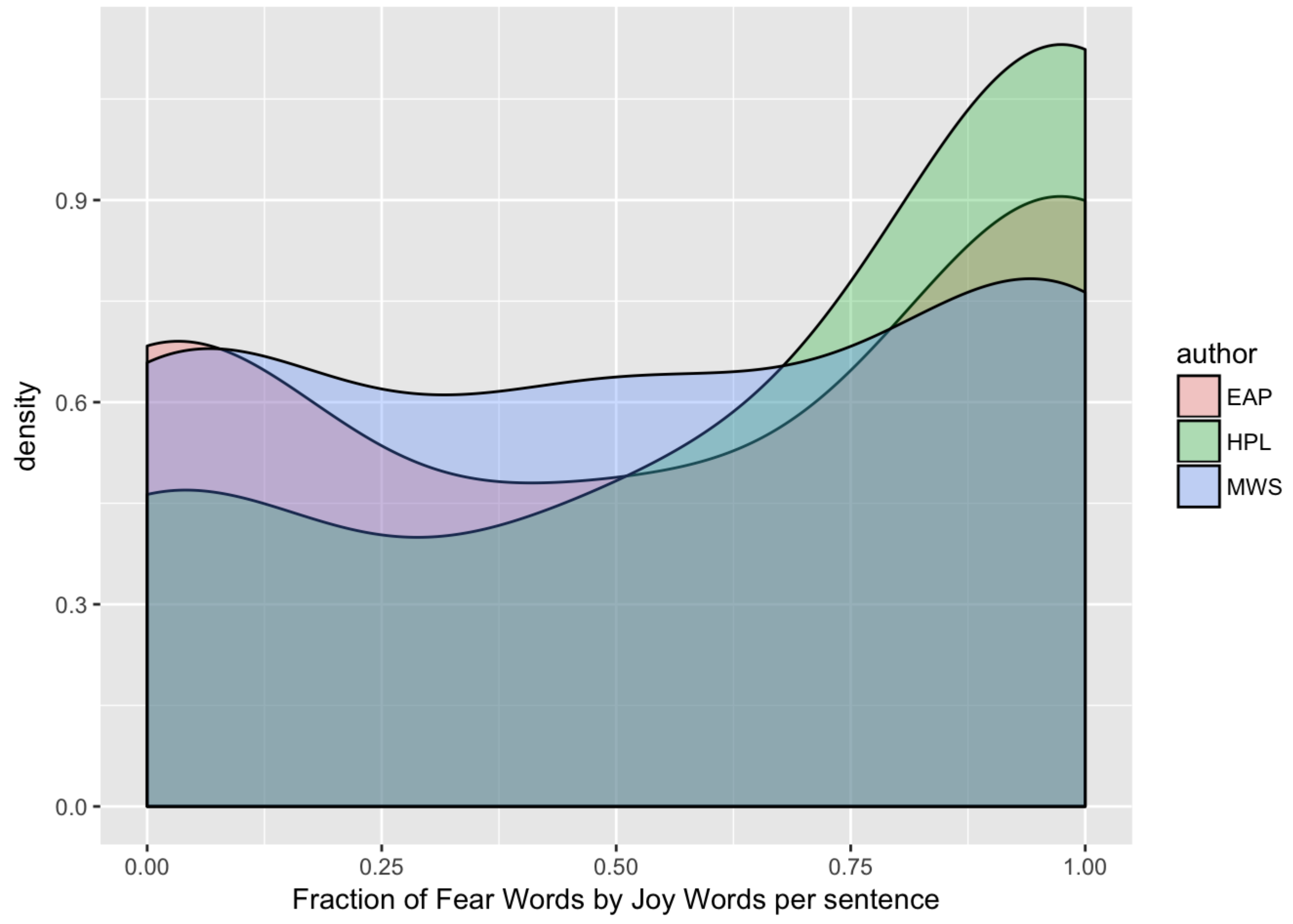
For the joy words, I find that HP doesn’t like to use the word ‘love’. However, Mary uses it very often. The frequency of these joy words used by Mary and Edgar are very similar.

Fraction of fear words

Here I want to get the fraction of fear words by joy words persentence. First, I choose all the fear words and count their fraction among the sample of fear terms plus joy terms, which is # fear / (# fear + # joy). Then I plot the distribution of the fraction of fear words for the three authors:

```
spooky_wrd %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  group_by(author, id, sentiment) %>%
  count() %>%
  spread(sentiment, n, fill = 0) %>%
  group_by(author, id) %>%
  summarise(fear = sum(fear),
            joy = sum(joy)) %>%
  arrange(id) %>%
  mutate(frac_fear = fear/(fear + joy)) %>%
  ggplot(aes(frac_fear, fill = author)) +
  geom_density(bw = .2, alpha = 0.3) +
  theme(legend.position = "right") +
  labs(x = "Fraction of Fear Words by Joy Words per sentence")
```

```
## Warning: Removed 5210 rows containing non-finite values (stat_density).
```



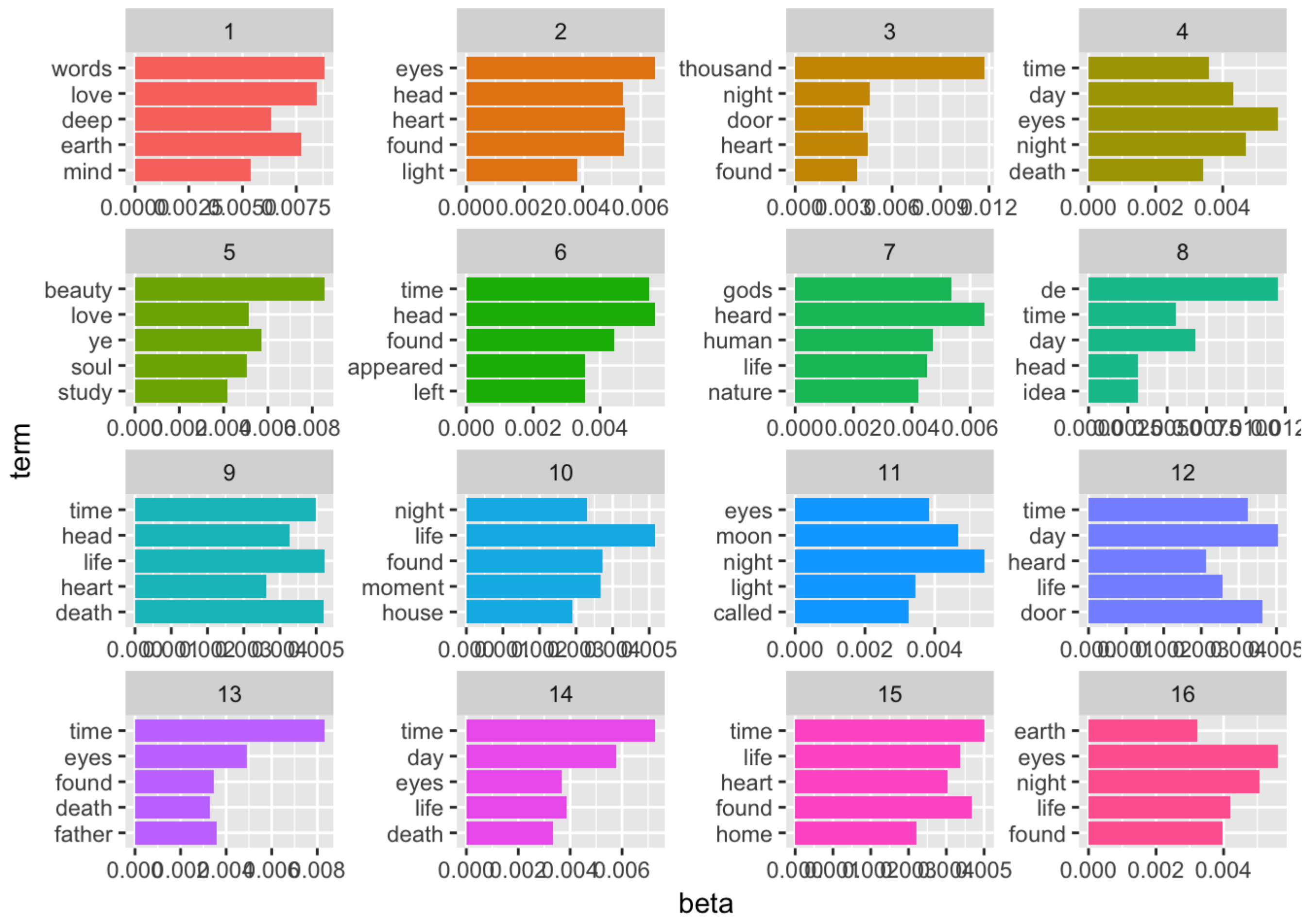
The distribution of the fraction of fear words per sentence is clearly skewed towards larger values for HP (green) than in the case of MWS and EAP. The difference between Shelley and Poe is more subtle.

Part 6: Topic modeling

A ‘topic’ consists of a cluster of words that frequently occur together. In this part, I use topic modeling to connect words with similar meanings and distinguish between uses of words with multiple meanings.

LDA and DTM

```
# Counts how many times each word appears in each sentence
sent_wrd_freqs <- count(spooky_wrd, id, word)

# Creates a DTM matrix
spooky_wrd_tm <- cast_dtm(sent_wrd_freqs, id, word, n)
spooky_wrd_lda    <- LDA(spooky_wrd_tm, k = 16, control = list(seed = 1234))
spooky_wrd_topics <- tidy(spooky_wrd_lda, matrix = "beta")

spooky_wrd_topics %>%
  group_by(topic) %>%
  top_n(5, beta) %>%
  ungroup() %>%
  arrange(topic, -beta) %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free", ncol = 4) +
  coord_flip()
```

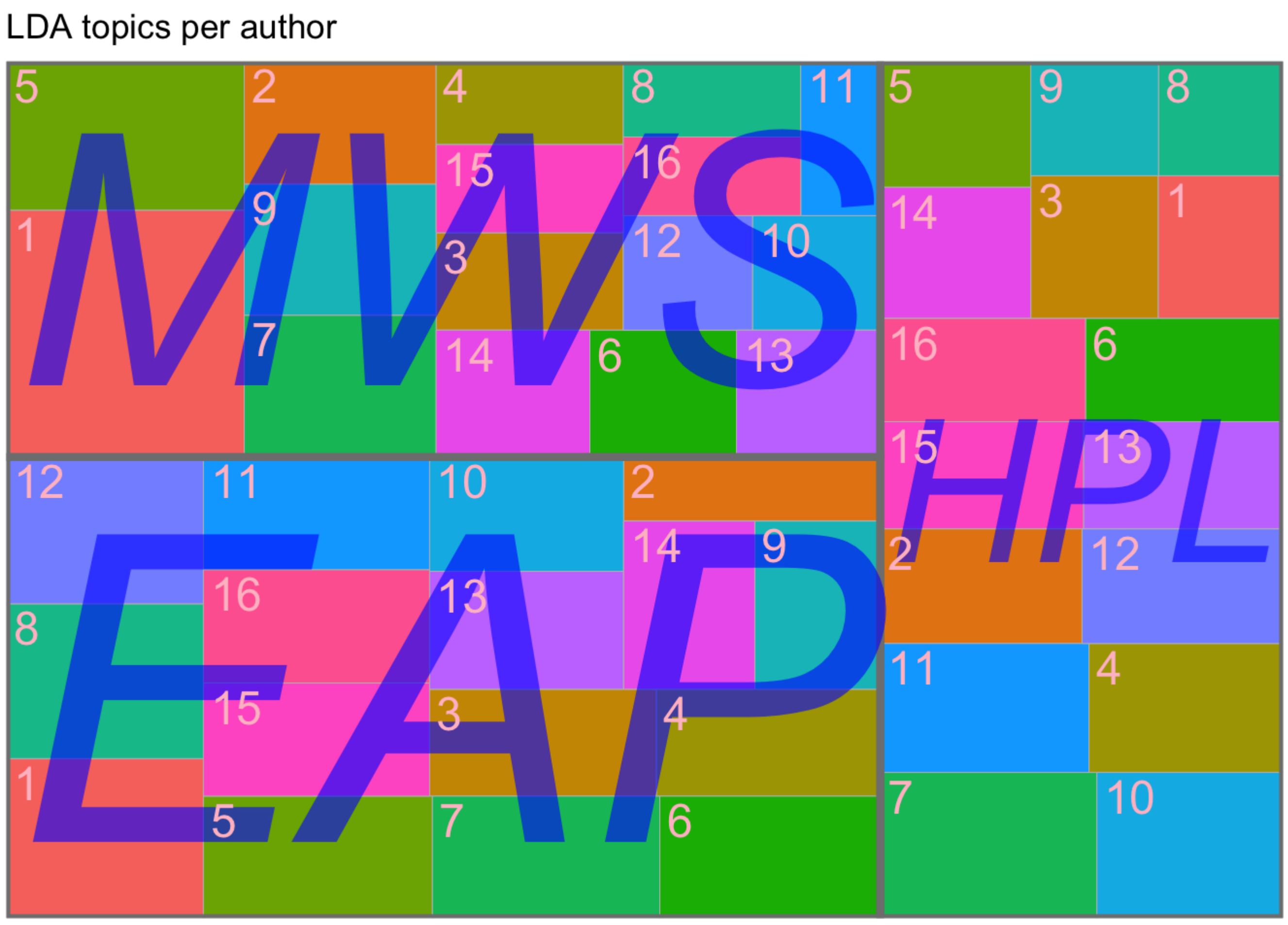


I grab 5 words for each topic and here are 16 top topics. I see that for instance topic 1 contains words such as “love” and “mind”, while topic 4 prominently features “death” and topic 5 has the word “beauty”. We also see several overlaps between topics such as “found” or “time”.

Sentence topics

I am also curious about how how these topics relate to our three authors. So I build a treemap here.

```
t1_tdocs <- tidy(spooky_wrd_lda, matrix = "gamma")


t1_tdocs %>%
  left_join(spooky_wrd, by = c("document" = "id")) %>%
  select(-word) %>%
  mutate(topic = as.factor(topic)) %>%
  group_by(document) %>%
  top_n(1, gamma) %>%
  ungroup() %>%
  group_by(author, topic) %>%
  count() %>%
  ungroup() %>%
  ggplot(aes(area = n, fill = topic, label = topic, subgroup = author)) +
  geom_treemap() +
  geom_treemap_subgroup_border() +
  geom_treemap_subgroup_text(place = "centre", grow = T, alpha = 0.6, colour =
                             "blue", fontface = "italic", min.size = 0) +
  geom_treemap_text(colour = "pink", place = "topleft", reflow = T) +
  theme(legend.position = "null") +
  ggtitle("LDA topics per author")
```

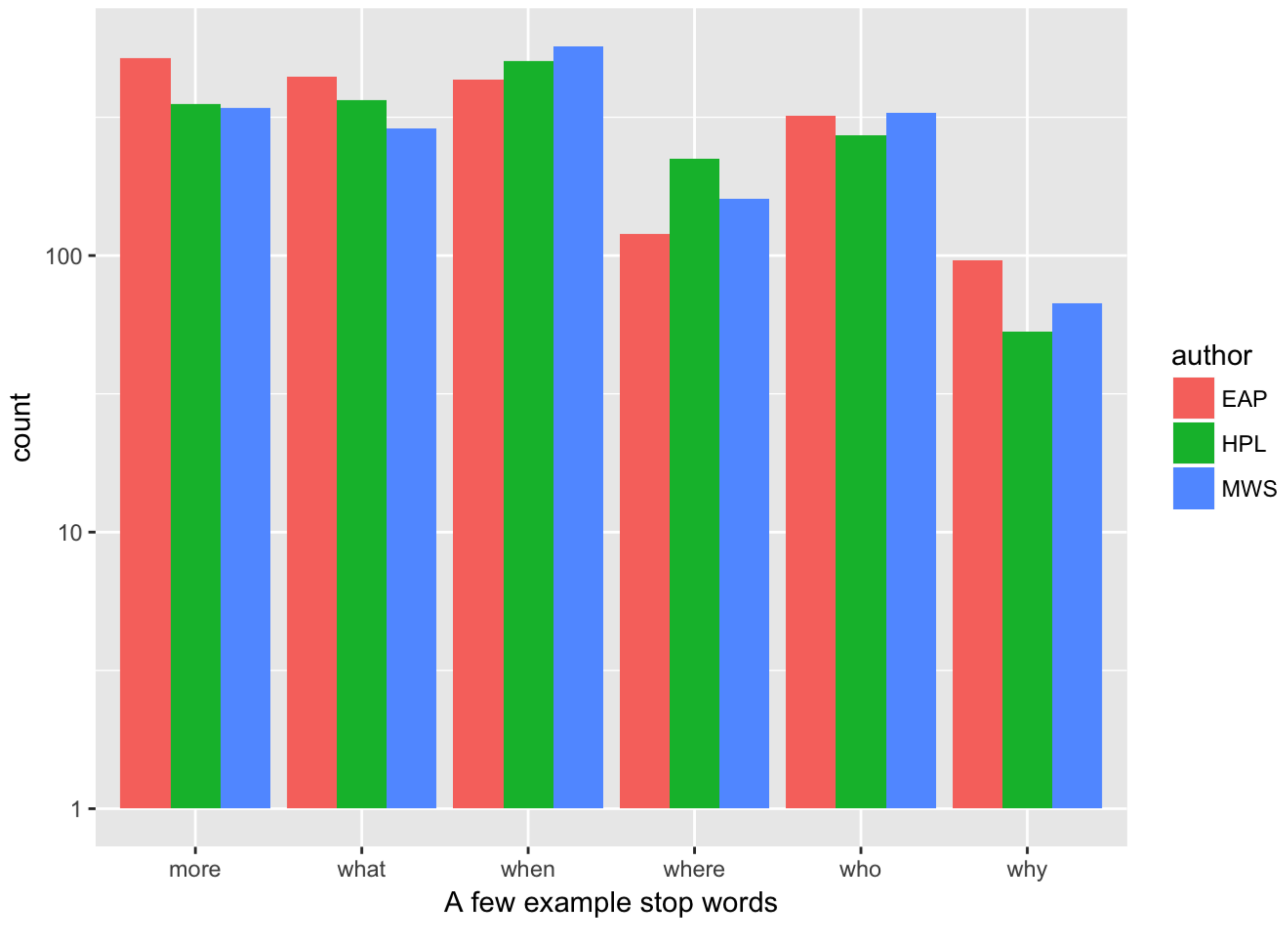


There are 17 topic related to Mary; 16 topics related to HP; 16 topics related to Edgar. I find that topic 1 and topic 5 are more related to Mary's novel ;topic 8 and topic 15 are more realted to Edgar's nove.

Part 7: Stop words

In the end, I want to find some interesting things about the stop words. Also I choose 6 stop words randomly and see the frequency of them among three authors.

```
# I choose 'why', 'what', 'who', 'when', 'where' and 'more'
my_stop <- c("why", "what", "who", "when", "where", "more")

spooky %>%
  unnest_tokens(word, text) %>%
  filter(word %in% my_stop)%>%
  ggplot(aes(word, fill = author)) +
  geom_bar(position = "dodge") +
  scale_y_log10() +
  labs(x = "A few example stop words")
```



I can not find too many differences on the use of these different stop words among the authors. Thus, it is reasonable to filter them out at first, as what I did.

png("/Users/xiangyu/Documents/GitHub/spring2018-project1-ellip123/figs/stop_words.png")