An R Notebook Data Story on Horror Stories

Toby

01/30/2018

# Section 0 Preparations, install and load the needed packages

packages.used <- c("ggplot2", "dplyr", "tibble", "tidyr", "stringr", "tidytext", "topicmodels", "wordcloud", "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)

##   
## 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

library(tibble)  
library(tidyr)  
library(stringr)  
library(tidytext)  
library(topicmodels)  
library(wordcloud)

## Loading required package: RColorBrewer

library(ggridges)  
  
  
source("../lib/multiplot.R")

# Section 1: Data Preparation

## Data load and overview

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

Data overview

head(spooky)

## id  
## 1 id26305  
## 2 id17569  
## 3 id11008  
## 4 id27763  
## 5 id12958  
## 6 id22965  
## text  
## 1 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.  
## 2 It never once occurred to me that the fumbling might be a mere mistake.  
## 3 In his left hand was a gold snuff box, from which, as he capered down the hill, cutting all manner of fantastic steps, he took snuff incessantly with an air of the greatest possible self satisfaction.  
## 4 How lovely is spring As we looked from Windsor Terrace on the sixteen fertile counties spread beneath, speckled by happy cottages and wealthier towns, all looked as in former years, heart cheering and fair.  
## 5 Finding nothing else, not even gold, the Superintendent abandoned his attempts; but a perplexed look occasionally steals over his countenance as he sits thinking at his desk.  
## 6 A youth passed in solitude, my best years spent under your gentle and feminine fosterage, has so refined the groundwork of my character that I cannot overcome an intense distaste to the usual brutality exercised on board ship: I have never believed it to be necessary, and when I heard of a mariner equally noted for his kindliness of heart and the respect and obedience paid to him by his crew, I felt myself peculiarly fortunate in being able to secure his services.  
## author  
## 1 EAP  
## 2 HPL  
## 3 EAP  
## 4 MWS  
## 5 HPL  
## 6 MWS

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",...

change the data type of author

sum(is.na(spooky))

## [1] 0

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

This notebook was prepared with the following environmental settings.

print(R.version)

## \_   
## platform x86\_64-w64-mingw32   
## arch x86\_64   
## os mingw32   
## system x86\_64, mingw32   
## status   
## major 3   
## minor 4.3   
## year 2017   
## month 11   
## day 30   
## svn rev 73796   
## language R   
## version.string R version 3.4.3 (2017-11-30)  
## nickname Kite-Eating Tree

# Section 2: Data Cleaning

We use the unnest\_token to drop the punctuation and transform the words to lower cases, remove the stop words from the data to focus on the really important words. Besides, we will have the data related to certain author.

spooky\_wrd<- unnest\_tokens(spooky,word,text)  
spooky\_wrd\_withstop<-spooky\_wrd # data with stop words  
spooky\_wrd<-anti\_join(spooky\_wrd,stop\_words,by="word") # without stop words  
EAP<-filter(spooky\_wrd\_withstop,author=="EAP")  
MWS<-filter(spooky\_wrd\_withstop,author=="MWS")  
HPL<-filter(spooky\_wrd\_withstop,author=="HPL")  
EAP\_nstop<-filter(spooky\_wrd,author=="EAP")  
MWS\_nstop<-filter(spooky\_wrd,author=="MWS")  
HPL\_nstop<-filter(spooky\_wrd,author=="HPL")

## Word Colud Generation

We generate one word cloud graph of the whole spooky file.

png('../figs/whole\_cloud.png')  
 spooky\_wrd %>%  
 count(word) %>%  
 with(wordcloud(word, n, max.words = 50, color = c("purple4", "red4", "black")))  
dev.off()

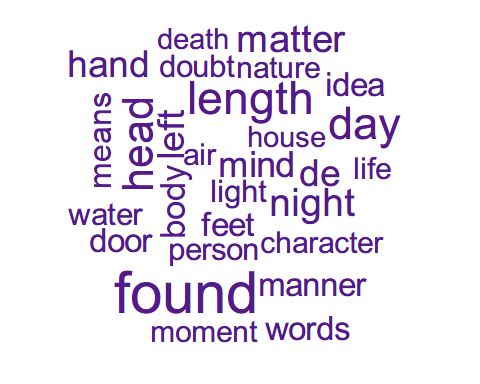
## png   
## 2

We draw the worldcloud for EAP

words\_EAP <- count(group\_by(EAP\_nstop, word))$word  
freqs\_EAP <- count(group\_by(EAP\_nstop, word))$n  
#png('../figs/EAP\_cloud.png')  
wordcloud(words\_EAP, freqs\_EAP, max.words = 30, color = c("purple4"))

## Warning in wordcloud(words\_EAP, freqs\_EAP, max.words = 30, color =  
## c("purple4")): time could not be fit on page. It will not be plotted.

## Warning in wordcloud(words\_EAP, freqs\_EAP, max.words = 30, color =  
## c("purple4")): eyes could not be fit on page. It will not be plotted.



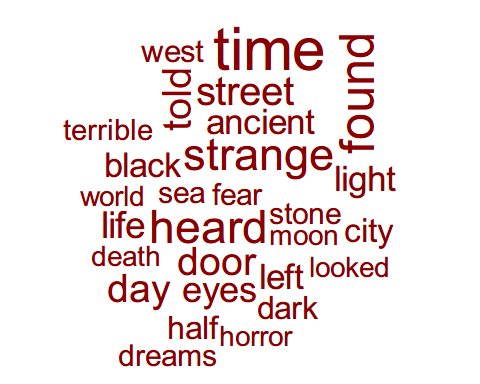
#dev.off()

We draw the worldcloud for HPL

words\_HPL <- count(group\_by(HPL\_nstop, word))$word  
freqs\_HPL <- count(group\_by(HPL\_nstop, word))$n  
#png('../figs/HPL\_cloud.png')  
wordcloud(words\_HPL, freqs\_HPL, max.words = 30, color = c("red4"))

## Warning in wordcloud(words\_HPL, freqs\_HPL, max.words = 30, color =  
## c("red4")): night could not be fit on page. It will not be plotted.

## Warning in wordcloud(words\_HPL, freqs\_HPL, max.words = 30, color =  
## c("red4")): house could not be fit on page. It will not be plotted.



#dev.off()

We draw the worldcloud for MWS

words\_MWS <- count(group\_by(MWS\_nstop, word))$word  
freqs\_MWS <- count(group\_by(MWS\_nstop, word))$n  
#png('../figs/MWS\_cloud.png')  
wordcloud(words\_MWS, freqs\_MWS, max.words = 30, color = c("blue4"))



#dev.off()

We can see that different authors have different preferences for using words. For Edgar Allan Poe, he most frequently use words like “mind” and “manner”. HP Lovecraft likes to use words “strange” and “horror”. While for Mary Shelley, her most frequent words are “love” and “death”.These frequent words are pretty normal for the horror authors.

Data Visualization and comparation, We have done some numerical summaries of the data to provide some nice visualizations.

p1 <- ggplot(spooky) +  
 geom\_bar(aes(author, fill = author)) +  
 theme(legend.position = "none") # the whole spooky data analysis  
spooky$sen\_length <- str\_length(spooky$text)  
spooky$author2<-as.character(spooky$author)  
head(spooky$sen\_length)

## [1] 231 71 200 206 174 468

EAPsen\_length<-mean(((filter(spooky,author2=="EAP"))$sen\_length))  
MWSsen\_length<-mean(((filter(spooky,author2=="MWS"))$sen\_length))  
HPLsen\_length<-mean(((filter(spooky,author2=="HPL"))$sen\_length))  
dt1=as.data.frame(matrix(c("EAP","MWS","HPL"),nrow=3,ncol=1))  
colnames(dt1)<-"Author"  
dt1$sentence\_length<-c(EAPsen\_length,MWSsen\_length,HPLsen\_length)  
p2 <- ggplot(dt1,aes(x=Author,y=sentence\_length,fill=Author))+  
 geom\_bar(stat="identity", position=position\_dodge())  
spooky\_wrd$word\_length <- str\_length(spooky\_wrd$word)  
EAPwrd\_length<-mean(((filter(spooky\_wrd,author=="EAP"))$word\_length))  
MWSwrd\_length<-mean(((filter(spooky\_wrd,author=="MWS"))$word\_length))  
HPLwrd\_length<-mean(((filter(spooky\_wrd,author=="HPL"))$word\_length))  
dt1$word\_length<-c(EAPwrd\_length,MWSwrd\_length,HPLwrd\_length)  
p3 <- ggplot(dt1,aes(x=Author,y=word\_length,fill=Author))+  
 geom\_bar(stat="identity", position=position\_dodge())  
layout <- matrix(c(1, 2, 1, 3), 2, 2, byrow = TRUE)  
png('../figs/author\_compare.png')  
multiplot(p1, p2, p3, layout = layout)

## Loading required package: grid

dev.off()

## png   
## 2

So in the spooky data, Edgar Allan Poe has the most sentences and HP Lovecraft has the least. As for the average sentence length, HP Lovecraft writes the longgest sentence while Edgar Allan Poe writes the shortest sentence. When it comes to the average word length, we can see that Edgar Allan Poe uses the longgest word and there is no clear difference in word length between HP Lovecraft and Mary Shelley.

we decide to take a look at the He/She, to explore over the gender differenc. (This requires the stop words not be removed as “he” or “she” are usually taken as the stop words)

male<-c("he","him","his")  
female<-c("she","her")  
heEAP<-dim(filter(EAP,word %in% male))[1]  
sheEAP<-dim(filter(EAP,word %in% female))[1]  
heHPL<-dim(filter(HPL,word %in% male))[1]  
sheHPL<-dim(filter(HPL,word %in% female))[1]  
heMWS<-dim(filter(MWS,word %in% male))[1]  
sheMWS<-dim(filter(MWS,word %in% female))[1]  
dt2=as.data.frame(matrix(c(rep("male",3),rep("female",3)),nrow=6,ncol=1))  
colnames(dt2)<-"Gender"  
dt2$var<-c(rep(c("EAP","HPL","MWS"),2))  
dt2$count<-c(heEAP,sheEAP,heHPL,sheHPL,heMWS,sheMWS)  
png('../figs/gender\_difference.png')  
ggplot(dt2, aes(x = var, y = count, fill = Gender))+   
 geom\_bar(stat="identity", position=position\_dodge())  
dev.off()

## png   
## 2

Both Edgar Allan Poe and HP Lovecraft have very clear preference in using masculine third-person word or feminine third-person word. Edgar Allan Poe uses the masculine third-person word more often (he,him,his) while HP Lovecraft uses the feminine third-person word (she,her) more often. Mary Shelley uses masculine third-person word and feminine third-person word almost equally (A bit more in masculine). #Sentiment Analysis In the sentiment analysis part, we want to measure what is the proportion of the sentiment was positive or negative. We first use the nrc lexicons.

nrc\_filter <- filter(get\_sentiments('nrc'), sentiment %in% c("positive","negative"))  
sentiments\_nrc <- inner\_join(spooky\_wrd, nrc\_filter, by = "word")  
EAP\_nrc<-filter(sentiments\_nrc,author=="EAP")  
HPL\_nrc<-filter(sentiments\_nrc,author=="HPL")  
MWS\_nrc<-filter(sentiments\_nrc,author=="MWS")  
EAP\_pos<-dim(filter(EAP\_nrc,sentiment=="positive"))[1]  
EAP\_neg<-dim(EAP\_nrc)[1]-EAP\_pos  
HPL\_pos<-dim(filter(HPL\_nrc,sentiment=="positive"))[1]  
HPL\_neg<-dim(HPL\_nrc)[1]-HPL\_pos  
MWS\_pos<-dim(filter(MWS\_nrc,sentiment=="positive"))[1]  
MWS\_neg<-dim(MWS\_nrc)[1]-HPL\_pos  
#plot pie graph for EAP  
dt3 = data.frame(A = c(EAP\_pos, EAP\_neg), B = c('Positive','Negative'))  
  
myLabel = as.vector(dt3$B)   
myLabel = paste(myLabel, "(", round(dt3$A / sum(dt3$A) \* 100, 2), "%) ", sep = "")   
  
p4 = ggplot(dt3, aes(x = "", y = A, fill = B)) +   
 geom\_bar(stat = "identity", width = 1) +   
 coord\_polar(theta = "y") +   
 labs(x = "", y = "EAP", title = "") +   
 theme(axis.ticks = element\_blank()) +   
 theme(legend.title = element\_blank(), legend.position = "top") +   
 scale\_fill\_discrete(breaks = dt3$B, labels = myLabel)+  
 theme(axis.text.x = element\_blank())  
#plot pie graph for HPL  
dt4 = data.frame(A = c(HPL\_pos, HPL\_neg), B = c('Positive','Negative'))  
  
myLabel = as.vector(dt4$B)   
myLabel = paste(myLabel, "(", round(dt4$A / sum(dt4$A) \* 100, 2), "%) ", sep = "")   
  
p5 = ggplot(dt4, aes(x = "", y = A, fill = B)) +   
 geom\_bar(stat = "identity", width = 1) +   
 coord\_polar(theta = "y") +   
 labs(x = "", y = "HPL", title = "") +   
 theme(axis.ticks = element\_blank()) +   
 theme(legend.title = element\_blank(), legend.position = "top") +   
 scale\_fill\_discrete(breaks = dt4$B, labels = myLabel)+  
 theme(axis.text.x = element\_blank())  
#plot pie graph for MWS  
dt5 = data.frame(A = c(MWS\_pos, MWS\_neg), B = c('Positive','Negative'))  
  
myLabel = as.vector(dt5$B)   
myLabel = paste(myLabel, "(", round(dt5$A / sum(dt5$A) \* 100, 2), "%) ", sep = "")   
  
p6 = ggplot(dt5, aes(x = "", y = A, fill = B)) +   
 geom\_bar(stat = "identity", width = 1) +   
 coord\_polar(theta = "y") +   
 labs(x = "", y = "MWS", title = "") +   
 theme(axis.ticks = element\_blank()) +   
 theme(legend.title = element\_blank(), legend.position = "top") +   
 scale\_fill\_discrete(breaks = dt5$B, labels = myLabel)+  
 theme(axis.text.x = element\_blank())  
png('../figs/nrc\_pos.png')  
multiplot(p4,p5,p6,cols=2)  
dev.off()

## png   
## 2

Overall we can see that in different authors’ text, the negative sentiment accounts more than the positive sentiment. In Edgar Allan Poe’s article, he has more positive parts.

Now we repeat what we did in last part again, but this time we use the “bing” lexicons, we use the setiment analysis package of bing

sentiments\_bing<-inner\_join(spooky\_wrd, get\_sentiments('bing'), by = "word")  
EAP\_bing<-filter(sentiments\_bing,author=="EAP")  
HPL\_bing<-filter(sentiments\_bing,author=="HPL")  
MWS\_bing<-filter(sentiments\_bing,author=="MWS")  
EAP\_pos<-dim(filter(EAP\_bing,sentiment=="positive"))[1]  
EAP\_neg<-dim(EAP\_bing)[1]-EAP\_pos  
HPL\_pos<-dim(filter(HPL\_bing,sentiment=="positive"))[1]  
HPL\_neg<-dim(HPL\_bing)[1]-HPL\_pos  
MWS\_pos<-dim(filter(MWS\_bing,sentiment=="positive"))[1]  
MWS\_neg<-dim(MWS\_bing)[1]-HPL\_pos  
#plot pie graph for EAP  
dt3 = data.frame(A = c(EAP\_pos, EAP\_neg), B = c('Positive','Negative'))  
  
myLabel = as.vector(dt3$B)   
myLabel = paste(myLabel, "(", round(dt3$A / sum(dt3$A) \* 100, 2), "%) ", sep = "")   
  
p4 = ggplot(dt3, aes(x = "", y = A, fill = B)) +   
 geom\_bar(stat = "identity", width = 1) +   
 coord\_polar(theta = "y") +   
 labs(x = "", y = "EAP", title = "") +   
 theme(axis.ticks = element\_blank()) +   
 theme(legend.title = element\_blank(), legend.position = "top") +   
 scale\_fill\_discrete(breaks = dt3$B, labels = myLabel)+  
 theme(axis.text.x = element\_blank())  
#plot pie graph for HPL  
dt4 = data.frame(A = c(HPL\_pos, HPL\_neg), B = c('Positive','Negative'))  
  
myLabel = as.vector(dt4$B)   
myLabel = paste(myLabel, "(", round(dt4$A / sum(dt4$A) \* 100, 2), "%) ", sep = "")   
  
p5 = ggplot(dt4, aes(x = "", y = A, fill = B)) +   
 geom\_bar(stat = "identity", width = 1) +   
 coord\_polar(theta = "y") +   
 labs(x = "", y = "HPL", title = "") +   
 theme(axis.ticks = element\_blank()) +   
 theme(legend.title = element\_blank(), legend.position = "top") +   
 scale\_fill\_discrete(breaks = dt4$B, labels = myLabel)+  
 theme(axis.text.x = element\_blank())  
#plot pie graph for MWS  
dt5 = data.frame(A = c(MWS\_pos, MWS\_neg), B = c('Positive','Negative'))  
  
myLabel = as.vector(dt5$B)   
myLabel = paste(myLabel, "(", round(dt5$A / sum(dt5$A) \* 100, 2), "%) ", sep = "")   
  
p6 = ggplot(dt5, aes(x = "", y = A, fill = B)) +   
 geom\_bar(stat = "identity", width = 1) +   
 coord\_polar(theta = "y") +   
 labs(x = "", y = "MWS", title = "") +   
 theme(axis.ticks = element\_blank()) +   
 theme(legend.title = element\_blank(), legend.position = "top") +   
 scale\_fill\_discrete(breaks = dt5$B, labels = myLabel)+  
 theme(axis.text.x = element\_blank())  
png('../figs/bing\_pos.png')  
multiplot(p4,p5,p6,cols=2)  
dev.off()

## png   
## 2

Using the “bing” lexicon, the result actually doesn’t change. In text,the negative sentiment accounts more than the positive sentiment. Besides, we can see that now the neagtive sentiment is accounting more part than before. Now in Edgar Allan Poe’s article, he has more negative parts.

In this part we will try to find what are the top few emotions represented by each author and the difference

sentiments <- inner\_join(spooky\_wrd, get\_sentiments('nrc'), by = "word")  
dt6<-as.data.frame(count(sentiments, author, sentiment))  
EAP\_emotion<-filter(dt6,author=="EAP")  
HPL\_emotion<-filter(dt6,author=="HPL")  
MWS\_emotion<-filter(dt6,author=="MWS")  
p7<-ggplot(EAP\_emotion,aes(x=reorder(sentiment,n),y=n,fill=sentiment))+  
 geom\_bar(stat="identity", position=position\_dodge())+  
 coord\_flip()+  
 labs(x = NULL, y = "count")+  
 facet\_wrap(~ author)+  
 theme(legend.position = "none")  
p8<-ggplot(HPL\_emotion,aes(x=reorder(sentiment,n),y=n,fill=sentiment))+  
 geom\_bar(stat="identity", position=position\_dodge())+  
 coord\_flip()+  
 labs(x = NULL, y = "count")+  
 facet\_wrap(~ author)+  
 theme(legend.position = "none")  
p9<-ggplot(MWS\_emotion,aes(x=reorder(sentiment,n),y=n,fill=sentiment))+  
 geom\_bar(stat="identity", position=position\_dodge())+  
 coord\_flip()+  
 labs(x = NULL, y = "count")+  
 facet\_wrap(~ author)+  
 theme(legend.position = "none")  
png('../figs/emotion\_difference.png')  
multiplot(p7,p8,p9,cols=3)  
dev.off()

## png   
## 2

We can see that the top emotions are pretty different among these authors. For Edgar Allan Poe, his top emotions are “trust, fear and anticipation”. HP Lovecraft’s top emotions are “fear, sadness and trust”. Mary Shelley’s top emotions are “fear, trust and sadness”.

# Topic Modelling

In the topic modelling part, we try to visualize author topics and we choose 6 topics.

#EAP  
sent\_wrd\_freqs\_EAP <- count(filter(spooky\_wrd,author=="EAP"), id, word)  
spooky\_wrd\_tm\_EAP <- cast\_dtm(sent\_wrd\_freqs\_EAP, id, word, n)  
spooky\_wrd\_lda\_EAP <- LDA(spooky\_wrd\_tm\_EAP, k = 6, control = list(seed = 1234))  
spooky\_wrd\_topics\_EAP <- tidy(spooky\_wrd\_lda\_EAP, matrix = "beta")  
spooky\_wrd\_topics\_5\_EAP <- ungroup(top\_n(group\_by(spooky\_wrd\_topics\_EAP, topic), 5, beta))  
spooky\_wrd\_topics\_5\_EAP <- arrange(spooky\_wrd\_topics\_5\_EAP, topic, -beta)  
spooky\_wrd\_topics\_5\_EAP <- mutate(spooky\_wrd\_topics\_5\_EAP, term = reorder(term, beta))  
png('../figs/EAP\_topic.png')  
ggplot(spooky\_wrd\_topics\_5\_EAP) +  
 geom\_col(aes(term, beta, fill = factor(topic)), show.legend = FALSE) +  
 facet\_wrap(~ topic, scales = "free", ncol = 3) +  
 coord\_flip()  
dev.off()

## png   
## 2

#HPL  
sent\_wrd\_freqs\_HPL <- count(filter(spooky\_wrd,author=="HPL"), id, word)  
spooky\_wrd\_tm\_HPL <- cast\_dtm(sent\_wrd\_freqs\_HPL, id, word, n)  
spooky\_wrd\_lda\_HPL <- LDA(spooky\_wrd\_tm\_HPL, k = 6, control = list(seed = 1234))  
spooky\_wrd\_topics\_HPL <- tidy(spooky\_wrd\_lda\_HPL, matrix = "beta")  
spooky\_wrd\_topics\_5\_HPL <- ungroup(top\_n(group\_by(spooky\_wrd\_topics\_HPL, topic), 5, beta))  
spooky\_wrd\_topics\_5\_HPL <- arrange(spooky\_wrd\_topics\_5\_HPL, topic, -beta)  
spooky\_wrd\_topics\_5\_HPL <- mutate(spooky\_wrd\_topics\_5\_HPL, term = reorder(term, beta))  
png('../figs/HPL\_topic.png')  
ggplot(spooky\_wrd\_topics\_5\_HPL) +  
 geom\_col(aes(term, beta, fill = factor(topic)), show.legend = FALSE) +  
 facet\_wrap(~ topic, scales = "free", ncol = 3) +  
 coord\_flip()  
dev.off()

## png   
## 2

#MWS  
sent\_wrd\_freqs\_MWS <- count(filter(spooky\_wrd,author=="MWS"), id, word)  
spooky\_wrd\_tm\_MWS <- cast\_dtm(sent\_wrd\_freqs\_MWS, id, word, n)  
spooky\_wrd\_lda\_MWS <- LDA(spooky\_wrd\_tm\_MWS, k = 6, control = list(seed = 1234))  
spooky\_wrd\_topics\_MWS <- tidy(spooky\_wrd\_lda\_MWS, matrix = "beta")  
spooky\_wrd\_topics\_5\_MWS <- ungroup(top\_n(group\_by(spooky\_wrd\_topics\_MWS, topic), 5, beta))  
spooky\_wrd\_topics\_5\_MWS <- arrange(spooky\_wrd\_topics\_5\_MWS, topic, -beta)  
spooky\_wrd\_topics\_5\_MWS <- mutate(spooky\_wrd\_topics\_5\_MWS, term = reorder(term, beta))  
png('../figs/MWS\_topic.png')  
ggplot(spooky\_wrd\_topics\_5\_MWS) +  
 geom\_col(aes(term, beta, fill = factor(topic)), show.legend = FALSE) +  
 facet\_wrap(~ topic, scales = "free", ncol = 3) +  
 coord\_flip()  
dev.off()

## png   
## 2

In the above, we see that for Edgar Allan Poe, the first topic is characterized by words like “doubt”, “time”, and the third topic includes the word “death”, and the fifth topic the word “individual”. For HP Lovecraft, the first topic is characterized by words like “strange”, “house”, and the third topic includes the word “death”, and the sixth topic the word “time”,“life. For Mary Shelley, she first topic is characterized by words like”heart“,”love“, and the third topic includes the word”time" and “life”, and the fifth topic the word “hand”. Note that the words “eyes”, “time” and “life” appear in many topics.