bigrams\_separated<-separate(bigrams,bigram,c("word1", "word2"),sep = " ")  
bigrams\_filtered<-bigrams\_separated %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
# new bigram counts:  
bigram\_counts<-bigrams\_filtered %>%   
 count(word1,word2,sort=T)  
head(bigram\_counts)

## # A tibble: 6 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 lord raymond 27  
## 2 fellow creatures 22  
## 3 ha ha 22  
## 4 main compartment 21  
## 5 madame lalande 20  
## 6 chess player 18

bigrams\_HPL\_separated<-separate(bigrams\_HPL,bigram,c("word1", "word2"),sep = " ")  
bigrams\_HPL\_filtered<-bigrams\_HPL\_separated %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
# new bigram counts:  
bigram\_HPL\_counts<-bigrams\_HPL\_filtered %>%   
 count(word1,word2,sort=T)  
head(bigram\_HPL\_counts)

## # A tibble: 6 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 heh heh 17  
## 2 shunned house 16  
## 3 tempest mountain 14  
## 4 brown jenkin 13  
## 5 herbert west 13  
## 6 yog sothoth 12

bigrams\_MWS\_separated<-separate(bigrams\_MWS,bigram,c("word1", "word2"),sep = " ")  
bigrams\_MWS\_filtered<-bigrams\_MWS\_separated %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
# new bigram counts:  
bigram\_MWS\_counts<-bigrams\_MWS\_filtered %>%   
 count(word1,word2,sort=T)  
head(bigram\_MWS\_counts)

## # A tibble: 6 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 lord raymond 27  
## 2 fellow creatures 22  
## 3 native country 14  
## 4 natural philosophy 10  
## 5 poor girl 10  
## 6 human race 9

bigrams\_EAP\_separated<-separate(bigrams\_EAP,bigram,c("word1", "word2"),sep = " ")  
bigrams\_EAP\_filtered<-bigrams\_EAP\_separated %>%  
 filter(!word1 %in% stop\_words$word) %>%  
 filter(!word2 %in% stop\_words$word)  
  
# new bigram counts:  
bigram\_EAP\_counts<-bigrams\_EAP\_filtered %>%   
 count(word1,word2,sort=T)  
head(bigram\_EAP\_counts)

## # A tibble: 6 x 3  
## word1 word2 n  
## <chr> <chr> <int>  
## 1 ha ha 22  
## 2 main compartment 21  
## 3 madame lalande 20  
## 4 chess player 18  
## 5 left arm 13  
## 6 tea pot 13

We can see that these phrases are the most common pairs in spooky data set.

In other analyses, we may want to work with the recombined words. tidyr’s unite() function is the inverse of separate(), and lets us recombine the columns into one. Thus, “separate/filter/count/unite” let us find the most common bigrams not containing stop-words.

bigrams\_united<-bigrams\_filtered %>%  
 unite(bigram, word1, word2, sep = " ")  
head(bigrams\_united)

## id author bigram  
## 1 id00002 HPL hateful modernity  
## 2 id00002 HPL accursed city  
## 3 id00003 EAP dark valley  
## 4 id00004 EAP unusual clearness  
## 5 id00004 EAP necessarily lost  
## 6 id00004 EAP lost sight