## Step2: Do sentiment analysis at sentense level

## Step2: Do sentiment analysis at sentense level

spooky<-read.csv('../data/spooky.csv',as.is=T)  
spooky.sentense<-spooky%>%  
 mutate(sentiment = get\_sentiment(text))

## Warning in split\_warn(text.var, "sentiment\_by", ...): Each time  
## `sentiment\_by` is run it has to do sentence boundary disambiguation when  
## a raw `character` vector is passed to `text.var`. This may be costly of  
## time and memory. It is highly recommended that the user first runs the raw  
## `character` vector through the `get\_sentences` function.

count(spooky.sentense, sentiment)

## # A tibble: 8,704 x 2  
## sentiment n  
## <dbl> <int>  
## 1 -2.42 1  
## 2 -2.15 1  
## 3 -1.92 1  
## 4 -1.66 1  
## 5 -1.65 1  
## 6 -1.59 1  
## 7 -1.56 1  
## 8 -1.53 1  
## 9 -1.48 1  
## 10 -1.48 1  
## # ... with 8,694 more rows

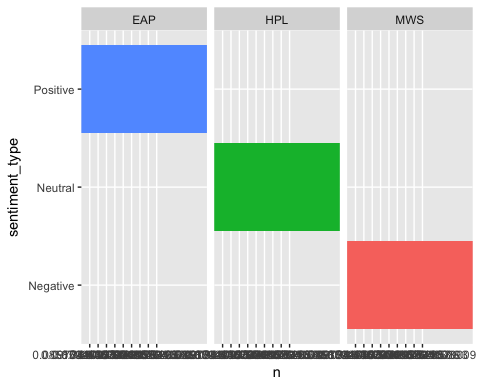
count(spooky.sentense, author, sentiment)

## # A tibble: 11,370 x 3  
## author sentiment n  
## <chr> <dbl> <int>  
## 1 EAP -2.42 1  
## 2 EAP -2.15 1  
## 3 EAP -1.66 1  
## 4 EAP -1.56 1  
## 5 EAP -1.45 1  
## 6 EAP -1.42 1  
## 7 EAP -1.40 1  
## 8 EAP -1.35 1  
## 9 EAP -1.33 1  
## 10 EAP -1.32 1  
## # ... with 11,360 more rows

spooky.sentense.data<-spooky.sentense %>%  
 mutate(sentiment\_type = if\_else(sentiment >0, "Positive", if\_else(sentiment <0, "Negative", "Neutral")))%>%  
 select(sentiment, sentiment\_type,text,author)  
order.spooky.sentense<-spooky.sentense.data[order(spooky.sentense.data$sentiment),]  
positive.rate<-sum(spooky.sentense.data$sentiment\_type=='Positive')/nrow(spooky.sentense.data)  
positive.rate

## [1] 0.4305634

count.whole.table<-count(spooky.sentense.data%>%group\_by(author))  
interger.EAP<-as.integer(count.whole.table[count.whole.table$author=='EAP',]$n)  
interger.HPL<-as.integer(count.whole.table[count.whole.table$author=='HPL',]$n)  
interger.MWS<-as.integer(count.whole.table[count.whole.table$author=='MWS',]$n)  
count.table<-count(spooky.sentense.data%>%group\_by(sentiment\_type, author))   
frequency.EAP<-count.table[count.table$author=='EAP',]$n/  
 as.integer(count.whole.table[count.whole.table$author=='EAP',]$n)  
frequency.HPL<-count.table[count.table$author=='HPL',]$n/  
 as.integer(count.whole.table[count.whole.table$author=='HPL',]$n)  
frequency.MWS<-count.table[count.table$author=='MWS',]$n/  
 as.integer(count.whole.table[count.whole.table$author=='MWS',]$n)  
n<-c(frequency.MWS,frequency.HPL,frequency.EAP)  
author<-c('MWS','MWS','MWS','HPL','HPL','HPL','EAP','EAP','EAP')  
sentiment\_type<-c('Negative','Negative','Negative','Neutral','Neutral','Neutral',  
 'Positive','Positive','Positive')  
frequency.table<-as.data.frame(cbind(sentiment\_type,author,n))  
ggplot(frequency.table)+geom\_col(aes(sentiment\_type, n, fill = sentiment\_type)) +   
 facet\_wrap(~ author) +  
 coord\_flip() +  
 theme(legend.position = "none")

 Proportion of sentences are ‘postive’ is 43%, and for each author based on sensitive level, Poe, Lovecraft,Shelly are positive, neutral, and negative compare to each other.