# assignment4plus

Keliang Xu

12/7/2021

## Sentence-level Analysis

### Tnum ingester

```
tnum.authorize("mssp1.bu.edu")
tnum.setSpace("test2")

source("Book2TN-v6A-1.R")

Holmes<-gutenberg_download(2852)
#write.table(Holmes, file = "holmes.txt", sep = "\t",row.names = F, col.names = T)

Holmes_txt<-read.table("Holmes.txt",header = T)
```

I have the same method in assignment4 to get the holmes text from TN and gutenberg. This time I will put some visualization into report.

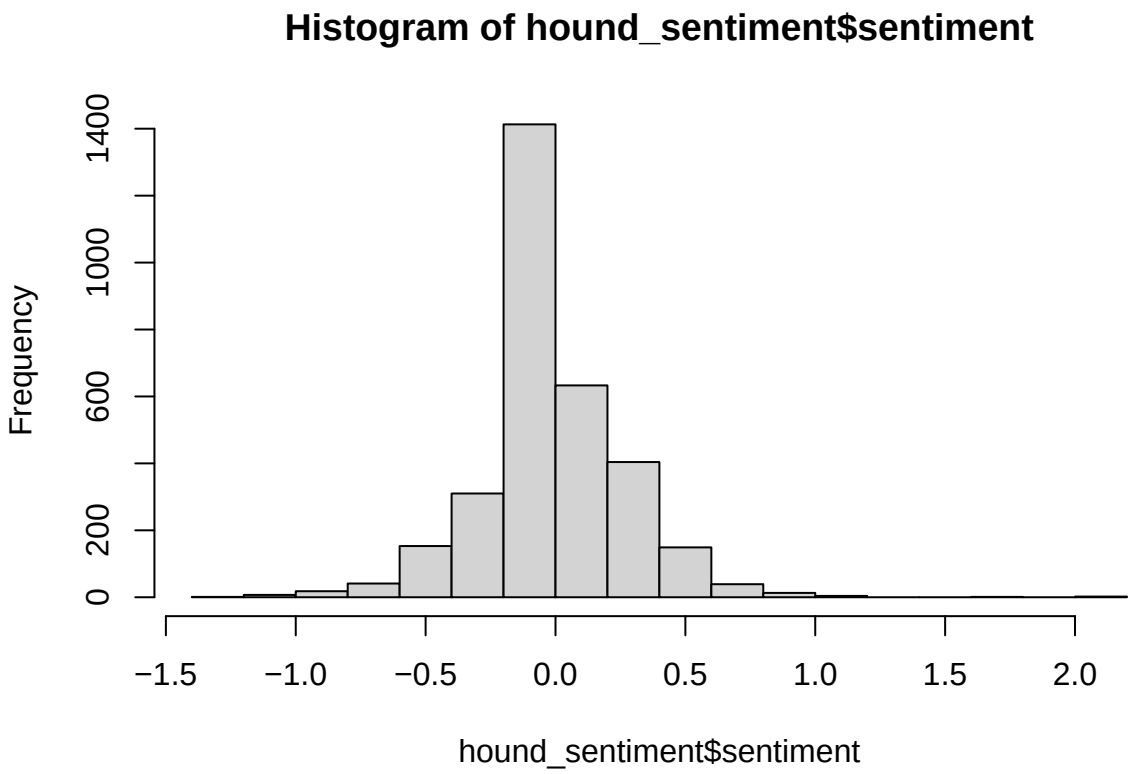
### Sentimerntr

The average sentiment score of the sentences in the review. The most interesting variable is the ave_sentiment, which is the sentiment of the review in one number. The number can take positive or negative values and expresses the valence and the polarity of the sentiment.

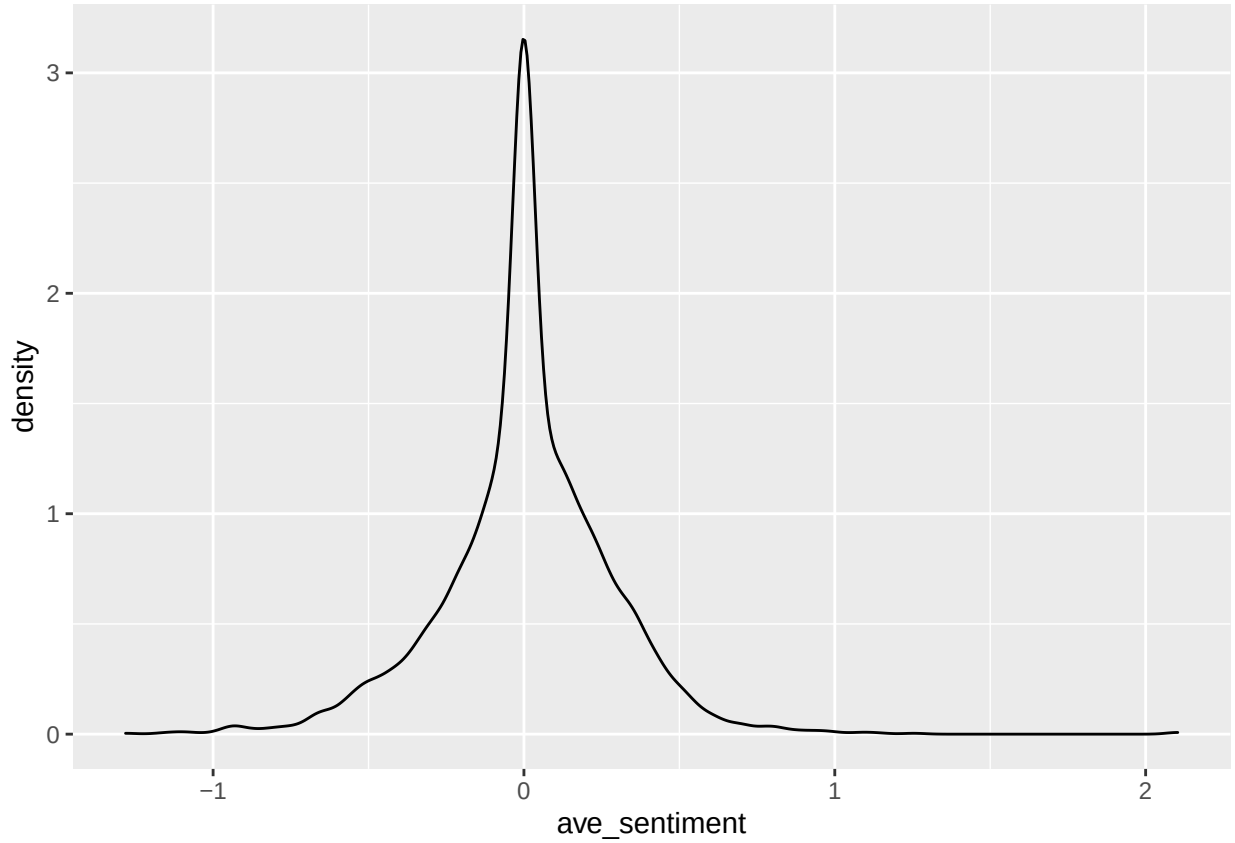
```
q<- tnum.query("holmes/hound# has text", max = 2900)
```

```
## Returned 1 thru 2865 of 2865 results
```

```
df <-tnum.objectsToDf(q)
para_text <- df %>% pull(string.value) %>%
                    str_replace_all("\"","") %>%
                    str_flatten(collapse = " ")

hound<-get_sentences(para_text)
hound_sentiment<-sentiment(hound)

hist(hound_sentiment$sentiment)
```




```
df %>%
  get_sentences() %>%
  sentiment_by(by = NULL) %>% #View()
  ggplot() + geom_density(aes(ave_sentiment))
```




Next analyze the sentiment scores and look at some summary statistics of the calculated sentiment scores.

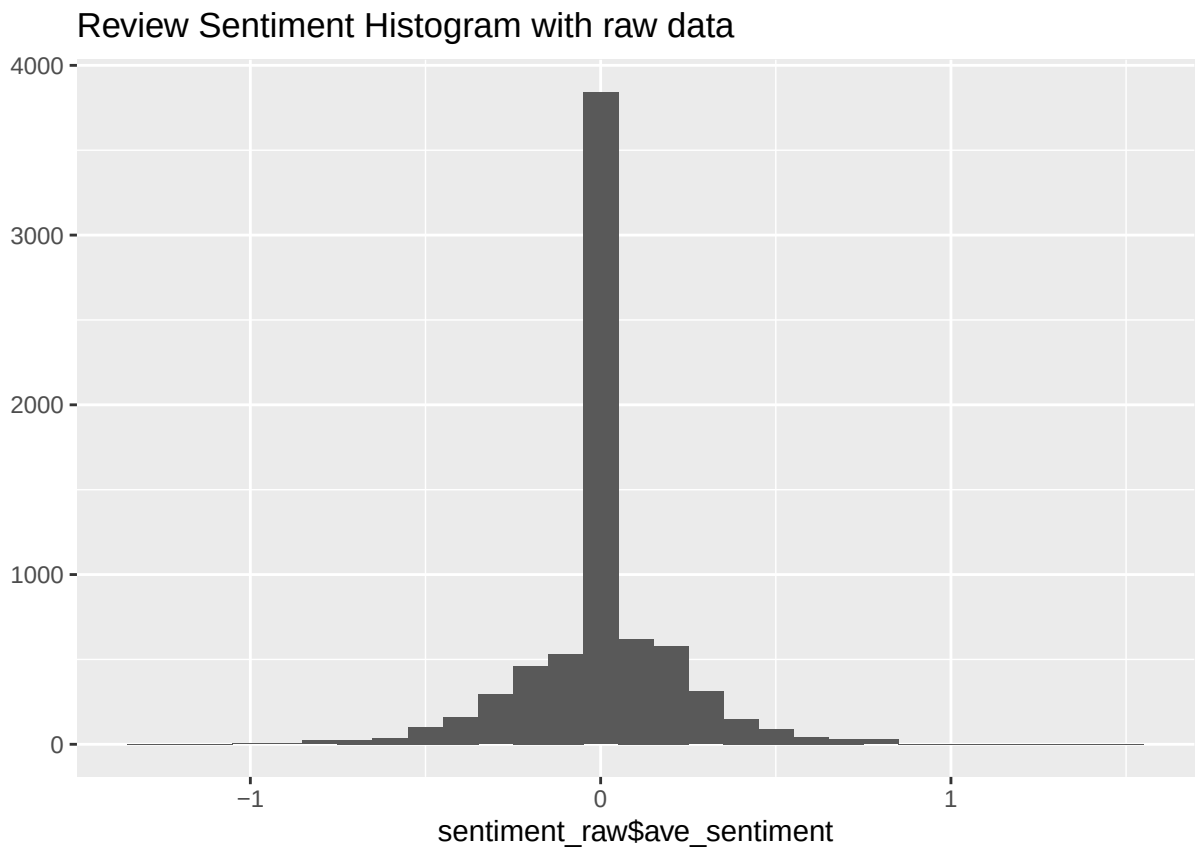
```
sentiment_raw<-sentiment_by(Holmes$text)
sentiment_by(para_text)
```

```
##    element_id word_count        sd ave_sentiment
## 1:          1      63519 0.2713711    0.01311054
```

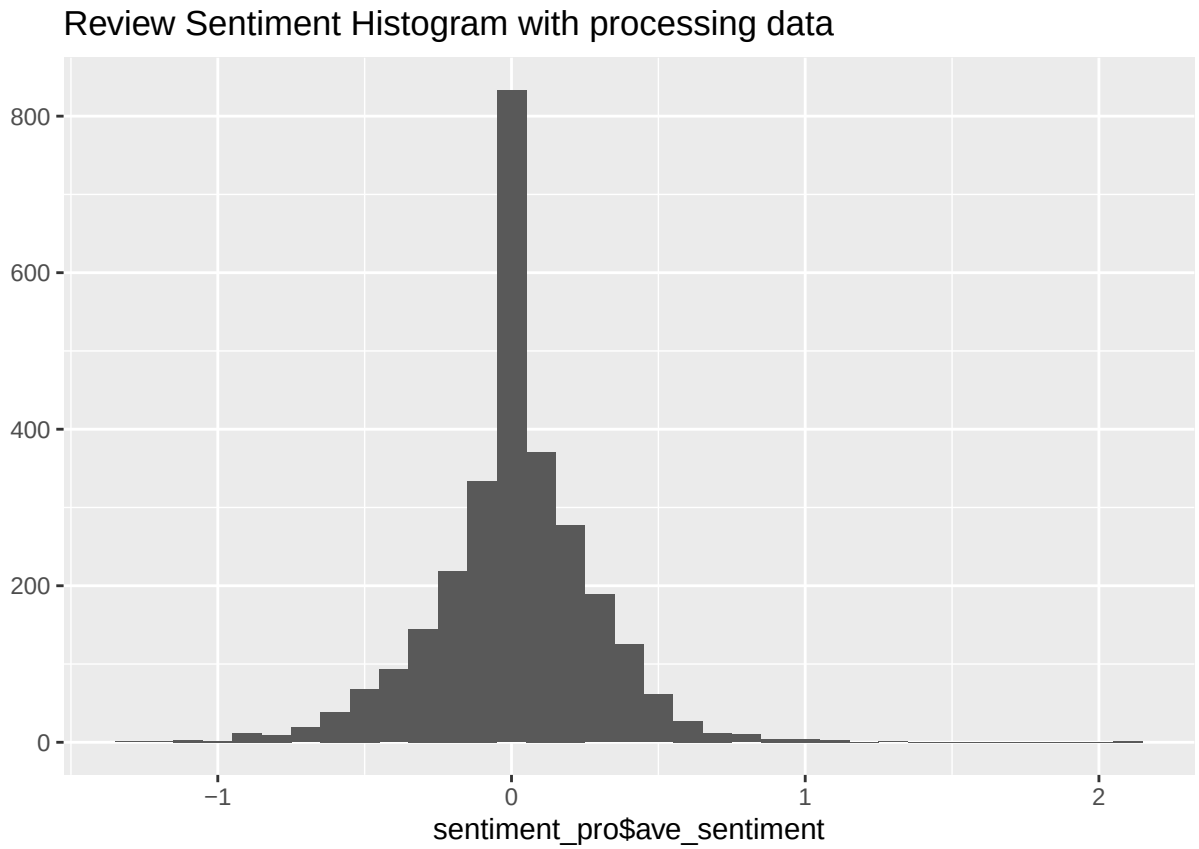
```
summary(sentiment_raw$ave_sentiment)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.280722  0.000000  0.000000  0.004662  0.055470  1.546753
```

```
qplot(sentiment_raw$ave_sentiment,
geom="histogram",binwidth=0.1,main="Review Sentiment Histogram with raw data")
```




```
sentiment_pro <-df %>%
  get_sentences() %>%
  sentiment_by(by = NULL)
qplot(sentiment_pro$ave_sentiment,
geom="histogram",binwidth=0.1,main="Review Sentiment Histogram with processing data")
```




Comparing these two plots, we can find that if the text is not processed, there will be a lot of 0 values, because it is similar to a chapter or a blank line will be doubled as 0.

## Sentence analysis compare to Task two

### Sentence analysis by BING sentiment lexicon

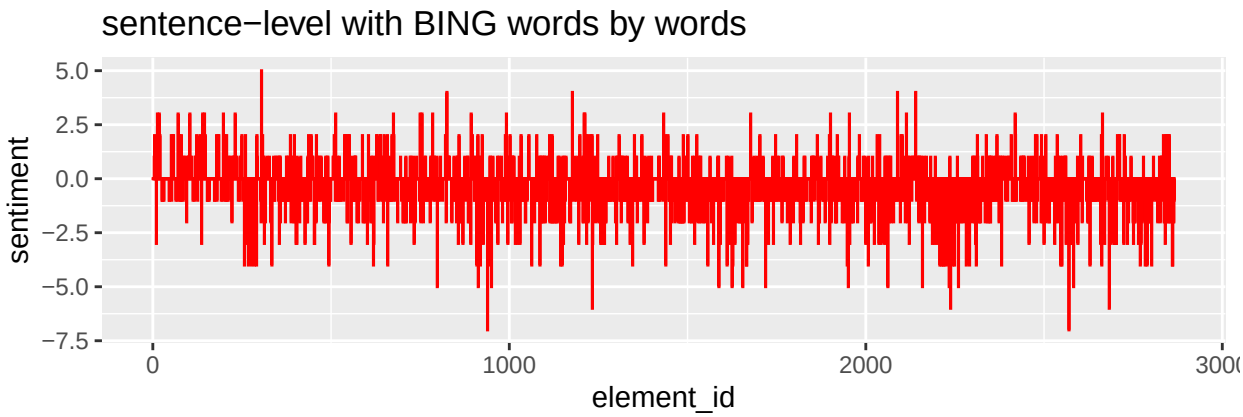
I analyzed each sentence word by word and got sentiment score for word analysis. Comparing with Bing sentiment lexicon word analysis.

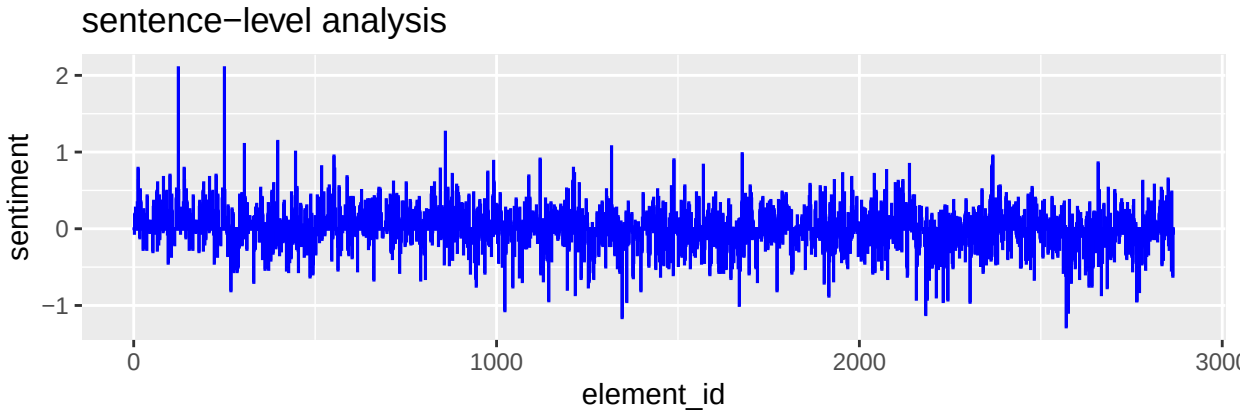
```
word_df<-df %>% select(subject,property,string.value)
word_df<-cbind(word_df,bing=0)
tmp<-NA
for(i in 1:length(df$subject)){
  tmp<-word_df$string.value[i]
  tmp<-data.frame(tmp)

   tidy_tmp <- tmp %>%
    mutate(
      chapter = cumsum(str_detect(tmp, regex("^chapter [\\divxlc]", ignore_case = TRUE)))
  )%>%
  unnest_tokens(word,tmp) %>%
  anti_join(stop_words)

  tmp_sentiment <- tidy_tmp %>%
  inner_join(get_sentiments("bing"))
```

```
  word_df$bing[i]<-sum(tmp_sentiment$sentiment=="positive")-sum(tmp_sentiment$sentiment=="negative")
}
```

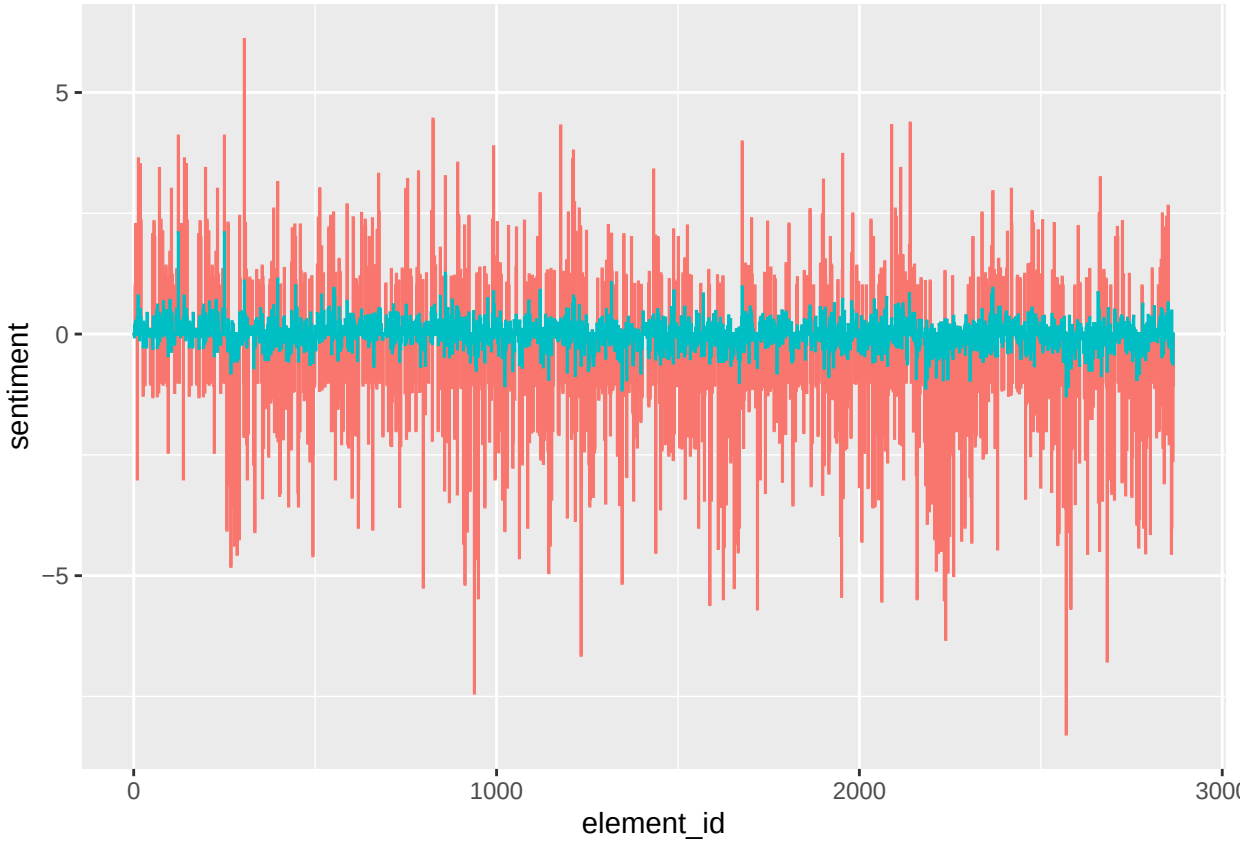
Next I'll use Task two method to analysis the sentence. I will separate the words from each sentence and compare to which just use function sentiment_by().

```
plot_holmes<- cbind(element_id=sentiment_pro$element_id,sentiment=word_df$bing)
plot_holmes<- cbind(plot_holmes,group="bing")
plot_2<-cbind(element_id=sentiment_pro$element_id,sentiment=sentiment_pro$ave_sentiment)
plot_2<-cbind(plot_2,group="Sentimerntr")
plot_holmes<-rbind(plot_holmes,plot_2)

plot_holmes<-data.frame(plot_holmes)

for(i in 1:length(plot_holmes$element_id)){
  plot_holmes$sentiment[i]<-round(as.numeric(plot_holmes$sentiment[i]),2)
}
plot_holmes$element_id<-as.numeric(plot_holmes$element_id)
plot_holmes$sentiment<-as.numeric(plot_holmes$sentiment)

p1<-ggplot(plot_holmes[1:2865,] ) +
  geom_col(aes(element_id, sentiment),show.legend = FALSE,color="RED") +
  ggtitle("sentence-level with BING words by words")

p2<-ggplot(plot_holmes[2866:5730,])+
  geom_col(aes(element_id, sentiment) ,show.legend = FALSE,color="BLUE")  +
  ggtitle("sentence-level analysis")
ggpubr::ggarrange(p1,p2,nrow=2,ncol=1)
```




```
ggplot(plot_holmes, aes(element_id, sentiment,color=group)) +
  geom_col(show.legend = FALSE)
```

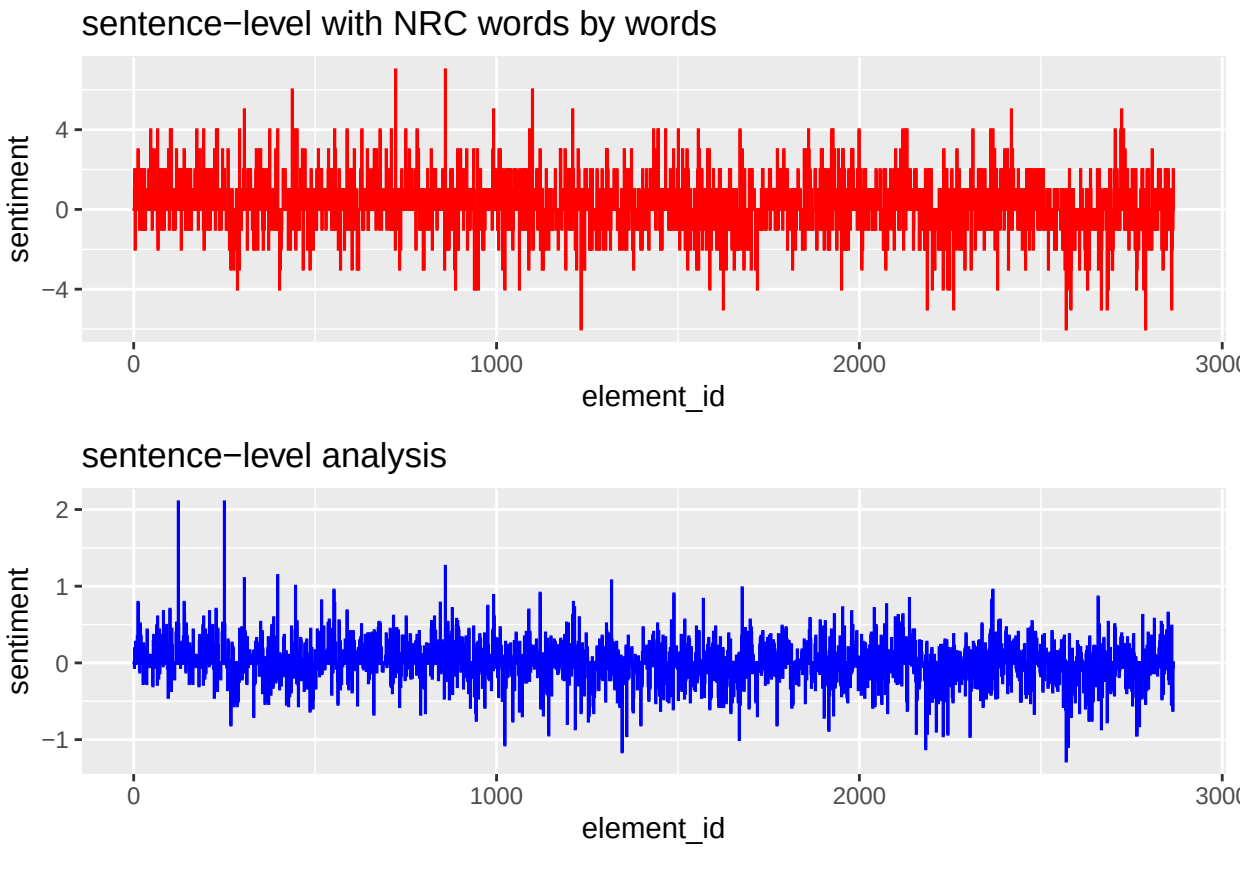



**Sentence analysis by NRC sentiment lexicon**

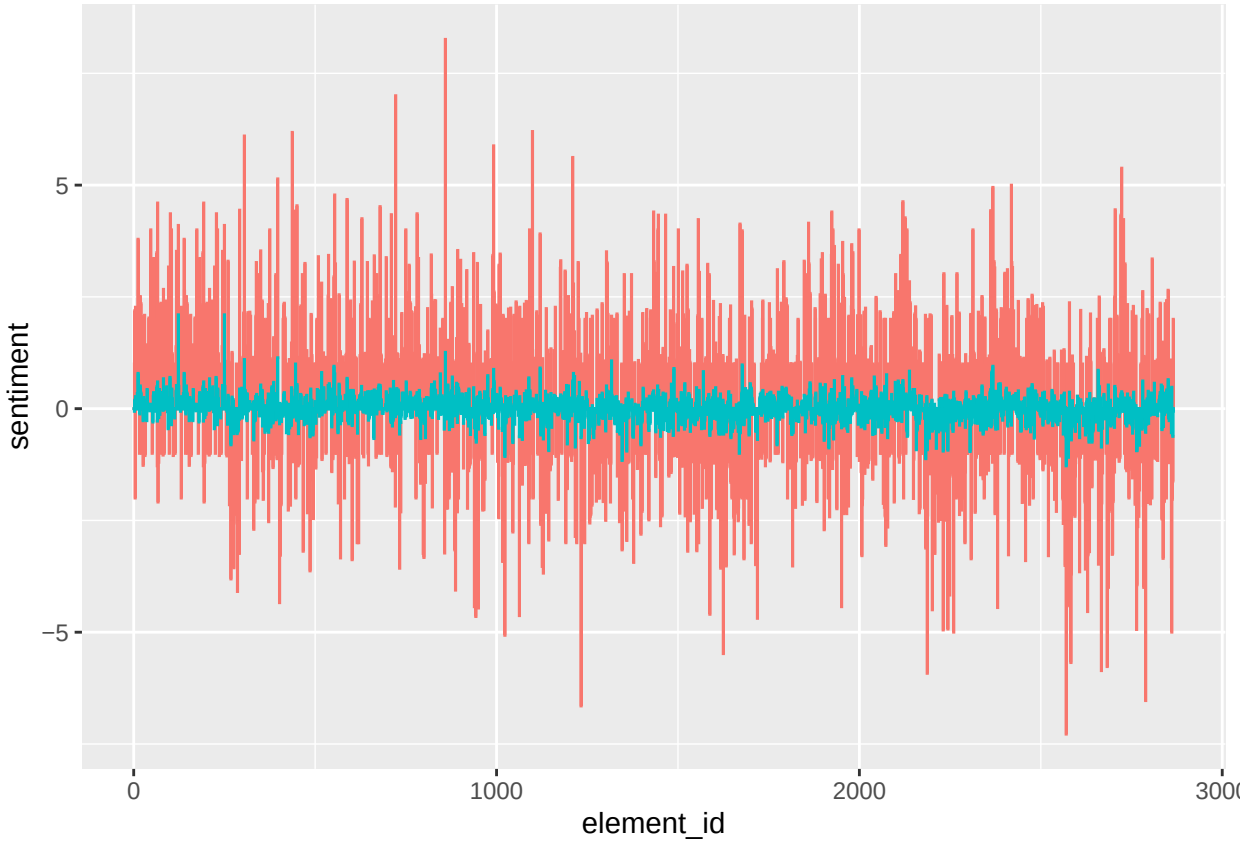
Comparing with NRC index word analysis.

```
word_df<-df %>% select(subject,property,string.value)
word_df<-cbind(word_df,nrc=0)
tmp<-NA
for(i in 1:length(df$subject)){
  tmp<-word_df$string.value[i]
  tmp<-data.frame(tmp)
  tidy_tmp <- tmp %>%
    mutate(
      chapter = cumsum(str_detect(tmp, regex("^chapter [\\divxlc]", ignore_case = TRUE)))
  )%>%
  unnest_tokens(word,tmp) %>%
  anti_join(stop_words)

  tmp_sentiment <- tidy_tmp %>%
  inner_join(get_sentiments("nrc"))  %>%
                filter(sentiment %in% c("positive",
                                        "negative"))

  word_df$nrc[i]<-sum(tmp_sentiment$sentiment=="positive")-sum(tmp_sentiment$sentiment=="negative")
}
```

```
plot_holmes<- cbind(element_id=sentiment_pro$element_id,sentiment=word_df$nrc)
plot_holmes<- cbind(plot_holmes,group="nrc")
plot_2<-cbind(element_id=sentiment_pro$element_id,sentiment=sentiment_pro$ave_sentiment)
plot_2<-cbind(plot_2,group="Sentimerntr")
plot_holmes<-rbind(plot_holmes,plot_2)

plot_holmes<-data.frame(plot_holmes)

for(i in 1:length(plot_holmes$element_id)){
  plot_holmes$sentiment[i]<-round(as.numeric(plot_holmes$sentiment[i]),2)
}
plot_holmes$element_id<-as.numeric(plot_holmes$element_id)
plot_holmes$sentiment<-as.numeric(plot_holmes$sentiment)

p1<-ggplot(plot_holmes[1:2865,] ) +
  geom_col(aes(element_id, sentiment),show.legend = FALSE,color="RED") +
  ggtitle("sentence-level with NRC words by words")

p2<-ggplot(plot_holmes[2866:5730,])+
  geom_col(aes(element_id, sentiment) ,show.legend = FALSE,color="BLUE")  +
  ggtitle("sentence-level analysis")
ggpubr::ggarrange(p1,p2,nrow=2,ncol=1)
```




```
ggplot(plot_holmes, aes(element_id, sentiment,color=group)) +
  geom_col(show.legend = FALSE)
```

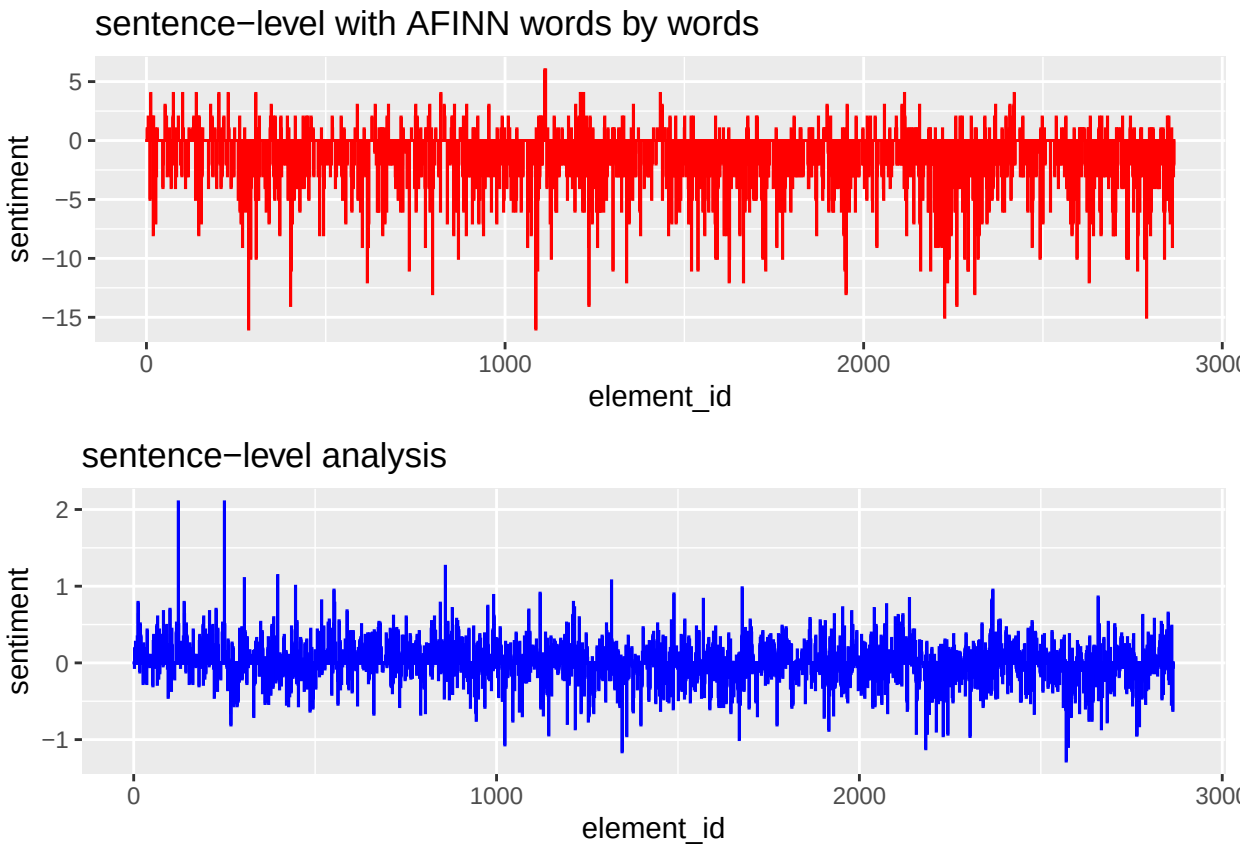



**Sentence analysis by AFINN sentiment lexicon**

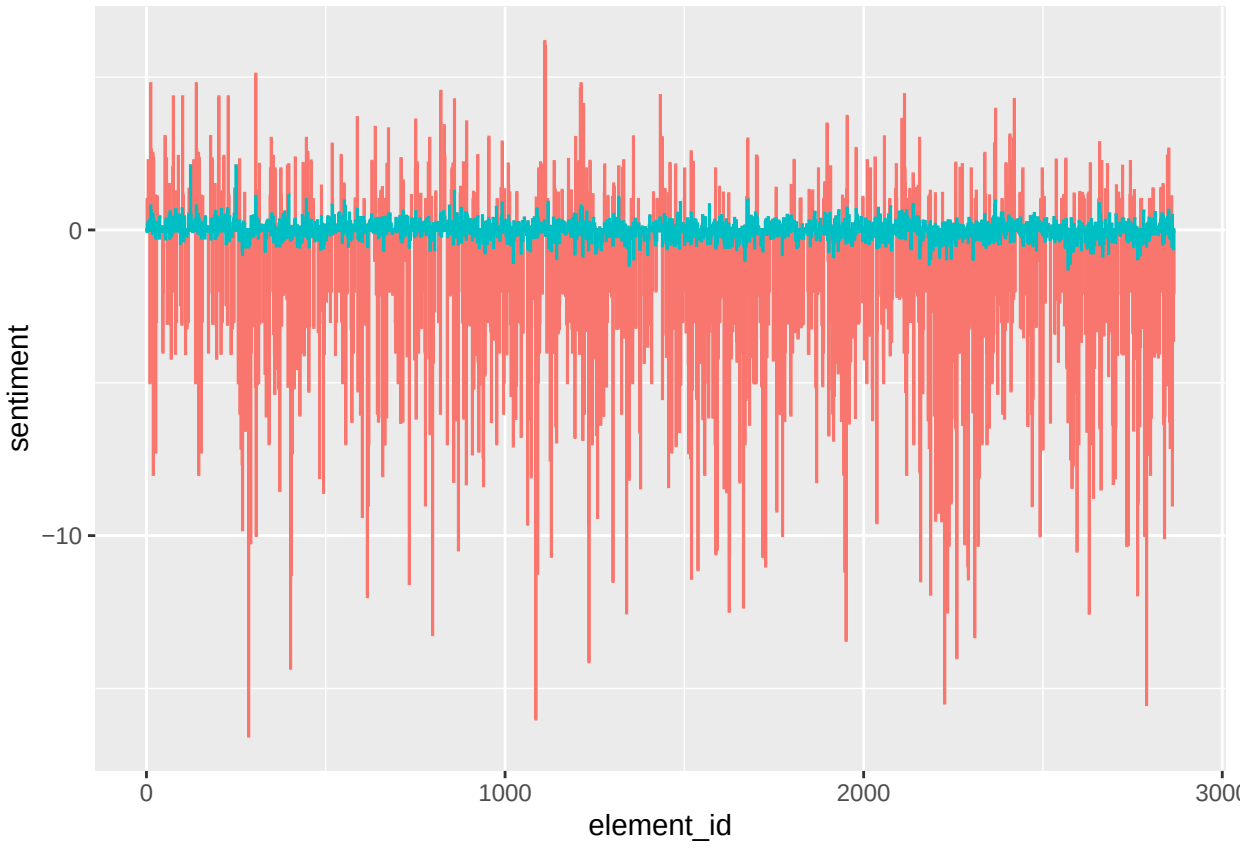
Comparing with AFINN index word analysis.

```
word_df<-df %>% select(subject,property,string.value)
word_df<-cbind(word_df,afinn=0)
tmp<-NA
for(i in 1:length(df$subject)){
  tmp<-word_df$string.value[i]
  tmp<-data.frame(tmp)
  tidy_tmp <- tmp %>%
    mutate(
      chapter = cumsum(str_detect(tmp, regex("^chapter [\\divxlc]", ignore_case = TRUE)))
  )%>%
  unnest_tokens(word,tmp) %>%
  anti_join(stop_words)

  tmp_sentiment <- tidy_tmp %>%
  inner_join(get_sentiments("afinn"))


  word_df$afinn[i]<-sum(tmp_sentiment$value)-length(tmp_sentiment$value)
}
```

```
plot_holmes<- cbind(element_id=sentiment_pro$element_id,sentiment=word_df$afinn)
plot_holmes<- cbind(plot_holmes,group="affin")
plot_3<-cbind(element_id=sentiment_pro$element_id,sentiment=sentiment_pro$ave_sentiment)
plot_3<-cbind(plot_3,group="Sentimerntr")
plot_holmes<-rbind(plot_holmes,plot_3)

plot_holmes<-data.frame(plot_holmes)

for(i in 1:length(plot_holmes$element_id)){
  plot_holmes$sentiment[i]<-round(as.numeric(plot_holmes$sentiment[i]),2)
}
plot_holmes$element_id<-as.numeric(plot_holmes$element_id)
plot_holmes$sentiment<-as.numeric(plot_holmes$sentiment)

p1<-ggplot(plot_holmes[1:2865,] ) +
  geom_col(aes(element_id, sentiment),show.legend = FALSE,color="RED") +
  ggtitle("sentence-level with AFINN words by words")

p2<-ggplot(plot_holmes[2866:5730,])+
  geom_col(aes(element_id, sentiment) ,show.legend = FALSE,color="BLUE")  +
  ggtitle("sentence-level analysis")
ggpubr::ggarrange(p1,p2,nrow=2,ncol=1)
```




```
ggplot(plot_holmes, aes(element_id, sentiment,color=group)) +
  geom_col(show.legend = FALSE)
```

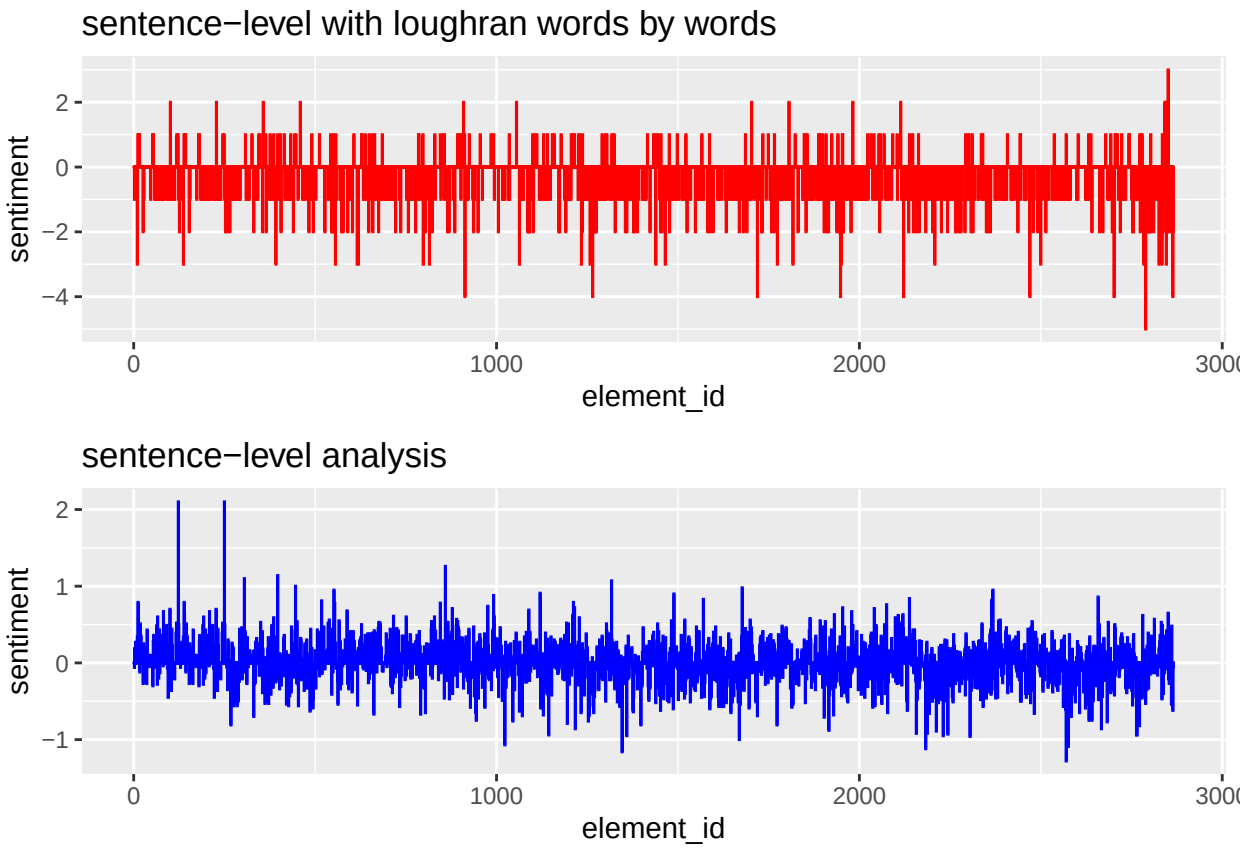



**Sentence analysis by LOUGHRAN sentiment lexicon**

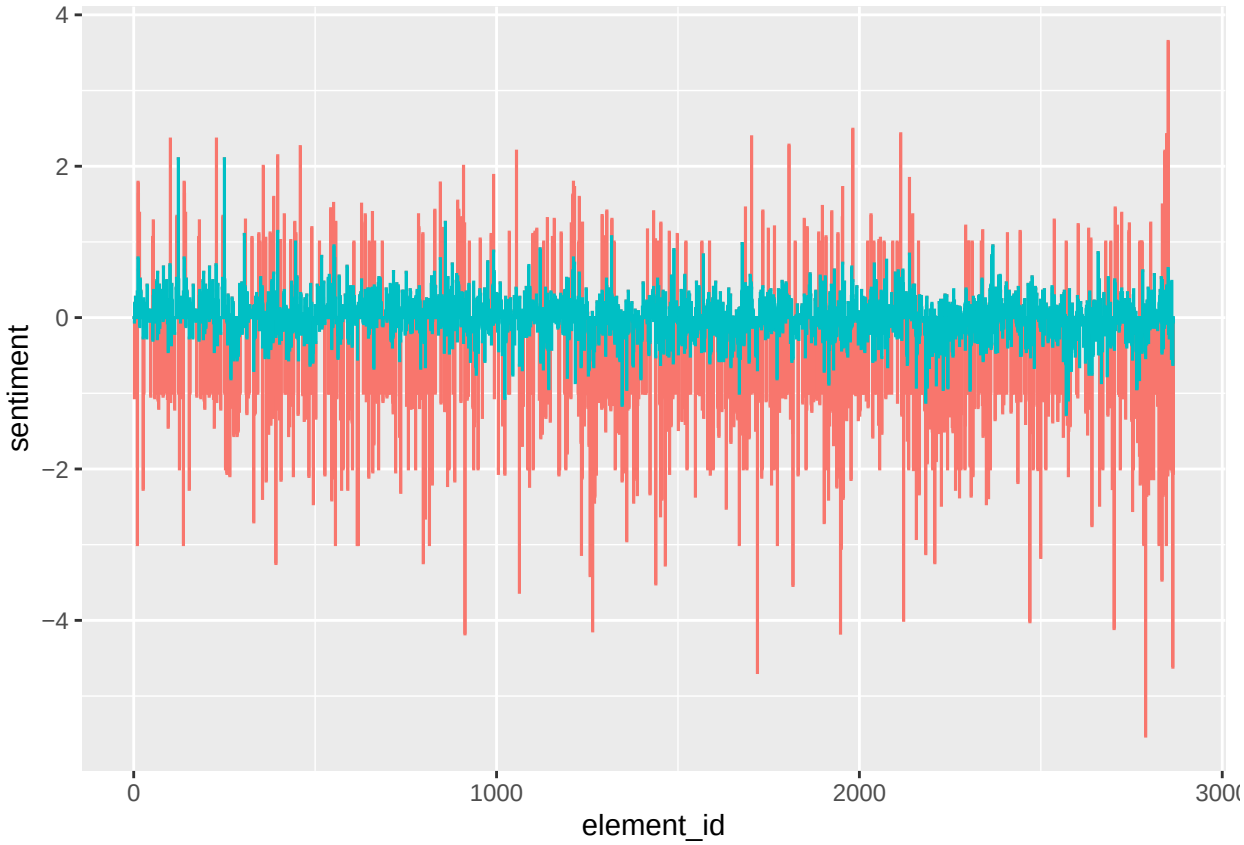
Comparing with LOUGHRAN index word analysis.

```
word_df<-df %>% select(subject,property,string.value)
word_df<-cbind(word_df,loughran=0)
tmp<-NA
for(i in 1:length(df$subject)){
  tmp<-word_df$string.value[i]
  tmp<-data.frame(tmp)
  tidy_tmp <- tmp %>%
    mutate(
      chapter = cumsum(str_detect(tmp, regex("^chapter [\\divxlc]", ignore_case = TRUE)))
  )%>%
  unnest_tokens(word,tmp) %>%
  anti_join(stop_words)

  tmp_sentiment <- tidy_tmp %>%
  inner_join(get_sentiments("loughran")) %>%
                filter(sentiment %in% c("positive",
                                        "negative"))

  word_df$loughran[i]<-sum(tmp_sentiment$sentiment=="positive")-sum(tmp_sentiment$sentiment=="negative")
}
```

```
plot_holmes<- cbind(element_id=sentiment_pro$element_id,sentiment=word_df$loughran)
plot_holmes<- cbind(plot_holmes,group="loughran")
plot_4<-cbind(element_id=sentiment_pro$element_id,sentiment=sentiment_pro$ave_sentiment)
plot_4<-cbind(plot_4,group="Sentimerntr")
plot_holmes<-rbind(plot_holmes,plot_4)

plot_holmes<-data.frame(plot_holmes)

for(i in 1:length(plot_holmes$element_id)){
  plot_holmes$sentiment[i]<-round(as.numeric(plot_holmes$sentiment[i]),2)
}
plot_holmes$element_id<-as.numeric(plot_holmes$element_id)
plot_holmes$sentiment<-as.numeric(plot_holmes$sentiment)

p1<-ggplot(plot_holmes[1:2865,] ) +
  geom_col(aes(element_id, sentiment),show.legend = FALSE,color="RED") +
  ggtitle("sentence-level with loughran words by words")

p2<-ggplot(plot_holmes[2866:5730,])+
  geom_col(aes(element_id, sentiment) ,show.legend = FALSE,color="BLUE")  +
  ggtitle("sentence-level analysis")
ggpubr::ggarrange(p1,p2,nrow=2,ncol=1)
```




```
ggplot(plot_holmes, aes(element_id, sentiment,color=group)) +
  geom_col(show.legend = FALSE)
```


4
2
0
−2
−4
sentiment
0
1000
2000
element_id