Sentiment analysis of Shakespeare's tragedies

Peer Christensen

3/4/2018

Outline:
1. Simple sentiment analysis of one play (Hamlet)
2. Same method on all Shakespeare's tragedies **3. complex sentiment analysis**
4. Simple temporal sentiment analysis I
5. Simple temporal sentiment analysis II

Install and load packages

```
## Loading required package: pacman
```

Sentiment analysis of one play (Hamlet)

First, we load in 'Hamlet' and do some cleaning of the text.
No need to remove any words.

```
text = glue(read_file("Hamlet.txt"))

tokens = tibble(text = tolower(text)) %>% unnest_tokens(word, text)

tokens
```

```
## # A tibble: 32,197 x 1
##    word
##    <chr>
##  1 act
##  2 i
##  3 scene
##  4 i
##  5 elsinore
##  6 a
##  7 platform
##  8 before
##  9 the
## 10 castle
## # ... with 32,187 more rows
```

We then do the following with our tokens:

1. **subset 'sentiment words' using the 'bing' lexicon**
2. **count positive and negative words**
3. **switch to wide format**
4. **create a new variable calculating N positive - negative words**

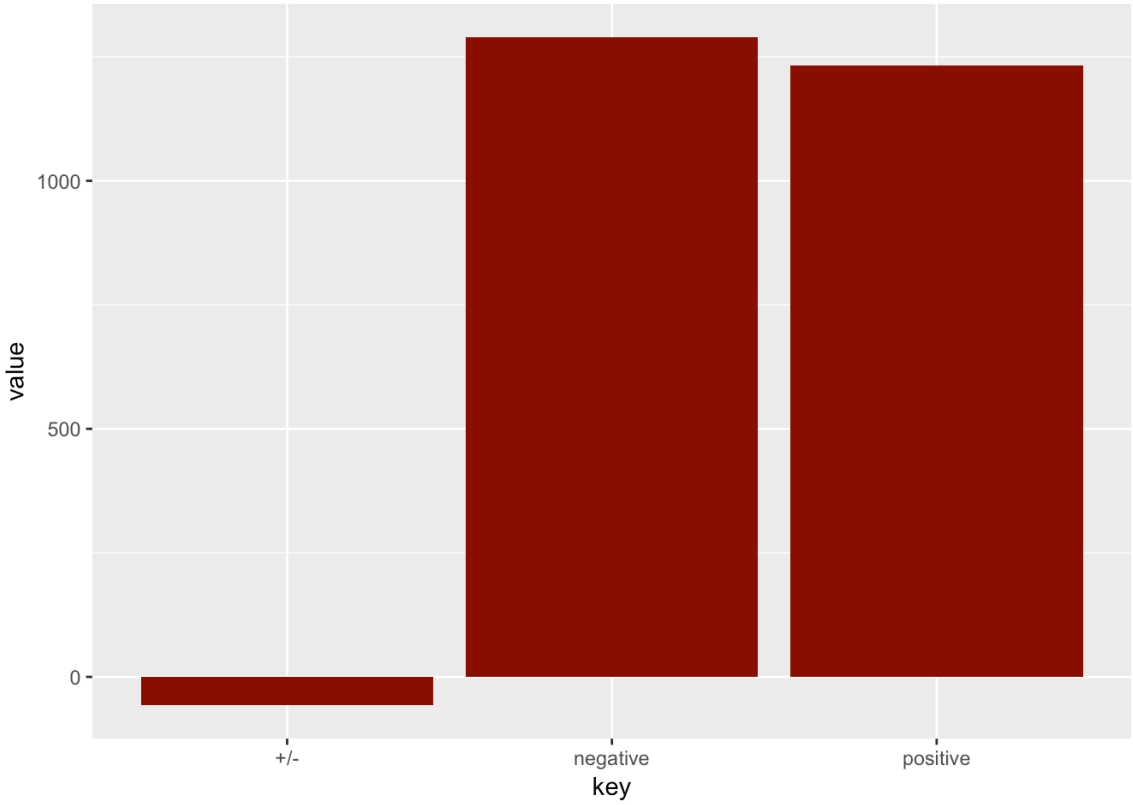
5. switch back to long format (for plotting)

```
sentiments = tokens %>%
  inner_join(get_sentiments("bing")) %>%
  count(sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate("+/-" = positive - negative) %>%
  gather()

sentiments
```

```
## # A tibble: 3 x 2
##   key        value
##   <chr>      <dbl>
## 1 negative  1290
## 2 positive  1233
## 3 +/-      -  57.0
```

Now we can plot the results!

```
sentimentsBars = ggplot(sentiments, aes(x=key, y=value)) +
  geom_bar(stat="identity",fill="darkred")

sentimentsBars
```



But what's in 'bing' and the other available sentiment lexica?

```
# bing
get_sentiments("bing")[sample(nrow(get_sentiments("bing")),10),]
```

```
## # A tibble: 10 x 2
##    word           sentiment
##    <chr>          <chr>
##  1 polluters      negative
##  2 bashing        negative
##  3 dishonorablely negative
##  4 rascals        negative
##  5 humiliating    negative
##  6 polarisation   negative
##  7 lividly        negative
##  8 senseless      negative
##  9 misguidance    negative
## 10 bothered       negative
```

```
# nrc
get_sentiments("nrc")[sample(nrow(get_sentiments("nrc")),10),]
```

```
## # A tibble: 10 x 2
##    word            sentiment
##    <chr>           <chr>
##  1 ripen           anticipation
##  2 intelligent     trust
##  3 disillusionment anger
##  4 vote            anticipation
##  5 alarming        surprise
##  6 actionable      disgust
##  7 pastor          trust
##  8 enchanting      anticipation
##  9 cross           anger
## 10 nourishment     positive
```

```
# afinn
get_sentiments("afinn")[sample(nrow(get_sentiments("afinn")),10),]
```

```
## # A tibble: 10 x 2
##    word          score
##    <chr>         <int>
##  1 fad              -2
##  2 invincible        2
##  3 stalling         -2
##  4 mourns           -2
##  5 protected         1
##  6 perturbed        -2
##  7 disillusioned    -2
##  8 insult           -2
##  9 endorsed          2
## 10 postpones        -1
```

```
# loughran
get_sentiments("loughran")[sample(nrow(get_sentiments("loughran")),10),]
```

```
## # A tibble: 10 x 2
##    word           sentiment
##    <chr>          <chr>
##  1 diverts        negative
##  2 nullifications negative
##  3 accusation     negative
##  4 ascendants     litigious
##  5 slowed         negative
##  6 succeeding     positive
##  7 jeopardize     negative
##  8 variables      uncertainty
##  9 sentencing     negative
## 10 acquit         negative
```

```
unique(get_sentiments("loughran")$sentiment)
```

```
## [1] "negative"     "positive"     "uncertainty"  "litigious"
## [5] "constraining" "superfluous"
```

Sentiment analysis of all Shakespeare's tragedies

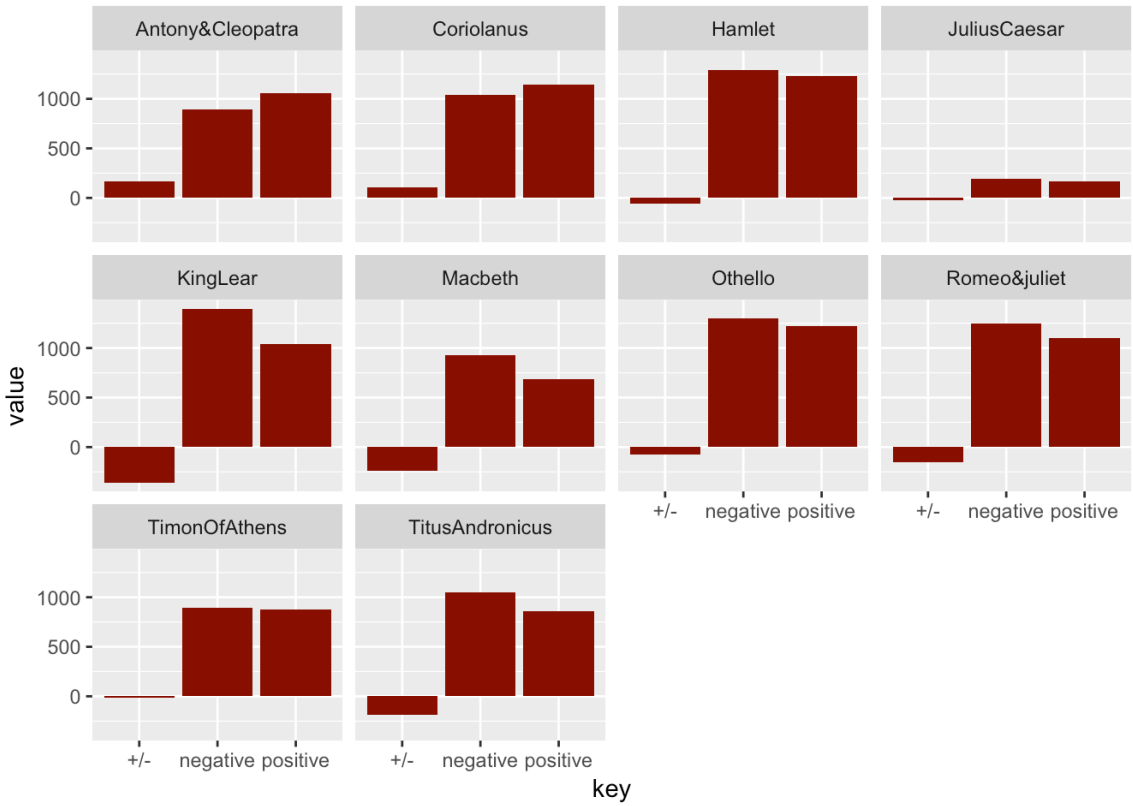
% unnest_tokens(word, text)

sentiments = tokens %>% inner_join(get_sentiments("bing")) %>% count(sentiment) %>% spread(sentiment, n, fill = 0) %>% mutate("+/-" = positive - negative) %>% gather() %>% mutate(Play = play)

df = rbind(df,sentiments) }

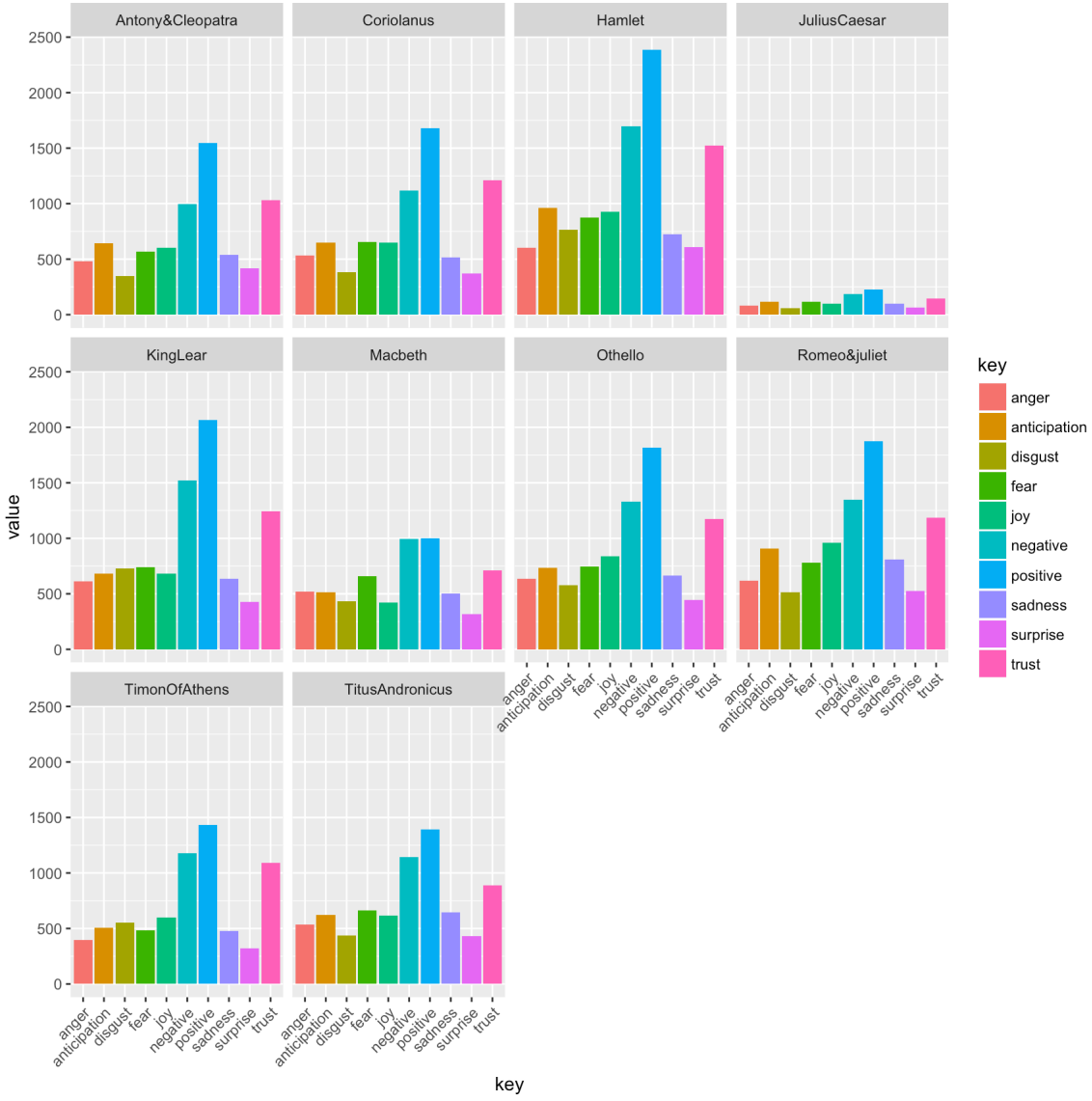
sentimentsBarsAll = ggplot(df, aes(x=key, y=value)) + geom_bar(stat="identity", fill = "darkred") + facet_wrap(~gsub(".txt","",Play))

sentimentsBarsAll ```



Complex sentiment analysis

Same code as above, except that we choose “nrc” instead of “bing”, and change the angle of the x-axis labels so that they all fit, i.e. adding:
theme(axis.text.x = element_text(angle = 45, hjust = 1))



Simple temporal sentiment analysis I

We create a variable called polarity with values -1 to 1 corresponding to “negative”" and “positive”.
We then use this variable to create another variable containing the “rolling mean” of 50 sentiment words.

```
for (play in playList){

  text=glue(read_file(play))

  tokens = tibble(text = tolower(text)) %>% unnest_tokens(word, text)
  tokens <- rowid_to_column(tokens, "ID") # create row numbers

  sentiments = tokens %>%
  inner_join(get_sentiments("bing"))

  sentiments$polarity = c()
  sentiments$polarity[sentiments$sentiment=="negative"] = -1
  sentiments$polarity[sentiments$sentiment=="positive"] =  1

  rollMean<-rollmean(sentiments$polarity, 50,fill = list(NA, NULL, NA))
  sentiments$rollMean=rollMean

  plot = ggplot(sentiments) +
    aes(ID,polarity, fill= sentiment) +
    geom_col() +
    geom_line(aes(ID,rollMean)) +
    ggtitle(gsub(".txt","",play)) +
    theme_minimal()
    print(plot)
}
```

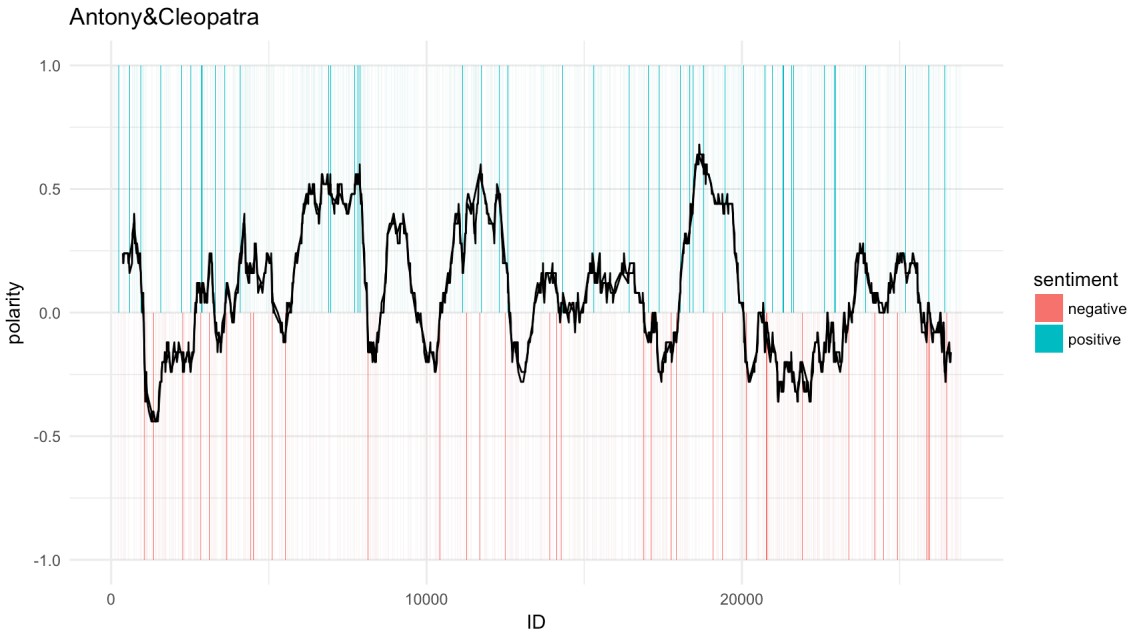
```
## Joining, by = "word"
```

```
## Warning: Unknown or uninitialised column: 'polarity'.
```

```
## Warning: Removed 49 rows containing missing values (geom_path).
```

```
## Joining, by = "word"
```

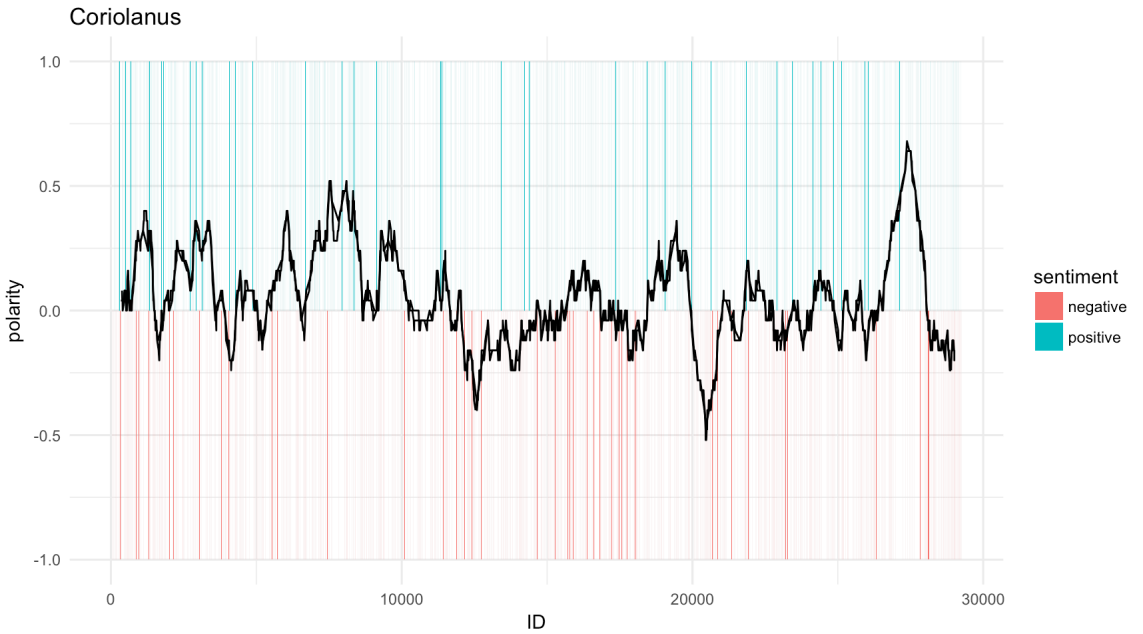
```
## Warning: Unknown or uninitialised column: 'polarity'.
```




```
## Warning: Removed 49 rows containing missing values (geom_path).
```

```
## Joining, by = "word"
```

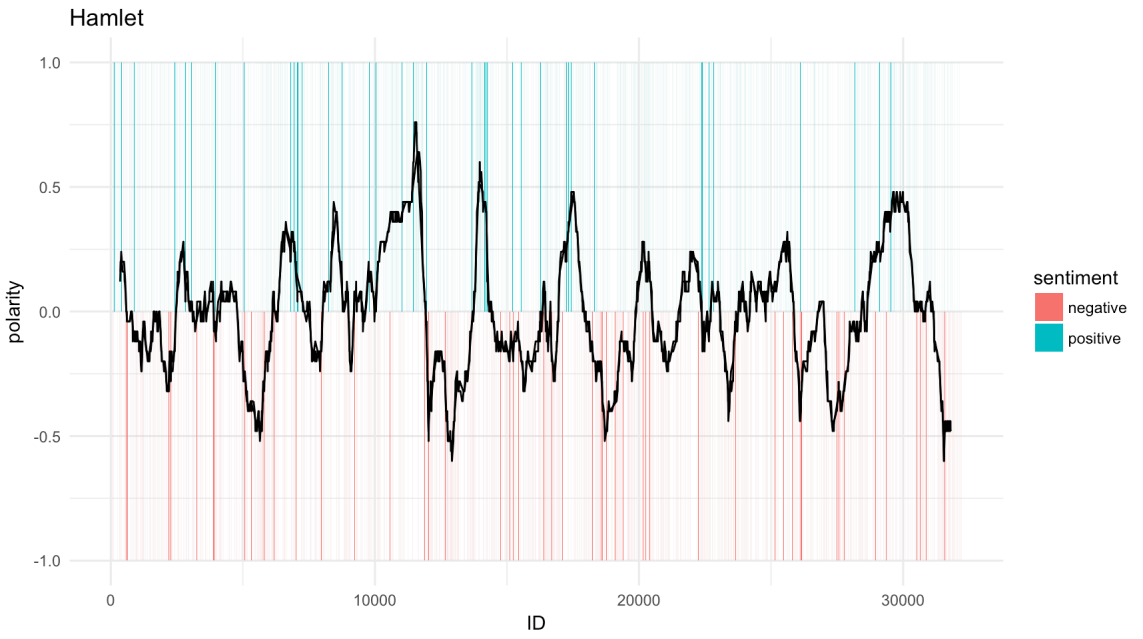
```
## Warning: Unknown or uninitialised column: 'polarity'.
```




```
## Warning: Removed 49 rows containing missing values (geom_path).
```

```
## Joining, by = "word"
```

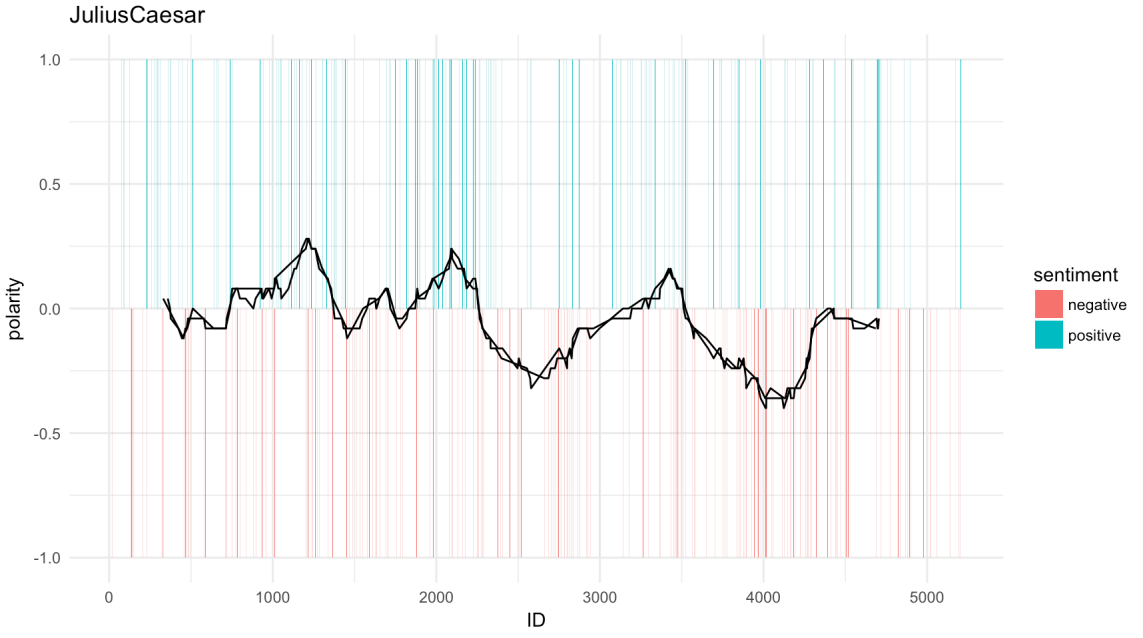
```
## Warning: Unknown or uninitialised column: 'polarity'.
```




```
## Warning: Removed 49 rows containing missing values (geom_path).
```

```
## Joining, by = "word"
```

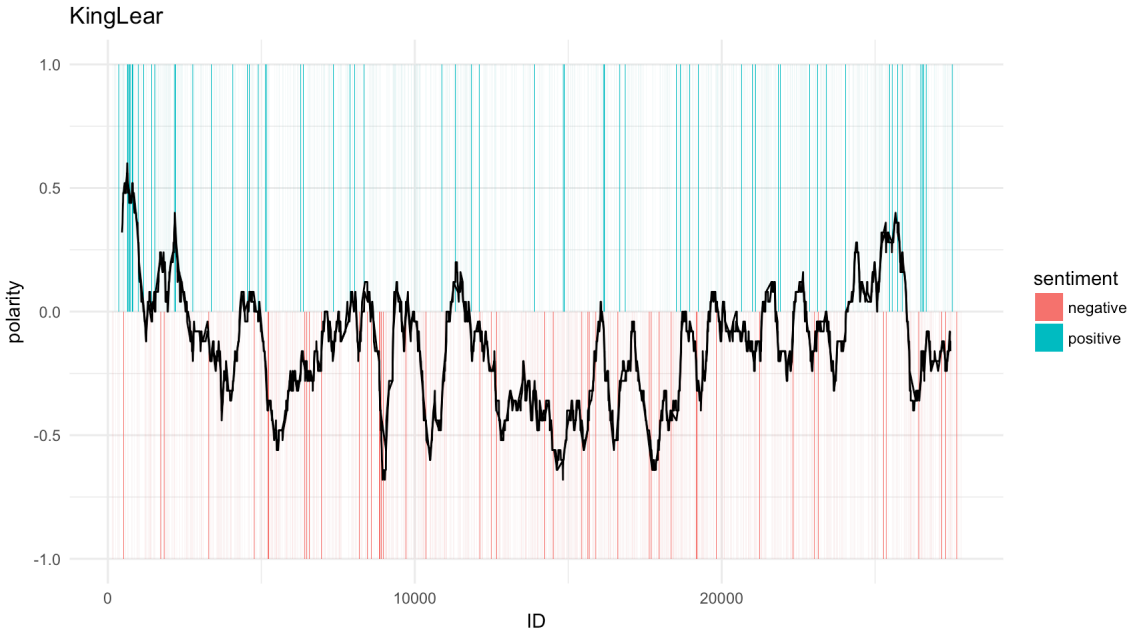
```
## Warning: Unknown or uninitialised column: 'polarity'.
```




```
## Warning: Removed 49 rows containing missing values (geom_path).
```

```
## Joining, by = "word"
```

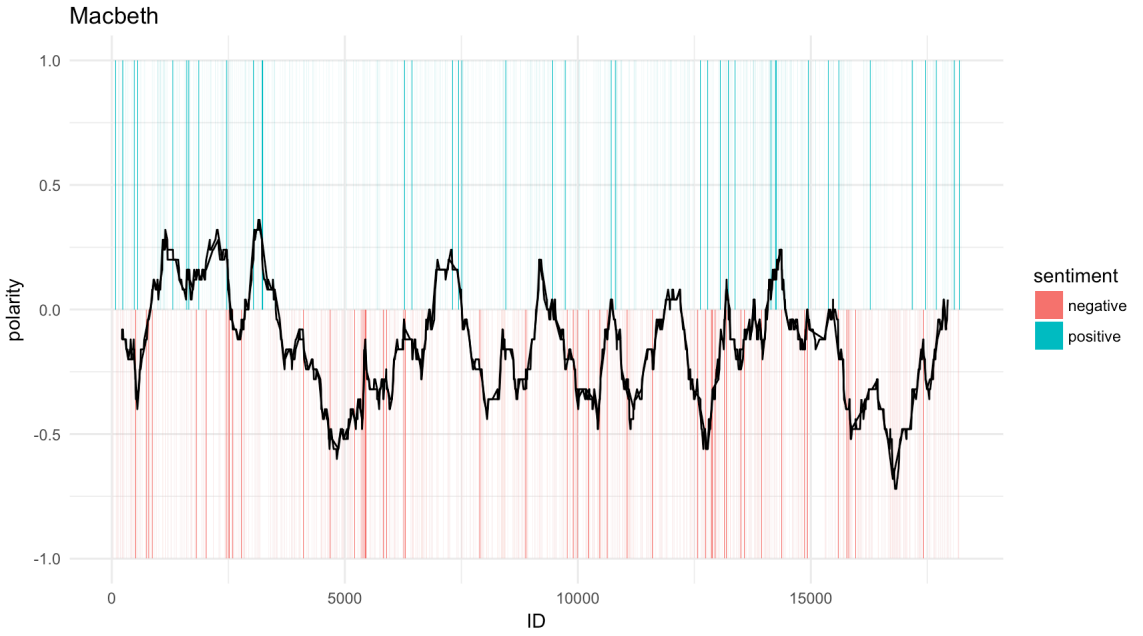
```
## Warning: Unknown or uninitialised column: 'polarity'.
```




```
## Warning: Removed 49 rows containing missing values (geom_path).
```

```
## Joining, by = "word"
```

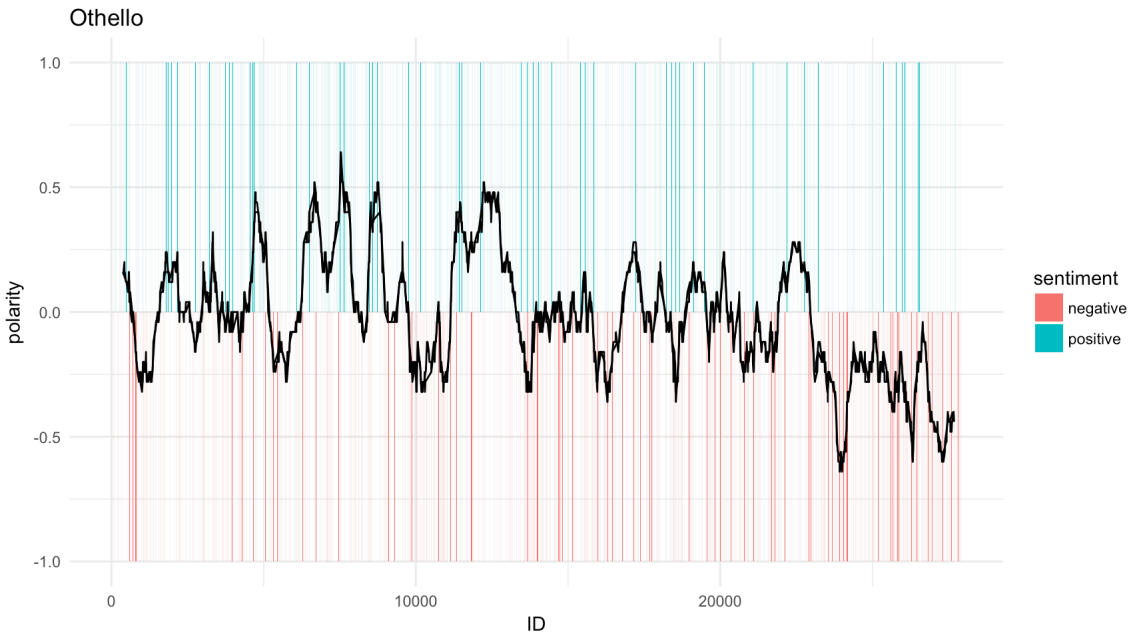
```
## Warning: Unknown or uninitialised column: 'polarity'.
```



```
## Warning: Removed 49 rows containing missing values (geom_path).
```

```
## Joining, by = "word"
```

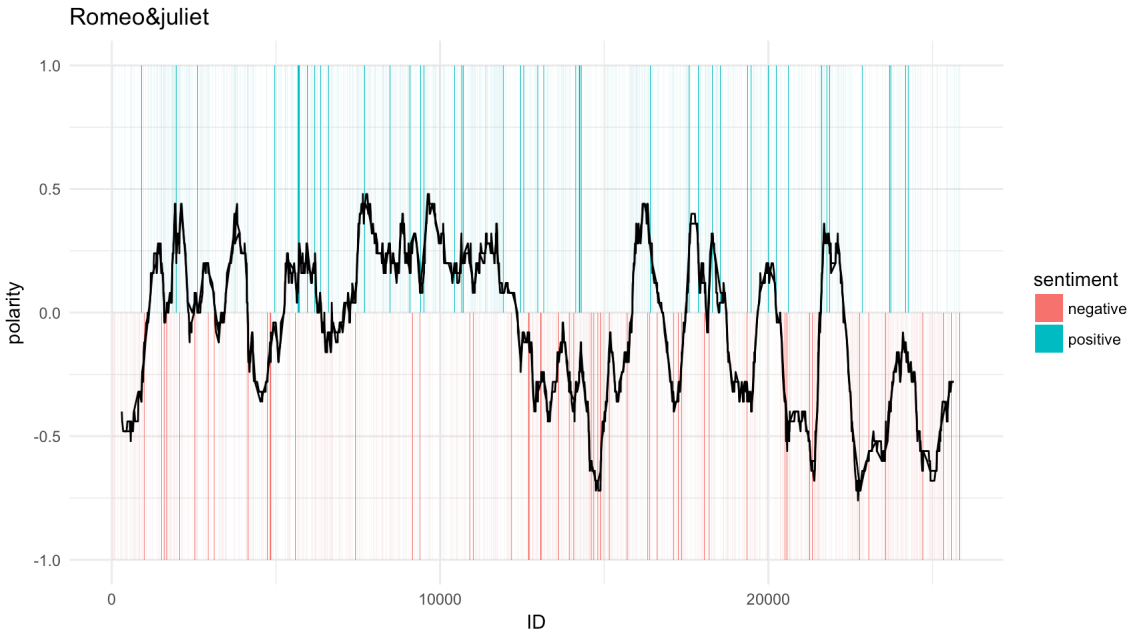
```
## Warning: Unknown or uninitialised column: 'polarity'.
```




```
## Warning: Removed 49 rows containing missing values (geom_path).
```

```
## Joining, by = "word"
```

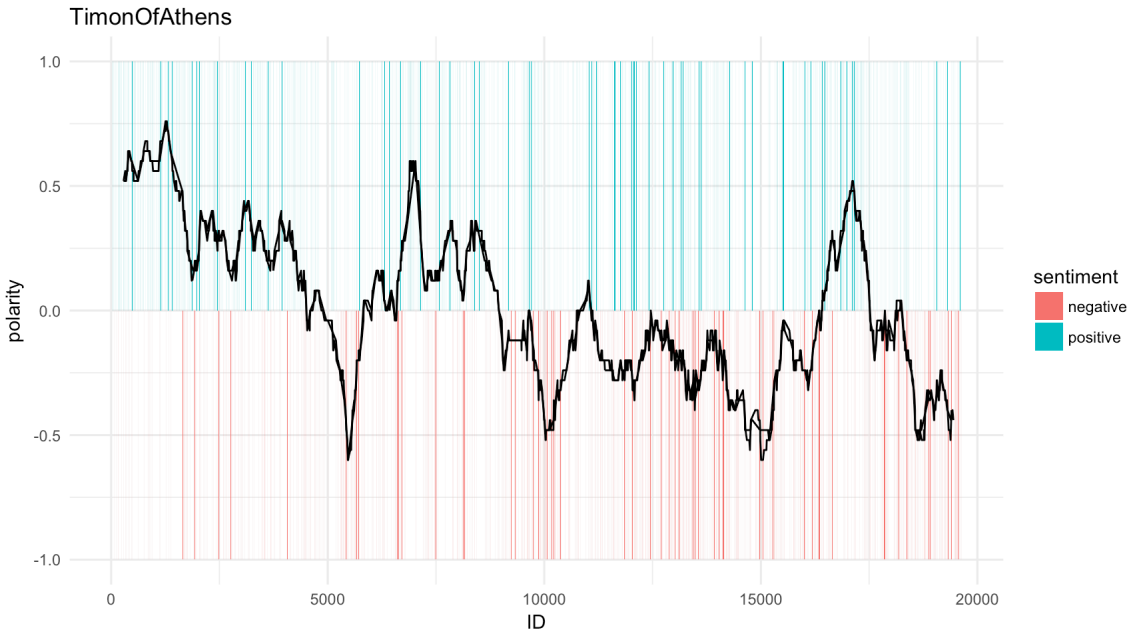
```
## Warning: Unknown or uninitialised column: 'polarity'.
```



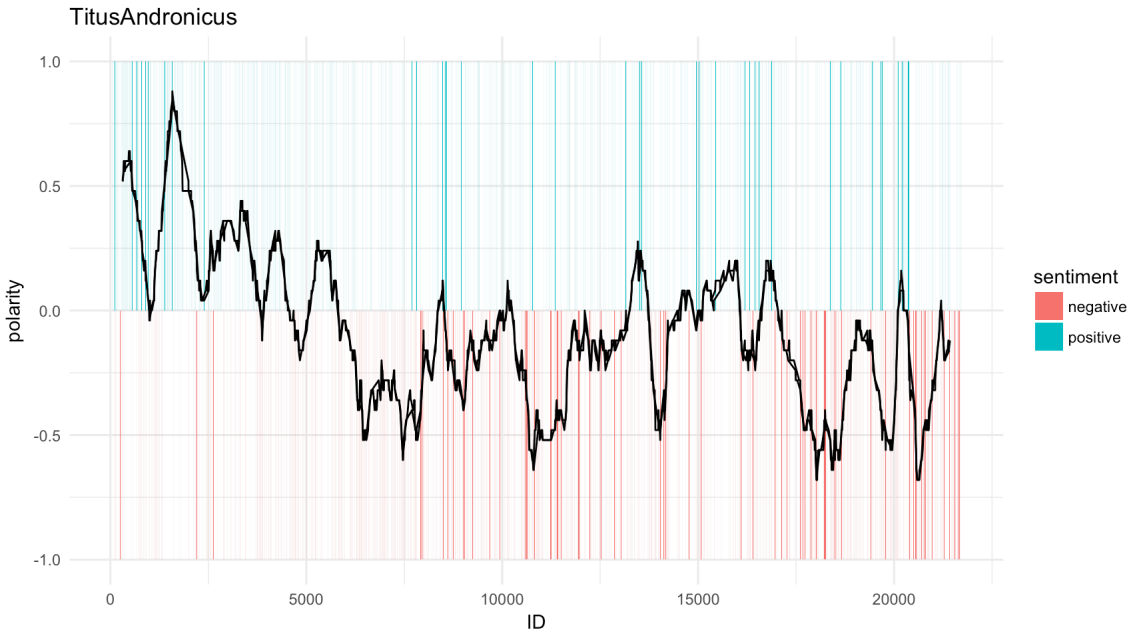

```
## Warning: Removed 49 rows containing missing values (geom_path).
```

```
## Joining, by = "word"
```

```
## Warning: Unknown or uninitialised column: 'polarity'.
```




```
## Warning: Removed 49 rows containing missing values (geom_path).
```



Each bar represents a negative or positive word in chronological order.
The lines represent the rolling means.

Simple temporal sentiment analysis II

Here, we aggregate our data and compute a mean for every 30 observations.

```
for (play in playList){

text=glue(read_file(play))

tokens = tibble(text = tolower(text)) %>% unnest_tokens(word, text)
tokens = rowid_to_column(tokens, "ID")

sentiments = tokens %>%
  inner_join(get_sentiments("bing"))

sentiments$polarity = NULL
sentiments$polarity[sentiments$sentiment=="negative"] = -1
sentiments$polarity[sentiments$sentiment=="positive"] =  1

means=colMeans(matrix(sentiments$polarity, nrow=30))

df=tibble(row=seq(1:length(means)),means)

plot = df %>%
  ggplot() +
  ggtitle(gsub(".txt","",play)) +
  theme_dark()+
  aes(row,means,fill=means) +
  geom_col() +
  ylim(-1,1) +
  scale_fill_gradient2(low = "red", mid = "white",
                       high = "blue", midpoint = 0, space = "Lab",
                       na.value = "grey50", guide = "colourbar")
  print(plot)
}
```

```
## Joining, by = "word"
```

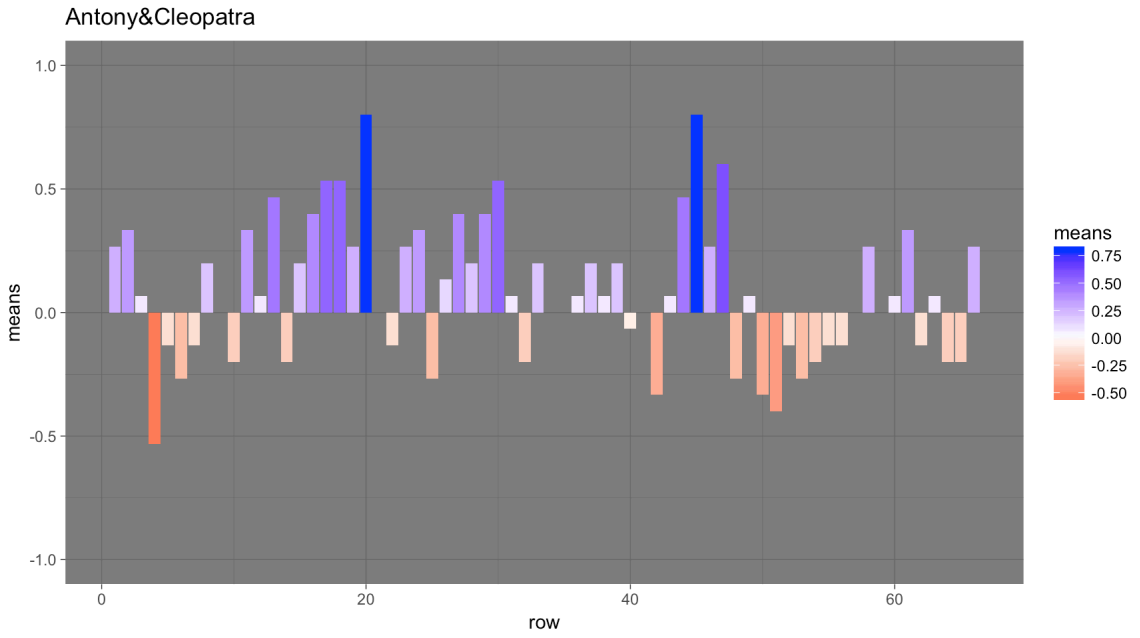
```
## Warning: Unknown or uninitialised column: 'polarity'.
```

```
## Warning in matrix(sentiments$polarity, nrow = 30): data length [1951] is
## not a sub-multiple or multiple of the number of rows [30]
```

```
## Joining, by = "word"
```

```
## Warning: Unknown or uninitialised column: 'polarity'.
```

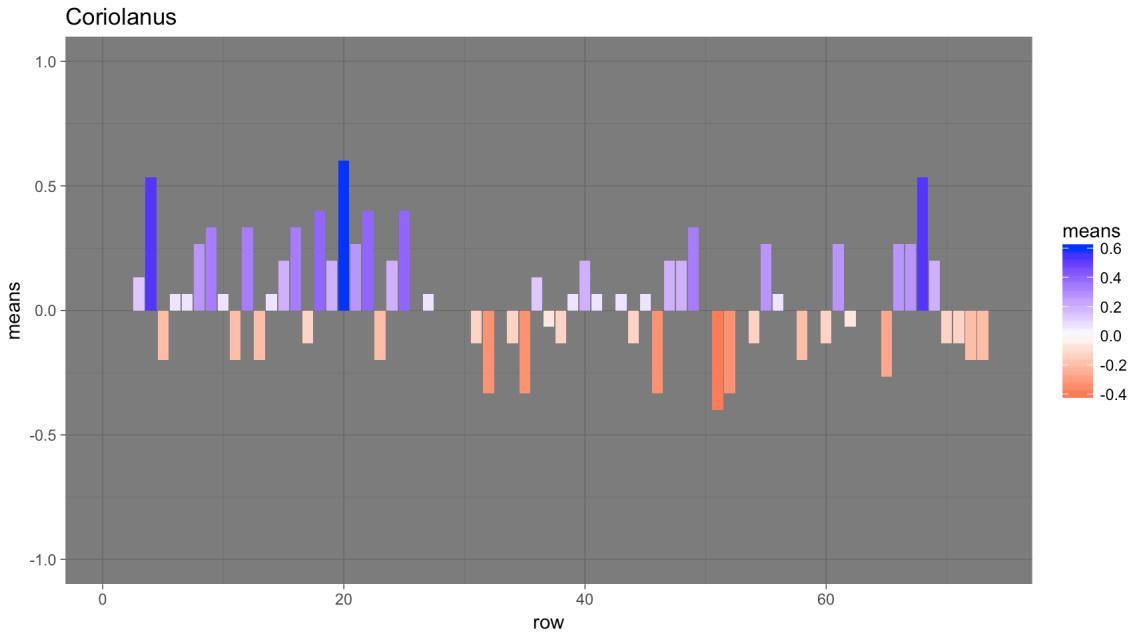
```
## Warning in matrix(sentiments$polarity, nrow = 30): data length [2184] is
## not a sub-multiple or multiple of the number of rows [30]
```




```
## Joining, by = "word"
```

```
## Warning: Unknown or uninitialised column: 'polarity'.
```

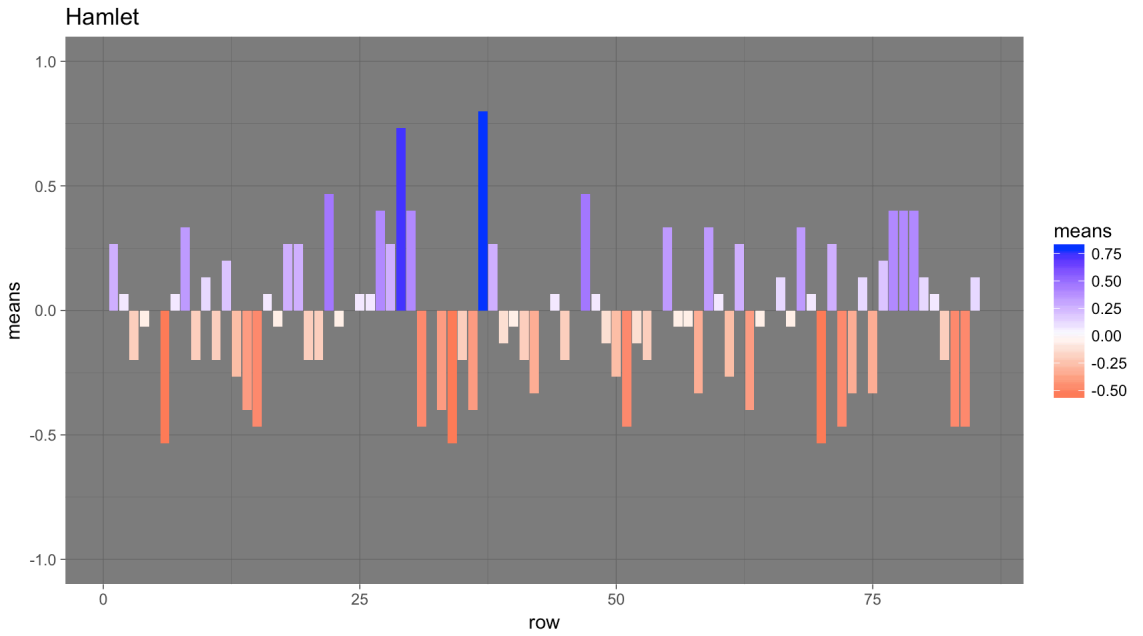
```
## Warning in matrix(sentiments$polarity, nrow = 30): data length [2523] is
## not a sub-multiple or multiple of the number of rows [30]
```




```
## Joining, by = "word"
```

```
## Warning: Unknown or uninitialised column: 'polarity'.
```

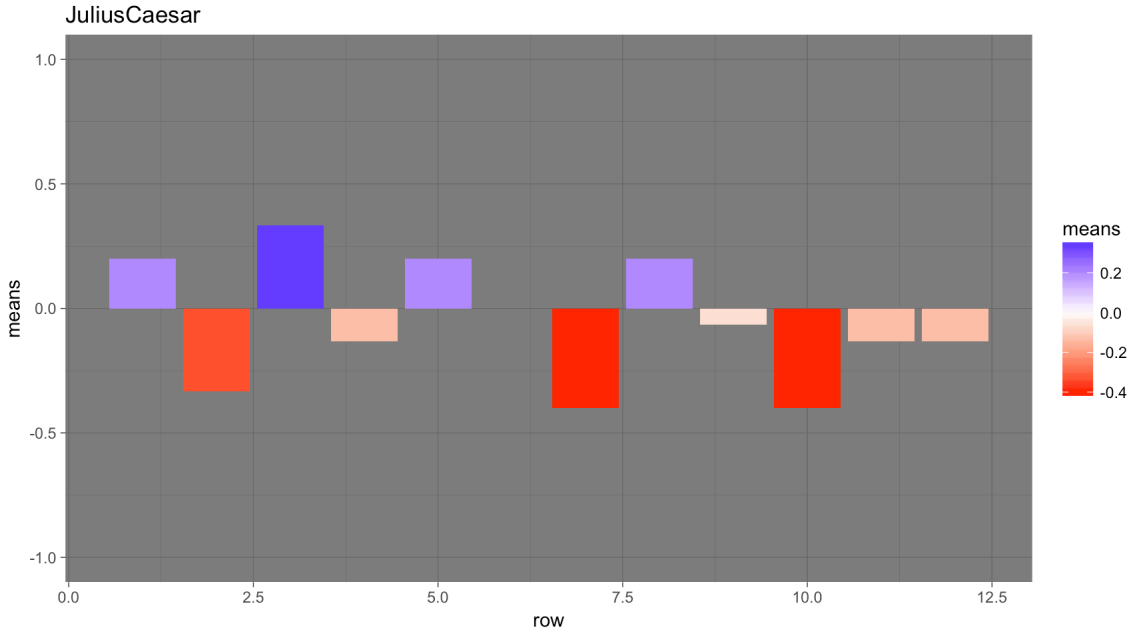
```
## Warning in matrix(sentiments$polarity, nrow = 30): data length [359] is not
## a sub-multiple or multiple of the number of rows [30]
```




```
## Joining, by = "word"
```

```
## Warning: Unknown or uninitialised column: 'polarity'.
```

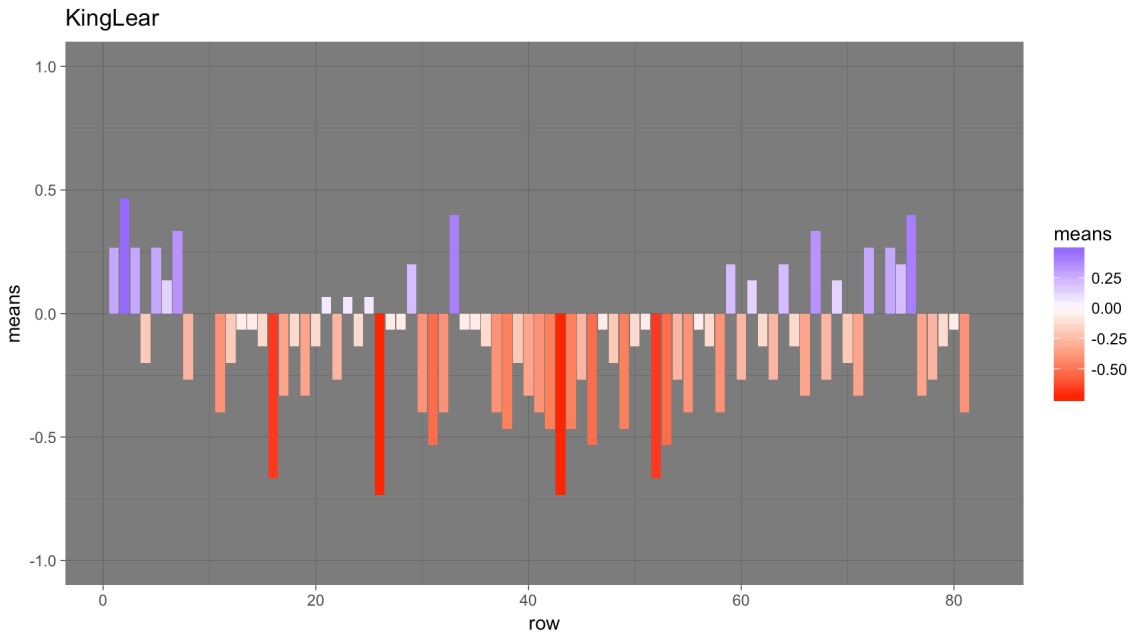
```
## Warning in matrix(sentiments$polarity, nrow = 30): data length [2437] is
## not a sub-multiple or multiple of the number of rows [30]
```



```
## Joining, by = "word"
```

```
## Warning: Unknown or uninitialised column: 'polarity'.
```

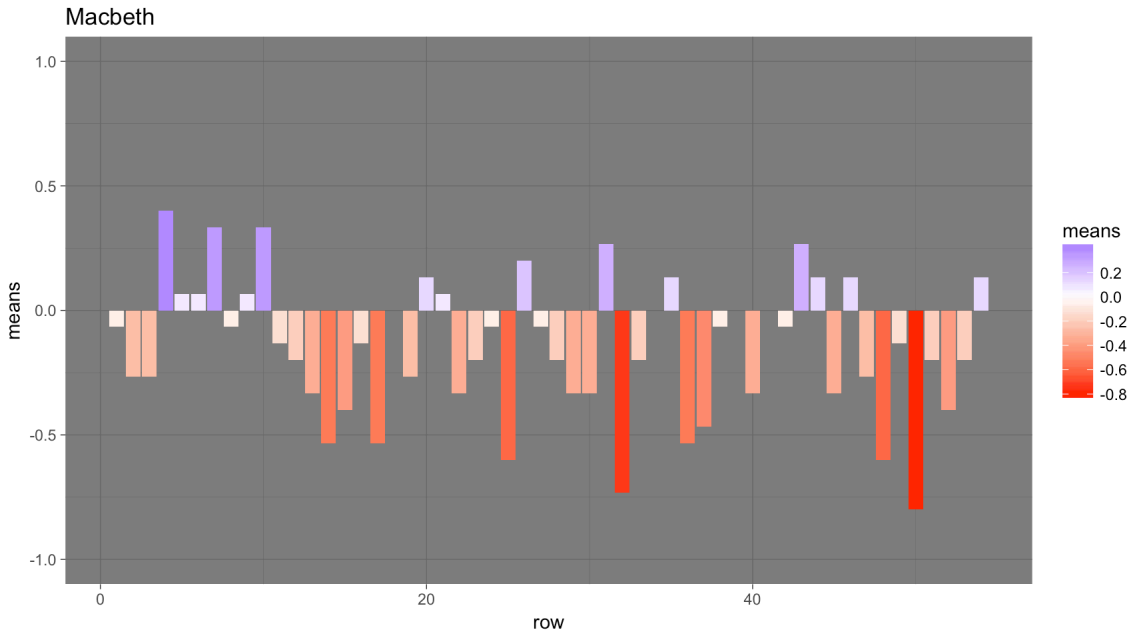
```
## Warning in matrix(sentiments$polarity, nrow = 30): data length [1612] is
## not a sub-multiple or multiple of the number of rows [30]
```




```
## Joining, by = "word"
```

```
## Warning: Unknown or uninitialised column: 'polarity'.
```

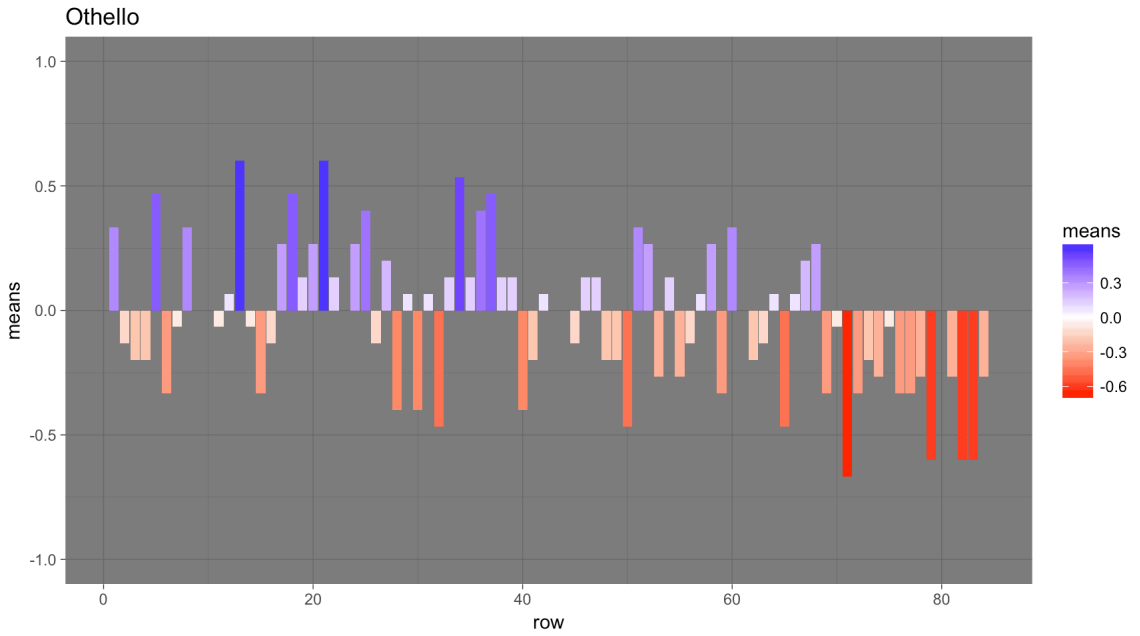
```
## Warning in matrix(sentiments$polarity, nrow = 30): data length [2515] is
## not a sub-multiple or multiple of the number of rows [30]
```




```
## Joining, by = "word"
```

```
## Warning: Unknown or uninitialised column: 'polarity'.
```

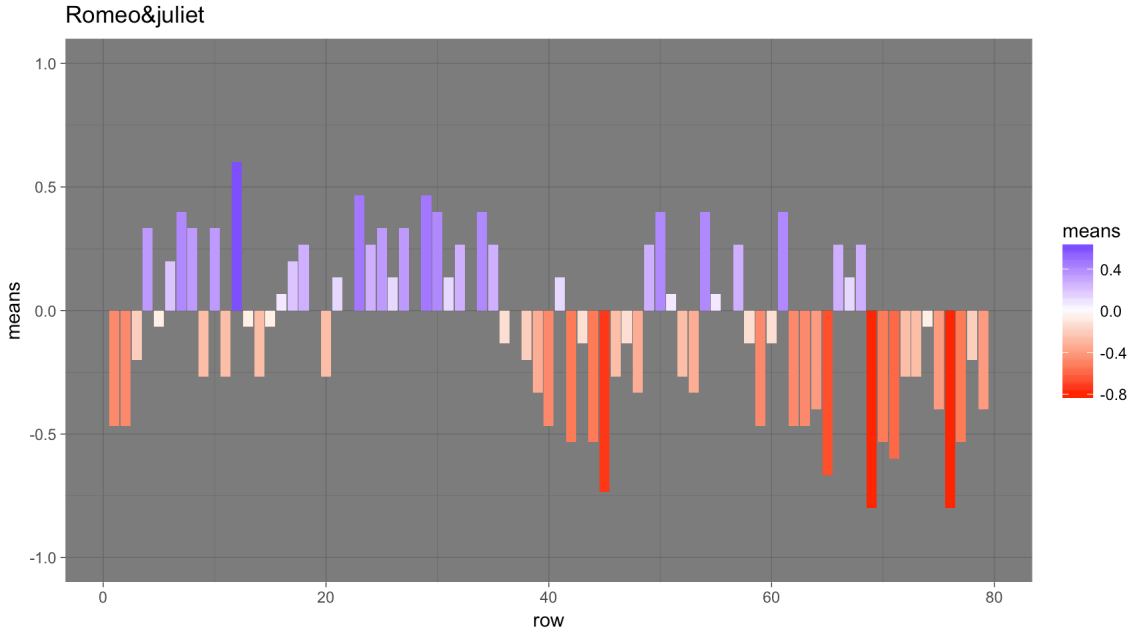
```
## Warning in matrix(sentiments$polarity, nrow = 30): data length [2350] is
## not a sub-multiple or multiple of the number of rows [30]
```




```
## Joining, by = "word"
```

```
## Warning: Unknown or uninitialised column: 'polarity'.
```

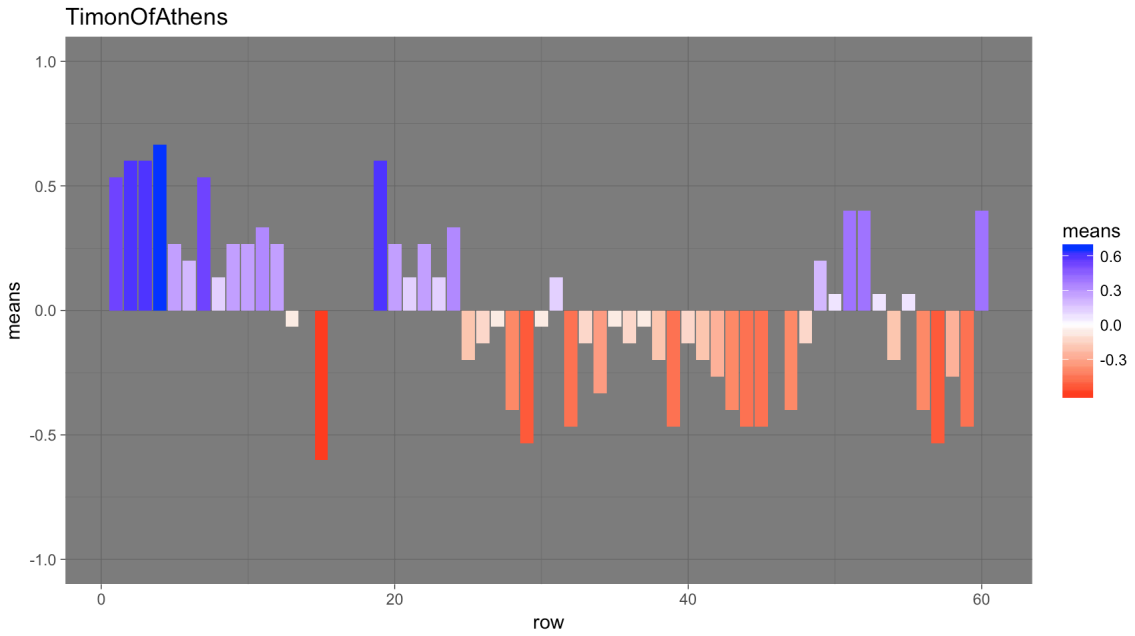
```
## Warning in matrix(sentiments$polarity, nrow = 30): data length [1777] is
## not a sub-multiple or multiple of the number of rows [30]
```

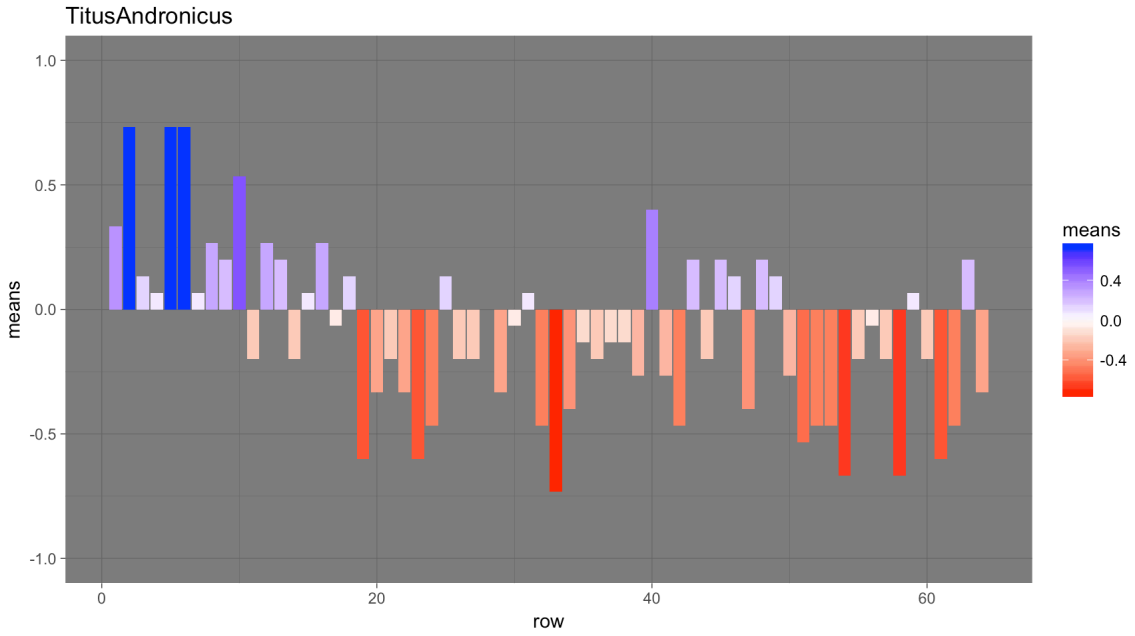



```
## Joining, by = "word"
```

```
## Warning: Unknown or uninitialised column: 'polarity'.
```

```
## Warning in matrix(sentiments$polarity, nrow = 30): data length [1916] is
## not a sub-multiple or multiple of the number of rows [30]
```


TimonOfAthens
means
1.0
0.5
0.0
-0.5
-1.0
0
20
40
60
row
means
0.6
0.3
0.0
-0.3


TitusAndronicus
means
1.0
0.5
0.0
-0.5
-1.0
0
20
40
60
row
means
0.4
0.0
-0.4

BONUS!

```
library(wordcloud2)

df = tibble()

for (play in playList) {

  text = glue(read_file(play))
  tokens = tibble(text = tolower(text)) %>% unnest_tokens(word, text)
  df=rbind(df,tokens)
}

sentiments = df %>%
  group_by(word) %>%
  count(word) %>%
  inner_join(get_sentiments("bing")) %>%
  arrange(desc(n))
```

```
## Joining, by = "word"
```

```
set.seed(1112)
wordcloud2(sentiments, figPath = "silh2.png", size = 1.5, color = "snow", backgro
undColor="black")
```