Sentiment Analysis Social Media Communities & Violence

AUTHOR: SALVATORE CANALE

Project scope

Is sentiment analysis of Twitter data a valid way to predict the likelihood of violence at a protest/rally?

- Obtain Twitter data related to protests/rallies
  - Data obtained through hashag
  - Tested: #unitetheright, #FreeEdNow, and #STLverdict
- #unitetheright and #STLverdict were protests that involved violence
- #FreeEdNow was a non-violent protest
- Use algorithms that analyze words within tweet and rates the tweet “positive” or “negative”
  - If positive, negative, and neutral were considered, this would create ambiguity
- Wordcloud also created and analyzed to ensure context is not skewing data

Libraries

- Bing and NRC libraries are dictionary-based sentiment analysis tools that add up the total number of positive and negative words in each set of texts in order to give a sentiment score
  - Positive score is a positive sentiment
  - Negative score is a negative sentiment
- Bing is a binary (positive or negative) dictionary
  - Named after creator Bing Liu and collaborators
- NRC is dictionary based on a range of emotions
  - Emotions include positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, trust
  - Created by Saif Mohammad and Peter Turney

Analysis Violent Protest: #unitetheright

- Violence tally
  - 3 deaths
    - 1 vehicular homicide
    - 2 State Troopers killed in helicopter crash
  - 38 non-fatal injuries
    - 19 injured during vehicle ramming
    - 14+ injured in general fighting/violence
  - 11 arrests

Analysis Violent Protest: #unitetheright

- Bing library
  - Negative 44
  - Positive 24
  - Sentiment score of -20

```
> unq.charRally_text <- charRally_text[!duplicated(charRally_text)]
> #remove any dollar signs (they're special characters in R)
> unq.charRally_text <- gsub("\\$", "", unq.charRally_text)
>
> #get rid of any trailing spaces
> unq.charRally_text <- trimws(unq.charRally_text)
>
> #tokenize
> tokens <- data_frame(text = unq.charRally_text) %>% unnest_tokens(word,text)
>
> #get the sentiment from the first text:
> tokens %>%
+     inner_join(get_sentiments("bing")) %>% # pull out only sentiment words
+     count(sentiment) %>% # count the # of positive & negative words
+     spread(sentiment, n, fill = 0) %>% # made data wide rather than narrow
+     mutate(sentiment = positive - negative)
Joining, by = "word"
# A tibble: 1 x 3
  negative positive sentiment
     <dbl>    <dbl>     <dbl>
1       44       24       -20
```

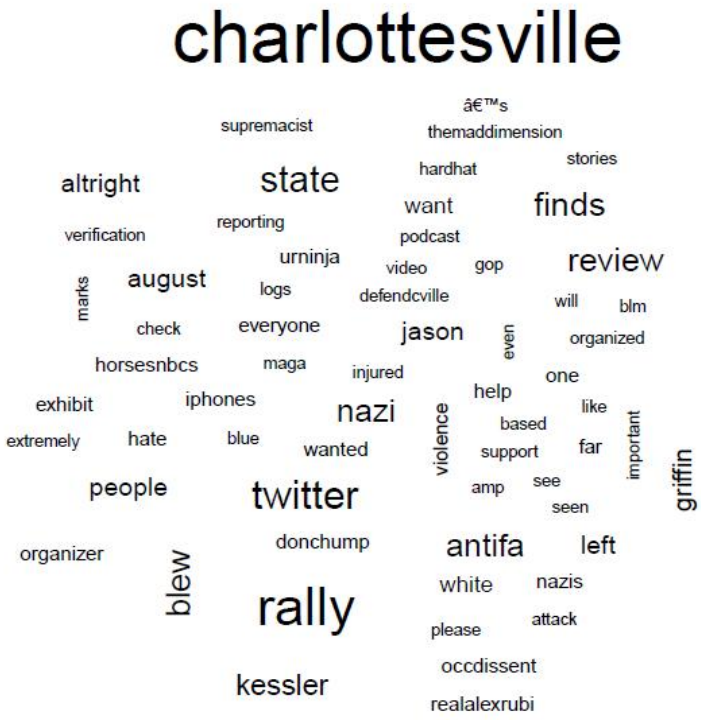
Analysis Violent Protest: #unitetheright

- NRC library
  - Anger 29
  - Anticipation 40
  - Disgust 24
  - Fear 54
  - Joy 24
  - Negative 47
  - Positive 80
  - Sadness 29
  - Surprise 13
  - Trust 56
  - Sentiment 33

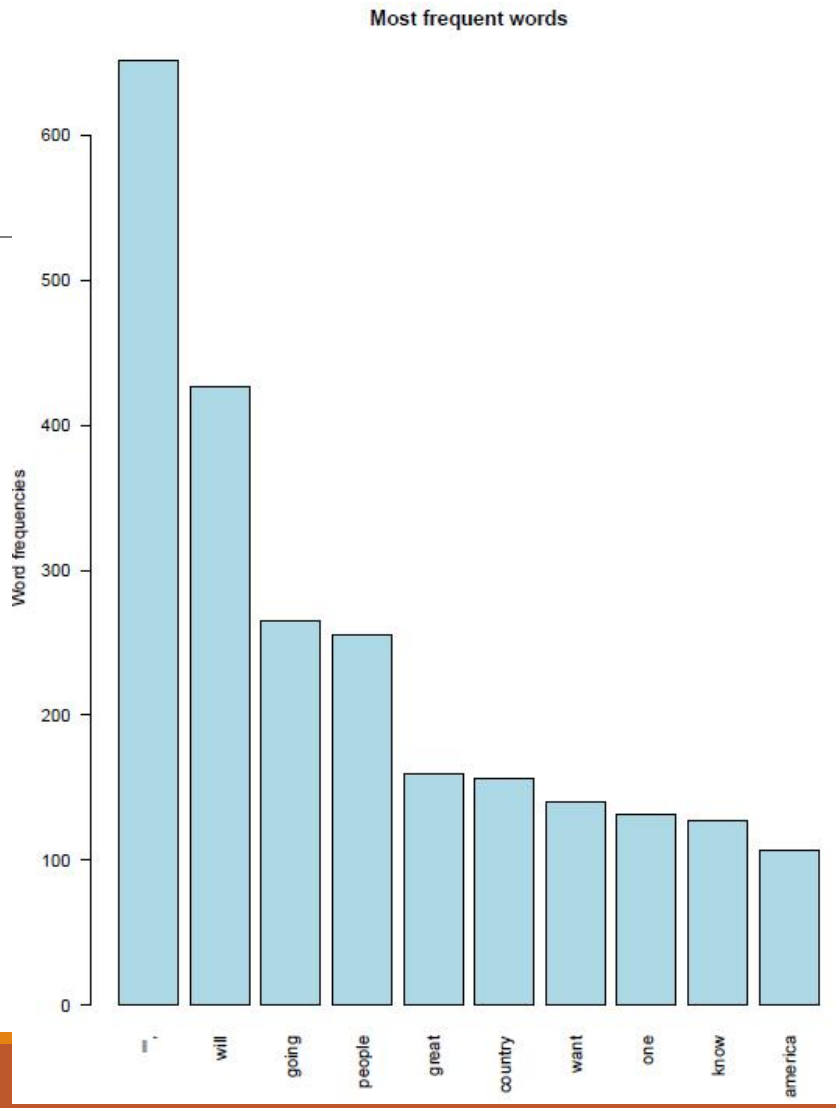
```
> #remove any dollar signs (they're special characters in R)
> unq.charRally_text <- gsub("\\$", "", unq.charRally_text)
>
> #get rid of any trailing spaces
> unq.charRally_text <- trimws(unq.charRally_text)
>
> #tokenize
> tokens <- data_frame(text = unq.charRally_text) %>% unnest_tokens(word,text)
>
> #get the sentiment from the first text:
> tokens %>%
+     inner_join(get_sentiments(lexicon = c("nrc"))) %>% # pull out only sentiment words
+     count(sentiment) %>% # count the # of positive & negative words
+     spread(sentiment, n, fill = 0) %>% # made data wide rather than narrow
+     mutate(sentiment = positive - negative)
Joining, by = "word"
# A tibble: 1 x 11
  anger anticipation disgust  fear   joy negative positive sadness surprise trust sentiment
  <dbl>        <dbl>   <dbl> <dbl> <dbl>    <dbl>    <dbl>   <dbl>    <dbl> <dbl>     <dbl>
1    29           40      24    54    24       47       80      29       13    56        33
>
```

Analysis Violent Protest: #unitetheright

- Human perceived negative words
  - nazi
  - antifa
  - altright
  - violence
  - injured
  - hate
  - attack
- Human perceived positive words
  - support
  - Organized (maybe ?)



Analysis Violent Protest: #unitetheright



Analysis Violent Protest: #STLverdict

- Violence tally
  - 120+ arrests
  - 11 injured law enforcement officers
  - Broken windows in local businesses and library
  - Mayors house vandalized

Analysis Violent Protest: #STLverdict

- Bing library
  - Negative 177
  - Positive 93
  - Sentiment score of -84

```
> # remove any dollar signs (they're special characters in R)
> unq.StlRally_text <- gsub("\\$", "", unq.StlRally_text)
>
> # get rid of any sneaky trailing spaces
> unq.StlRally_text <- trimws(unq.StlRally_text)
>
> #tokenize
> tokens3 <- data_frame(text = unq.StlRally_text) %>% unnest_tokens(word, text)
>
> # get the sentiment from the first text:
> tokens3 %>%
+     inner_join(get_sentiments("bing")) %>% # pull out only sentiment words
+     count(sentiment) %>% # count the # of positive & negative words
+     spread(sentiment, n, fill = 0) %>% # made data wide rather than narrow
+     mutate(sentiment = positive - negative) # # of positive words - # ofnegative words
Joining, by = "word"
# A tibble: 1 x 3
  negative positive sentiment
     <dbl>    <dbl>     <dbl>
1      177       93       -84
```

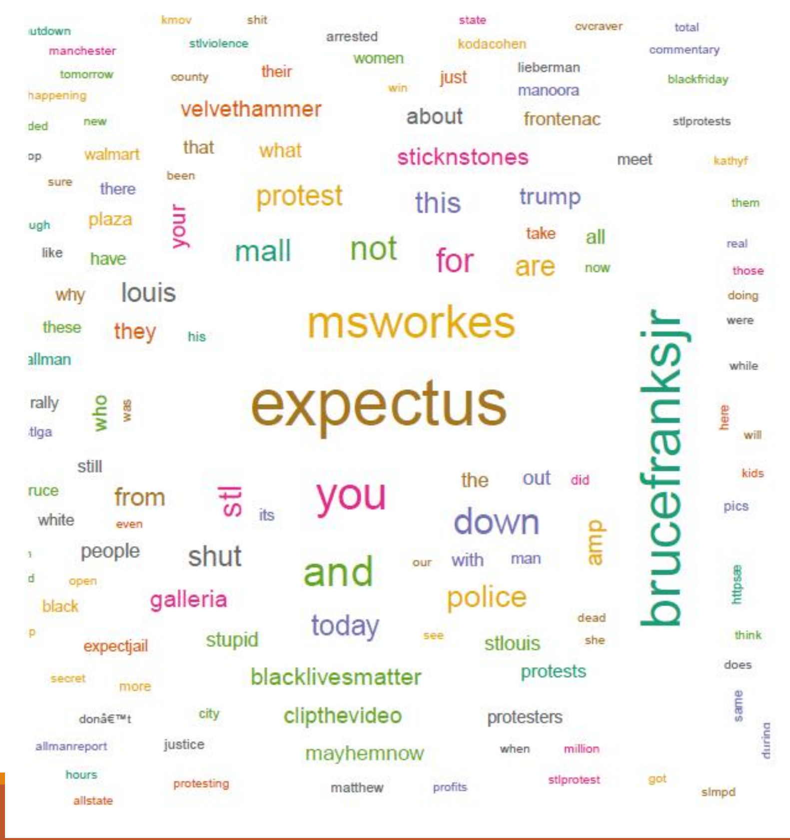
Analysis Violent Protest: #STLverdict

- NRC library
  - Anger 65
  - Anticipation 57
  - Disgust 36
  - Fear 90
  - Joy 41
  - Negative 147
  - Positive 145
  - Sadness 56
  - Surprise 49
  - Trust 109
  - Sentiment -2

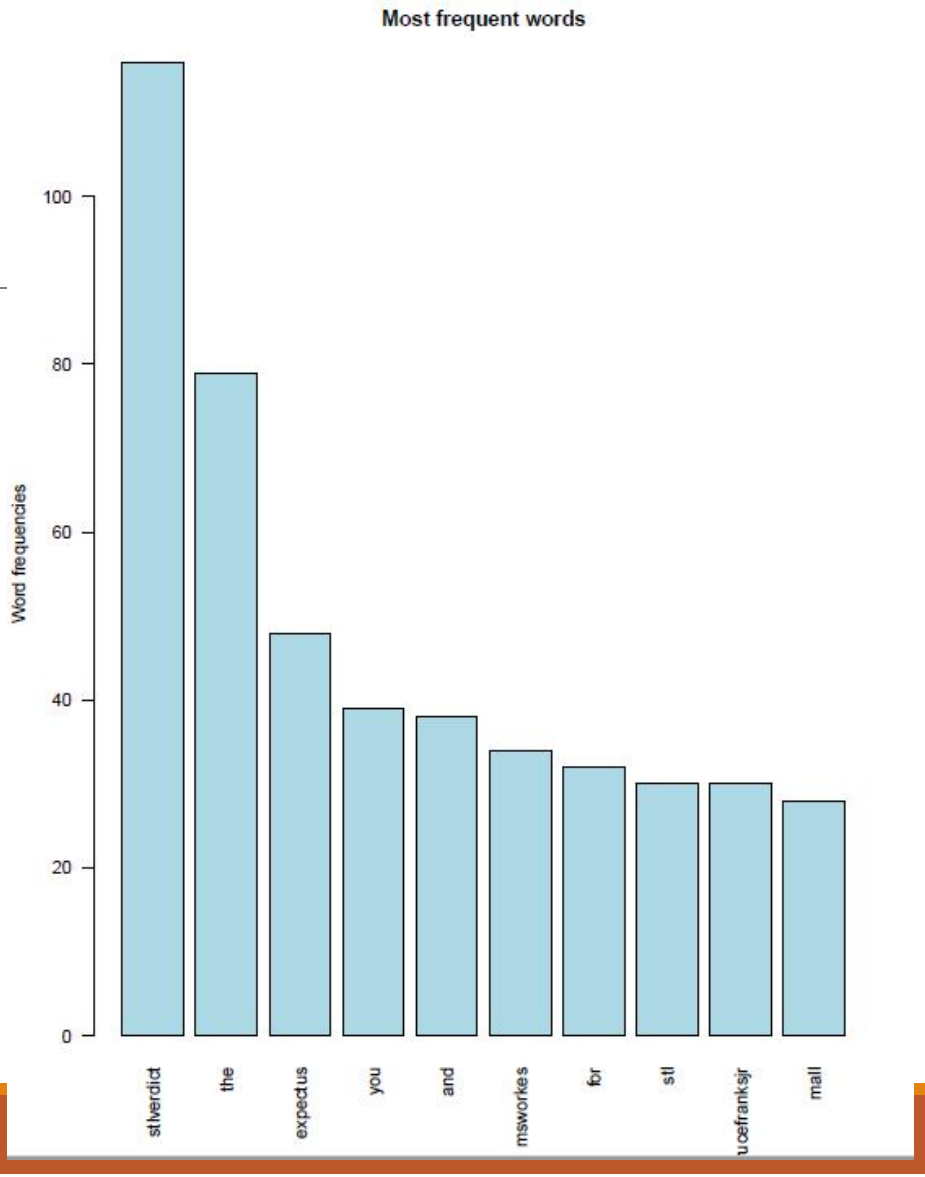
```
> # remove any dollar signs (they're special characters in R)
> unq.StlRally_text <- gsub("\\$", "", unq.StlRally_text)
>
> # get rid of any sneaky trailing spaces
> unq.StlRally_text <- trimws(unq.StlRally_text)
>
> #tokenize
> tokens3 <- data_frame(text = unq.StlRally_text) %>% unnest_tokens(word, text)
>
> # get the sentiment from the first text:
> tokens3 %>%
+     inner_join(get_sentiments(lexicon = c("nrc"))) %>% # pull out only sentiment words
+     count(sentiment) %>% # count the # of positive & negative words
+     spread(sentiment, n, fill = 0) %>% # made data wide rather than narrow
+     mutate(sentiment = positive - negative) # # of positive words - # ofnegative words
Joining, by = "word"
# A tibble: 1 x 11
  anger anticipation disgust  fear   joy negative positive sadness surprise trust sentiment
  <dbl>        <dbl>   <dbl> <dbl> <dbl>    <dbl>    <dbl>   <dbl>    <dbl> <dbl>     <dbl>
1    65           57      36    90    41      147      145      56       49   109        -2
>
```

Analysis Violent Protest: #STLverdict

- Human perceived negative words
  - protest
  - trump
  - arrested
  - expectus
  - expectjail
  - stupid
  - police
  - protesters
  - stlviolence
  - stlprotests
- Human perceived positive words
  - justice



Analysis Violent Protest: #STLverdict



Analysis Nonviolent Protest: #FreeEdNow

- Violence tally
  - 1 fight between a protestor and a rival demonstrator

Analysis Nonviolent Protest: #FreeEdNow

- Bing library
  - Negative 5
  - Positive 16
  - Sentiment score of a positive 11

```
> # remove any dollar signs (they're special characters in R)
> unq.FreeEdRally_text <- gsub("\\$", "", unq.FreeEdRally_text)
>
> # get rid of any sneaky trailing spaces
> unq.FreeEdRally_text <- trimws(unq.FreeEdRally_text)
>
> #tokenize
> tokens2 <- data_frame(text = unq.FreeEdRally_text) %>% unnest_tokens(word, text)
>
> # get the sentiment from the first text:
> tokens2 %>%
+     inner_join(get_sentiments("bing")) %>% # pull out only sentiment words
+     count(sentiment) %>% # count the # of positive & negative words
+     spread(sentiment, n, fill = 0) %>% # made data wide rather than narrow
+     mutate(sentiment = positive - negative) # # of positive words - # ofnegative words
Joining, by = "word"
# A tibble: 1 x 3
  negative positive sentiment
     <dbl>    <dbl>     <dbl>
1        5       16        11
```

Analysis Nonviolent Protest: #FreeEdNow

- NRC
  - Anger 3
  - Anticipation 3
  - Disgust 3
  - Fear 2
  - Negative 7
  - Positive 9
  - Sadness 3
  - Sentiment 2

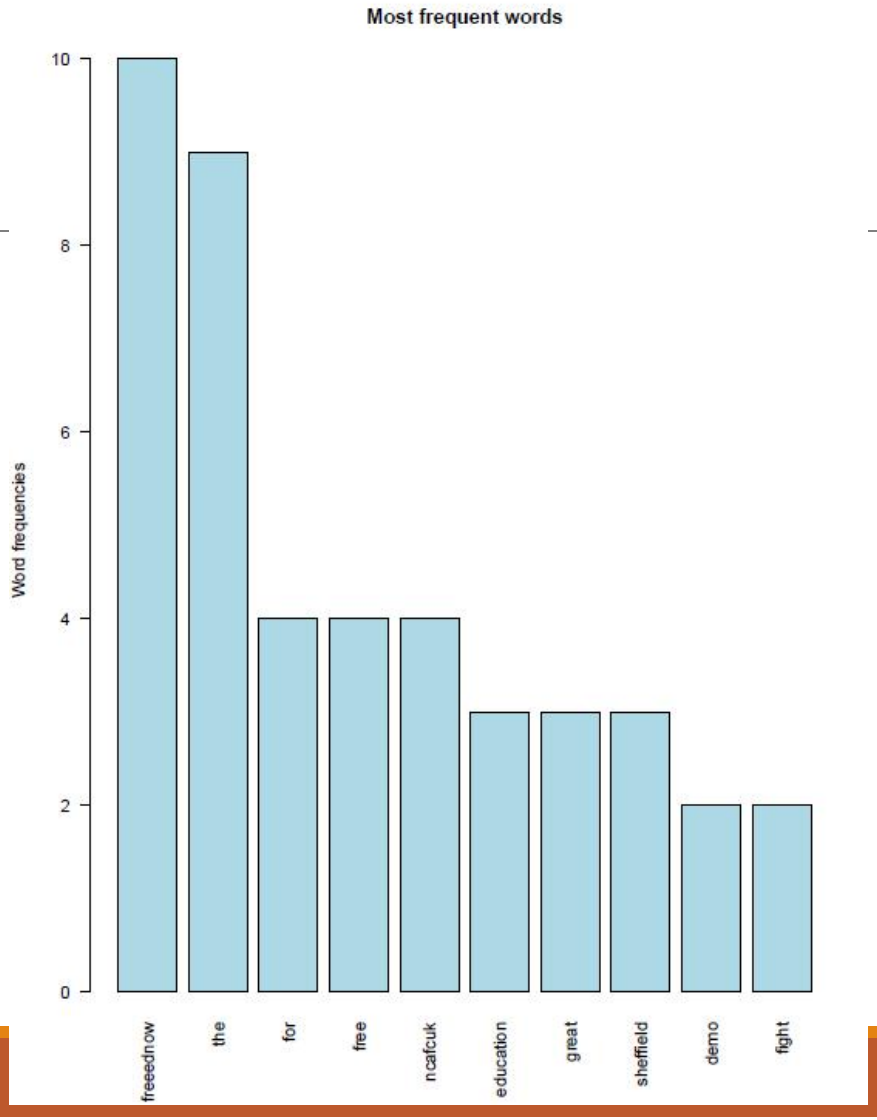
```
> # remove any dollar signs (they're special characters in R)
> unq.FreeEdRally_text <- gsub("\\$", "", unq.FreeEdRally_text)
>
> # get rid of any sneaky trailing spaces
> unq.FreeEdRally_text <- trimws(unq.FreeEdRally_text)
>
> #tokenize
> tokens2 <- data_frame(text = unq.FreeEdRally_text) %>% unnest_tokens(word, text)
>
> # get the sentiment from the first text:
> tokens2 %>%
+     inner_join(get_sentiments(lexicon = c("nrc"))) %>% # pull out only sentiment words
+     count(sentiment) %>% # count the # of positive & negative words
+     spread(sentiment, n, fill = 0) %>% # made data wide rather than narrow
+     mutate(sentiment = positive - negative) # # of positive words - # ofnegative words
Joining, by = "word"
# A tibble: 1 x 8
  anger anticipation disgust  fear negative positive sadness sentiment
  <dbl>        <dbl>   <dbl> <dbl>    <dbl>    <dbl>   <dbl>     <dbl>
1     3            3       3     2        7        9       3         2
```

Analysis Nonviolent Protest: #FreeEdNow

- Human perceived negative words
  - fight
  - sick

- Human perceived positive words
  - free
  - helping
  - great

Analysis Nonviolent Protest: #FreeEdNow



Final Thoughts

- Analyzed each dataset with 2 different algorithms
  - Sentiment score was consistent 2 out of the 3 times
- Human sentiment analysis through Wordcloud consistent with machine sentiment analysis
- Loss in data accuracy
  - Ambiguous words
  - Words that could be different depending on context
- Struggles
  - Obtaining large enough data set
  - Twitter developer api only pulls from more recent Tweets
- Twitter data is a valid way to predict the likelihood of violence at a protest/rally

End

Questions?

1) Connect to Twitter Developer API

```
#twitter API authorization
consumer_key <- 'UErWwe1a3aTpb1UcxVA63iZ6a'
consumer_secret <- '3yQDTxki1WTtGLkeAs41t5xcHkVVjYDI9QA758gWfBaRGZeG8B'
access_token <- '619686486-wydSYycZEHzkmHHjrRFX5s3GNSq51eZkiHIAwoaa'
access_secret <- '661COl1iiI668wRAyXhG2vAAK0pqLFeidAJmTEGKEZKII'

setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

##################################################################################
```

2)Obtain data based on keywords or twitter #hashtags

```
#############################################################################

#rally that resulted in violence

#searches most recent tweets
#charRally <- searchTwitter('#unitetheright', lang="en", n=500, resultType="recent")

#Grabs 400 tweets
charRally <- searchTwitter('#unitetheright', 400)

class(charRally)

charRally_text <- sapply(charRally, function(x) x$getText())
str(charRally_text)


unq.charRally_text <- charRally_text[!duplicated(charRally_text)]
```

3)Clean the data by removing symbols, unnecessary words such as pronouns, trailing spaces, and make formatting consistent

```
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""

#remove any dollar signs (they're special characters in R)
unq.charRally_text <- gsub("\\$", "", unq.charRally_text)

#get rid of any trailing spaces
unq.charRally_text <- trimws(unq.charRally_text)
```

4)Run sentiment analysis algorithm and report sentiment analysis score

```
> unq.charRally_text <- charRally_text[!duplicated(charRally_text)]
> #remove any dollar signs (they're special characters in R)
> unq.charRally_text <- gsub("\\$", "", unq.charRally_text)
>
> #get rid of any trailing spaces
> unq.charRally_text <- trimws(unq.charRally_text)
>
> #tokenize
> tokens <- data_frame(text = unq.charRally_text) %>% unnest_tokens(word,text)
>
> #get the sentiment from the first text:
> tokens %>%
+     inner_join(get_sentiments("bing")) %>% # pull out only sentiment words
+     count(sentiment) %>% # count the # of positive & negative words
+     spread(sentiment, n, fill = 0) %>% # made data wide rather than narrow
+     mutate(sentiment = positive - negative)
Joining, by = "word"
# A tibble: 1 x 3
  negative positive sentiment
     <dbl>    <dbl>     <dbl>
1       44       24       -20
```

```
> #remove any dollar signs (they're special characters in R)
> unq.charRally_text <- gsub("\\$", "", unq.charRally_text)
>
> #get rid of any trailing spaces
> unq.charRally_text <- trimws(unq.charRally_text)
>
> #tokenize
> tokens <- data_frame(text = unq.charRally_text) %>% unnest_tokens(word,text)
>
> #get the sentiment from the first text:
> tokens %>%
+     inner_join(get_sentiments(lexicon = c("nrc"))) %>% # pull out only sentiment words
+     count(sentiment) %>% # count the # of positive & negative words
+     spread(sentiment, n, fill = 0) %>% # made data wide rather than narrow
+     mutate(sentiment = positive - negative)
Joining, by = "word"
# A tibble: 1 x 11
  anger anticipation disgust  fear   joy negative positive sadness surprise trust sentiment
  <dbl>        <dbl>   <dbl> <dbl> <dbl>    <dbl>    <dbl>   <dbl>    <dbl> <dbl>     <dbl>
1    29           40      24    54    24       47       80      29       13    56        33
>
```

6)Organize sentiment analysis data into charts and wordclouds to visually report results of data