Final Project

Chris Obermeier

12/11/2019

##Hypthosis

The null hypothesis for a 2-sample t-test of this question is

H_0: Twitter is able to show the same rankings of popularity for party debates as polling can during the time period.

H_1: The two are different and not equivalent.

Note:This is a slight deviation from the original plan since I had originally hoped to use the data during the debate using the twitter search attributes: since = '2019-11-20', until = '2019-11-21', however technical difficulties ahve prevented me from doing this.

##Packages

Packages loaded here.

```r
library("twitteR")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:twitteR':
##
##     id, location

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library('sentimentr')
library('ggplot2')
```

Setup TWitter

Setup Twitter Keys and OAuth here.

```r
origop <- options("httr_oauth_cache")
options(httr_oauth_cache=TRUE)

consumerKey = "qx2TnLPN1ti4ftGYGnjxch0ig"
consumerSecret = "HUg2IRUmbYlRkpGgxxakxB2WNnnYkHL3H4FVd10x8C9pxWoYao"
```

```
accessToken ="94134854-JXQGar3FCkKEi5DvF2LKv4DYJHkMQFyhZH9vMZAMp"
accessTokenSecret = "ZRIvJyWoiWD6F4qSdfGhP8OKx4FkqvLBibpcZUPyjPxqR"

setup_twitter_oauth(consumerKey,consumerSecret,accessToken,accessTokenSecret)
```

```
## [1] "Using direct authentication"
```

```
options(httr_oauth_cache=origop)
```

Grab twtitter data

Use Twitter's API to pull tweets that have the two most populat hashtags for the democratic debate: #demdebate or #democraticdebate.

```
hashtag1 = "#demdebate"
hashtag2 = "#democraticdebate"

tw1 = twitteR::searchTwitter(hashtag1, n=1e3, retryOnRateLimit = 1e3)
d1 = twitteR::twListToDF(tw1)

tw2 = twitteR::searchTwitter(hashtag2, n=1e3, retryOnRateLimit = 1e3)
d2 = twitteR::twListToDF(tw2)
```

#Check Twitter Data

Combine dataframes of both hashtags

```
#Combine the hashtag dataframes
df <- rbind(d1,d2)

write.csv(df, file="DemocraticDebate_TweetData.csv")
```

##set keywords for each candidate

```
joebiden = "joe|biden"
elizabethwarren = "elizabeth|warren|liz"
berniesanders = "bernie|sanders"
petebuttigieg = "pete|buttigieg"
corybooker = "cory|booker"
amyklobuchar = "amy|klobuchar"
mikebloomberg = "mike|bloomberg"
andrewyang = "andrew|yang"
```

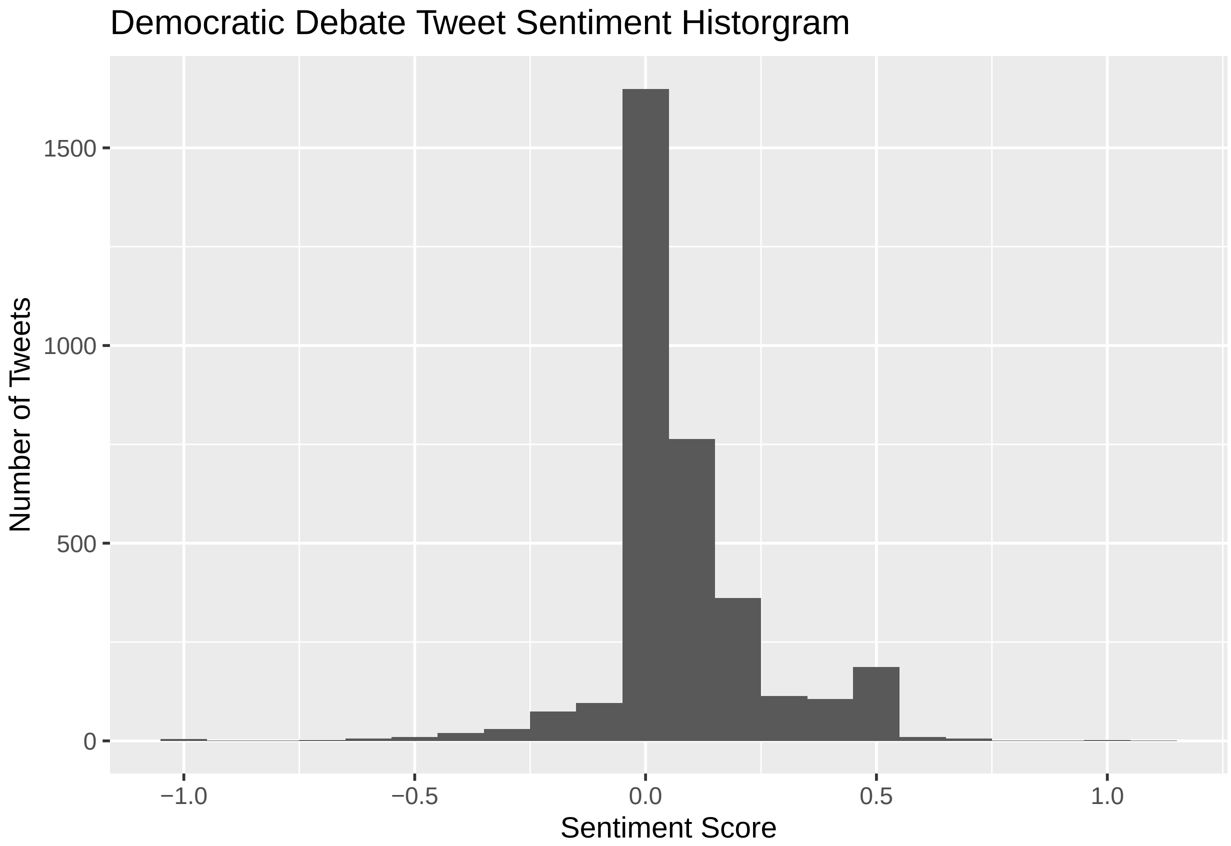
##Search for keywords and assign logical values to dataframe row for each candidate if they appear in the tweet or not

```
df$joebiden <- grepl(joebiden, df$text, ignore.case = TRUE)
df$elizabethwarren <- grepl(elizabethwarren, df$text, ignore.case = TRUE)
df$berniesanders <- grepl(berniesanders, df$text, ignore.case = TRUE)
df$petebuttigieg <- grepl(petebuttigieg, df$text, ignore.case = TRUE)
```

```
df$corybooker <- grepl(corybooker, df$text, ignore.case = TRUE)
df$amyklobuchar <- grepl(amyklobuchar, df$text, ignore.case = TRUE)
df$mikebloomberg <- grepl(mikebloomberg, df$text, ignore.case = TRUE)
df$andrewyang <- grepl(andrewyang, df$text, ignore.case = TRUE)
```

##Overall sentiment analysis of R, and Histogram of scoring

```
sentiment_df <- sentiment(get_sentences(df$text))

qplot(sentiment_df$sentiment, geom="histogram", xlab = "Sentiment Score", ylab = "Number of Tweets", bi
```

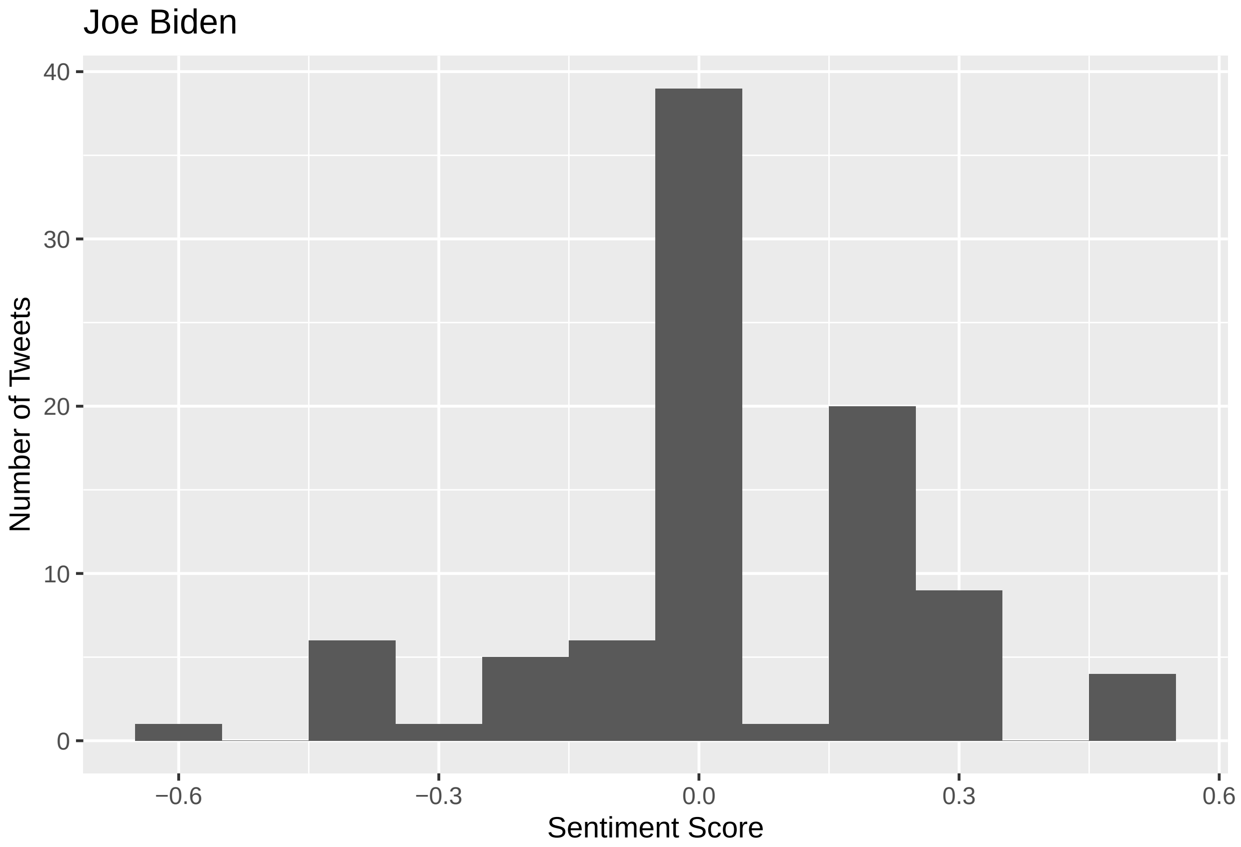


##Create function for Individual Candidate Sentiment Analysis Graphing

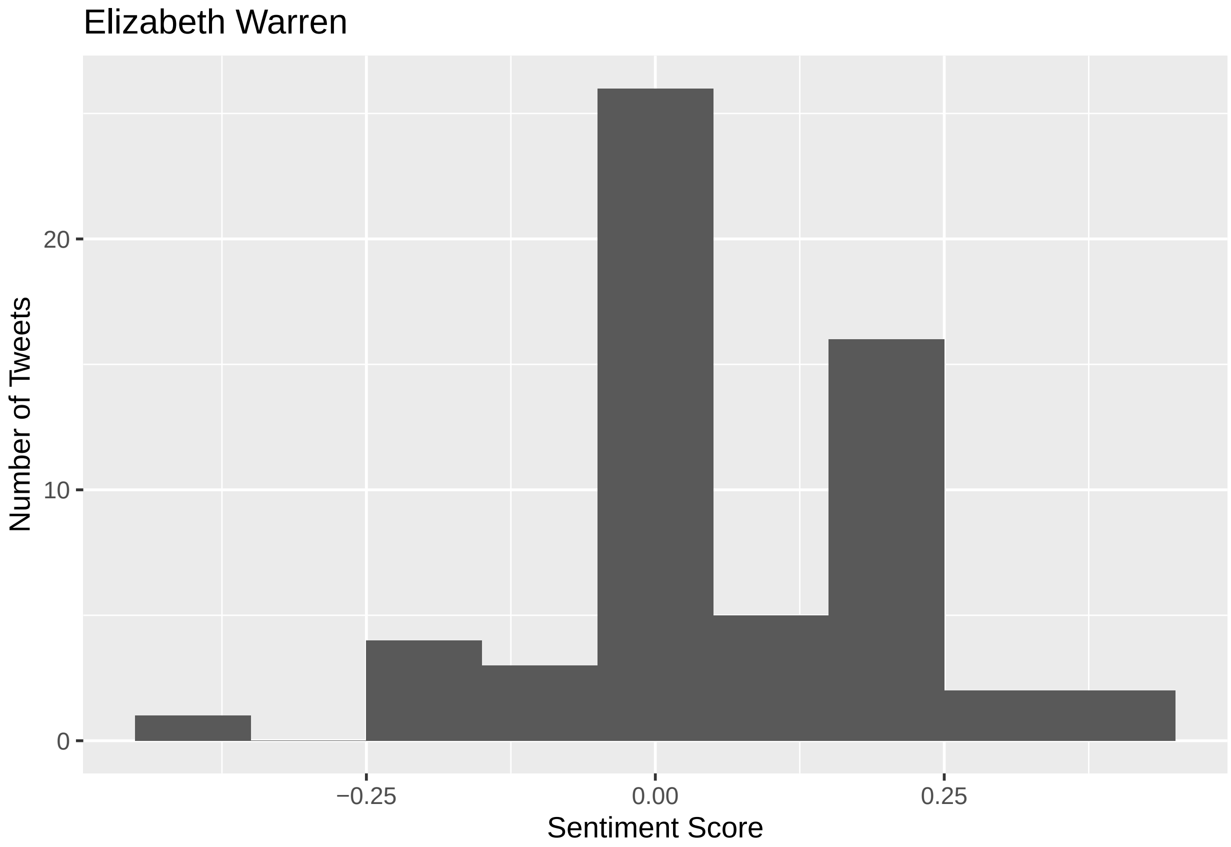
```
sentiment_analysis <- function(sentiment, candidateString){

  qplot(sentiment, geom="histogram", xlab = "Sentiment Score", ylab = "Number of Tweets", binwidth=0.1,

}
```

##Candidate Tweet Sentiment Analysis

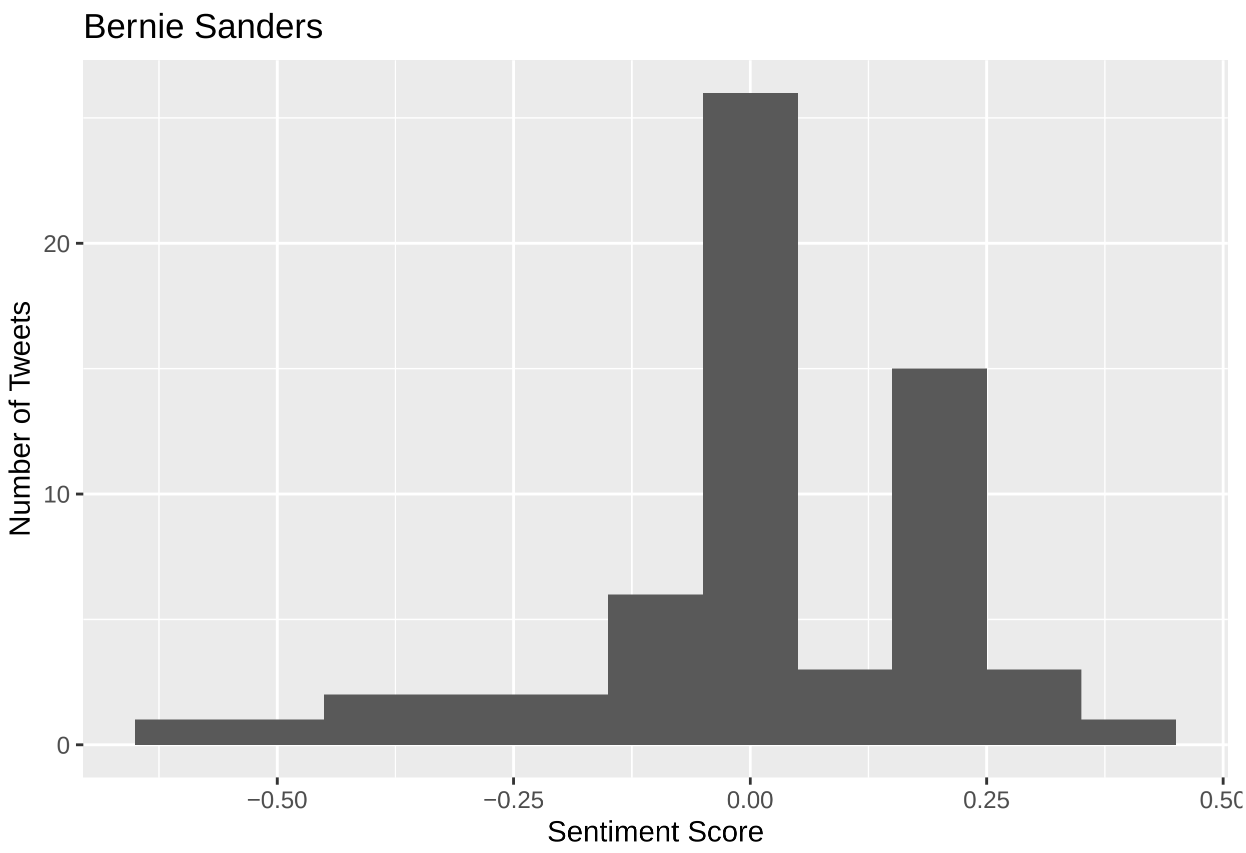
```
sentiment_mean_list <- list()

sentiment <- sentiment(get_sentences(df$text[df$joebiden]))
sentiment_analysis(sentiment$sentiment, "Joe Biden")
```



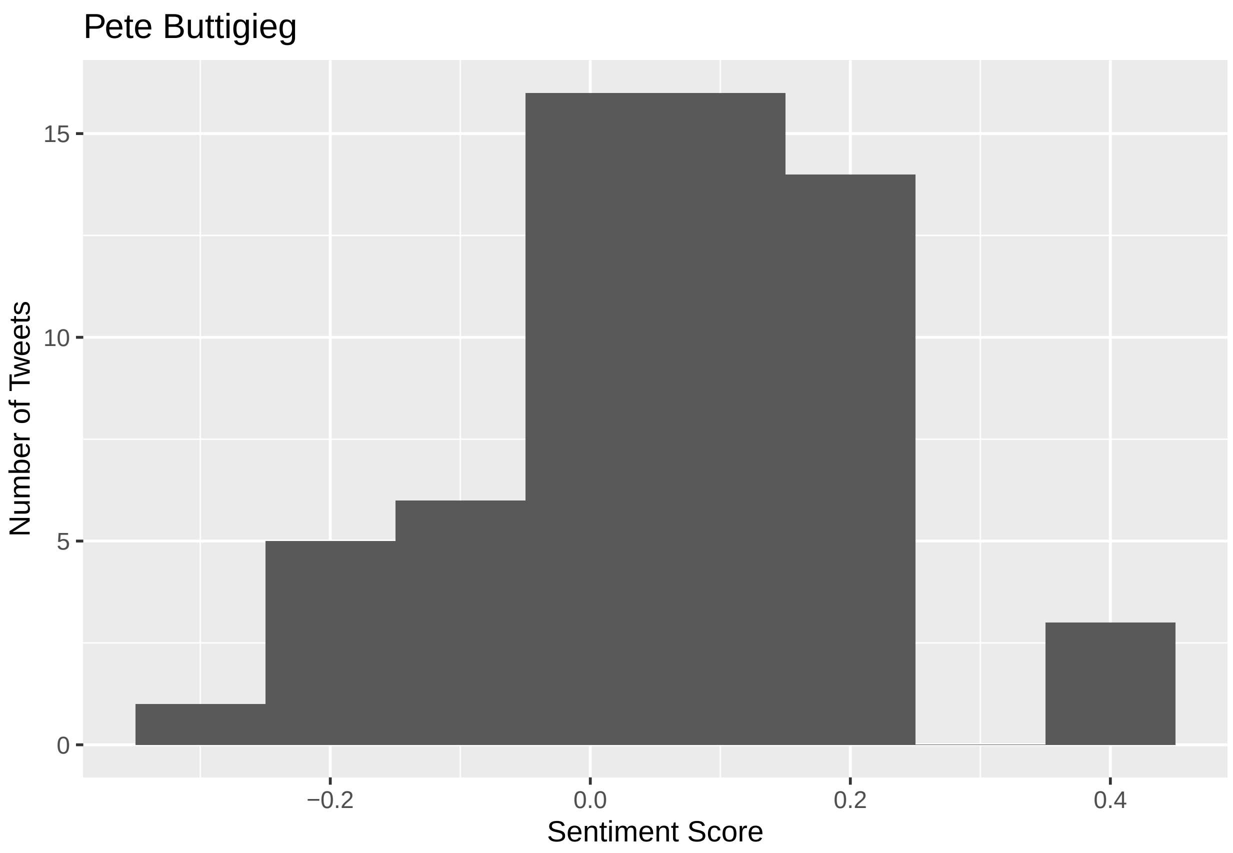
```
sentiment_mean_list <- append(sentiment_mean_list, mean(sentiment$sentiment))

sentiment <- sentiment(get_sentences(df$text[df$elizabethwarren]))
sentiment_analysis(sentiment$sentiment, "Elizabeth Warren")
```



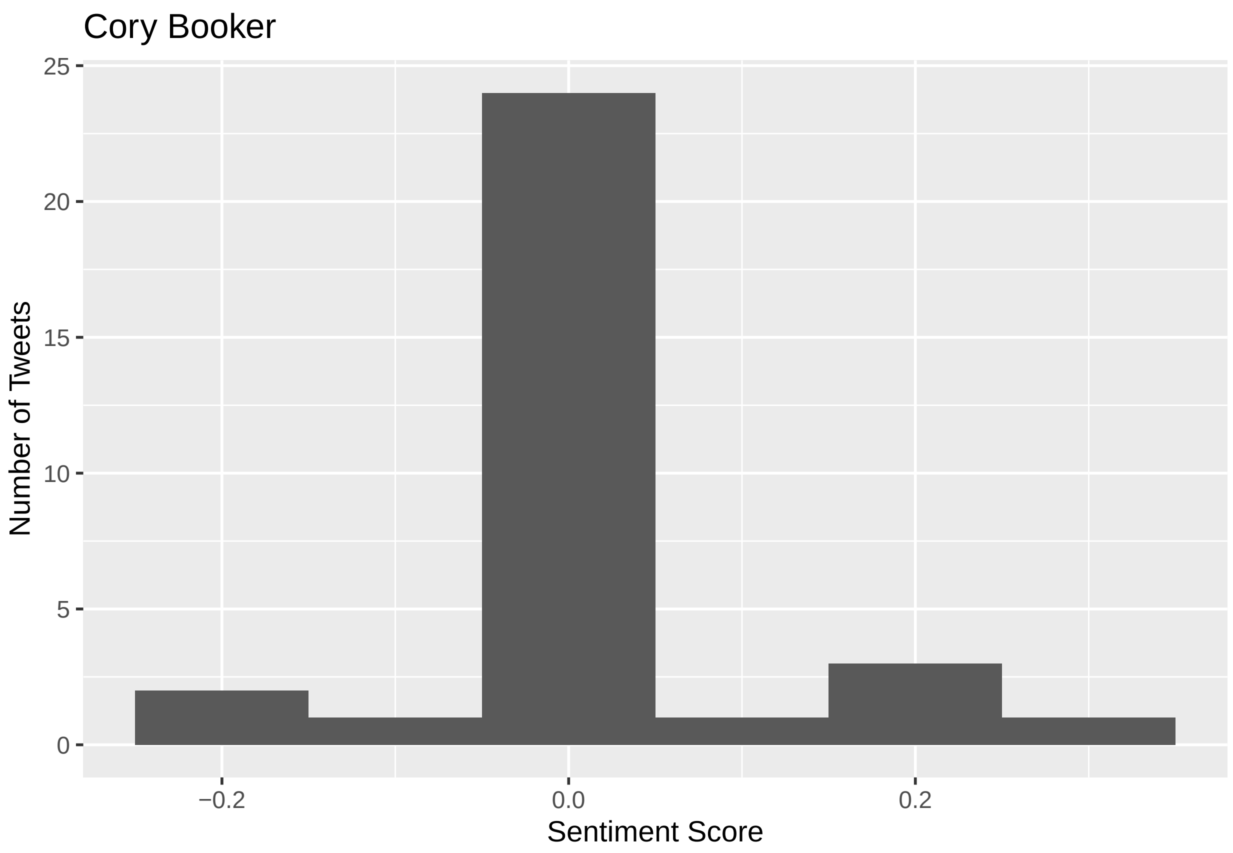
```r
sentiment_mean_list <- append(sentiment_mean_list, mean(sentiment$sentiment))

sentiment <- sentiment(get_sentences(df$text[df$berniesanders]))
sentiment_analysis(sentiment$sentiment, "Bernie Sanders")
```



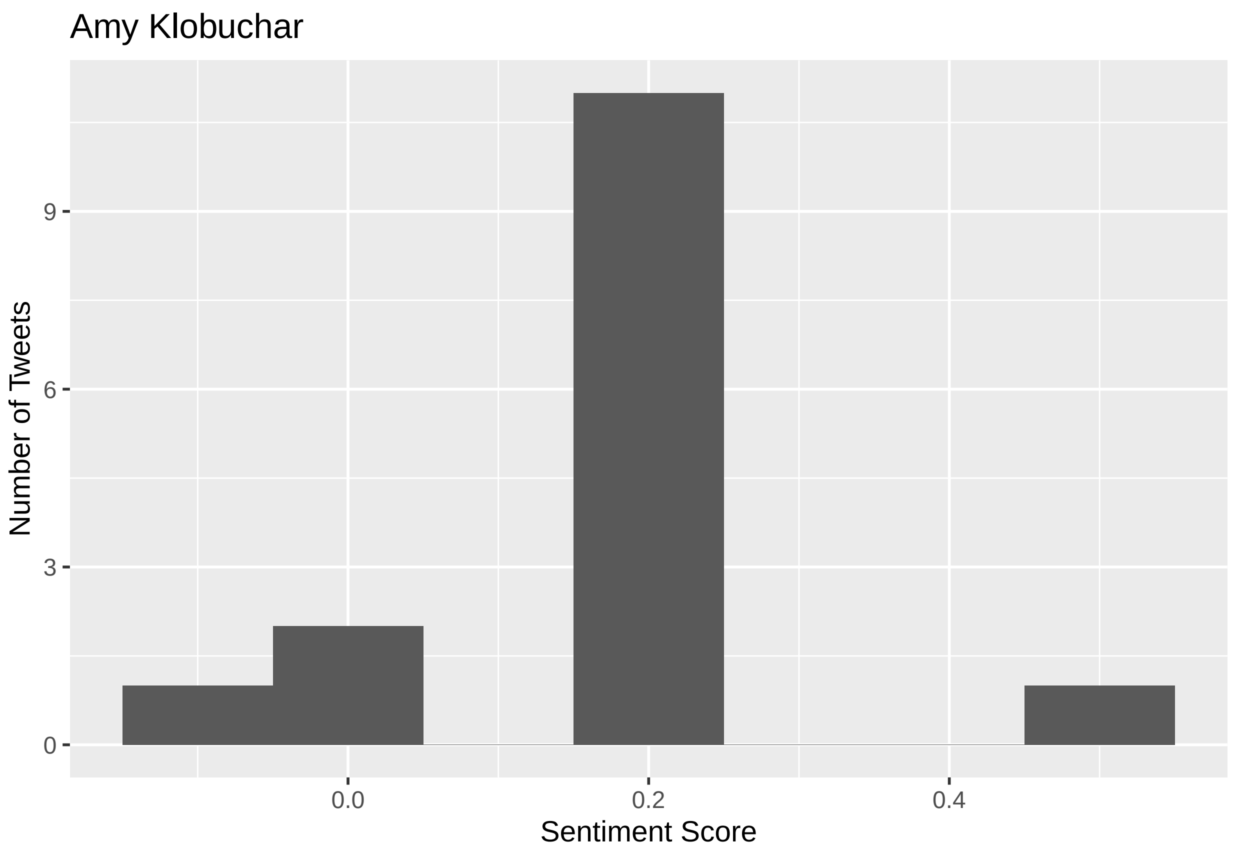
```
sentiment_mean_list <- append(sentiment_mean_list, mean(sentiment$sentiment))

sentiment <- sentiment(get_sentences(df$text[df$petebuttigieg]))
sentiment_analysis(sentiment$sentiment, "Pete Buttigieg")
```



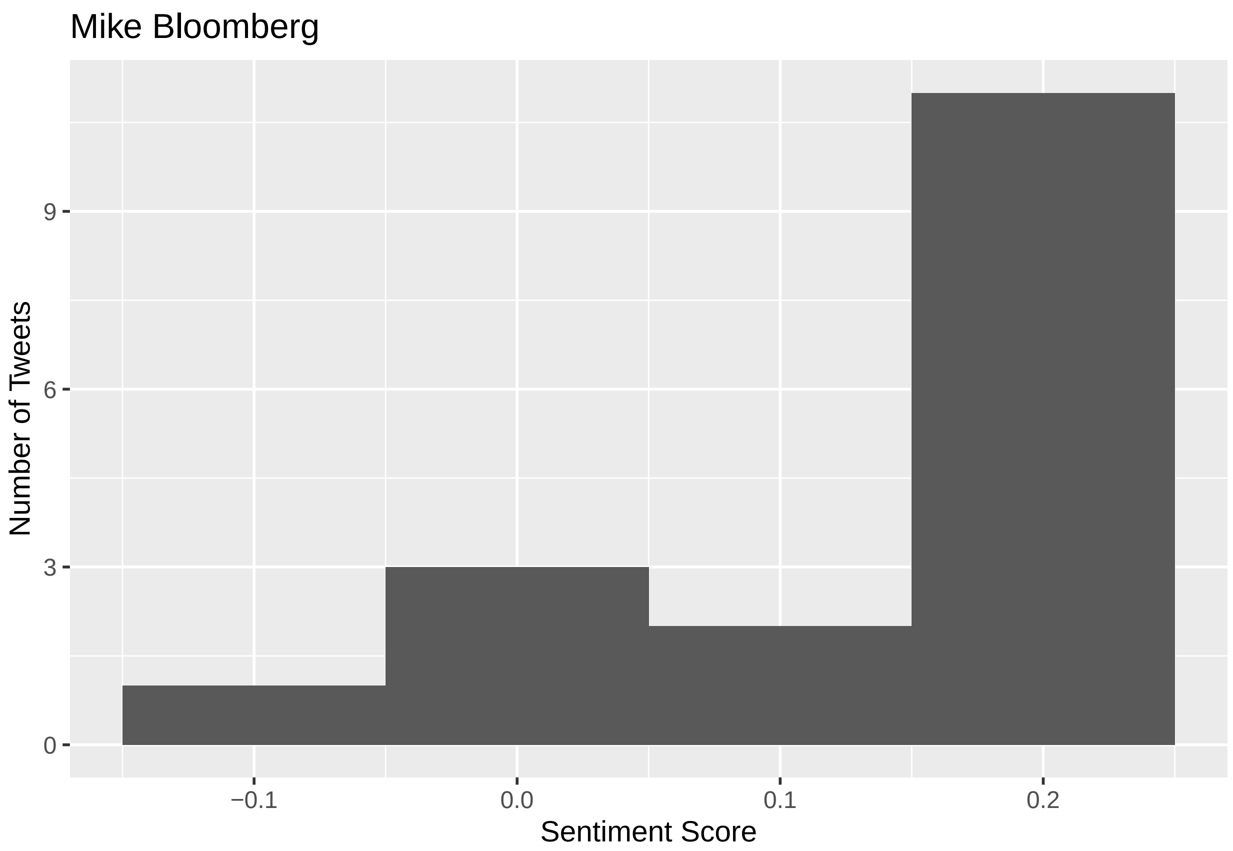
```
sentiment_mean_list <- append(sentiment_mean_list, mean(sentiment$sentiment))

sentiment <- sentiment(get_sentences(df$text[df$corybooker]))
sentiment_analysis(sentiment$sentiment, "Cory Booker")
```



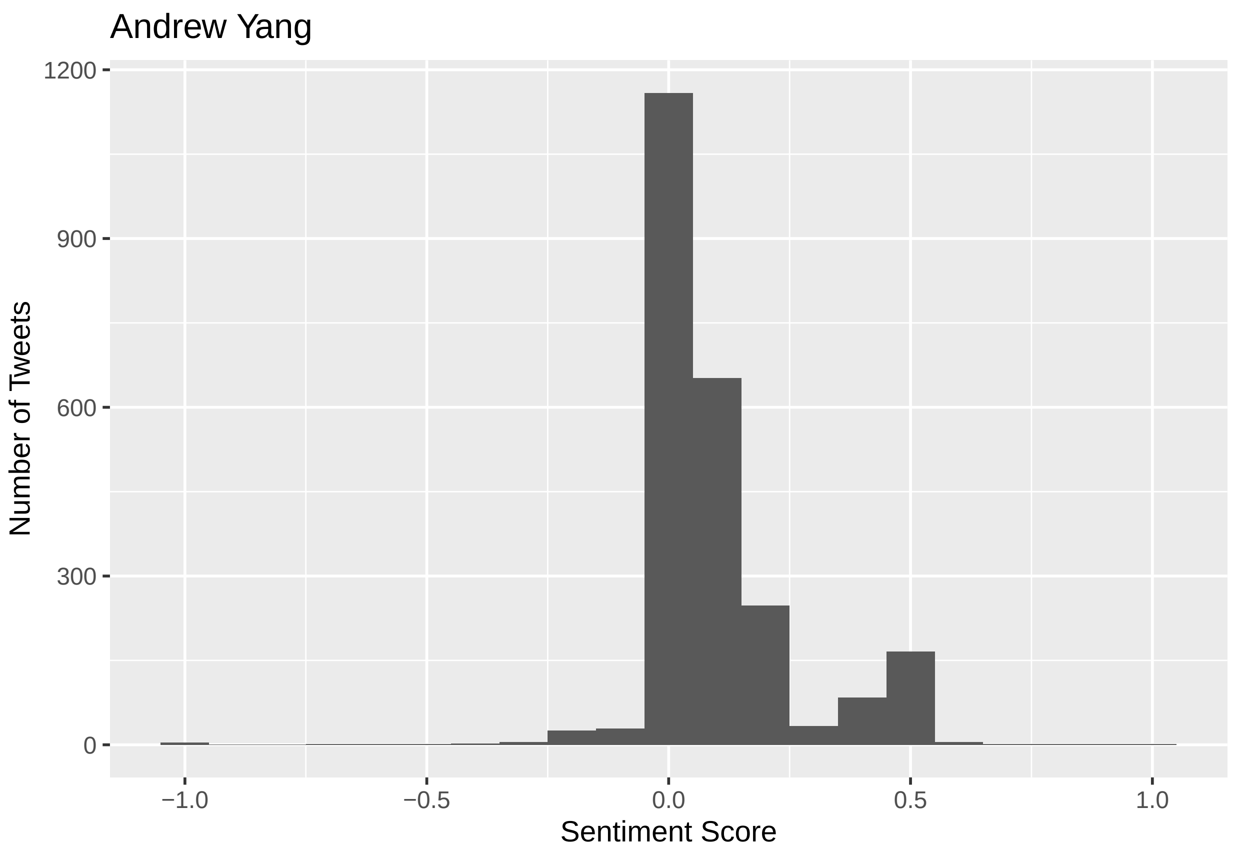
```
sentiment_mean_list <- append(sentiment_mean_list, mean(sentiment$sentiment))

sentiment <- sentiment(get_sentences(df$text[df$amyklobuchar]))
sentiment_analysis(sentiment$sentiment, "Amy Klobuchar")
```



```
sentiment_mean_list <- append(sentiment_mean_list, mean(sentiment$sentiment))

sentiment <- sentiment(get_sentences(df$text[df$mikebloomberg]))
sentiment_analysis(sentiment$sentiment, "Mike Bloomberg")
```



```
sentiment_mean_list <- append(sentiment_mean_list, mean(sentiment$sentiment))

sentiment <- sentiment(get_sentences(df$text[df$andrewyang]))
sentiment_analysis(sentiment$sentiment, "Andrew Yang")
```



```
sentiment_mean_list <- append(sentiment_mean_list, mean(sentiment$sentiment))
```

#Create Temp List and get Sentiment Mean

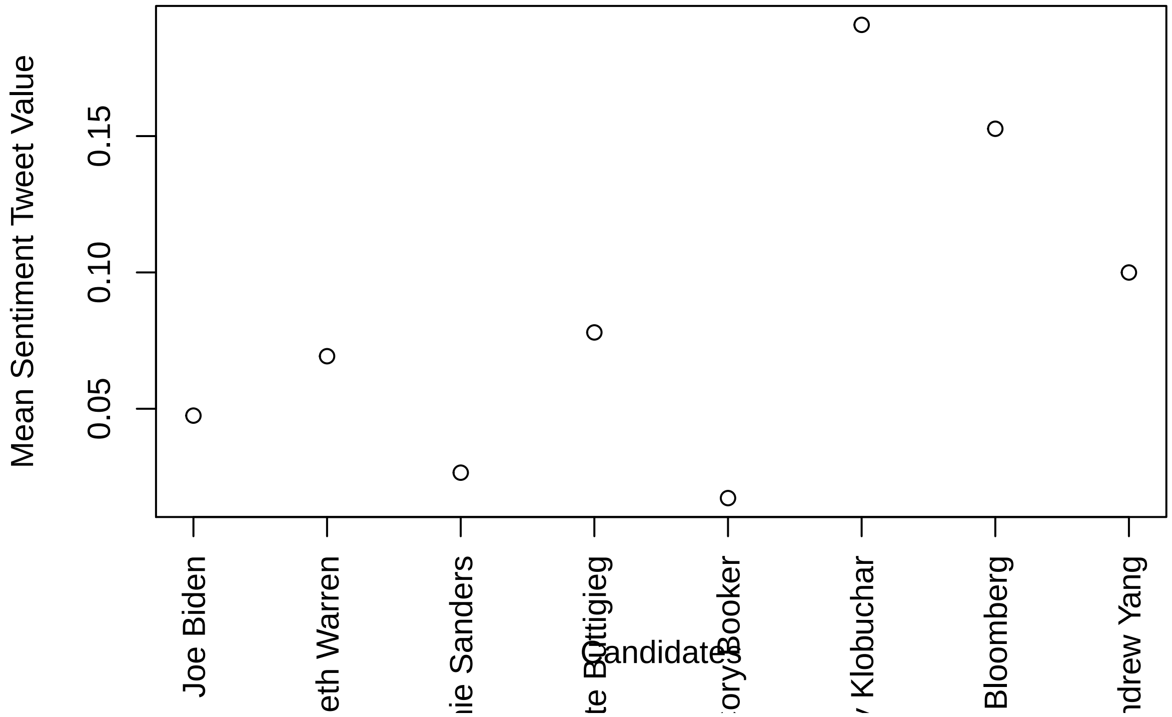
```
tmp_mean = unlist(sentiment_mean_list, F)

sentiment_mean <-mean(tmp_mean)
```

#Create two entries for each candidate with name and sentiment mean

```
candidates_list = list(name = "Joe Biden", "Elizabeth Warren", "Bernie Sanders", "Pete Buttigieg", "Cory
```

##Add candidate entries to graphable list

```
plot(tmp_mean, axes=FALSE, xlab="Candidates", ylab = "Mean Sentiment Tweet Value")
axis(2)
axis(1, at=seq_along(tmp_mean),labels=as.character(candidates_list), las=2)
box()
```

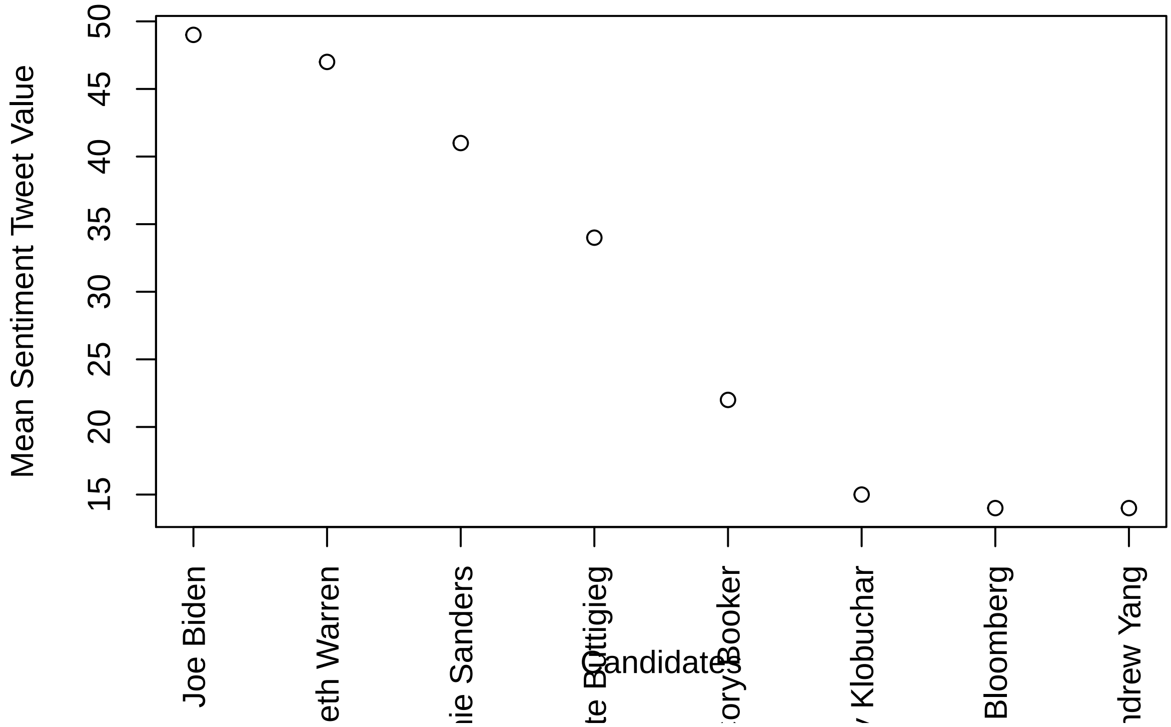


##Pull econTabReport.csv data created from the econTabReport.pdf starting on page 164 of the file.

```
PollingData <- read.csv(file="econTabReport.csv", header=TRUE, sep=",")
```

##Create graph on polling data overall values

```
plot(PollingData$Total, axes=FALSE, xlab="Candidates", ylab = "Mean Sentiment Tweet Value")
axis(2)
axis(1, at=seq_along(PollingData$Total),labels=as.character(candidates_list), las=2)
box()
```



##T-Test

We want to use R to assess whether Twitter is a good subset representation of public sentiment compared to how people are polled afterwards. In other words, can we see by Twitter's activities what we would see by otherwise going through all of the steps of polling?

```
t.test(PollingData$Total,tmp_mean)
```

```
##
##  Welch Two Sample t-test
##
## data:  PollingData$Total and tmp_mean
## t = 5.5272, df = 7.0002, p-value = 0.0008807
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  16.83071 41.99880
## sample estimates:
##   mean of x   mean of y
## 29.50000000  0.08524556
```

##T-Test results The T-Test results show that the p-value is 0.008809 which means that there is strong evidence against the hypthosis that the values would be related. That means we can reject the hypothesis and claim there is no real strong correlation.

##Conclusion

H_0: The data that was used in the test was nowhere near even close. You can see by simply eyeballing the results that the tweets' sentiment score had really no relation to how well the candidates polled.