Sentiment Analysis: A Demonstration of Python Text Analytics

Stephen Schneider, Caitlin McOsker, Aaliyah Smith, Caleb Vandersall

Information Technology Department
Georgia Southern University

Table of Contents

Introduction

Abstract

Our goal was to compare traditional political polls to the opinions of social media users using Python and sentiment analysis. We did this with two different data mining methods, a “static” method that pulled data for a search term over a set time period and a “dynamic” method that pulled a set number of tweets for a certain search term. We then used Python to perform sentiment analysis on the data. finally, we analyzed that data using a t-test, as well as through some exploratory analysis of the data.

Feasibility Study and Schedule

To better understand if the goals of our project were reasonable, we conducted a feasibility study. We looked at three perspectives: economic, operational, and technical feasibility.

The tools utilized for this project were Python, Tableau, and Microsoft Excel. We used open-source libraries and used the school’s licenses for any other software. Overall, there were no direct costs, making the project economically feasible. In terms of operational feasibility, all of the group members are IT majors with specializations in information management or data science. We are all fluent in Python and have experience with the above-mentioned applications, and this made us confident that we could complete the technical requirement for the project. Finally, we assessed whether we had access to all of the necessary tools listed above. Once we were confident that we had acquired all of these, we determined that our project was feasible for the time frame and goals.

After the feasibility study, we created a schedule to gauge our progress over the course of the semester. **Figures 1-1** and **1-2** detail the initial plan and the group members assigned to each task. In the end, we did not stick strictly to this schedule because we ran into challenges along the way. The challenges and problems we had can be found in our Challenges section on page 19.

Static vs. Dynamic Methodology

The first program we wrote in Python (**Figure 2-1**) was designed to pull data from Twitter and compile it into a single dataset that could be analyzed. From there, the data had to be cleaned, formatted, and combined in Excel. This approach was more hands-on and allowed us a more up-close view of our data.

The second program we wrote (**Figure 2-2**) used a more dynamic approach, extracting, cleaning, and visualizing data in a single script. This approach allowed us to quickly gauge online opinion on any topic without having to externalize the data cleansing and visualization process in Excel.

Static Method

Introduction

Our static method for pulling data was how we initially mined a data set of Tweets during a live democratic primary debate. We performed sentiment analysis on this data set to explore candidate. For this initial data set, we pulled tweets for four hours during the October 15 Democratic Debate, as well as several weeks later for four hours on November 3. We then compared these to national polls from those time periods.

Methodology

Pulling data

To pull the data during the debate, we used the following Python libraries:

- Tweetpy: to access the Twitter API
- JSON: to unpack the JSON data structure
- Pandas: for filtering data, creating dataframe and later, CSV

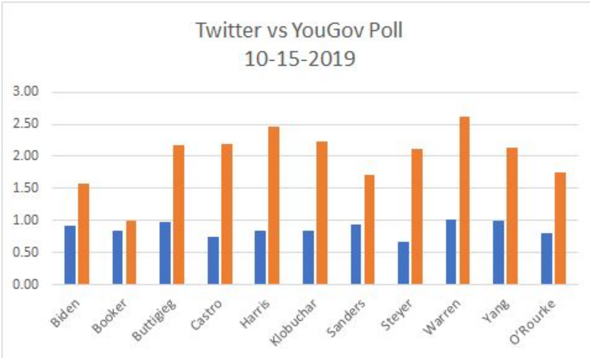
The code we wrote can be found in **Figures 2-1** and **2-2.**

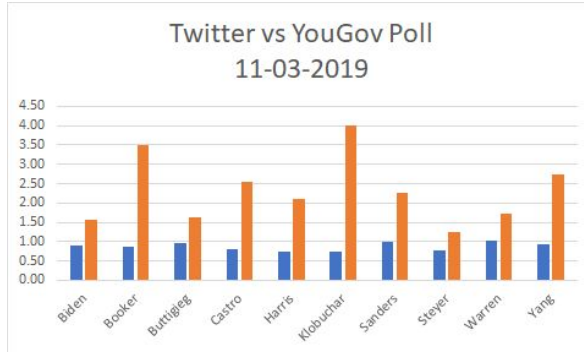
Analyze data

First, we assigned candidates to tweets based on the 'mentions' column. Then, we used these values to group our data into 4 groups in Excel (**Figures 3-1** and **3-2**).

- Very Favorable: 0.5 <= Polarity <= 1.0
- Somewhat Favorable: Polarity < 0.5
- Neutral: Polarity = 0
- Somewhat Unfavorable: 0 > Polarity > -0.5
- Very Unfavorable: -0.5 >= Polarity >= -1.0

We could not map the 'Neutral' classification to 'Don't Know' category of the poll, so we investigated the positive/negative ratio for the data. Finally, we compared these results to the Poll Data (see Polls). Below is a summarization of this comparison.





The blue bars represent the YouGov poll and the orange bars represent our twitter data. A value of 1.0 indicates a closer split between positive and negative. The traditional poll in both data sets has ratios closer to 1.0, while our Twitter data returns ratios well above 1.0; this indicates that we received a lot more positive than negative hits from our sentiment model.

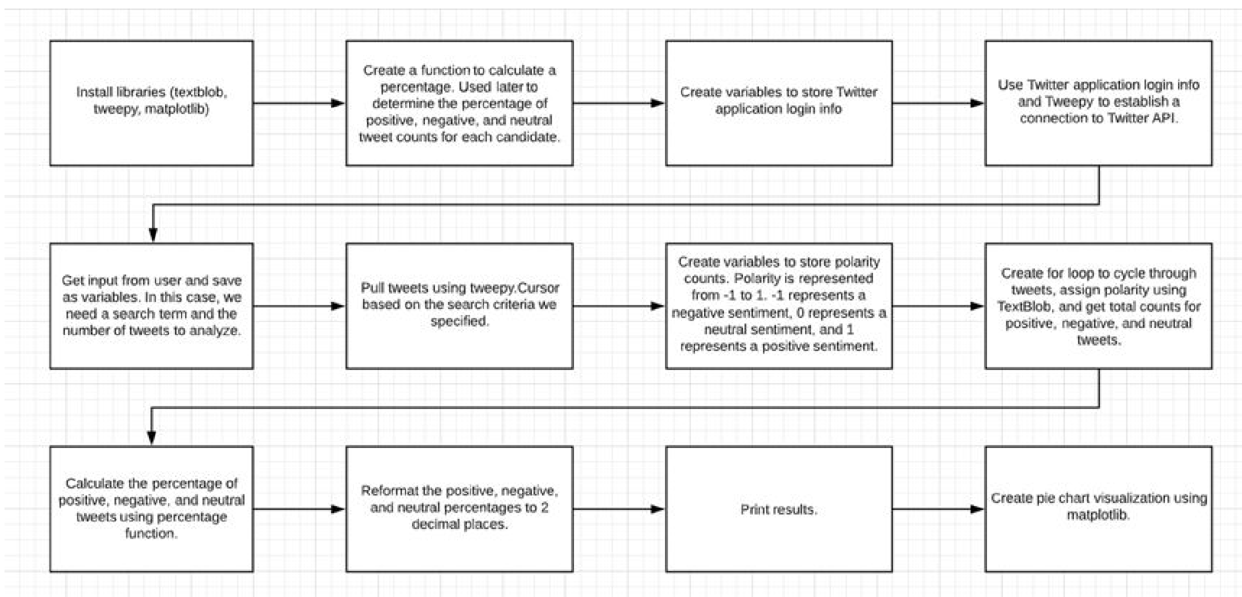
Dynamic Method

Introduction

Our dynamic method allowed us to create an adaptive Python tool to quickly gauge online opinion on any topic. We demonstrated the utility of Python and its libraries by performing sentiment analysis of Twitter data regarding opinions on 2020 Democratic primary candidates and evaluated the tool by comparing results with traditional polling methods.

How the code works

Below is a graphic showing how the dynamic code works. See **Figure 2-3** for the entire script.



Sample Size

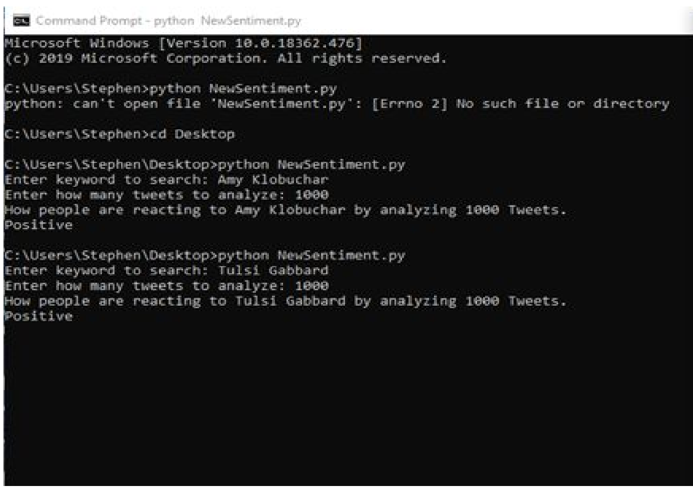
We decided to use 1,000 tweets as our sample size for each candidate. Assuming our population is approximately 372,000,000 (the population of the United States), 1,000 tweets per candidate will give us a 3.10% margin of error with a 95% confidence level for each sentiment analysis.

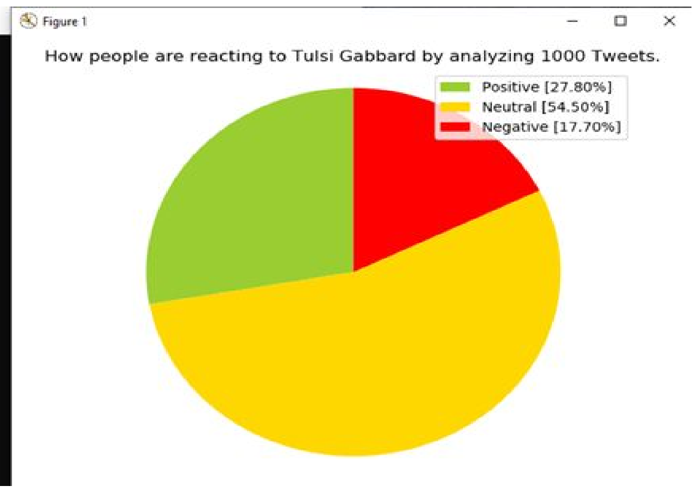
Uncertain Factors

We believe our tool is a good predictor of online public opinion, however, there are still some factors that our code cannot account for. The authenticity of Tweets is a legitimate concern with the rising issue of Twitter bots and online trolling. Unfortunately, this just happens to be the nature of social media data.

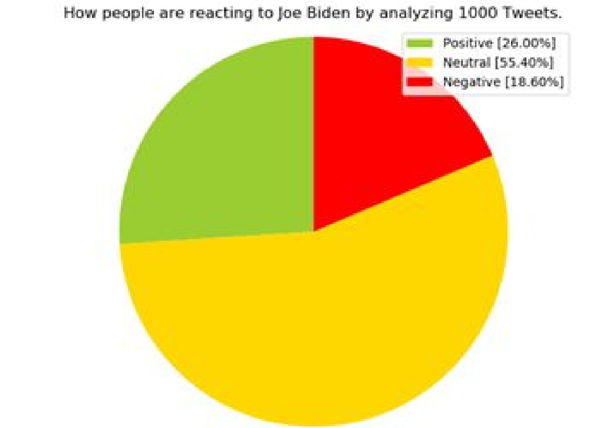
Running the Code

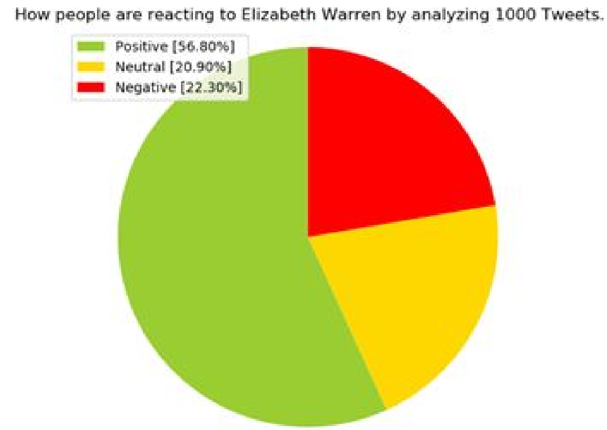
Our code can be run in the command prompt. Simply run the file, enter the search criteria, and press enter.

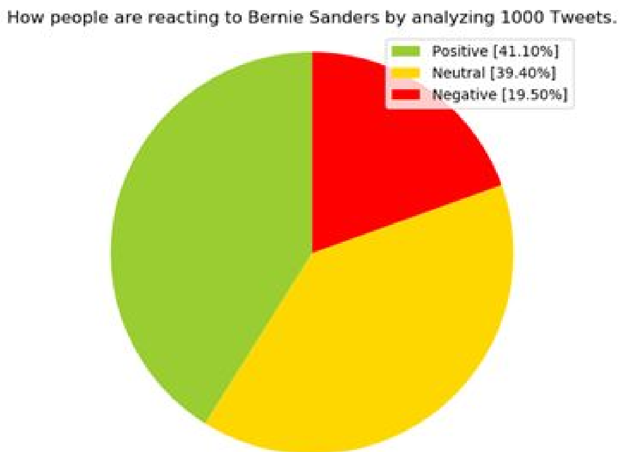


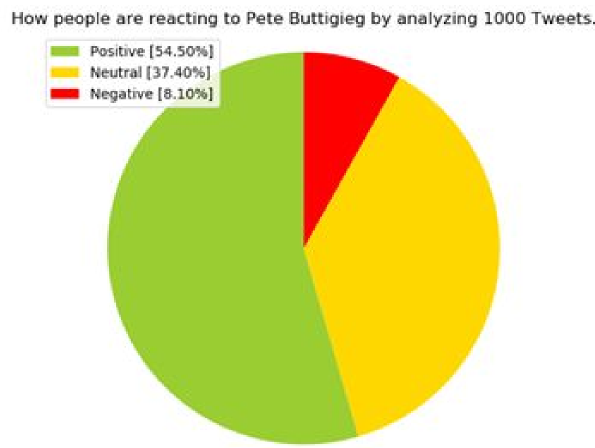


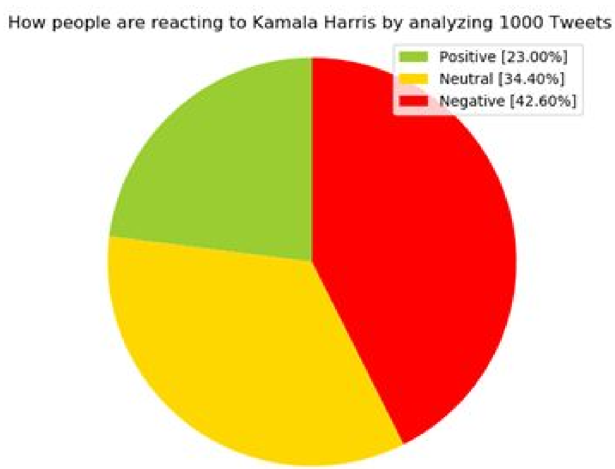
Raw Output

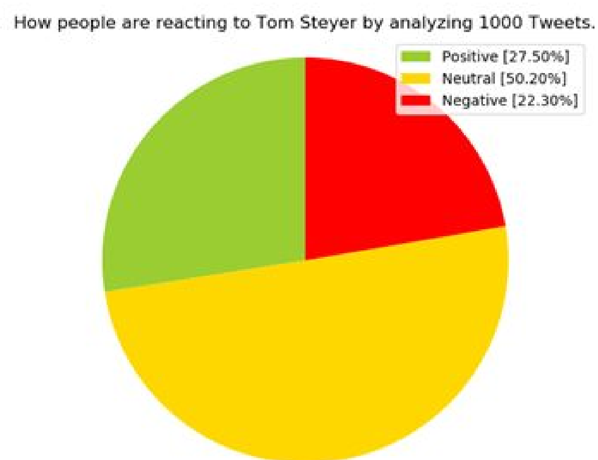

How people are reacting to Joe Biden by analyzing 1000 Tweets.
Positive [26.00%]
Neutral [55.40%]
Negative [18.60%]


How people are reacting to Elizabeth Warren by analyzing 1000 Tweets.
Positive [56.80%]
Neutral [20.90%]
Negative [22.30%]

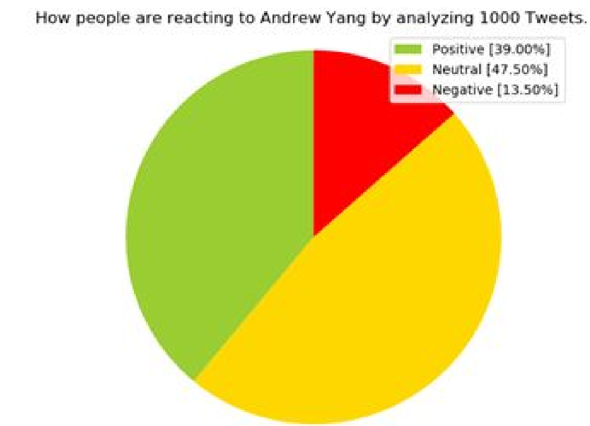

How people are reacting to Bernie Sanders by analyzing 1000 Tweets.
Positive [41.10%]
Neutral [39.40%]
Negative [19.50%]

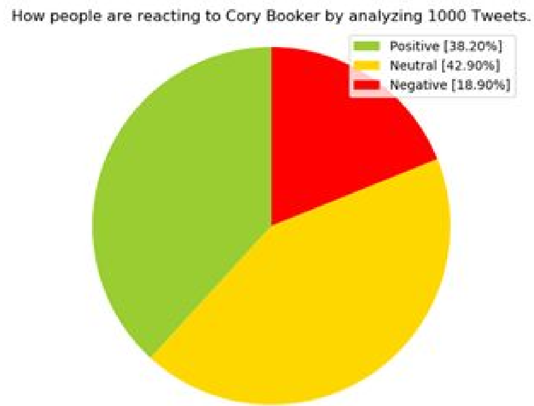

How people are reacting to Pete Buttigieg by analyzing 1000 Tweets.
Positive [54.50%]
Neutral [37.40%]
Negative [8.10%]

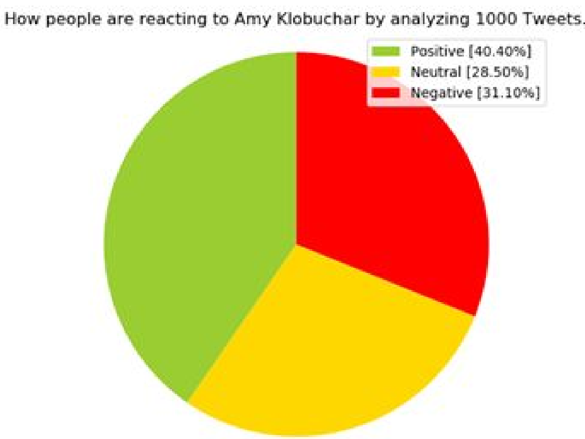

How people are reacting to Kamala Harris by analyzing 1000 Tweets
Positive [23.00%]
Neutral [34.40%]
Negative [42.60%]

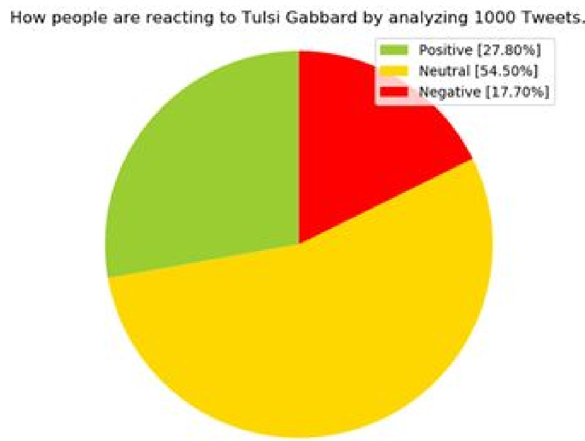

How people are reacting to Tom Steyer by analyzing 1000 Tweets.
Positive [27.50%]
Neutral [50.20%]
Negative [22.30%]

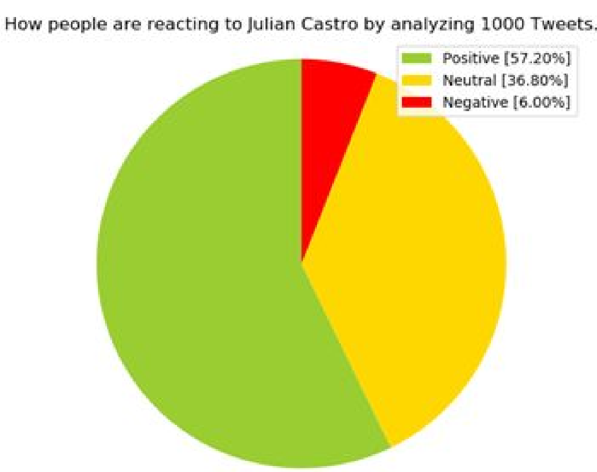
Raw Output (part 2)

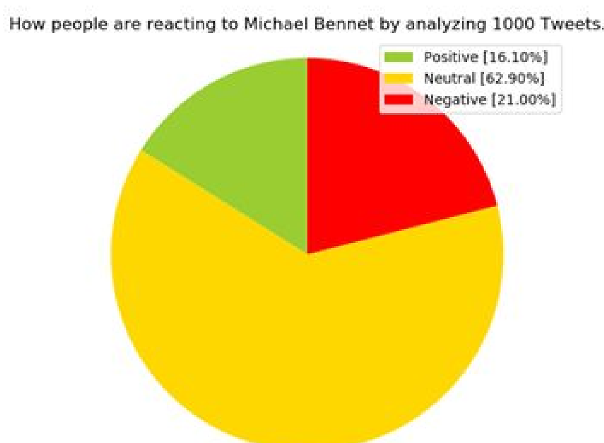








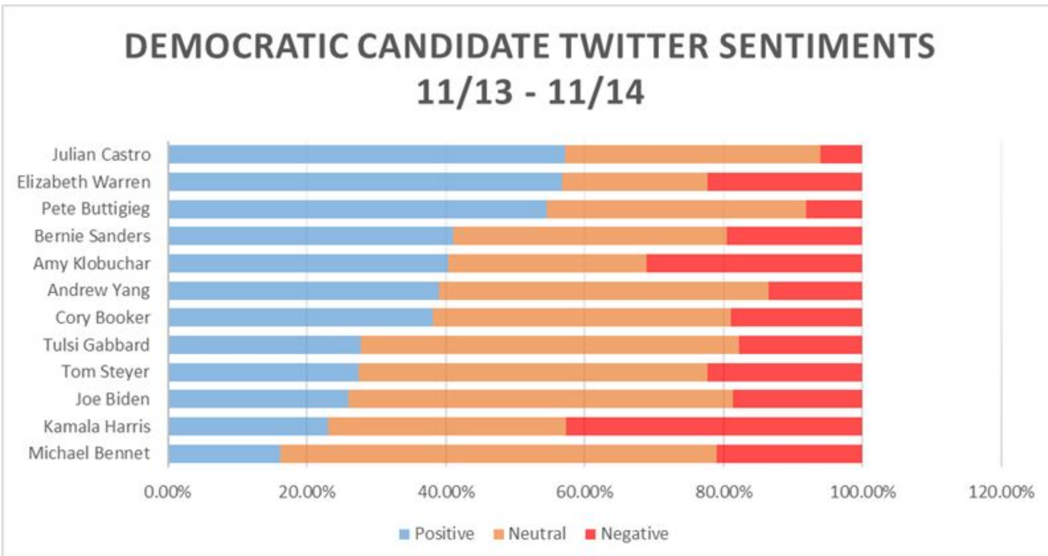


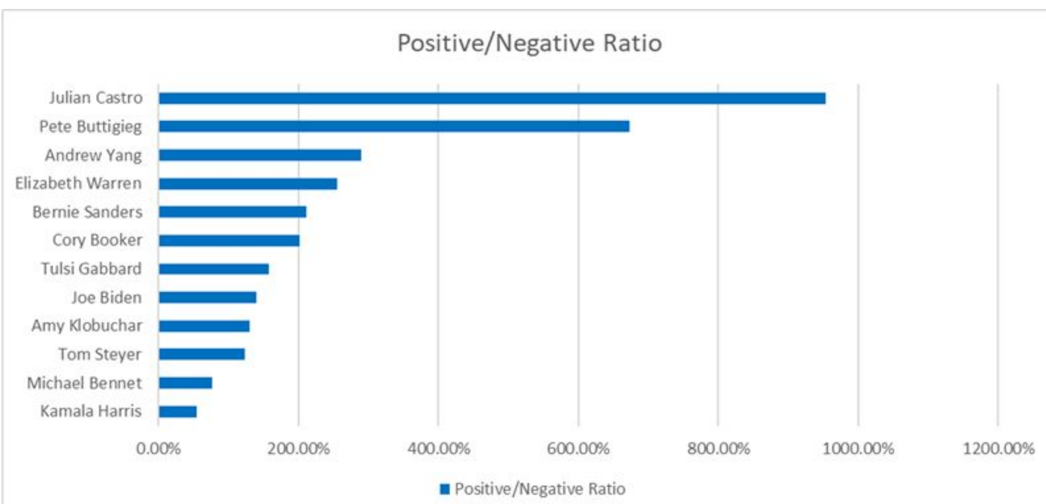


Dynamic Findings (summarized)

- Most Positive Candidate: Julian Castro
- Most Neutral Candidate: Michael Bennet
- Most Negative Candidate: Kamala Harris
- Best Positive/Negative Ratio: Julian Castro

Sentiments Visualized





The positive-negative ratio chart looks at which candidates have the most positive tweets and least negative tweets about them. Surprisingly, Julian Castro, Pete Buttigieg, and Andrew Yang were the top performers by this metric.

Comparison With National Poll

RealClearPolitics National Democratic Poll (11/13)

2020 Democratic Presidential Nomination

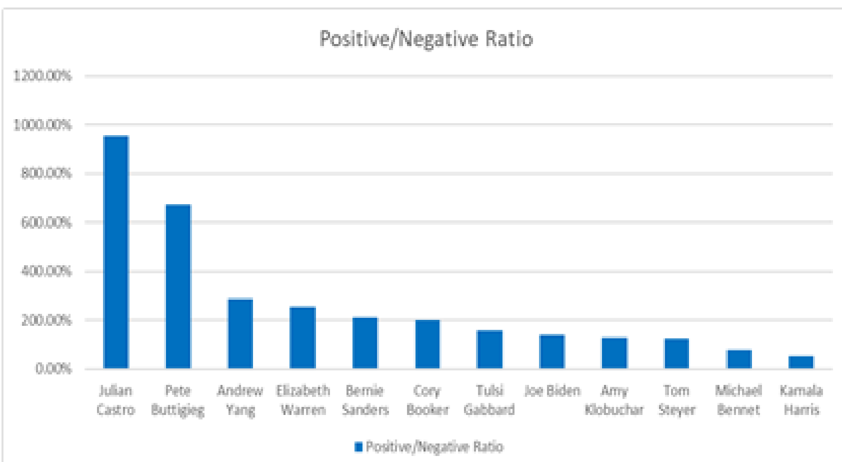
National | Iowa | New Hampshire | Nevada | South Carolina | California | Texas | Massachusetts | General Election Polls

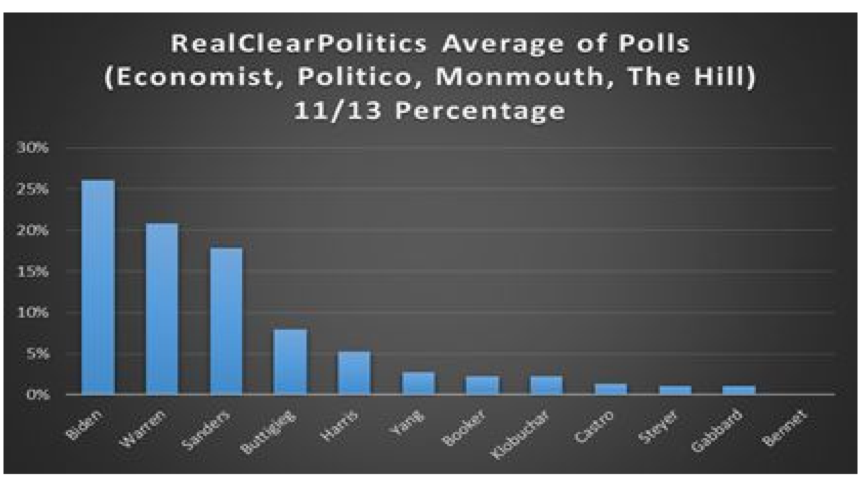
Polling Data

Poll	Date	Biden	Warren	Sanders	Buttigieg	Harris	Yang	Klobuchar	Booker	Castro	Gabbard	Steyer	Bullock	Delaney	Spread
RCP Average	10/30 - 11/12	26.0	20.8	17.8	8.0	5.3	2.8	2.3	2.3	1.3	1.0	1.0	0.8	0.8	Biden +5.2
Economist/YouGov	11/10 - 11/12	23	26	17	9	5	4	2	1	2	2	1	1	1	Warren +3
Politico/Morning Consult	11/4 - 11/10	32	19	20	8	5	3	2	3	1	1	1	1	1	Biden +12
Monmouth	10/30 - 11/3	23	23	20	9	5	3	2	3	0	0	1	0	0	Tie
The Hill/HarrisX	11/1 - 11/2	26	15	14	6	6	1	3	2	2	1	1	1	1	Biden +11

All 2020 Democratic Presidential Nomination Polling Data

RCP Poll vs. Twitter Data





Understanding the Data

Sentiment Analysis

TextBlob Polarity

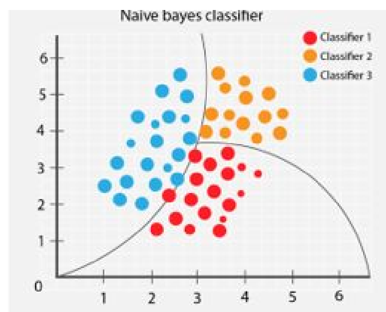
TextBlob is a library for processing textual data. The TextBlob NaiveBayesAnalyzer uses a Naive Bayes classification algorithm based on a pre-programmed training data set based on the English language in order to determine the polarity of a given sentence.

From thatware.co:
“In machine learning, naive Bayes classifiers are a family of simple ‘probabilistic classifiers’ based on applying Bayes’ theorem with strong (naive) independence assumptions between features

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

using Bayesian probability terminology, the above equation can be written as

$$\text{Posterior} = \frac{\text{prior x likelihood}}{\text{evidence}}$$



How It Works
To classify our data as positive, negative, or neutral, we used TextBlob’s sentiment analysis tool for both our static and dynamic methods. We were interested in the polarity and the subjectivity of our Tweets. The polarity has a range of [-1.0,1.0] and indicates how negative or positive the text’s sentiment is. The subjectivity has a range of [0.0,1.0] and indicates how objective or subjective the text is. For our static analysis, we split the range of subjectivity into quarter ranges, i.e. [0,0.25], [0.26,0.50], etc., and assigned each range a particular sentiment. [0,0.25] was Strongly Rational with Rational, Emotional, Strongly Emotional being the others in order.

In our data, this information can be found in the columns labeled 'Sentiment' and 'Polarity.' See **Figure 3-1** and **Figure 3-2.**

Reason For Use

Twitter data is largely unstructured and uncategorized data, so in order to categorize the pulled data and discover insights, sentiment analysis was the most applicable way for us to do it. It's scalable to different sizes of data sources and if we wanted to, we would be able to break down the tweets to different levels of analysis. The most important reason that it was used is that it provides a consistent base to determine a tweets sentiment. Every individual interprets meaning behind words differently, so having clear and concise rules of what constitutes positive and negative emotion is important for accurate analysis.

Traditional Polls - Economist/YouGov

Reliability

YouGov, which we used as our traditional polls is an "international research data and analytics group" and has "one of the world's largest research networks" [1]. They work to provide a constant stream of relevant data as a "public resource." The Pew Research Centersaid that YouGov "consistently outperforms competitors on accuracy," and they are the "market research pioneer of Multilevel Regression with Post-stratification (MRP)" [1]. Because of these qualifications and the consistency of poll output (every other week), we decided that this was a good example of a traditional poll to use in our project. We have included links to all three polls in the References section [2] [3] [4].

Analysis

Analysis Explanation

We wanted to determine whether the comparison of our data to traditional polls was significant. To do this, we conducted a T-Test on the consolidated percentages of the top three candidates for all three polls (Biden, Warren, and Sanders).

After looking at the data, we noticed that there were some interesting patterns within it. Therefore, we also did some exploratory data analysis of our datasets, which is elaborated on below.

T-Statistic Analysis

Method

First, we took the consolidated percentage data for our polls (**Figures 4-1**, **4-2**, and **4-3**) and our datasets (**Figure 3-3**, **3-4**, and **3-5**) and compiled it into the following table:

	Favorable Data	Sample Size (n1)	Frequency	Relative Frequency (p1)	Sample Size (n2)	Frequency	Relative Frequency (p2)
15-Oct	Biden	1486	609	41%	456	116	25%
	Warren	1484	608	41%	380	89	23%
	Sanders	1489	610	41%	447	110	25%
3-Nov	Biden	1485	594	40%	303	86	28%
	Warren	1487	595	40%	301	82	27%
	Sanders	1490	641	43%	266	79	30%
14-Nov	Biden	1483	578	39%	1000	260	26%
	Warren	1481	563	38%	1000	568	57%
	Sanders	1484	608	41%	1000	410	41%

All of our calculations for this analysis were done in Excel. We used the following formula to perform a t-test on the consolidated data:

$$p = \frac{(p_1 * n_1) + (p_2 * n_2)}{n + n_2}$$

We then used this number to calculate the Standard Error:

$$SE = \sqrt{p * (1 - p) * (\frac{1}{n_1} + \frac{1}{n_2})}$$

Once we had the Standard error, we were able to compute the z-score:

$$z - score = \frac{p_1 - p_2}{SE}$$

From there, we were able to compute the p-value. We did this using Excel's NORM.S.DIST function. This function returns the probability from the standard normal distribution. We subtracted this value from 1.

The following table has all of these calculations for the top three candidates of each poll.

		t-test			
	Favorable Data	p	SE	z-stat	p-value

15-Oct	Biden	0.37346035	0.0258958	6.009238588	9.32E-10
	Warren	0.37416309	0.0278212	6.318552559	1.32E-10
	Sanders	0.372153926	0.0260699	6.287517525	1.61E-10
3-Nov	Biden	0.380313199	0.0306024	3.796155525	7.35E-05
	Warren	0.37852349	0.0306552	4.161605671	1.58E-05
	Sanders	0.409851936	0.0327357	4.063066213	2.42E-05
14-Nov	Biden	0.337643979	0.0193505	6.718159631	9.20E-12
	Warren	0.455775897	0.0203845	-9.22268622	0
	Sanders	0.41	0.0201223	2.76E-15	0.5

Results

For our project, we declared P statistically significant when P < 0.05. Our null hypothesis and alternate hypothesis were:

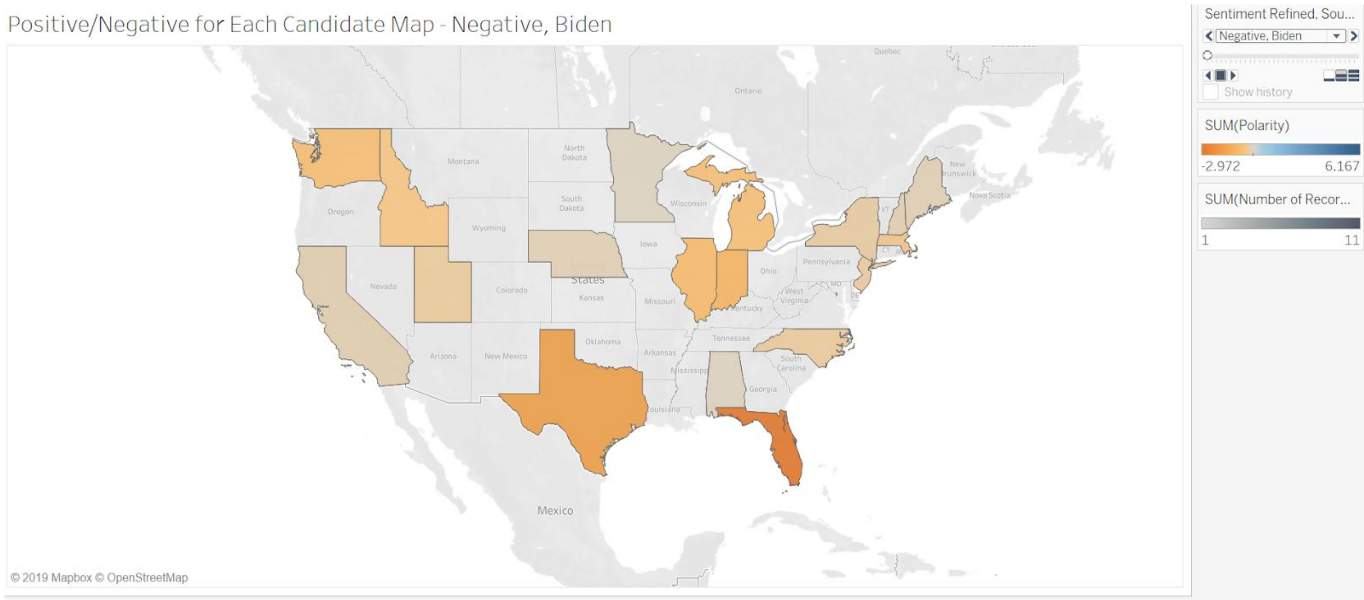
$$H_0 : \mu_1 \neq \mu_2$$
$$H_a : \mu_1 = \mu_2$$

The only instance where the null hypothesis was rejected was with Sanders on November 14th, and it was the only instance that had the same percentage for the poll and our twitter data.

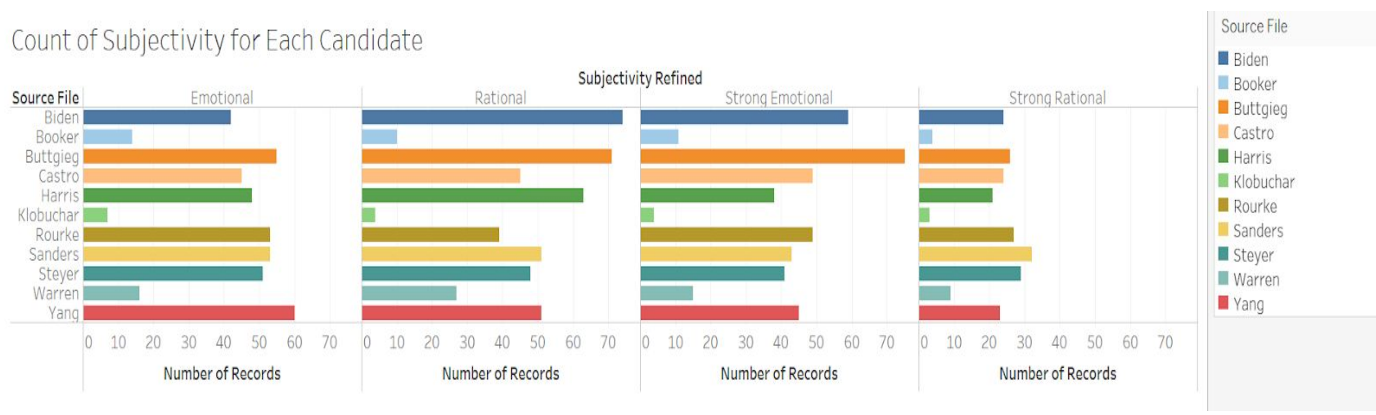
Exploratory Data Analysis

Static Method

We created visualizations using Tableau to try and help identify any particularly useful and interesting insights about our Twitter data. Initially, we wanted to use location data from each tweet and map polarity across the US for each candidate. We were hoping this would highlight states that showed a particularly strong like or dislike of a candidate. While it did show certain states had more of an affinity with certain candidates, the data pool was not large enough to truly be a representative sample. Also, since users can enter practically whatever they want as their location on twitter, the accuracy of the data itself is questionable. An example of a location map is included below.



Next, we decided to look at the sentiment of each candidate to see if there were any candidates whose tweets leaned more towards a more rational tweet or an emotional tweet. While some had a slight lean towards a particular sentiment, the majority of them were well rounded and had quite a few from both sides. Biden stood out by having the largest tweet pool for us to look at and while it was indeed still well rounded, there was a slight favor towards more emotional tweets both negatively and positively.



Dynamic Method

Our dynamic code allowed us to create an exploratory data analysis tool. From the command prompt, we can simply run our Python script, enter a keyword and number of Tweets to analyze, and get a fast sentiment analysis visualization in the form of a MatPlotLib pie chart. This useful visualization highlights the positive, negative, and neutral sentiments being expressed about the given topic on Twitter. We believe our code is a great starter tool for performing exploratory data analysis on social media data.

While not a perfect comparison, our analysis does show a stark difference between online opinion about candidates and their performance in standard national polls.

Julian Castro, Pete Buttigieg, and Andrew Yang have the best positive/negative ratio of tweets during the period of our analysis. We believe this may be a result of increased social media outreach by these small-time candidates in order to make up for their lack of support in traditional polls.

Our data also shows a potential disparity between the demographics of traditional polling and that of Twitter users. This begs the question if traditional polling methodologies may be ignoring certain demographics.

There are many factors that could be in play here, including campaign social media outreach, the news cycle, and other complex political and behavioral influences. We believe further analysis is warranted to investigate the connection between social media sentiments and those of traditional polls.

Conclusion

In conclusion, despite encountering several obstacles, we were able to complete our goal of comparing social media data with more traditional data and demonstrating the utility of Python and its libraries for exploratory data analysis. Our static and dynamic approaches of data analysis made good case studies for the different kinds of analytics projects that are often seen in the world of business and academia.

Challenges

Twitter Data

We ran into several problems due to the nature of social media data. Traditional polls are able to organize their data based on the demographics of their sample population. They are also able to target only US citizens that can vote. With social media data, specifically Tweets, there is no reliable way to determine location or age. It is also difficult to eliminate sarcasm, trolling, and illegitimate accounts from our analysis. In order to differentiate between genuine and other forms of tweets we would require a much more sophisticated model.

Static Sentiment Analysis

The sentiment analysis model used was not as accurate as we needed. Polarity was often inaccurate for records that were sarcastic or included special characters. We should have trained our own model to account for these issues.

Data Cleaning

When eliminating duplicates, we deleted tweets that mentioned multiple candidates after the first occurrence. This skewed the data for later candidates. We solved this problem by allowing tweets to count towards more than one candidate, but moving forward it is important to be aware of this.

Moving Forward

For future additions to the project, groups could possibly apply for a grant to use the paid version of the Twitter API. This would increase the amount of tweets they could pull at one time, and it would result in a better data set. Groups could also spread into different social media platforms that are easy to mine data from, like reddit or Facebook.

Something interesting future groups could look at is the relationship between records' subjectivity and polarity. this was an area we did not get to explore as much, and we think something interesting could be found there.

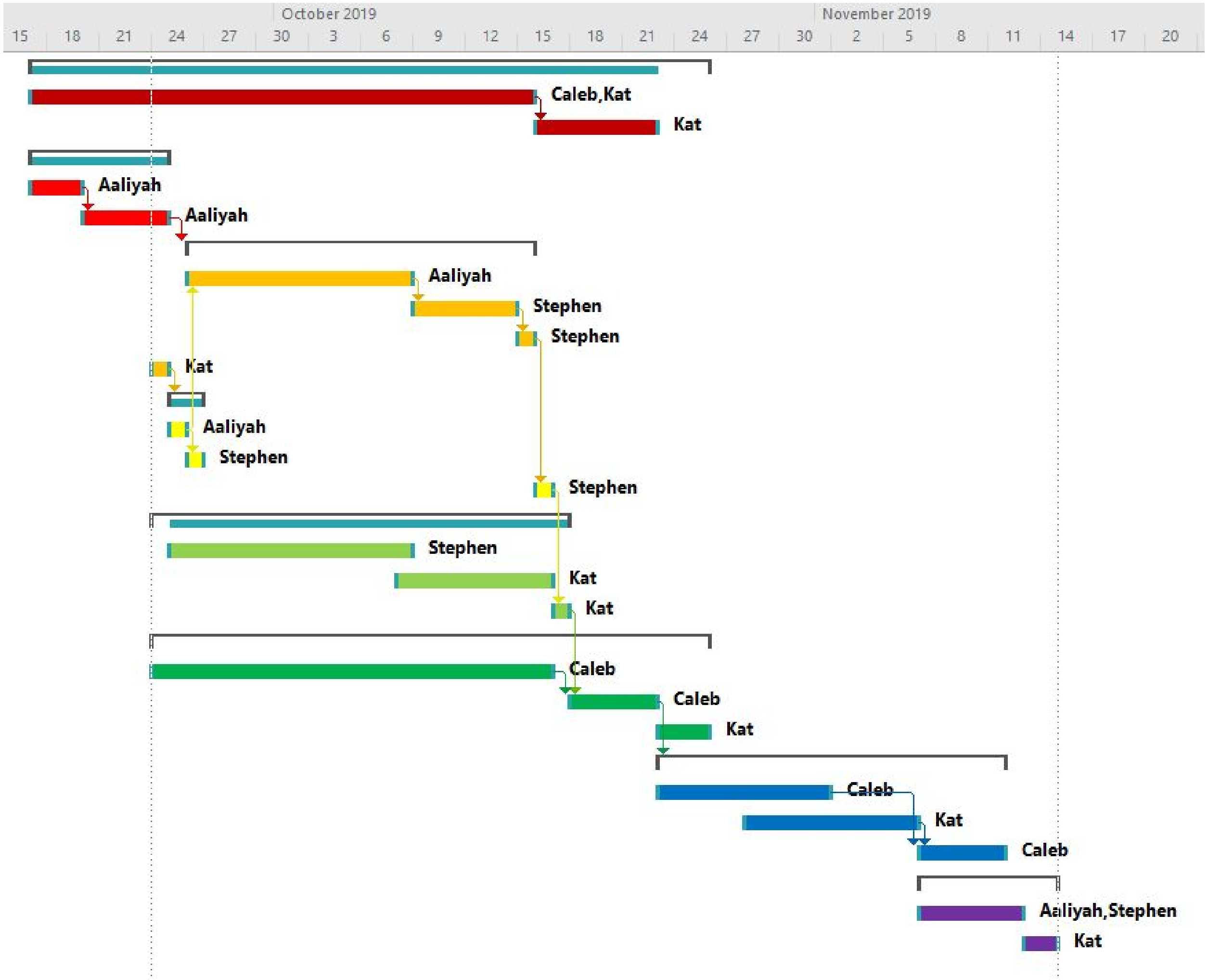
Appendices

Gantt Chart

Figure 1-1: Breakdown of Tasks

	Task Name
1	**Understand traditional polling**
2	Research traditional methods
3	Choose poll(s) that best fits our use
4	**Acquire test data**
5	Edit base code
6	Acquire test data
7	**Acquire Debate data**
8	Edit script to model/preprocessing requirements
9	Test final script
10	Acquire Debate data
11	Determine necessary preprocessing
12	**Preprocess test data**
13	Write code to test
14	Test with test data
15	Preprocess Debate data
16	**Build and implement model**
17	Write code to build model
18	Test model with test data and make necessary changes
19	Apply model to Debate Data
20	**Analyse Debate Data**
21	Research Twitter demographics/background
22	Examine Debate data for interesting trends
23	Compare to traditional poll(s)
24	**Visualization**
25	Use Tableau to create interesting visualizations of Twitter data
26	Create/implement visualizations of traditonal poll methods/data
27	Combine visulaizations to compare twitter vs. traditional
28	**Create poster to present**
29	Use visualizations, methodology, and background information to support fin
30	Print and submit poster

Figure 1-2: Task Assignment



Static Code

Figure 2-1 Import Data

```
#Required modules
import tweepy
#this module is used for accessing the twitter api
import json
#with this module we are able to unpack this data structure
import pandas as pd
#pandas will be used for furthor filtering the data and later creating a csv
from pandas import DataFrame
auth =
tweepy.OAuthHandler('nT8KP0LQX0p7sxqgUawojrpxd','lZFejHEMD5mA7Lwjxc47DuHe6gUekG
4wddo270UIRbzigUj6fq')
auth.set_access_token('1046519830461984768-JiGfq4e0VNGgewONGM8A5ixBcFWUT3',
'ITvIwjGNdJY28gmJRlJM9ZdKmiVbRMILgmLAGgKnLBSYS')
api = tweepy.API(auth)



#--------------------- get tweet objects in json format
------------------------------------
json_list = []
user_list = []
entities_list = []
hashtag_list = []
mention_list = []
# ------------------------------------- SEARCH STRING GOES HERE
---------------------------------------
# Search strings make sure that the query is run in the correct time frame
(since/until) while also filtering out retweets
#This search string list also make sure that 100 tweets are pulled from each
candidate based on the filter

querylist = ['"@JoeBiden"AND -filter:retweets AND since:2019-11-03 AND
until:2019-11-05','"@BernieSanders" AND -filter:retweets AND since:2019-11-03
AND until:2019-11-05','"@CoreyBooker" AND -filter:retweets AND since:2019-11-03
AND until:2019-11-05','"@PeteButtigieg" AND -filter:retweets AND
since:2019-11-03 AND until:2019-11-05','"@JulianCastro"AND -filter:retweets AND
since:2019-11-03 AND until:2019-11-05','"@KamalaHarris"AND -filter:retweets AND
since:2019-11-03 AND until:2019-11-05','"@SenAmyKlobuchar"AND -filter:retweets
AND since:2019-11-03 AND until:2019-11-05','"@BetoORourke"AND -filter:retweets
AND since:2019-11-03 AND until:2019-11-05','"@TomSteyer"AND -filter:retweets
AND since:2019-11-03 AND until:2019-11-05','"@ewarren"AND -filter:retweets AND
```

```
since:2019-11-03 AND until:2019-11-05','"@AndrewYang"AND -filter:retweets AND
since:2019-11-03 AND until:2019-11-05' ]

for query in querylist :
    #When extracting tweets from twitter there is a maximum limit of 100
    status_list = api.search(q=query, count=100)
    for status in status_list:

        json_str = json.dumps(status._json)
        json_data = json.loads(json_str)
        user = json_data['user']
        del user['entities']
        user_list.append(user)
        entities = json_data['entities']

        hashtag_text = []
        hashtags = entities['hashtags']
        for hashtag in hashtags:
            hashtag_text.append(hashtag['text'])
        hashtag_list.append(hashtag_text)

        screen_names = []
        user_mentions = entities['user_mentions']
        for mention in user_mentions:
            screen_names.append(mention['screen_name'])
        mention_list.append(screen_names)

        entities_list.append(entities)

        del json_data['user']
        del json_data['entities']
        json_list.append(json_data)

        json_df = pd.DataFrame(json_list)

        # Note: status list might be empty because we use the Standard Search
API which limits our search history to within the past 7 days
        # ---------------------- create a data frame of the list of tweets
----------------------------


        # ---------------------- extract required fields
----------------------------------------------
        json_df =
json_df[['id_str','text','retweet_count','favorite_count','created_at','coordin
```

```
ates', 'in_reply_to_screen_name']]

        user_df = pd.DataFrame(user_list)
        hashtag_series = pd.Series(hashtag_list)
        mention_series = pd.Series(mention_list)
        user_df =  user_df[['id_str', 'location', 'screen_name']]
        # entities_df = pd.DataFrame(entities_list)

        json_df['user_id_str'] = user_df['id_str']
        json_df['user_location'] = user_df['location']
        json_df['user_screen_name'] = user_df['screen_name']
        json_df['hashtags'] = hashtag_series
        json_df['mentions'] = mention_series
    #since some tweets mention more than one candidate there will be duplicates
that require removal
    json_2 = json_df.drop_duplicates(keep = False, subset = "id_str")


    # ---------------------- save as a csv file grouped by candidate while
appending new tweets each run  --------------------------------------------
#print(json_df)
#check = json_2['id_str'] == 1180839899731120000
#print(json_2[check])
json_2.to_csv('nov112019.csv',header = True,mode = 'a')
#KamalaHarris = json_2.mentions.str.contains('KamalaHarris',regex = False)
#print(KamalaHarris.head())
# KamalaHarrisDF= json_2[KamalaHarris]
# KamalaHarrisDF.to_csv('Harris.csv',header = True, mode = 'a')
# JoeBiden = json_2.mentions.str.contains('JoeBiden',regex = False)
# JoeBidenDF = json_2[JoeBiden]
# JoeBidenDF.to_csv('Biden.csv',header = True, mode = 'a')
# BernieSanders = json_2.mentions.str.contains('BernieSanders',regex = False)
# BernieSandersDF = json_2[BernieSanders]
# BernieSandersDF.to_csv('Sanders.csv',header = True, mode = 'a')
# CoreyBooker = json_2.mentions.str.contains('CoreyBooker',regex = False)
# CoreyBookerDF = json_2[CoreyBooker]
# CoreyBookerDF.to_csv('Booker.csv',header = True, mode = 'a')
# PeteButtigieg = json_2.mentions.str.contains('PeteButtigieg',regex = False)
# PeteButtigiegDF = json_2[PeteButtigieg]
# PeteButtigiegDF.to_csv('Buttigieg.csv',header = True,mode = 'a')
# JulianCastro = json_2.mentions.str.contains('JulianCastro',regex = False)
# JulianCastroDF= json_2[JulianCastro]
# JulianCastroDF.to_csv('Castro.csv',header = True, mode = 'a')
# AmyKlobuchar = json_2.mentions.str.contains('SenAmyKlobuchar',regex = False)
```

```
# AmyKlobucharDF = json_2[AmyKlobuchar]
# AmyKlobucharDF.to_csv('Klobuchar.csv',header = True, mode = 'a')
# BetoORourke = json_2.mentions.str.contains('BetoORourke',regex = False)
# BetoORourkeDF = json_2[BetoORourke]
# BetoORourkeDF.to_csv('Rourke.csv',header = True, mode = 'a')
# TomSteyer = json_2.mentions.str.contains('TomSteyer',regex = False)
# TomSteyerDF = json_2[TomSteyer]
# TomSteyerDF.to_csv('Steyer.csv',header = True, mode = 'a')
# ElizabethWarren = json_2.mentions.str.contains('ewarren',regex = False)
# ElizabethWarrenDF = json_2[ElizabethWarren]
# ElizabethWarrenDF.to_csv('Warren.csv',header = True, mode = 'a')
# AndrewYang = json_2.mentions.str.contains('AndrewYang',regex = False)
# AndrewYangDF = json_2[AndrewYang]
# AndrewYangDF.to_csv('Yang.csv',header = True, mode = 'a')

print('done')
```

Figure 2-2 Sentiment

```
import csv
from textblob import TextBlob
import pandas as pd
from pandas import DataFrame
#morethantweets is a dataframe that was created from the output of cinnamon2.py
#tweetdf is a duplicate of morethantweets,however it only has two columns
(id_str,tweetid)
morethantweets = pd.read_csv("nov112019.csv")
tweetdf = pd.read_csv("nov112019tweets.csv")
#These 4 lists were created the store some of the values for easier appending
to the dataframe
sentimentpolaritylist = []
sentimentsubjectivitylist = []
sentimentlist = []
subjectivitylist = []
sentence = tweetdf['text']
#.sentiment.polarity and .sentiment.subjectivity was used to keep the values
separate instead of using .sentiment
for sen in sentence:
    pepper = TextBlob(sen)
    pepperpolarity = pepper.sentiment.polarity
    peppersubjectivity = pepper.sentiment.subjectivity
    #in this loop after the sentiment of each sentence("tweet") is analyzed the
```

```
polarity and subjectivity is then appending to the applicable lists
    sentimentpolaritylist.append(pepperpolarity)
    sentimentsubjectivitylist.append(peppersubjectivity)

#after the loop is finished, both of the lists are inserted into the
morethantweets df and named respectivetly, Polarity & Subjectivity
morethantweets.insert(2,"Polarity",sentimentpolaritylist)
morethantweets.insert(3,"Subjectivity",sentimentsubjectivitylist)
#With the polarity value, we are now able to determine the sentiment of the
tweet. Each of the sentiment are appended to a sentiment list
for pole in morethantweets["Polarity"]:
    if pole > 0:
        sentimentlist.append("Positive")
    elif pole < 0:
        sentimentlist.append("Negative")
    else:
        sentimentlist.append("Neutral")
#Similar logic for Subjectivity
for subj in morethantweets["Subjectivity"]:
    if subj > .25 and subj <= 0.5:
        subjectivitylist.append("Rational")
    elif subj > .5 and subj <= .75:
        subjectivitylist.append("Emotional")
    elif subj > .75 and subj <= 1:
        subjectivitylist.append("Strong Emotional")
    else:
        subjectivitylist.append("Strong Rational")
#Once again after the loop is completed the lists are appended into
morethantweets
morethantweets.insert(4,"Sentiment Refined",sentimentlist)
morethantweets.insert(5, "Subjectivity Refined",subjectivitylist)
#morethantweets is then extracted to a csv for visualization purposes and
further examination
morethantweets.to_csv("Completed.csv")
```

Dynamic Code

Figure 2-3 All-in-One

```
#NewSentiment
#Stephen Schneider
#Capstone Python Text Analytics Project

from textblob import TextBlob
import sys, tweepy
import matplotlib.pyplot as plt

#create a function to calculate the percentage. Takes 2 arguments, part and
whole.
def percentage(part, whole):
      return 100 * float(part)/float(whole)

#Twitter app info
consumerKey = "nT8KP0LQX0p7sxqgUawojrpxd"
consumerSecret = "lZFejHEMD5mA7Lwjxc47DuHe6gUekG4wddo270UIRbzigUj6fq"
accessToken = "1046519830461984768-JiGfq4e0VNGgewONGM8A5ixBcFWUT3"
accessTokenSecret = "ITvIwjGNdJY28gmJRlJM9ZdKmiVbRMILgmLAGgKnLBSYS"

#establish connection to Twitter API
auth = tweepy.OAuthHandler(consumerKey, consumerSecret)
auth.set_access_token(accessToken, accessTokenSecret)
api = tweepy.API(auth)

#Get input from user, what to search and how many tweets to analyze
searchTerm = input("Enter keyword to search: ")
noOfSearchTerms = int(input("Enter how many tweets to analyze: "))

#make tweets based on search criteria
tweets = tweepy.Cursor(api.search, q=searchTerm).items(noOfSearchTerms)

#TextBlob rates a sentence from -1 to 1. -1 being extremely negative. 1 being
extremely positive.
#Test: a = TextBlob("I am the worst programmer ever."
#a.sentiment.polarity
# -1
#b = TextBlob("I am the best programmer ever.")
#b.sentiment.polarity
# 1
#c = TextBlob("I am a programmer.")
#c.sentiment.polarity
```

```
# 0

#create 3 variables to store the polarity
positive = 0
negative = 0
neutral = 0
polarity = 0

#get total polarity for all the tweets
for tweet in tweets:
	#print(tweet.text)
	analysis = TextBlob(tweet.text)
	polarity += analysis.sentiment.polarity

	if(analysis.sentiment.polarity == 0):
		neutral += 1

	elif(analysis.sentiment.polarity < 0.00):
		negative += 1

	elif(analysis.sentiment.polarity > 0.00):
		positive += 1

#calculate percentage of positive, negative, and neutral
positive = percentage(positive, noOfSearchTerms)
negative = percentage(negative, noOfSearchTerms)
neutral = percentage(neutral, noOfSearchTerms)

#reformat to 2 decimal places
positive = format(positive, '.2f')
negative = format(negative, '.2f')
neutral = format(neutral, '.2f')

print("How people are reacting to " +searchTerm + " by analyzing "
+str(noOfSearchTerms) + " Tweets.")

#print consensus
if (polarity == 0):
	print("Neutral")
elif (polarity < 0):
	print("Negative")
elif (polarity > 0):
	print("Positive")
```

```
#create pie chart
labels = ['Positive ['+str(positive)+ '%]', 'Neutral ['+str(neutral)+ '%]',
'Negative ['+str(negative)+ '%]']
sizes = [positive, neutral, negative]
colors = ['yellowgreen', 'gold', 'red']
patches, texts = plt.pie(sizes, colors=colors, startangle=90)
plt.legend(patches,labels, loc="best")
plt.title("How people are reacting to " +searchTerm + " by analyzing "
+str(noOfSearchTerms) + " Tweets.")
plt.axis('equal')
plt.tight_layout()
plt.show()
```

Data Set

Figure 3-1: Static - October 15 Snapshot

text	Polarity	Subjectivity	Sentiment	created_at	coordinates	in_reply_to_screen_	user_id_str	user_location	user_screen_name	hashtags	mentions	Source File
@BetoORourke @Di	0.7	0.6	Positive	Oct 16 00:53:44 +0000 2019		BetoORourke	268776039		njjcv	[]	['BetoORourke', 'Diar	Rourke
@JoeBiden Mr Bider	0.0625	0.9166666667	Positive	Oct 15 23:59:53 +0000 2019		JoeBiden	167924087	Alabama	RoseDru63	[]	['JoeBiden']	Biden
@JoeBiden you can'	0.01875	0.7625	Positive	Oct 16 00:53:45 +0000 2019		JoeBiden	484367959	Alabama	WayneCain	[]	['JoeBiden']	Biden
@PeteButtigieg YOU	1	0.8	Positive	Oct 15 23:58:02 +0000 2019		PeteButtigieg	1.06E+18	Alabama	Jennife97449851	[]	['PeteButtigieg']	Buttgieg
@PeteButtigieg ðŸ'	0	0	Neutral	Oct 15 23:59:10 +0000 2019		PeteButtigieg	1.06E+18	Alabama	Jennife97449851	[]	['PeteButtigieg']	Buttgieg
#MayorPete is kickin SORRY, NOT SORR GO PETE!!!!! #DemocraticDebate @PeteButtigieg #DemDebate	0	1	Neutral	Oct 16 01:31:47 +0000 2019			1714142790	Alabama	KerriKac	['MayorPete', 'Democ	['PeteButtigieg']	Buttgieg
@KamalaHarris Over	0	0	Neutral	Oct 16 01:48:26 +0000 2019		KamalaHarris	1.05E+18	Alabama	FrontPorchRep	['DemDebate']	['KamalaHarris']	Harris
@BernieSanders The	0	0	Neutral	Oct 16 01:31:26 +0000 2019		BernieSanders	83925756	Alabama	CUBUNO1	[]	['BernieSanders']	Sanders
@CiaranDeFaoite @	0	0	Neutral	Oct 15 23:53:01 +0000 2019		CiaranDeFaoite	876598638	Alabama	1goalnmind	[]	['CiaranDeFaoite', 'T	Steyer
Totally dig @TomSt #DemDebate #Demc	0	0.75	Neutral	Oct 16 01:49:33 +0000 2019			2273161893	Alabama	lizzythelesi	['CitizensUnited', 'De	['TomSteyer']	Steyer
@TomSteyer why? E	-0.1458333333	0.2208333333	Negative	Oct 15 23:49:44 +0000 2019		WTR4U	539166324	Alabama	WTR4U	[]	['TomSteyer']	Steyer
@AndrewYang ðŸ'	0	0	Neutral	Oct 15 23:58:36 +0000 2019		AndrewYang	1.06E+18	Alabama	Jennife97449851	[]	['AndrewYang']	Yang
@pushforward40 @K	0	0	Neutral	Oct 16 01:32:10 +0000 2019		pushforward40	9.30E+17	Alabama	LaTonya49927569	[]	['pushforward40', 'Ka	Harris
@BetoORourke They	0	0	Neutral	Oct 16 00:55:09 +0000 2019		BetoORourke	101579962	Alabama	p11b30	[]	['BetoORourke']	Rourke
@BernieSanders Wh	0.6	1	Positive	Oct 15 23:59:38 +0000 2019		BernieSanders	3918728308	Alabama	AITrumpTrain	[]	['BernieSanders']	Sanders
@LindseyGrahamSC	-0.1071428571	0.6428571429	Negative	Oct 16 01:31:56 +0000 2019		LindseyGrahamSC	475057018	Alabama	stanford_david	[]	['LindseyGrahamSC',	Sanders
#DemDebates Mode	-0.075	0.55	Negative	Oct 16 00:53:59 +0000 2019			43261063	Alabama	gcrump21	['DemDebates']	['JoeBiden']	Biden
@bmoschetti @Elair	0	0	Neutral	Oct 16 01:48:36 +0000 2019		bmoschetti	1.15E+18	Alaska	BadKarm59255794	[]	['bmoschetti', 'Elaine	Biden
@DailyCaller @JoeE	0	0	Neutral	Oct 16 01:31:51 +0000 2019		DailyCaller	272237835	Arizona	Mydesigningmind	[]	['DailyCaller', 'JoeBic	Biden
@JoeBiden Hoping y	0	0	Neutral	Oct 15 23:58:37 +0000 2019		JoeBiden	353956711	Arizona	Na68er	[]	['JoeBiden']	Biden
@JoeBiden we want	0	0	Neutral	Oct 16 00:53:44 +0000 2019		JoeBiden	1.14E+18	Arizona	PhxYangGang	['FreedomDividend',	['JoeBiden']	Biden
@MSNBC @RachelMaddow @	0	0	Neutral	Oct 15 13:23:24 +0000 2019		MSNBC	114373177	Arizona	TamPhotog	[]	['MSNBC', 'rachelma	Biden
@ElizabethWarren @	0.5	0.8888888889	Positive	Oct 15 16:56:36 +0000 2019		elizabethwarren	114373177	Arizona	TamPhotog	[]	['elizabethwarren', 'J	Biden
@CoreyBooker is RIG	0.1785714286	0.7678571429	Positive	Oct 16 01:14:42 +0000 2019		CoreyBooker	636559543	Arizona	kitcatlyon	['Democrat', 'Unity']	['CoreyBooker']	Booker
@davidaxelrod @Pe	0	0	Neutral	Oct 16 01:48:47 +0000 2019		davidaxelrod	16917851	Arizona	7407413a	[]	['davidaxelrod', 'Pete	Buttgieg
@VanJones68 @Pet	0.0625	0.35	Positive	Oct 15 23:59:54 +0000 2019		VanJones68	138996169	Arizona	herekitty67	[]	['VanJones68', 'Petel	Buttgieg

Figure 3-2: Static - November 3 Snapshot

text	Polarity	Subjectivity	Sentiment Refined	Subjectivity Refined	retweet_count	favorite_count	created_at	coordinates	in_reply_to_screen_	user_id_str	user_location	user_screen_name
@W_Jamie @MattB	0.125	0.1875	Positive	Strong Rational	0	0	Mon Nov 04 23:59:58 +0000 2019		W_Jamie	31649984	Cincinnati	BillyDonovan
@SelmaTerzic1 @J	-0.333333333	0.583333333	Negative	Emotional	0	0	Mon Nov 04 23:59:55 +0000 2019		SelmaTerzic1	755688480		bathtubmoses
@JoeBiden Because	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:51 +0000 2019		JoeBiden	110048166	Connecticut, USA	plays2many_RPGs
Only two outcomes,	-0.044444444	0.9	Negative	Strong Emotional	0	0	Mon Nov 04 23:59:48 +0000 2019			303925826	Florida, NY	fred9038
@JoeBiden Because	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:48 +0000 2019		JoeBiden	4184373254		lvitag1
@T_S_P_O_O_K_Y	-1	1	Negative	Strong Emotional	1	1	Mon Nov 04 23:59:45 +0000 2019		T_S_P_O_O_K_Y	1.11E+18	Brooklyn, NY	BoursiquotAnge
@JoeBiden https://t.	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:45 +0000 2019		JoeBiden	326891196	Abilene, TX	jadanosh
@AskMe_idareyou @	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:44 +0000 2019		AskMe_idareyou	9.17E+17		the1chainsawguy
@h_ghosn @MarkDi	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:43 +0000 2019		h_ghosn	1.11E+18		BobConroy4
@JoeBiden 27 years	-0.5	1	Negative	Strong Emotional	0	4	Mon Nov 04 23:59:43 +0000 2019		JoeBiden	7.13E+17		dcostanzo4444
@JoeBiden A nickna	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:39 +0000 2019		JoeBiden	9.51E+17	Michigan, USA	RoscoOriginal
@JoeBiden You're s	0.15	0.475	Positive	Rational	0	0	Mon Nov 04 23:59:38 +0000 2019		JoeBiden	25380274		joeg1117
@HillaryClinton @Jo	-0.15	0.6	Negative	Emotional	0	0	Mon Nov 04 23:59:37 +0000 2019		HillaryClinton	1.05E+18		libtards_no
@steph93065 @See	-0.15	0.75	Negative	Emotional	0	0	Mon Nov 04 23:59:35 +0000 2019		steph93065	1.03E+18	West Virginia, USA	OATH101STUSARI
@JoeBiden Yes we v	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:30 +0000 2019		JoeBiden	100048541	Illinois, USA	T_Raffety
@reallyabu @SebG	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:24 +0000 2019		reallyabu	63170544		sofiegeorge
@T_S_P_O_O_K_Y	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:22 +0000 2019		T_S_P_O_O_K_Y	411764646	Texas, USA	TanyaLaPree
@jsolomonReports C	0	0	Neutral	Strong Rational	0	2	Mon Nov 04 23:59:18 +0000 2019		jsolomonReports	3019003122	Parkland, FL	LilLubbie
@JoeBiden @senate	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:17 +0000 2019		JoeBiden	9.29E+17	Oregon	Route0660
@BrianRecca @karg	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:15 +0000 2019		BrianRecca	1.16E+18	United States	TheAmericanPa10
@JoeBiden Sooo, ly	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:15 +0000 2019		JoeBiden	1.18E+18		The4thJDM
@JoeBiden Because	0	0	Neutral	Strong Rational	0	1	Mon Nov 04 23:59:13 +0000 2019		JoeBiden	847839458		emonkey79
@CharityChicken @	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:12 +0000 2019		CharityChicken	7.60E+17	The 49th District of (	CA49thDistrict
@T_S_P_O_O_K_Y	0.65	0.95	Positive	Strong Emotional	0	0	Mon Nov 04 23:59:08 +0000 2019		T_S_P_O_O_K_Y	1.02E+18		JackManeJr31
Poor @joebiden. He	0.05	0.675	Positive	Emotional	0	0	Mon Nov 04 23:59:06 +0000 2019			3377336925	TheDanCrenshawWa	B13_suMCovfefe
@JoeBiden @senate	0	0	Neutral	Strong Rational	0	0	Mon Nov 04 23:59:06 +0000 2019		JoeBiden	93760197		jknightj
@JoeBiden You hav #idiot	0.133333333	0.683333333	Positive	Emotional	0	0	Mon Nov 04 23:59:04 +0000 2019		JoeBiden	9.49E+17	South Carolina, USA	iamjumpingin
@T_S_P_O_O_K_Y	0.392857143	0.712301587	Positive	Emotional	0	0	Mon Nov 04 23:59:03 +0000 2019		T_S_P_O_O_K_Y	9.51E+17	USA	NomadsWife6973
@mrglenn @Marlen	0	0.5	Neutral	Rational	0	5	Mon Nov 04 23:59:01 +0000 2019		mrglenn	260435415	New Brunswick	Fem0947

Figure 3-3: October 15 Consolidated Data

COUNT of Data											
	Biden	Booker	Buttigieg	Castro	Harris	Klobuchar	Sanders	Steyer	Warren	Yang	ORourke
Very Favorable	47	9	56	43	47	4	31	33	31	52	41
Somewhat Favorable	69	2	96	67	71	16	79	79	58	72	67
Somewhat Unfavorable	52	4	46	41	36	7	44	41	25	46	47
Very Unfavorable	22	7	24	9	12	2	20	12	9	12	15
Neutral	266	75	258	199	266	62	273	177	257	235	234
Total	456	97	480	359	432	91	447	342	380	417	404
Total w/o Neutral	190	22	222	160	166	29	174	165	123	182	170
Percentages of Data											
Very Favorable	10	9	12	12	11	4	7	10	8	12	10
Somewhat Favorable	15	2	20	19	16	18	18	23	15	17	17
Somewhat Unfavorable	11	4	10	11	8	8	10	12	7	11	12
Very Unfavorable	5	7	5	3	3	2	4	4	2	3	4
Neutral	58	77	54	55	62	68	61	52	68	56	58

Percentages w/o Neutral											
Very Favorable	25	41	25	27	28	14	18	20	25	29	24
Somewhat Favorable	36	9	43	42	43	55	45	48	47	40	39
Somewhat Unfavorable	27	18	21	26	22	24	25	25	20	25	28
Very Unfavorable	12	32	11	6	7	7	11	7	7	7	9
Favorable	61	50	68	69	71	69	63	68	72	68	64
Unfavorable	39	50	32	31	29	31	37	32	28	32	36

Figure 3-4: November 3 Consolidated Data

COUNT of Data										
	Biden	Booker	Buttigieg	Castro	Harris	Klobuchar	Sanders	Steyer	Warren	Yang
Very Favorable	35	5	42	23	38	15	23	43	16	35
Somewhat Favorable	51	2	60	51	53	25	56	27	66	74
Somewhat Unfavorable	35	2	53	23	34	9	29	41	32	31
Very Unfavorable	20	0	10	6	9	1	6	15	16	9
Neutral	162	19	186	143	198	102	187	127	136	151
Total	303	28	351	246	332	152	301	253	266	300
Total w/o Neutral	141	9	165	103	134	50	114	126	130	149
Percentages of Data										
Very Favorable	12	18	12	9	11	10	8	17	6	12
Somewhat Favorable	17	7	17	21	16	16	19	11	25	25
Somewhat Unfavorable	12	7	15	9	10	6	10	16	12	10
Very Unfavorable	7	0	3	2	3	1	2	6	6	3
Neutral	53	68	53	58	60	67	62	50	51	50
Percentages w/o Neutral										
Very Favorable	25	56	25	22	28	30	20	34	12	23
Somewhat Favorable	36	22	36	50	40	50	49	21	51	50
Somewhat Unfavorable	25	22	32	22	25	18	25	33	25	21
Very Unfavorable	14	0	6	6	7	2	5	12	12	6
Favorable	61	78	62	72	68	80	69	56	63	73

Unfavorable	39	22	38	28	32	20	31	44	37	27

Figure 3-5: November 14 Consolidated Data

Polls

Figure 4-1: The Economist/YouGov October 13-15, 2019 Summarization of Poll

Poll 10-15	Biden	Booker	Buttigieg	Castro	Harris	Klobuchar	Sanders	Steyer	Warren	Yang	O'Rourke
Very Favorable	18	10	14	7	12	9	19	6	22	8	8
Somewhat Favorable	23	21	18	20	22	19	22	13	19	23	23
Somewhat unfavorable	15	12	11	12	11	13	12	11	8	14	11
Very unfavorable	30	25	22	24	29	20	32	17	32	17	28
don't know	15	32	35	37	25	40	15	53	19	38	30
total	101	100	100	100	99	101	100	100	100	100	100
Favorable	41	31	32	27	34	28	41	19	41	31	31
Unfavorable	45	37	33	36	40	33	44	28	40	31	39
Don't know	15	32	35	37	25	40	15	53	19	38	30
Total	101	100	100	100	99	101	100	100	100	100	100

Figure 4-2: The Economist/YouGov November 3-5, 2019 Summarization of Poll

Poll 11-03	Biden	Booker	Buttigieg	Castro	Harris	Klobuchar	Sanders	Steyer	Warren	Yang
Very Favorable	18	10	14	8	13	8	20	6	22	8
Somewhat Favorable	22	20	17	19	18	16	23	14	18	20
Somewhat unfavorable	15	11	10	12	13	14	13	10	9	14
Very unfavorable	30	24	22	22	29	18	30	16	30	16
don't know	15	35	37	39	27	43	13	53	21	41
total	100	100	100	100	100	99	99	99	100	99

Favorable	40	30	31	27	31	24	43	20	40	28
Unfavorable	45	35	32	34	42	32	43	26	39	30
Don't know	15	35	37	39	27	43	13	53	21	41
Total	100	100	100	100	100	99	99	99	100	99

Figure 4-3: The Economist/YouGov November 10-12, 2019 Summarization of Poll

Poll 11-03	Biden	Booker	Buttigieg	Castro	Harris	Klobuchar	Sanders	Steyer	Warren	Yang
Very Favorable	16	11	14	8	12	9	20	6	21	8
Somewhat Favorable	23	21	18	19	21	18	21	15	17	21
Somewhat unfavorable	13	10	11	12	11	13	11	11	9	14
Very unfavorable	32	26	23	25	30	20	32	19	32	16
don't know	16	32	33	37	26	40	16	51	21	40
Total	100	100	100	100	100	99	99	99	100	99
Favorable	39	32	32	27	33	27	41	21	38	29
Unfavorable	45	36	34	37	41	33	43	30	41	30
Don't know	16	32	33	37	26	40	16	51	21	40
Total	100	100	100	100	100	99	99	99	100	99

References

Polls

[1] YouGov website: https://today.yougov.com/about/

[2] October 13 - 15:
https://d25d2506sfb94s.cloudfront.net/cumulus_uploads/document/dt74ean119/weeklytrackingreport.pdf

[3] November 3 - 5:
https://d25d2506sfb94s.cloudfront.net/cumulus_uploads/document/cseozthmrp/econTabReport.pdf

[4] November 10 -12:
https://d25d2506sfb94s.cloudfront.net/cumulus_uploads/document/7umtlf80ov/econTabReport.pdf