Contents

Table of Contents

1 Final Project Phase 4

1.1 Team Information

- Group 2: Pied Piper
- President Trump Twitter Analysis & Election Simulat
- Alex Bzdel - abzdel@bryant.edu (mailto:abzdel@bry
- Zach Galante - zgalante@bryant.edu (mailto:zgalan
- Robert Mitchell - rmitchell2@bryant.edu (mailto:rmit

2 Abstract

We plan to explore Donald Trump's approval
well as tweets regarding coronavirus. We w
n conjunction with these tweets. Our goal
d sentiment) of tweets with a keyword and
ump Administration's coronavirus timeline
is analysis and see how they correlate to

Part 1 will deal with Trump's approval rat
Part 2 will produce an election simulation

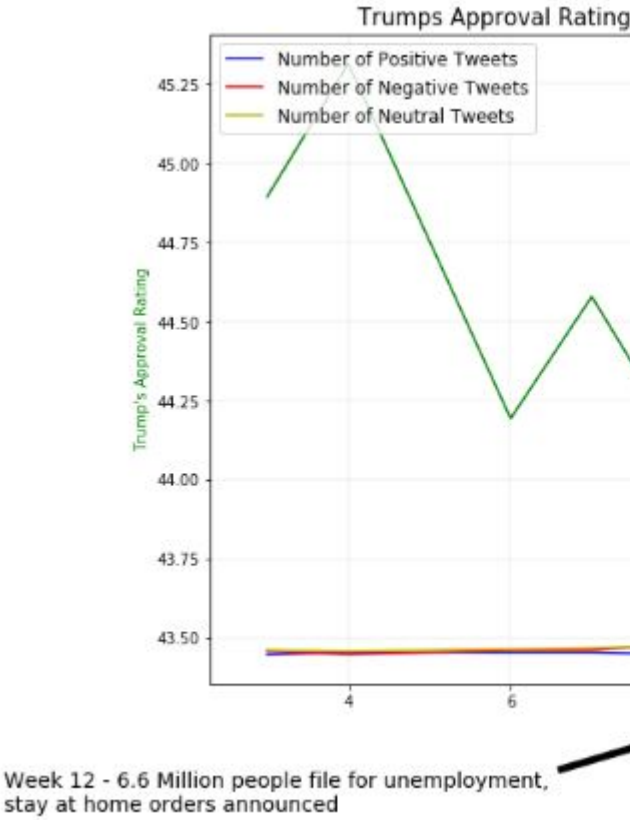
2.1 Keywords & hashtags:

- keywords: Trump, Stimulus Check, $1200, covid19,
- hashtags: #trump, #stimuluscheck, #money

2.2 Hypotheses

1) President Trump's approval rating has i
ut the stimulus checks distributed by the
hat have highly rated words as determined
al rating will be.
2) Sentiment ratings for the tweets will s
be distributed.
3) That Trump's approval rating is going t
also be reflected in the results of the 20

Contents



One of our final plots on Trump's Approval Rating vs Stin

```
Trump Wins Pennsy
Trump Vote Count
----------------
Biden Wins Illin
Biden Vote Count
----------------
Trump Wins Ohio
Trump Vote Count
----------------
Biden Wins Michi
Biden Vote Count
----------------
Trump Wins Georg
Trump Vote Count
----------------
Biden Wins North
Biden Vote Count
----------------
  *****Biden win
----------------
Trump's final sc
Biden's final sc
----------------
```

Final simulation example above

2 Part 1: Tracking Trump's A

In [146]:
```
# imports
import numpy as np
import pandas as pd
#import tweepy
#import twython

import json
import csv
import os
import codecs
import time
import gender_guesser.detector as gender
from textblob import TextBlob
from wordcloud import WordCloud, STOPWORDS
import seaborn as sns
import matplotlib.pyplot as plt
import re
%matplotlib inline
```

3.1 Description of the Data - Part

3.1.1 Scraper(s) used

Contents

Contents

In [147]:

```
#--------- NOTE THAT THIS CELL IS NOT SUPP
import time
from random import seed
from random import randint
# seed random number generator
seed(1)
from datetime import date, datetime, timec

def datespan(startDate, endDate, delta=tim
    currentDate = startDate
    while currentDate < endDate:
        yield currentDate
        currentDate += delta

text_query = "Stimulus Check"
start_of_period = date(2020, 1, 4)
end_of_period   = date(2020, 5, 2)
start_of_week = start_of_period
week_plus_1_start = start_of_period + time

for i, week_plus_1_start in enumerate(date
    print(f"week {i}, start of week: {star
    get_tweet_result(text_query = text_que
                     until_date = f"{week_plus
    time.sleep(randint(20, 90))

    start_of_week = week_plus_1_start
```

```
week 0, start of week: 2020-01-04 2020-01-

---------------------------------------------
NameError
<ipython-input-147-6a8d3fb3b268> in <modul
     21 for i, week_plus_1_start in enumer
days=7))):
     22     print(f"week {i}, start of wee
---> 23     get_tweet_result(text_query =
     24                      until_date = f"{w
     25     time.sleep(randint(20, 90))

NameError: name 'get_tweet_result' is not
```

3.1.2 Attempt at pulling user data (failed

Citation for scraper: https://github.com/taspinar/twitterscr

Contents

In [148]:

```
#--------- NOTE THAT THIS CELL IS NOT SUPP
from twitterscraper.query import query_use
import pandas as pd
from multiprocessing import Pool
import time
from IPython.display import display
import random


global twitter_user_info
twitter_user_info=[]


def get_user_info(twitter_user):
    """
    An example of using the query_user_inf
    :param twitter_user: the twitter user
    :return: twitter_user_data: returns a
    """
    user_info = query_user_info(user= twit
    twitter_user_data = {}
    twitter_user_data["user"] = user_info.
    twitter_user_data["fullname"] = user_i
    twitter_user_data["location"] = user_i
    twitter_user_data["blog"] = user_info.
    twitter_user_data["date_joined"] = use
    twitter_user_data["id"] = user_info.id
    twitter_user_data["num_tweets"] = user
    twitter_user_data["following"] = user_
    twitter_user_data["followers"] = user_
    twitter_user_data["likes"] = user_info
    twitter_user_data["lists"] = user_info

    return twitter_user_data


def main():
    start = time.time()
   # users = list(df.username.values) # pd
    #users = ['A3Patriot']
    users = ['A3Patriot', 'whaley1212', 'J
       'JLaroc318', 'MamaR130', 'cjstockto
       'VBrown13245591', 'bradyswenson', '
       'cajunvincent', 'grumpy_idiot', 'fv
       'LMenssen', 'AmericanEarlR', 'Frank
       'mtrwf11', 'Isaac_Visage', None]
    pool = Pool(8)
    for user in pool.map(get_user_info,use
        if True: # if/else meant to fix th
            twitter_user_info.append(user)
            time.sleep(random.randint(1,10
        elif False:
            print('an error has occurred')
            break

        # while true else break
```

```
    cols=['id','fullname','date_joined','l
        #cols=['id','date_joined','locatic

data_frame = pd.DataFrame(twitter_user_inf
        # save data_frame as a global vari
data_frame.index.name = "Users"
data_frame.sort_values(by="followers", asc
elapsed = time.time() - start
print(f"Elapsed time: {elapsed}")
display(data_frame)


if __name__ == '__main__':
    main()
```

```
---------------------------------------
ModuleNotFoundError
<ipython-input-148-72ca092c163d> in <modul
      1 #--------- NOTE THAT THIS CELL IS
----> 2 from twitterscraper.query import c
      3 import pandas as pd
      4 from multiprocessing import Pool
      5 import time

ModuleNotFoundError: No module named 'twit
```

3.1.3 Data Description

Our first analysis involves data with tweets involving keyv

```
* meant to analyze tweet counts and sentim
* drawback: no user data available (follow
```

- **We scraped the following tweet characteristics v**
  - Formatted Date of tweet
  - Author ID (user ID)
  - Username
  - User Location
  - Tweet Text
  - Hashtags
  - Mentions
  - to
    - who the tweet is addressed to - if anyone
  - Tweet URL
  - Replies
  - Retweets
  - Favorites
  - Permalink
- **We then added the following**:

- filtered_text
- retweet_flags
- Polarity_Score
- SubjectivityScore
- sentimentLabel

3.1.3.1 Third-Party Data

For our third party data, we will be using Donald Tru
will primarily be using this to compare the first datas
fivethirtyeight.com

3.1.3.2 Data Processing Tasks

For both sets of data, a big step in processing is sentime
different keywords in the context of sentiment. Additional
depict how sentiments (of both certain keywords and tota
creating three sub-DataFrames (one for each sentiment)
tweets vs. Trump's Approval Rating

3.1.3.3 When and How Long You Scraped Twitter

- we scraped twitter on 4/30 and 5/1 for an hour each

3.1.4 Load in Data

Contents

Contents

In [1]:

```
import pandas as pd
import numpy as np
hashtagTrump = pd.read_csv("data_files/Gro
hashtagTrump.columns = ['formatted_date',
new_money = pd.read_csv("data_files/Group_
new_money.columns = ['formatted_date', "au
a = new_money.append(hashtagTrump, ignore_
hash_stimCheck = pd.read_csv("data_files/G
hash_stimCheck.columns = ['formatted_date'
b = a.append(hash_stimCheck, ignore_index
coronavirus = pd.read_csv("data_files/Grou
coronavirus.columns = ['formatted_date', '
c = b.append(coronavirus, ignore_index = T
COVID19 = pd.read_csv("data_files/Group_2_
COVID19.columns = ['formatted_date', "auth
d = c.append(COVID19, ignore_index = True)
stim_check = pd.read_csv("data_files/Group
stim_check.columns = ['formatted_date', "a
e = d.append(stim_check, ignore_index = Tr
Trump = pd.read_csv("data_files/Group_2_Ph
Trump.columns= ['formatted_date', "author_
df = e.append(Trump, ignore_index = True)
df.head()
```

Out[1]:

	formatted_date	author_id	username	
0	Fri Apr 10 23:59:57 +0000 2020	2.716171e+08	ForeverMe_MsB	N
1	Fri Apr 10 23:59:48 +0000 2020	4.970938e+07	_carceeexoxo	N
2	Fri Apr 10 23:59:45 +0000 2020	1.239146e+08	bitcoinization	N
3	Fri Apr 10 23:59:44 +0000 2020	2.826924e+09	4rdaSquad	N
4	Fri Apr 10 23:59:39 +0000 2020	5.381119e+08	Felipe__	N

3.1.4.1 Approval rating data

In [150]:
```
approval_df = pd.read_csv('data_files/Grou
approval_df.head()
```

Out[150]:

	president	subgroup	modeldate	startdate	en
0	Donald Trump	Voters	4/30/2020	5/28/2019	4/3(
1	Donald Trump	Voters	4/30/2020	5/29/2019	4/3(
2	Donald Trump	Voters	4/30/2020	5/30/2019	4/29
3	Donald Trump	Voters	4/30/2020	5/31/2019	4/29
4	Donald Trump	Voters	4/30/2020	6/1/2019	4/29

5 rows × 22 columns

3.2 Data Cleaning

In [151]:
```
!pip install nltk
```

```
Requirement already satisfied: nltk in /us
Requirement already satisfied: six in /usr
```

In [152]:
```
# deal with nots
!pip install TextBlob
```

```
Requirement already satisfied: TextBlob in
Requirement already satisfied: nltk>=3.1 i
5)
Requirement already satisfied: six in /usr
(1.14.0)
```

Contents

In [153]:
```
import numpy as np
import pandas as pd
#import tweepy
#import twython

import json
import csv
import os
import codecs
import time
import gender_guesser.detector as gender
from textblob import TextBlob
from wordcloud import WordCloud, STOPWORDS
import seaborn as sns
import matplotlib.pyplot as plt
import re
%matplotlib inline
```

In [154]:
```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
```

```
[nltk_data] Downloading package stopwords
[nltk_data]   Package stopwords is already
[nltk_data] Downloading package punkt to /
[nltk_data]   Package punkt is already up-
```

Out[154]: True

In [155]:
```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import re
stop_words = set(stopwords.words('english'

def remove_url(txt):
    """Replace URLs found in a text string
    (i.e. it will remove the URL from the
    """
    return re.sub("([^0-9A-Za-z \t])|(\w+:

def remove_stop_words(txt):
    """lower case the text, and DROP stop
    return " ".join([w for w in word_toker

def preprocess_tweet_text(txt):
    return remove_stop_words(remove_url(t
```

In [156]:
```
df.text.isnull().sum()
df.text = df.text.fillna(' ') # fill nulls
```

In [157]:
```
# Remove URLs and Stop Words
df['filtered_text'] = df.text.apply(preprc

#retweet
df['retweet_flags'] = df.text.str.startswi
```

3.2.1 Create week bins

In [158]:
```
approval_df.enddate = pd.to_datetime(apprc
approval_df['week_bins'] = approval_df.enc
```

In [159]:
```
approval_df = approval_df.where(approval_c
```

In [160]:
```
df.formatted_date = pd.to_datetime(df.form
df['week_bins'] = df.formatted_date.dt.wee
```

3.2.2 Calculate Polarity

In [161]:
```
df['PolarityScore'] = df.filtered_text.app
df['SubjectivityScore'] = df.filtered_text
```

In [162]:
```
def sentiment_bins(data):
    if data <= -0.5:
        grouping = 'Strong-Negative'
    elif data > -0.5 and data < 0.0:
        grouping = 'Mild-Negative'
    elif data > 0.0 and data < 0.5:
        grouping = 'Mild Positive'
    elif data >=0.5:
        grouping = 'Strong-Positive'
    else:
        grouping = 'Neutral'
    return grouping

df['sentimentLabel'] = df['PolarityScore']
#These bins should be  used later for grap
```

In [163]:
```
df['sentiment'] = df.PolarityScore.apply(s
df.head()
```

Out[163]:

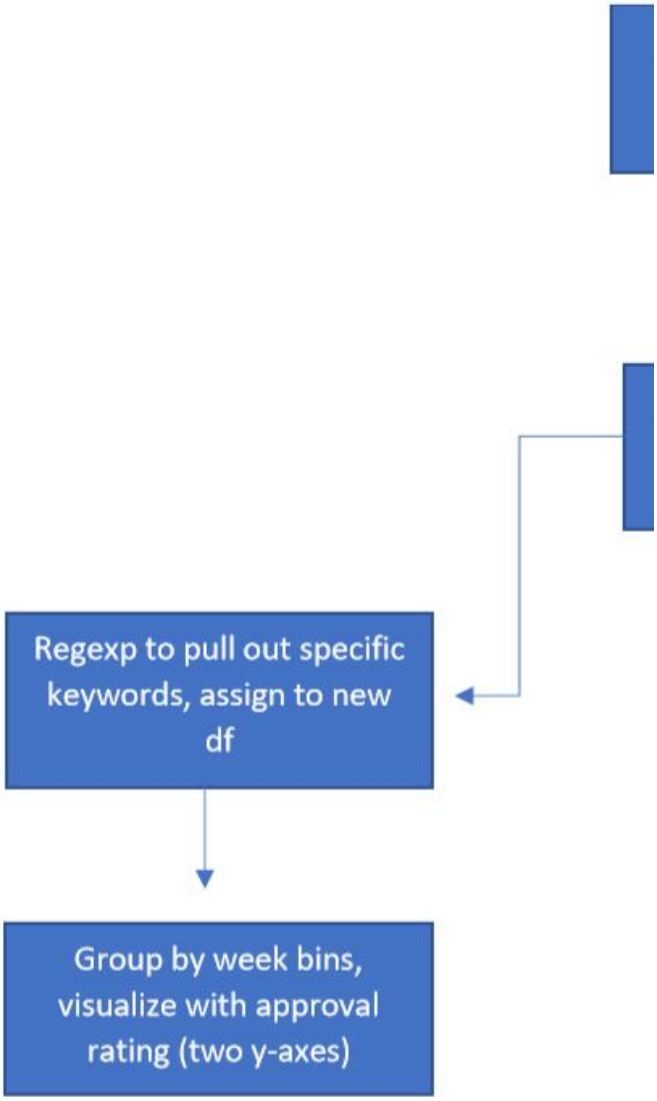
	formatted_date	author_id	username	
0	2020-04-10 23:59:57+00:00	2.716171e+08	ForeverMe_MsB	N
1	2020-04-10 23:59:48+00:00	4.970938e+07	_carceeexoxo	N
2	2020-04-10 23:59:45+00:00	1.239146e+08	bitcoinization	N
3	2020-04-10 23:59:44+00:00	2.826924e+09	4rdaSquad	N
4	2020-04-10 23:59:39+00:00	5.381119e+08	Felipe__	N

3.3 Workflow Diagrams

First, let's look at our planned workflow for how we will us

Contents

Contents



Here's how we plan to use our third party data about Tru

Contents

3.4 Visual EDA Prep - Part 1

3.4.1 Stimulus check tweets (regexp)

In [164]:
```
import re
df.text = df.text.fillna(" ")
stimuluscheck_df = df[df['text'].str.conta
```

In [165]:
```
stimuluscheck_df.groupby('week_bins')['te
```

Out[165]:
```
array([   2,    5,   16,   16,   18,   24,
        747,  980,  963, 1030, 1031,  477]
```

3.4.2 Coronavirus tweets

In [166]:
```
corona_df = df[df['text'].str.contains("12
arrcorona = corona_df.groupby('week_bins')

arrcorona= np.insert(arrcorona, 0, 0) #ins
```

3.4.3 Count of tweets for each sentiment

In [167]:
```
df_pos = df.where((df['sentimentLabel']=='
df_neg = df.where((df['sentimentLabel']=='
df_neut = df.where((df['sentimentLabel']==
```

In [168]:
```
df_pos = df_pos.dropna(subset=['sentimentL
df_neg = df_neg.dropna(subset=['sentimentL
df_neut = df_neut.dropna(subset=['sentimer
```

In [169]:
```
df_pos.groupby('week_bins')['sentimentLabe
df_neg.groupby('week_bins')['sentimentLabe
df_neut.groupby('week_bins')['sentimentLab
```

Out[169]:
```
week_bins
1.0        2
2.0     1101
3.0     1108
4.0     1179
5.0     1164
6.0     1200
7.0     1137
8.0     1138
9.0     1372
10.0    1333
11.0    1967
12.0    2002
13.0    1984
14.0    2069
15.0    2115
16.0    2022
17.0    1597
Name: sentimentLabel, dtype: int64
```

In [170]:
```
arrneg = df_neg.groupby('week_bins')['text
arrneg = np.insert(arrneg, 0, 0)

arrpos = df_pos.groupby('week_bins')['text

arrneut = df_neut.groupby('week_bins')['te
```

Contents

3.4.4 Stimulus check by sentiment

In [171]:
```
stimuluscheck_pos_df = df_pos[df_pos['text
stimuluscheck_neg_df = df_neg[df_neg['text
stimuluscheck_neut_df = df_neut[df_neut['t
```

Tweets with positive sentiment are missing week bins 1,

In [172]:
```
approval_temp_y = approval_df.groupby('wee
approval_temp_y = approval_temp_y.drop(lab

approval_temp_x = approval_df.groupby('wee
approval_temp_x = approval_temp_x.drop(lab

stimcheckneg = stimuluscheck_neg_df.groupb
stimcheckneg = stimcheckneg.drop(labels=[5

stimcheckneut = stimuluscheck_neut_df.grou
stimcheckneut = stimcheckneut.drop(labels=
```

3.4.5 Coronavirus tweets by sentiment

In [173]:
```
# search for coronavirus tweets by sentime

corona_pos_df = df_pos[df_pos['text'].str.
corona_neg_df = df_neg[df_neg['text'].str.
corona_neut_df = df_neut[df_neut['text'].s
```

Contents

In [174]:
```
corona_pos_df.groupby('week_bins')['sentim
corona_neg_df.groupby('week_bins')['sentim
corona_neut_df.groupby('week_bins')['senti
```

Out[174]:
```
week_bins
2.0     169
3.0     253
4.0     323
5.0     274
6.0     283
7.0     301
8.0     310
9.0     592
10.0    526
11.0    650
12.0    694
13.0    708
14.0    719
15.0    668
16.0    687
17.0    541
Name: sentimentLabel, dtype: int64
```

In [175]:
```
corona_temp_y = approval_df.groupby('week_
corona_temp_y = corona_temp_y.drop(labels=

corona_temp_x = approval_df.groupby('week_
corona_temp_x = corona_temp_x.drop(labels=
```

3.5 Visuals and EDA - Part 1

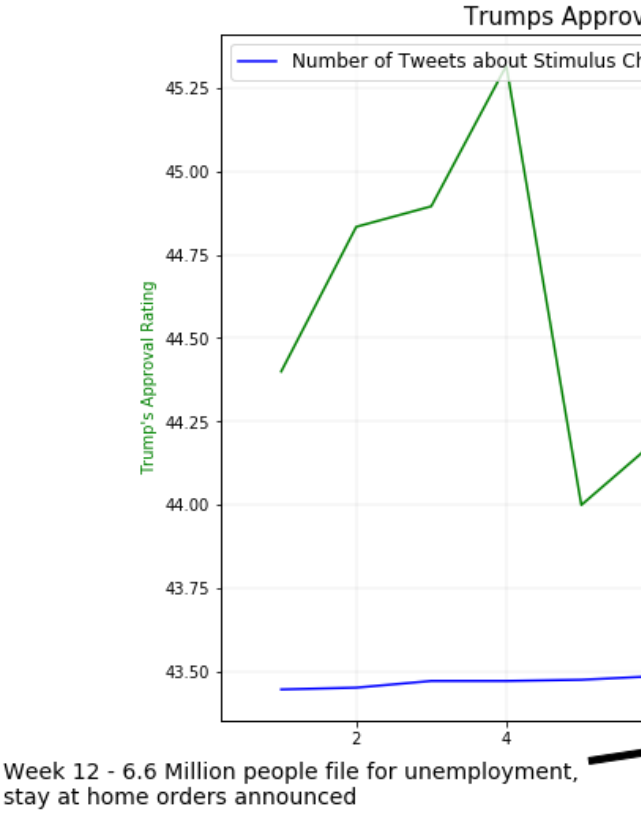
3.5.1 Number of Stimulus Check Tweets

First, let's explore how Trump's approval rating has chan
the year

Contents

In [176]:

```
import numpy as np
import matplotlib.pyplot as plt
x = approval_df.groupby('week_bins')['appr
y1 = approval_df.groupby('week_bins')['app
y2 = stimuluscheck_df.groupby('week_bins')
#y3 = arr1200

fig, ax1 = plt.subplots(figsize=(15,8))

ax1.set_title('Trumps Approval Rating vs.
ax1.grid(color='grey', linestyle='-', line

#duplicate ax1
ax2 = ax1.twinx()

#ax1 will plot Trump's approval rating, ax
ax1.plot(x, y1, 'g-')
ax2.plot(x, y2, 'b-')
#ax2.plot(x, y3, 'r-')

# set legends
ax1.legend(["Trump's Approval Rating"], fc
ax2.legend(["Number of Tweets about Stimul
#ax2.legend(bbox_to_anchor=(0.04, 0.82, 1.
            #handlelength=0.1, handletextp


ax1.set_xlabel('Week of Year')
ax1.set_ylabel("Trump's Approval Rating",
ax2.set_ylabel("Number of tweets", color='

# plt.text(6.4,3.3,"""Note on our visualiz
#          null value we replace it with t
#          on that platform's top 10 they
plt.text(14,200, """Weeks 15 and 16:
Talks of reopening the US""", fontsize = 1

plt.annotate("""Week 12 - 6.6 Million peop
stay at home orders announced""",
             xy = (12,15), xytext = (2,15)
             arrowprops=dict(facecolor='bl

plt.text(12,-200, """End of week 12:
Trump signs stimulus check bill""", fontsi




plt.show()

# print('notable dates:')
# print('Week 12 - 6.6 Million people file
```



Contents

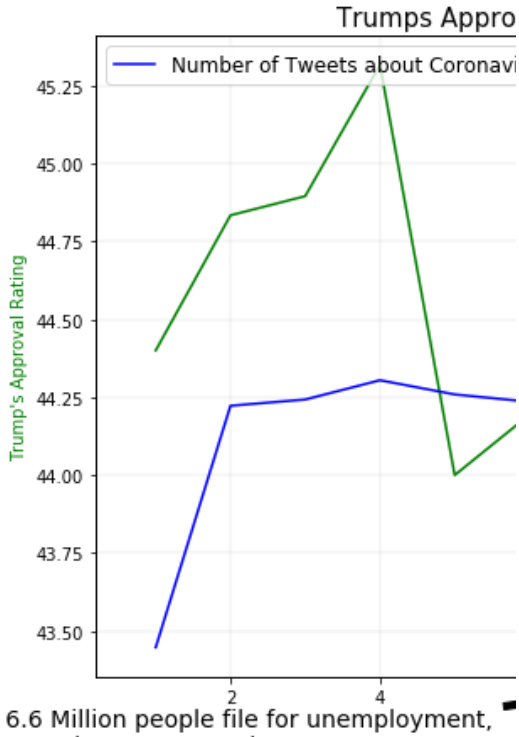
3.5.2 Number of Coronavirus Tweets

Now, let's look into how his rating changes with number c

Contents

In [177]:

```
import numpy as np
import matplotlib.pyplot as plt
x = approval_df.groupby('week_bins')['appr
y1 = approval_df.groupby('week_bins')['app
y2 = arrcorona

fig, ax1 = plt.subplots(figsize=(14,7))

ax1.set_title('Trumps Approval Rating vs.
ax1.grid(color='grey', linestyle='-', line

#duplicate ax1
ax2 = ax1.twinx()

#ax1 will plot Trump's approval rating, ax
ax1.plot(x, y1, 'g-')
ax2.plot(x, y2, 'b-')

# set legends
ax1.legend(["Trump's Approval Rating"], fc
ax2.legend(["Number of Tweets about Corona


plt.text(14,200, """Weeks 15 and 16:
Talks of reopening the US""", fontsize = 1

plt.annotate("""Week 12 - 6.6 Million peop
stay at home orders announced""",
             xy = (12,15), xytext = (2,15)
             arrowprops=dict(facecolor='bl

plt.text(12,-200, """End of week 12:
Trump signs stimulus check bill""", fontsi



ax1.set_xlabel('Week of Year')
ax1.set_ylabel("Trump's Approval Rating",
ax2.set_ylabel("Number of tweets", color='
#ax.legend((y1, y2), ('label1', 'label2'))
plt.show()
```



Week 12 - 6.6 Million people file for unemployment, stay at home orders announced

3.5.3 Number of Total Tweets by Sentime

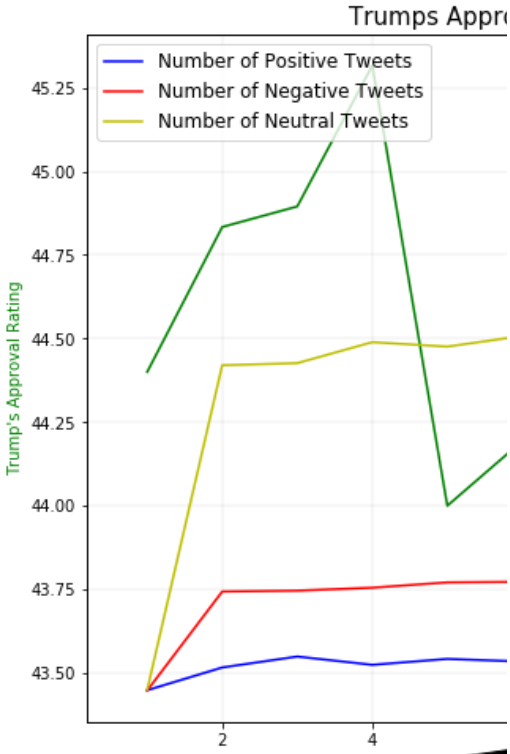
These give some interesting insights into how these figur
how this plot changes when we break it down by sentime

Contents

Contents

In [178]:

```
import numpy as np
import matplotlib.pyplot as plt
x = approval_df.groupby('week_bins')['appr
y1 = approval_df.groupby('week_bins')['app
y2 = arrpos
y3 = arrneg
y4 = arrneut

fig, ax1 = plt.subplots(figsize=(15,8))

ax1.set_title('Trumps Approval Rating vs.
ax1.grid(color='grey', linestyle='-', line

#duplicate ax1
ax2 = ax1.twinx()

#ax1 will plot Trump's approval rating, ax
ax1.plot(x, y1, 'g-')
ax2.plot(x, y2, 'b-')
ax2.plot(x, y3, 'r-')
ax2.plot(x, y4, 'y-')


# set legends
ax1.legend(["Trump's Approval Rating"], fc
ax2.legend(["Number of Positive Tweets", '

plt.text(14,300, """Weeks 15 and 16:
Talks of reopening the US""", fontsize = 1

plt.annotate("""Week 12 - 6.6 Million peop
stay at home orders announced""",
             xy = (12,15), xytext = (2,15)
             arrowprops=dict(facecolor='bl

plt.text(12,-400, """End of week 12:
Trump signs stimulus check bill""", fontsi




ax1.set_xlabel('Week of Year')
ax1.set_ylabel("Trump's Approval Rating",
ax2.set_ylabel("Number of tweets", color='
#ax.legend((y1, y2, y3), ('label1', 'label


plt.show()
```



Week 12 - 6.6 Million people file for unemployment, stay at home orders announced

3.5.4 Number of Stimulus Check Tweets

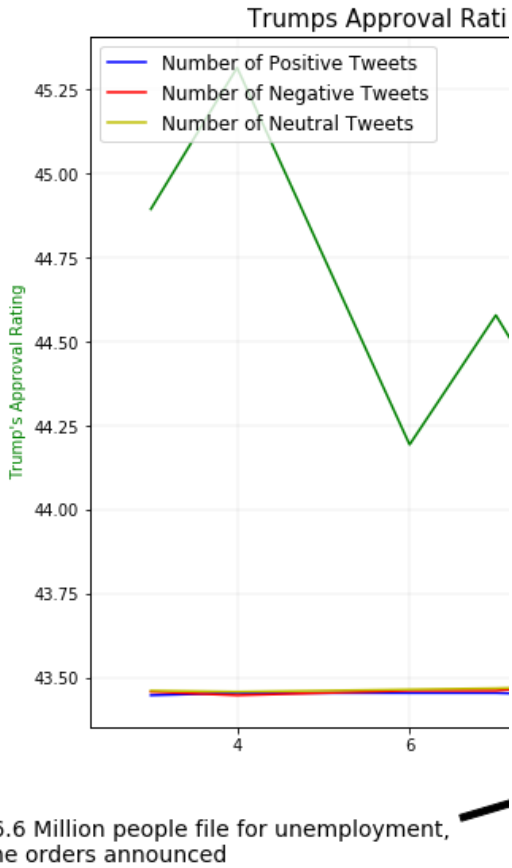
Let's take our positive, negative, and neutral tweets and

Contents

Contents

In [179]:

```
# import numpy as np
import matplotlib.pyplot as plt
x = approval_temp_x
y1 = approval_temp_y
y2 = stimuluscheck_pos_df.groupby('week_bi
y3 = stimcheckneg
y4 = stimcheckneut

fig, ax1 = plt.subplots(figsize=(15,8))

ax1.set_title('Trumps Approval Rating vs.
ax1.grid(color='grey', linestyle='-', line

#duplicate ax1
ax2 = ax1.twinx()

#ax1 will plot Trump's approval rating, ax
ax1.plot(x, y1, 'g-')
ax2.plot(x, y2, 'b-') # POSITIVE TWEETS
ax2.plot(x, y3, 'r-') # NEGATIVE TWEETS
ax2.plot(x, y4, 'y-') # NEUTRAL TWEETS

# set legends
ax1.legend(["Trump's Approval Rating"], fc
ax2.legend(["Number of Positive Tweets", '

plt.text(14,80, """Weeks 15 and 16:
Talks of reopening the US""", fontsize = 1

plt.annotate("""Week 12 - 6.6 Million peop
stay at home orders announced""",
             xy = (12,15), xytext = (2,15)
             arrowprops=dict(facecolor='bl

plt.text(12,-150, """End of week 12:
Trump signs stimulus check bill""", fontsi


ax1.set_xlabel('Week of Year')
ax1.set_ylabel("Trump's Approval Rating",
ax2.set_ylabel("Number of tweets", color='

plt.show()
```



3.5.5 Number of Coronavirus Tweets by S

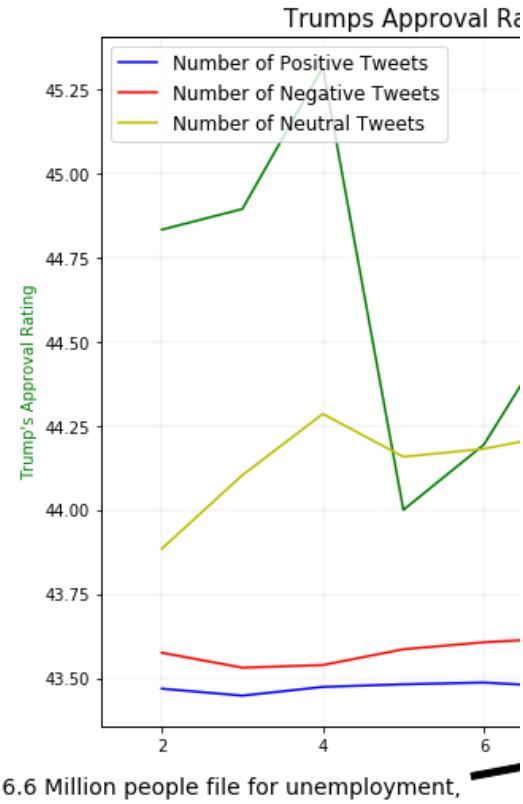
Let's take our positive, negative, and neutral tweets and

Contents

Contents

In [180]:

```
# import numpy as np
import matplotlib.pyplot as plt
x = corona_temp_x
y1 = corona_temp_y
y2 = corona_pos_df.groupby('week_bins')['s
y3 = corona_neg_df.groupby('week_bins')['s
y4 = corona_neut_df.groupby('week_bins')['

fig, ax1 = plt.subplots(figsize=(15,8))

ax1.set_title('Trumps Approval Rating vs.
ax1.grid(color='grey', linestyle='-', line

#duplicate ax1
ax2 = ax1.twinx()

#ax1 will plot Trump's approval rating, ax
ax1.plot(x, y1, 'g-')
ax2.plot(x, y2, 'b-') # POSITIVE TWEETS
ax2.plot(x, y3, 'r-') # NEGATIVE TWEETS
ax2.plot(x, y4, 'y-') # NEUTRAL TWEETS

# set legends
ax1.legend(["Trump's Approval Rating"], fc
ax2.legend(["Number of Positive Tweets", '

ax1.set_xlabel('Week of Year')
ax1.set_ylabel("Trump's Approval Rating",
ax2.set_ylabel("Number of tweets", color='



plt.text(14,115, """Weeks 15 and 16:
Talks of reopening the US""", fontsize = 1

plt.annotate("""Week 12 - 6.6 Million peop
stay at home orders announced""",
             xy = (12,15), xytext = (2,15)
             arrowprops=dict(facecolor='bl

plt.text(12,-150, """End of week 12:
Trump signs stimulus check bill""", fontsi



plt.show()
```



Week 12 - 6.6 Million people file for unemployment, stay at home orders announced

Further analysis of graph will be discussed in the "discus

4 Part 2: Predicting the 202(

4.1 Description Of The Data - Par

4.2 Scraper(s) Used

For this part of the project, we scraped for the same keyv users on current tweets.

Contents

Contents

In [181]:

```
from twython import TwythonStreamer
import csv
import json
import codecs
import time
from random import seed
from random import randint
# seed random number generator
seed(1)

tweets_filename = "twitter_output_Trump"

consumer_key = "ND4wRUZRSgRS1NBtb9vRbm96v"
consumer_secret = "z0zxb6QObq4AUU2bA8BBclr
access_token = "1250793114274598913-1mA5Lz
access_token_secret = "cqNiCEcr5OsbS5G1t3u

tweetabbr = []

# In this session we are using the Twitter
AllPDs = ['561106229', '34296669', '974277
          '35871927', '304847225']
# Filter out unwanted data for the CSV fil
def process_tweet(tweet):
    d = {}
    d['hashtags'] = [hashtag['text'] for h
    d['id'] = tweet['id']
    d['text'] = tweet['text']
    d['name'] = tweet['user']['name']
    d['user'] = tweet['user']['screen_name
    d['user_loc'] = tweet['user']['locatic
    d['user_desc'] = tweet['user']['descri
    d['user_followers'] = tweet['user']['f
    d['user_friends'] = tweet['user']['fri
    d['user_listed'] = tweet['user']['list
    d['user_created'] = tweet['user']['cre
    d['user_favs'] = tweet['user']['favour
    d['user_statuses'] = tweet['user']['st

    return d


# Create a class that inherits TwythonStre
class MyStreamer(TwythonStreamer):

    # Received data
    def on_success(self, data):

        # Save full JSON to file
        # TODO : save properly so we can l
        # A tweet JSON record per line
        with open(f'{tweets_filename}.jsor
            json.dump(data, jsonfile)
            jsonfile.write("\n")

        # Only save tweets in English
```

```
        if data['lang'] == 'en':
            tweet_data = process_tweet(dat
            self.save_to_csv(tweet_data)

    # Problem with the API
    def on_error(self, status_code, data):
        print(status_code, data)
        self.disconnect()

    # Save each tweet to csv file
    def save_to_csv(self, tweet):
        with open(f'{tweets_filename}.csv'
            writer = csv.writer(file)
            writer.writerow(list(tweet.val

while True:
    try:
        # Instantiate from our streaming c
        stream = MyStreamer(consumer_key,
                    access_token, access_t

        # Start the stream - this would cc
        # There are online tools to get th
        # stream.statuses.filter(follow=17

        # Start the stream - this would cc
        #stream.statuses.filter(follow=All

        # Start the stream - This stream l
        #stream.statuses.filter(track='@VA
        stream.statuses.filter(track='Trum
#track="Trump, Stimulus Check,stimulus che
    except (KeyboardInterrupt):
        print("Exiting")
        break
    except Exception as e:
        print("error - sleeping " + str(e)
        time.sleep(randint(30, 90)) #suspe
        continue
```

```
  File "<ipython-input-181-c91607ce482b>",
    stream.statuses.filter(track='Trump',
timuluscheck', '#money' )
                                         ^
SyntaxError: positional argument follows k
```

4.3 Data Description

Our second analysis involves data scraped from twitter A

- **We scraped the following tweet characteristics v**
  - Hashtags

- Tweet ID
- Tweet text
- User's display name
- Username
- User's location
- User's description
- Followers
- Friends
- Lists
- Account creation date
- Favorites
- User total tweets

- **We then added the following**:
  - filtered_text
  - retweet_flags
  - Polarity_Score
  - SubjectivityScore
  - sentimentLabel

4.3.1 Data Processing Tasks

For our second set of data, we will first have to perform s
saw how many neutral values there were, so we did a fu
user_description.

4.3.2 When And How Long You Scraped

For this part of the project, we scraped Twitter for about 8

4.4 Load in Data

In [182]:
```
import pandas as pd
import numpy as np

df= pd.read_csv("data_files/Group_2_Phase_
df.columns= ['Hashtags','ID','Tweet_Text',
```

4.4.1 Drop Duplicates

Contents

In [183]:
```
length  = len(df)
new_length = len(df.drop_duplicates())
print(f" The current length of our Datafra
print(f" The length of our Dataframe after
```

```
The current length of our Dataframe is 28
The length of our Dataframe after droppin
```

5 Simulation Data Cleaning/

Importing libraries and functions given to us from Profess

In [184]:
```
# Remove URLs and Stop Words
df['filtered_text'] = df.Tweet_Text.apply(

#retweet
df['retweet_flags'] = df.Tweet_Text.str.st

# TODO Add code analyse URLs, their domair
```

In [185]:
```
# deal with nots
!pip install TextBlob
```

```
Requirement already satisfied: TextBlob in
Requirement already satisfied: nltk>=3.1 i
5)
Requirement already satisfied: six in /usr
(1.14.0)
```

In [186]:
```
from textblob import TextBlob
df['PolarityScore'] = df.filtered_text.app
df['SubjectivityScore'] = df.filtered_text
```

In [187]:
```
def sentiment_bins(data):
    if data <= -0.5:
        grouping = 'Strong-Negative'
    elif data > -0.5 and data < 0.0:
        grouping = 'Mild-Negative'
    elif data > 0.0 and data < 0.5:
        grouping = 'Mild Positive'
    elif data >=0.5:
        grouping = 'Strong-Positive'
    else:
        grouping = 'Neutral'
    return grouping

df['sentimentLabel'] = df['PolarityScore']
```

In [188]:
```
# user locations
df.User_Location.value_counts(dropna = Tru
```

Out[188]:
```
United States                          716
USA                                    253
California, USA                        253
Texas, USA                             172
India                                  148
                                      ...
Time is a man made concept!!!            1
From NYC. Live in La Jolla, CA           1
Arizona USA                              1
Paddington, London                       1
society                                  1
Name: User_Location, Length: 8696, dtype:
```

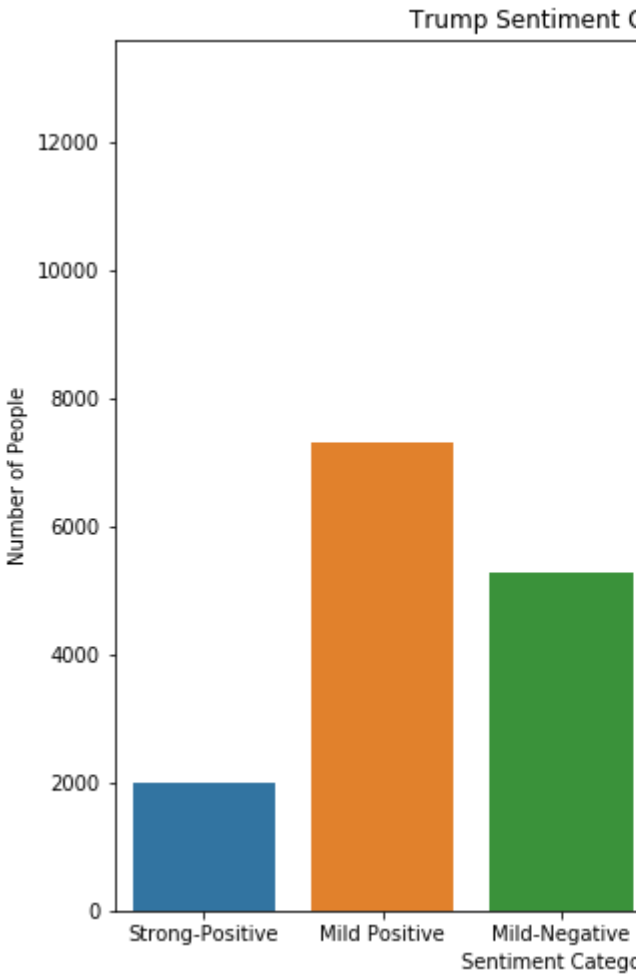
6 Visual EDA- Part 2

6.1 Further Filtering Neutral Valu

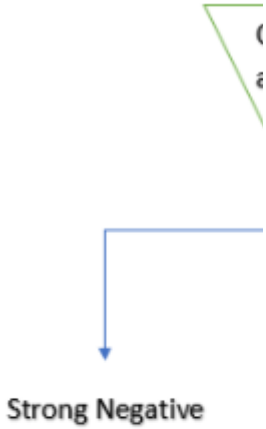
Contents

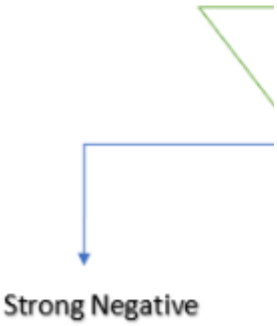
In [189]:
```
plt.figure(figsize = (8,8))
ax = sns.countplot(x = "sentimentLabel", d
ax.set(title = "Trump Sentiment Counts", x
plt.text(4.7, 10000,"We clearly see that w
```

Out[189]:
```
Text(4.7, 10000, 'We clearly see that we h
r down shortly')
```



From a basic level, this is how we are filtering neutral val





6.2 Creating the Neutral Datafram

In [190]:
```
Neutral= df[df['sentimentLabel'].str.conta
#Creating a datafrmae of all the Neutral t
Neutral['User_Description'] = Neutral.User
```

```
/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
  This is separate from the ipykernel pack
```

So here we are taking our dataframe of Neutral tweets ar

further filter them to be able to get rid of the number of N

After looking at some of the User_Descriptions we found political affiliation (ex. Patriot Pro-Trump Christian Englis

In [191]:

```
Neutral['PolarityScore'] = Neutral.User_De
Neutral['SubjectivityScore'] = Neutral.Use
#Actually creating the Polarity and Subjec
Neutral['sentimentLabel'] = Neutral['Polar
#Now putting them into bins
```

```
/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
  """Entry point for launching an IPython
/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)

/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
  after removing the cwd from sys.path.
```

We now have the neutral values with an assigned sentim

In [192]:
```
Neutral.head()
```

Out[192]:

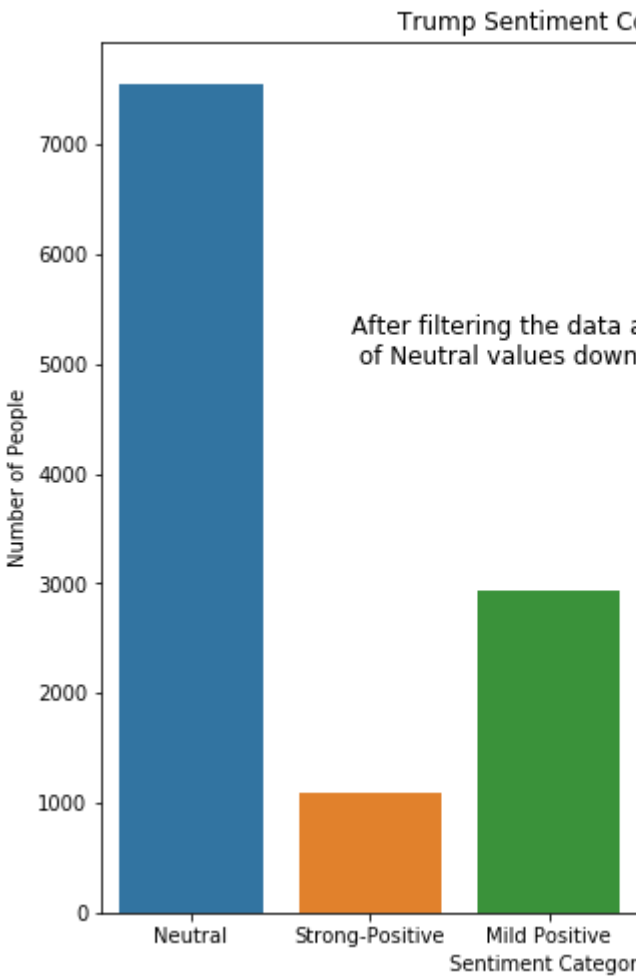
	Hashtags	ID	Tweet_Te
3	[]	1254972505367994368	F @RedWingGrip BREAKIN President Trun ha
4	[]	1254972505497808897	RT @mog754 Health exper from the Trum Ob
5	[]	1254972505544105984	F @costareport “U.S. intelligen agencies
9	[]	1254972505661616129	F @soledadobrie That ‘someboc is you, bab
10	[]	1254972505837764610	Trump Blam Plummeting P Numbers People

Contents

Contents

In [193]:
```
plt.figure(figsize = (8,8))
ax = sns.countplot(x = "sentimentLabel", d
ax.set(title = "Trump Sentiment Counts", x
plt.text(0.9,5000,"After filtering the dat
```

Out[193]:
```
Text(0.9, 5000, 'After filtering the data
40%')
```



Contents

7 EDA Part 2

7.1 2020 Election Simulation

This section includes an elaborate simulation of the 2020
We then looked at the outcomes of the previous election
was lower than 5% then we determined that the state wo
and belong to the other party in the upcoming election. T
race, and could ultimately be a 50/50 chance for each pa
small simulation of the actual election, and is determinig

In [194]:
```
topStates = pd.read_csv("data_files/Group_
topStates #A BLUE STATE= WON BY THE DEMOCR
```

Out[194]:

	State	Electoral Votes	Difference in % of V
0	California	53	28
1	Texas	36	9
2	New York	27	21
3	Florida	27	1
4	Pennsylvania	18	1
5	Illinois	18	
6	Ohio	16	8
7	Michigan	14	0
8	Georgia	14	5
9	North Carolina	13	3

7.2 How the simulation actually w

In summary, the simulation will follow these steps to dete

- PLEASE KEEP IN MIND THAT THIS IS FROM TRU
  Trump)
- Add up all of the positive and negative tweets for the
  label.
- Do the exact same for the original DataFrame to get
- Take the number of remaining Neutral values for eac

- It is now determined if a state is a swing state or not
- If the state is not a swing state, then the remaining n election. But if the state is indeed a swing state, it er chance of winning one of those states, a random nu are given to Joe Biden, but if that number comes out
- Then based off of that determination, we calculate w

Below is a screenshot for just one state, this then replica

```
Neutral['User_Location'] = df.User_Location.filln
#Creating positive and negative counts for the fi
FLN = Neutral[Neutral['User_Location'].str.contai
FLN1= FLN[FLN['sentimentLabel'].str.contains("Mil
FLN2= FLN[FLN['sentimentLabel'].str.contains("Str
FLN3 = FLN[FLN['sentimentLabel'].str.contains("Mi
FLN4 = FLN[FLN['sentimentLabel'].str.contains("St
NegativeN = len(FLN1)+len(FLN2)
PositiveN = len(FLN3)+len(FLN4)
#Now creating positive and negative counts for th
df['User_Location'] = df.User_Location.fillna('')
FL = df[df['User_Location'].str.contains("FLORIDA
FL1= FL[FL['sentimentLabel'].str.contains("Mild-N
FL2= FL[FL['sentimentLabel'].str.contains("Strong
FL3 = FL[FL['sentimentLabel'].str.contains("Mild-
FL4 = FL[FL['sentimentLabel'].str.contains("Stron
Negative = len(FL1)+len(FL2)
Positive = len(FL3)+len(FL4)
Only_Neutral = FLN[FLN['sentimentLabel'].str.cont
Neutral_Final = len(Only_Neutral)
#This for loop is entered becuase the state is a
#from the election before
import numpy as np
random = np.random.randint(low = 1, high = 10)
if (random%2) == 0:
        Negative_Final = Negative + NegativeN + N
        Positive_Final = Positive + PositiveN
else:
    Positive_Final = Positive + PositiveN + Neutr
    Negative_Final = Negative + NegativeN
if Negative_Final > Positive_Final:
    Biden_vote_count = Biden_vote_count+27
    print("Biden Wins Florida")
    print(f"Biden Vote Count: {Biden_vote_count}"
    print('-'*50)
#Updating the list to later analyze in our final
    trump_win.append(0)
    biden_win.append(1)
else:
    Trump_vote_count = Trump_vote_count+27
    print("Trump Wins Florida")
    print(f"Trump Vote Count: {Trump_vote_count}"
    print('-'*50)
    trump_win.append(1)
    biden_win.append(0)
```

7.3 The Actual Simulation

Please note this this is a live simulation, so each time this
output we got from this simulation.

Contents

Contents

In [195]:

```
Biden_vote_count = 0
Trump_vote_count = 0
trump_win = []
biden_win = []
# CALIFORNIA-----------------------------
Neutral['User_Location'] = df.User_Locatic
CALN = Neutral[Neutral['User_Location'].st
CALN1= CALN[CALN['sentimentLabel'].str.cor
CALN2= CALN[CALN['sentimentLabel'].str.cor
CALN3 = CALN[CALN['sentimentLabel'].str.cc
CALN4 = CALN[CALN['sentimentLabel'].str.cc
NegativeN = len(CALN1)+len(CALN2)
PositiveN = len(CALN3)+len(CALN4)
df['User_Location'] = df.User_Location.fil
CA = df[df['User_Location'].str.contains('
CAL1= CA[CA['sentimentLabel'].str.contains
CAL2= CA[CA['sentimentLabel'].str.contains
CAL3 = CA[CA['sentimentLabel'].str.contair
CAL4 = CA[CA['sentimentLabel'].str.contair
Negative = len(CAL1)+len(CAL2)
Positive = len(CAL3)+len(CAL4)
Only_Neutral = CALN[CALN['sentimentLabel']
Neutral_Final = len(Only_Neutral)
Positive_Final = Positive + PositiveN
Negative_Final = Negative + NegativeN+Neut
if Negative_Final > Positive_Final:
    Biden_vote_count = Biden_vote_count+53
    print("Biden Wins California")
    print(f"Biden Vote Count: {Biden_vote_
    print('-'*50)
    trump_win.append(0)
    biden_win.append(1)
else:
    Trump_vote_count = Trump_vote_count+53
    print("Trump Wins California")
    print(f"Trump Vote Count: {Trump_vote_
    trump_win.append(1)
    biden_win.append(0)
#TEXAS----------------------------------
Neutral['User_Location'] = df.User_Locatic
TXN = Neutral[Neutral['User_Location'].str
TXN1= TXN[TXN['sentimentLabel'].str.contai
TXN2= TXN[TXN['sentimentLabel'].str.contai
TXN3 = TXN[TXN['sentimentLabel'].str.conta
TXN4 = TXN[TXN['sentimentLabel'].str.conta
NegativeN = len(TXN1)+len(TXN2)
PositiveN = len(TXN3)+len(TXN4)
df['User_Location'] = df.User_Location.fil
Neutral['User_Location'] = df.User_Locatic
TX = df[df['User_Location'].str.contains('
TX1= TX[TX['sentimentLabel'].str.contains(
TX2= TX[TX['sentimentLabel'].str.contains(
TX3 = TX[TX['sentimentLabel'].str.contains
TX4 = TX[TX['sentimentLabel'].str.contains
Negative = len(TX1)+len(TX2)
Positive = len(TX3)+len(TX4)
```

Contents


```
Only_Neutral = TXN[TXN['sentimentLabel'].s
Neutral_Final = len(Only_Neutral)
Positive_Final = Positive + PositiveN+Neut
Negative_Final = Negative + NegativeN
Neutral_Final = len(Only_Neutral)
if Negative_Final > Positive_Final:
    Biden_vote_count = Biden_vote_count+36
    print("Biden Wins Texas")
    print(f"Biden Vote Count: {Biden_vote_
    print('-'*50)
    trump_win.append(0)
    biden_win.append(1)
else:
    Trump_vote_count = Trump_vote_count+36
    print("Trump Wins Texas")
    print(f"Trump Vote Count: {Trump_vote_
    print('-'*50)
    trump_win.append(1)
    biden_win.append(0)
# NEW YORK--------------------------------
Neutral['User_Location'] = df.User_Locatic
NYN = Neutral[Neutral['User_Location'].str
NYN1= NYN[NYN['sentimentLabel'].str.contai
NYN2= NYN[NYN['sentimentLabel'].str.contai
NYN3 = NYN[NYN['sentimentLabel'].str.conta
NYN4 = NYN[NYN['sentimentLabel'].str.conta
NegativeN = len(NYN1)+len(NYN2)
PositiveN = len(NYN3)+len(NYN4)
df['User_Location'] = df.User_Location.fil
NY = df[df['User_Location'].str.contains('
NY1= NY[NY['sentimentLabel'].str.contains(
NY2= NY[NY['sentimentLabel'].str.contains(
NY3 = NY[NY['sentimentLabel'].str.contains
NY4 = NY[NY['sentimentLabel'].str.contains
Negative = len(NY1)+len(NY2)
Positive = len(NY3)+len(NY4)
Only_Neutral = NYN[NYN['sentimentLabel'].s
Neutral_Final = len(Only_Neutral)
Positive_Final = Positive + PositiveN
Negative_Final = Negative + NegativeN+Neut
if Negative_Final > Positive_Final:
    Biden_vote_count = Biden_vote_count+27
    print("Biden Wins New York")
    print(f"Biden Vote Count: {Biden_vote_
    print('-'*50)
    trump_win.append(0)
    biden_win.append(1)
else:
    Trump_vote_count = Trump_vote_count+27
    print("Trump Wins New York")
    print(f"Trump Vote Count: {Trump_vote_
    print('-'*50)
    trump_win.append(1)
    biden_win.append(0)
# FLORIDA---------------------------------
Neutral['User_Location'] = df.User_Locatic
FLN = Neutral[Neutral['User_Location'].str
```

Contents


```
FLN1= FLN[FLN['sentimentLabel'].str.contai
FLN2= FLN[FLN['sentimentLabel'].str.contai
FLN3 = FLN[FLN['sentimentLabel'].str.conta
FLN4 = FLN[FLN['sentimentLabel'].str.conta
NegativeN = len(FLN1)+len(FLN2)
PositiveN = len(FLN3)+len(FLN4)
df['User_Location'] = df.User_Location.fil
FL = df[df['User_Location'].str.contains('
FL1= FL[FL['sentimentLabel'].str.contains(
FL2= FL[FL['sentimentLabel'].str.contains(
FL3 = FL[FL['sentimentLabel'].str.contains
FL4 = FL[FL['sentimentLabel'].str.contains
Negative = len(FL1)+len(FL2)
Positive = len(FL3)+len(FL4)
Only_Neutral = FLN[FLN['sentimentLabel'].s
Neutral_Final = len(Only_Neutral)
import numpy as np
random = np.random.randint(low = 1, high =
if (random%2) == 0:
        Negative_Final = Negative + Negati
        Positive_Final = Positive + Positi
else:
    Positive_Final = Positive + PositiveN
    Negative_Final = Negative + NegativeN
if Negative_Final > Positive_Final:
    Biden_vote_count = Biden_vote_count+27
    print("Biden Wins Florida")
    print(f"Biden Vote Count: {Biden_vote_
    print('-'*50)
    trump_win.append(0)
    biden_win.append(1)
else:
    Trump_vote_count = Trump_vote_count+27
    print("Trump Wins Florida")
    print(f"Trump Vote Count: {Trump_vote_
    print('-'*50)
    trump_win.append(1)
    biden_win.append(0)
#Pennsylvania-----------------------------
Neutral['User_Location'] = df.User_Locatic
PNN = Neutral[Neutral['User_Location'].str
PNN1= PNN[PNN['sentimentLabel'].str.contai
PNN2= PNN[PNN['sentimentLabel'].str.contai
PNN3 = PNN[PNN['sentimentLabel'].str.conta
PNN4 = PNN[PNN['sentimentLabel'].str.conta
NegativeN = len(PNN1)+len(PNN2)
PositiveN = len(PNN3)+len(PNN4)
df['User_Location'] = df.User_Location.fil
PN = df[df['User_Location'].str.contains('
PN1= PN[PN['sentimentLabel'].str.contains(
PN2= PN[PN['sentimentLabel'].str.contains(
PN3 = PN[PN['sentimentLabel'].str.contains
PN4 = PN[PN['sentimentLabel'].str.contains
Negative = len(PN1)+len(PN2)
Positive = len(PN3)+len(PN4)
Only_Neutral = PN[PN['sentimentLabel'].str
Neutral_Final = len(Only_Neutral)
```

Contents


```
import numpy as np
random = np.random.randint(low = 1, high =
if (random%2) == 0:
        Negative_Final = Negative + Negati
        Positive_Final = Positive + Positi
else:
    Positive_Final = Positive + PositiveN
    Negative_Final = Negative + NegativeN
if Negative_Final > Positive_Final:
    Biden_vote_count = Biden_vote_count+18
    print("Biden Wins Pennsylvania")
    print(f"Biden Vote Count: {Biden_vote_
    print('-'*50)
    trump_win.append(0)
    biden_win.append(1)
else:
    Trump_vote_count = Trump_vote_count+18
    print("Trump Wins Pennsylvania")
    print(f"Trump Vote Count: {Trump_vote_
    print('-'*50)
    trump_win.append(1)
    biden_win.append(0)
#ILLINOIS----------------------------------
Neutral['User_Location'] = df.User_Locatic
ILN = Neutral[Neutral['User_Location'].str
ILN1= ILN[ILN['sentimentLabel'].str.contai
ILN2= ILN[ILN['sentimentLabel'].str.contai
ILN3 = ILN[ILN['sentimentLabel'].str.conta
ILN4 = ILN[ILN['sentimentLabel'].str.conta
NegativeN = len(ILN1)+len(ILN2)
PositiveN = len(ILN3)+len(ILN4)
df['User_Location'] = df.User_Location.fil
IL = df[df['User_Location'].str.contains('
IL1= IL[IL['sentimentLabel'].str.contains(
IL2= IL[IL['sentimentLabel'].str.contains(
IL3 = IL[IL['sentimentLabel'].str.contains
IL4 = IL[IL['sentimentLabel'].str.contains
Negative = len(IL1)+len(IL2)
Positive = len(IL3)+len(IL4)
Only_Neutral = ILN[ILN['sentimentLabel'].s
Neutral_Final = len(Only_Neutral)
Positive_Final = Positive + PositiveN
Negative_Final = Negative + NegativeN+Neut
if Negative_Final > Positive_Final:
    Biden_vote_count = Biden_vote_count+18
    print("Biden Wins Illinois")
    print(f"Biden Vote Count: {Biden_vote_
    print('-'*50)
    trump_win.append(0)
    biden_win.append(1)
else:
    Trump_vote_count = Trump_vote_count+18
    print("Trump Wins Illinois")
    print(f"Trump Vote Count: {Trump_vote_
    trump_win.append(1)
    biden_win.append(0)
#OHIO--------------------------------------
```

Contents


```
Neutral['User_Location'] = df.User_Locatic
OHN = Neutral[Neutral['User_Location'].str
OHN1= OHN[OHN['sentimentLabel'].str.contai
OHN2= OHN[OHN['sentimentLabel'].str.contai
OHN3 = OHN[OHN['sentimentLabel'].str.conta
OHN4 = OHN[OHN['sentimentLabel'].str.conta
NegativeN = len(OHN1)+len(OHN2)
PositiveN = len(OHN3)+len(OHN4)
df['User_Location'] = df.User_Location.fil
OH = df[df['User_Location'].str.contains('
OH1= OH[OH['sentimentLabel'].str.contains(
OH2= OH[OH['sentimentLabel'].str.contains(
OH3 = OH[OH['sentimentLabel'].str.contains
OH4 = OH[OH['sentimentLabel'].str.contains
Negative = len(OH1)+len(OH2)
Positive = len(OH3)+len(OH4)
Only_Neutral = OHN[OHN['sentimentLabel'].s
Neutral_Final = len(Only_Neutral)
Positive_Final = Positive + PositiveN+Neut
Negative_Final = Negative + NegativeN
if Negative_Final > Positive_Final:
    Biden_vote_count = Biden_vote_count+16
    print("Biden Wins Ohio")
    print(f"Biden Vote Count: {Biden_vote_
    print('-'*50)
    trump_win.append(0)
    biden_win.append(1)
else:
    Trump_vote_count = Trump_vote_count+16
    print("Trump Wins Ohio")
    print(f"Trump Vote Count: {Trump_vote_
    print('-'*50)
    trump_win.append(1)
    biden_win.append(0)
#MICHIGAN---------------------------------
Neutral['User_Location'] = df.User_Locatic
MIN = Neutral[Neutral['User_Location'].str
MIN1= MIN[MIN['sentimentLabel'].str.contai
MIN2= MIN[MIN['sentimentLabel'].str.contai
MIN3 = MIN[MIN['sentimentLabel'].str.conta
MIN4 = MIN[MIN['sentimentLabel'].str.conta
NegativeN = len(MIN1)+len(MIN2)
PositiveN = len(MIN3)+len(MIN4)
df['User_Location'] = df.User_Location.fil
MI = df[df['User_Location'].str.contains('
MI1= MI[MI['sentimentLabel'].str.contains(
MI2= MI[MI['sentimentLabel'].str.contains(
MI3 = MI[MI['sentimentLabel'].str.contains
MI4 = MI[MI['sentimentLabel'].str.contains
Negative = len(MI1)+len(MI2)
Positive = len(MI3)+len(MI4)
Only_Neutral = MIN[MIN['sentimentLabel'].s
Neutral_Final = len(Only_Neutral)
import numpy as np
random = np.random.randint(low = 1, high =
if (random%2) == 0:
        Negative_Final = Negative + Negati
```

Contents


```
        Positive_Final = Positive + Positi
else:
    Positive_Final = Positive + PositiveN
    Negative_Final = Negative + NegativeN
if Negative_Final > Positive_Final:
    Biden_vote_count = Biden_vote_count+14
    print("Biden Wins Michigan")
    print(f"Biden Vote Count: {Biden_vote_
    print('-'*50)
    trump_win.append(0)
    biden_win.append(1)
else:
    Trump_vote_count = Trump_vote_count+14
    print("Trump Wins Michigan")
    print(f"Trump Vote Count: {Trump_vote_
    print('-'*50)
    trump_win.append(1)
    biden_win.append(0)
#GEORGIA----------------------------------
Neutral['User_Location'] = df.User_Locatic
GAN = Neutral[Neutral['User_Location'].str
GAN1= GAN[GAN['sentimentLabel'].str.contai
GAN2= GAN[GAN['sentimentLabel'].str.contai
GAN3 = GAN[GAN['sentimentLabel'].str.conta
GAN4 = GAN[GAN['sentimentLabel'].str.conta
NegativeN = len(GAN1)+len(GAN2)
PositiveN = len(GAN3)+len(GAN4)
df['User_Location'] = df.User_Location.fil
GA = df[df['User_Location'].str.contains('
GA1= GA[GA['sentimentLabel'].str.contains(
GA2= GA[GA['sentimentLabel'].str.contains(
GA3 = GA[GA['sentimentLabel'].str.contains
GA4 = GA[GA['sentimentLabel'].str.contains
Negative = len(GA1)+len(GA2)
Positive = len(GA3)+len(GA4)
Only_Neutral = GAN[GAN['sentimentLabel'].s
Neutral_Final = len(Only_Neutral)
Positive_Final = Positive + PositiveN+Neut
Negative_Final = Negative + NegativeN

if Negative_Final > Positive_Final:
    Biden_vote_count = Biden_vote_count+14
    print("Biden Wins Georgia")
    print(f"Biden Vote Count: {Biden_vote_
    print('-'*50)
    trump_win.append(0)
    biden_win.append(1)

else:
    Trump_vote_count = Trump_vote_count+14
    print("Trump Wins Georgia")
    print(f"Trump Vote Count: {Trump_vote_
    print('-'*50)
    trump_win.append(1)
    biden_win.append(0)
#North Carolina---------------------------
Neutral['User_Location'] = df.User_Locatic
```

Contents


```
NCN = Neutral[Neutral['User_Location'].str
NCN1= NCN[NCN['sentimentLabel'].str.contai
NCN2= NCN[NCN['sentimentLabel'].str.contai
NCN3 = NCN[NCN['sentimentLabel'].str.conta
NCN4 = NCN[NCN['sentimentLabel'].str.conta
NegativeN = len(NCN1)+len(NCN2)
PositiveN = len(NCN3)+len(NCN4)
df['User_Location'] = df.User_Location.fil
NC = df[df['User_Location'].str.contains('
NC1= NC[NC['sentimentLabel'].str.contains(
NC2= NC[NC['sentimentLabel'].str.contains(
NC3 = NC[NC['sentimentLabel'].str.contains
NC4 = NC[NC['sentimentLabel'].str.contains
Negative = len(NC1)+len(NC2)
Positive = len(NC3)+len(NC4)
Only_Neutral = NCN[NCN['sentimentLabel'].s
Neutral_Final = len(Only_Neutral)
import numpy as np
random = np.random.randint(low = 1, high =
if (random%2) == 0:
        Negative_Final = Negative + Negati
        Positive_Final = Positive + Positi
else:
    Positive_Final = Positive + PositiveN
    Negative_Final = Negative + NegativeN
if Negative_Final > Positive_Final:
    Biden_vote_count = Biden_vote_count+13
    print("Biden Wins North Carolina")
    print(f"Biden Vote Count: {Biden_vote_
    print('-'*50)
    trump_win.append(0)
    biden_win.append(1)
else:
    Trump_vote_count = Trump_vote_count+13
    print("Trump Wins North Carolina")
    print(f"Trump Vote Count: {Trump_vote_
    print('-'*50)
    trump_win.append(1)
    biden_win.append(0)

#----------------------------------------
if Trump_vote_count > Biden_vote_count:
    print("  *****Trump wins the 2020 elec
else:
    print("  *****Biden wins the 2020 elec
print('-'*50)
print(f"Trump's final score was {Trump_vot
print(f"Biden's final score was {Biden_vot
print('-'*50)
```

```
/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
```

```
/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)

Biden Wins California
Biden Vote Count: 53
------------------------------------------
Trump Wins Texas
Trump Vote Count: 36
------------------------------------------
Biden Wins New York
Biden Vote Count: 80
------------------------------------------
Trump Wins Florida
Trump Vote Count: 63
------------------------------------------

/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
/usr/local/lib/python3.6/site-packages/ipy
```

Contents

```
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)

Biden Wins Pennsylvania
Biden Vote Count: 98
----------------------------------------
Biden Wins Illinois
Biden Vote Count: 116
----------------------------------------
Trump Wins Ohio
Trump Vote Count: 79
----------------------------------------
Biden Wins Michigan
Biden Vote Count: 130
----------------------------------------
Trump Wins Georgia
Trump Vote Count: 93
----------------------------------------
Trump Wins North Carolina
Trump Vote Count: 106
----------------------------------------
  *****Biden wins the 2020 election*****
----------------------------------------
Trump's final score was 106
Biden's final score was 130
----------------------------------------

/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
/usr/local/lib/python3.6/site-packages/ipy
A value is trying to be set on a copy of a
Try using .loc[row_indexer,col_indexer] =

See the caveats in the documentation: http
ml#returning-a-view-versus-a-copy (https:/
returning-a-view-versus-a-copy)
```

7.4 Final DataFrame Of The Resu

In [196]:
```
topStates = pd.read_csv("data_files/Group_
topStates
topStates["Trump's States"] = trump_win
topStates["Trump's States"].replace(to_rep
topStates["Trump's States"].replace(to_rep
topStates["Biden's States"] = biden_win
topStates["Biden's States"].replace(to_rep
topStates["Biden's States"].replace(to_rep
topStates
```

Out[196]:

	State	Electoral Votes	Difference in % of V
0	California	53	28
1	Texas	36	9
2	New York	27	21
3	Florida	27	1
4	Pennsylvania	18	1
5	Illinois	18	
6	Ohio	16	8
7	Michigan	14	0
8	Georgia	14	5
9	North Carolina	13	3

7.5 Geo-Map

In [197]:
```
!pip install geopy
import geopy
from geopy.geocoders import Nominatim
from geopy.geocoders import Nominatim
from mpl_toolkits.basemap import Basemap
nom = Nominatim()
```

```
Requirement already satisfied: geopy in /u
Requirement already satisfied: geographicl
py) (1.50)

/usr/local/lib/python3.6/site-packages/ipy
the default "geopy/1.21.0" `user_agent` is
perations.osmfoundation.org/policies/nomin
m/) and may possibly cause 403 and 429 HTT
(user_agent="my-application")` or by overr
t_user_agent = "my-application"`. In geopy
```

Contents

In [198]:

```
location = df["User_Location"].value_count
new_loc = location.to_string()
location #Taking the frequency of each loc
locc = pd.DataFrame(location)
locc
e = df.User_Location.dropna()
ee = pd.DataFrame(e)
zg = ee.User_Location.value_counts().head(
ZG = zg.index
ZLG= pd.DataFrame(ZG)
ZLG.columns = ["Location"]
ZLG
e = df.User_Location.dropna()
ee = pd.DataFrame(e)
zg = ee.User_Location.value_counts().head(
count = pd.DataFrame(zg)
count
ZLG["Coordinates"] = ZLG["Location"].apply
ZLG.dropna()
```

Out[198]:

	Location	
1	United States	(United States
2	USA	(United States
3	California, USA	(California, Unite
4	Texas, USA	(Texas, United Sta
5	India	(भारत - Inc
...	...	
295	Islamabad, Pakistan	(اسلام آباد, A
296	Amsterdam, The Netherlands	(Amsterdam, Noord-H
297	Bay Area, CA	(San Francisco Bay
298	Montréal, Québec	(Montréal, Agglomé
299	Connecticut	(Connecticut, Unit

297 rows × 2 columns

In [199]:

```
ZLG["Latitude"] = ZLG["Coordinates"].apply
ZLG["Longitude"] = ZLG["Coordinates"].appl
Lat = ZLG["Latitude"]
Long = ZLG["Longitude"]
import numpy as np # linear algebra
import pandas as pd # data processing, CSV
import seaborn as sns #Statistical Data Vi
import matplotlib.pyplot as plt
%matplotlib inline
```

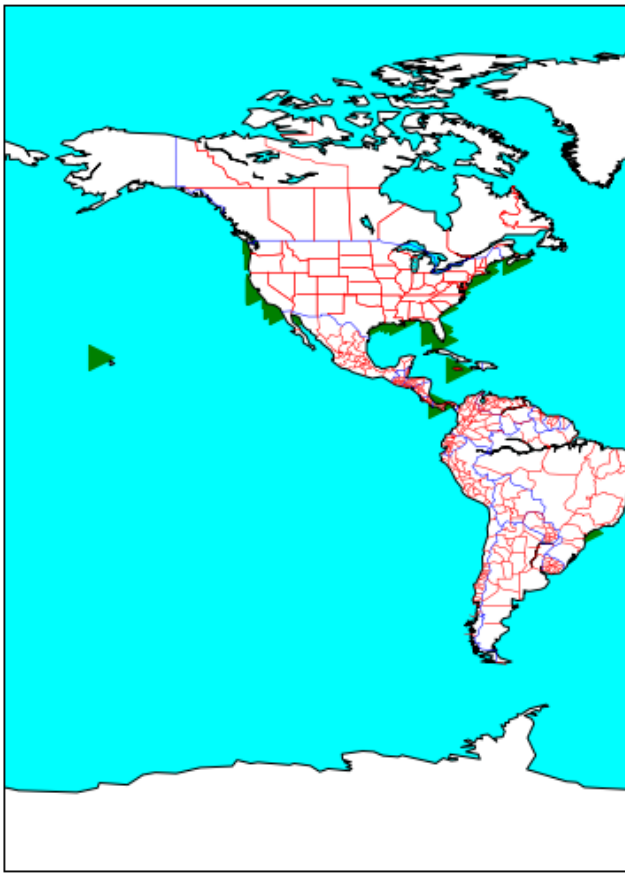
Contents

In [200]:

```
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(20,9))
m = Basemap(projection='gall',
            resolution = 'c')
m.drawcoastlines()
m.drawcountries(color= 'blue')
m.drawstates(color= 'red')
m.drawmapboundary(fill_color='aqua')
m.fillcontinents(color='white',lake_color=
plt.title("Tweet Location",fontsize = 30)
Lat = list(ZLG["Latitude"])
Long = list(ZLG["Longitude"])
m.scatter(Long,Lat,latlon=True,color="gree
plt.show()
```

```
/usr/local/lib/python3.6/site-packages/mpl
ue encountered in greater
  lonsin = np.where(lonsin > lon_0+180, lc
/usr/local/lib/python3.6/site-packages/mpl
ue encountered in less
  lonsin = np.where(lonsin < lon_0-180, lc
/usr/local/lib/python3.6/site-packages/mpl
ue encountered in greater_equal
  itemindex = len(lonsin)-np.where(londiff
/usr/local/lib/python3.6/site-packages/mpl
ue encountered in less
  mask = np.logical_or(lonsin<lon_0-180,lc
/usr/local/lib/python3.6/site-packages/mpl
ue encountered in greater
  mask = np.logical_or(lonsin<lon_0-180,lc
```

T



8 Discussions

Starting off with this project, we ran into quite a few roadl
metric and the user location to test our second and third
ways to pull user info from our tweets and joining it to ou
direction. However, we were still very interested in our fir
we went ahead and worked on this part without followers
ratings by week (our third party data) and spent most of t
polarity scores, or a combination of the two. We were the
be discussed below.

Another issue we ran into had to do with the number of tv
per week for 17 weeks with 9 keywords, which gave us a
size in our virtual containers proved to be extremely inefe
keywords and scrape less tweets per week to ensure we

9 Conclusions

Let's explore how each hypothesis turned out

1) President Trump’s approval rating has increased with
by the US government. The more tweets involving stimul
the higher we believe Trump’s approval rating will be. Th

- Given that the TextBlob tool pooled most of our twee
negative sentiment tweets specifically had a correlat
seems as if the more tweets there were about stimu
Trump's approval rating was. They seemed to move
rating, but they were just correlated. We did some re
coronavirus actions and added them to our plots as
- Despite our issues with the sentiment tool, we were
about stimulus checks and Trump's approval rating.
(approval rating) seem to generally move together. T
coronavirus (shown in section 3.5.5) as there appea
inversely with approval rating in this figure.
- Given the results from the simulation, we conclude t
he is still winning his historically Republican states, b
saw in Part 1, it appears that his approval rating is o

2) Sentiment ratings for the tweets will show an increase

- We were able to explore this via sections 3.5.3, 3.5.

Contents