Dealing with Data: Spring 2016

Analysis of Public Sentiment on Twitter Versus Debate Performance of Top GOP Candidates

May 16, 2016

Team 2: Ankur Chaudhary, Christina Wei, Janul Hernandez, and Peter Meijer

Executive Summary

The 2016 GOP race has become one of the most talked about topics of the year, garnering the public's attention with the debates becoming a televised spectacle. Fox News' debate in Detroit is the highest viewed debate of the year at 16.9 million viewers. This is no surprise with the interest Donald Trump has gathered from the general public. Trump's brash manner and lack of civility has ran counter to traditional political campaigns. Our project tried to see if there was a correlation between how Trump using negative rhetoric during the debates helped his standing with the public in contrast to his rivals.

We took a sentiment analysis of all GOP candidates participating in the last three debates of the GOP nominee trail and compared it to Twitter's response of each candidate. To our surprise we found that Trump's debate performances were not as negative in sentiment as his running mates. Additionally, our expectation for positive public sentiment on Twitter was wrong as well. What we did find was that Trump dominated the digital space in sheer volume of tweets and re-tweets.

Background

Our hypothesis was that Trump using negative rhetoric on the debate stage helped him garner positive sentiment from the public on social media specifically Twitter. Our datasets included:

Debate transcripts from the Washington Post website:

1) Debate 1 (Houston, TX, 02/25/2016): https://www.washingtonpost.com/news/the-fix/wp/2016/02/25/the-cnntelemundo-republican-debate-transcript-annotated/

2) Debate 2 (Detroit, MI, 03/03/2016): https://www.washingtonpost.com/news/the-fix/wp/2016/03/03/the-fox-news-gop-debate-transcript-annotated/

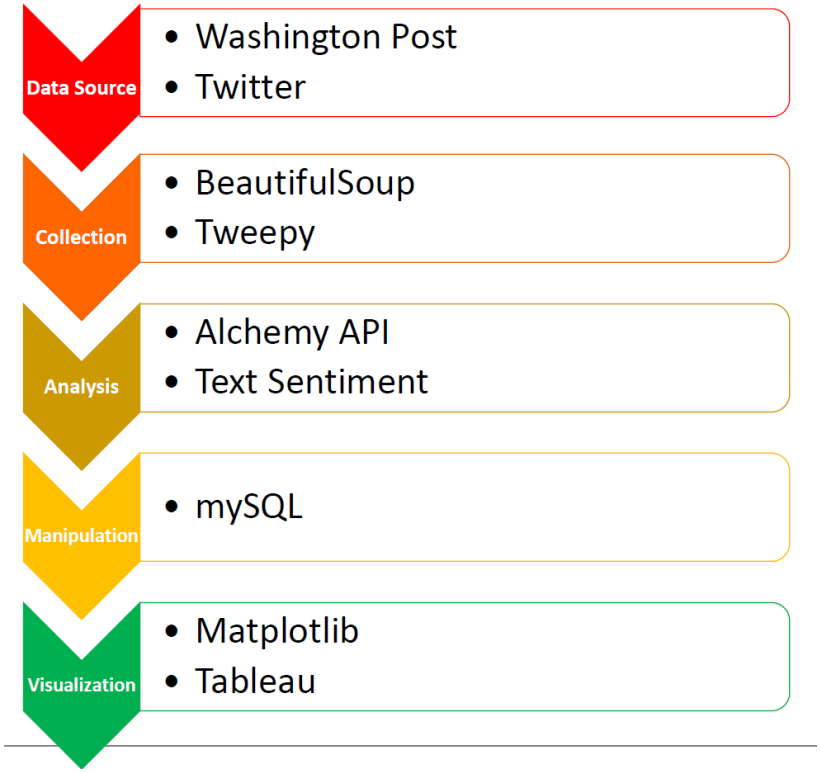
3) Debate 3 (Coral Gables,FL, 03/10/2016): https://www.washingtonpost.com/news/the-fix/wp/2016/03/10/the-cnn-miami-republican-debate-transcript-annotated/

Twitter data was extracted for different candidates using Tweepy API.

Project Description

The workflow process was as follows:

Figure 1: Workflow



1) We used transcripts from the Washington Post of three previous Republican Debates (Houston, Detroit, and Miami). With the help of Beautiful Soup (an HTML parser) we scraped these sites to build dictionaries of all words said by specific candidates during each debate.

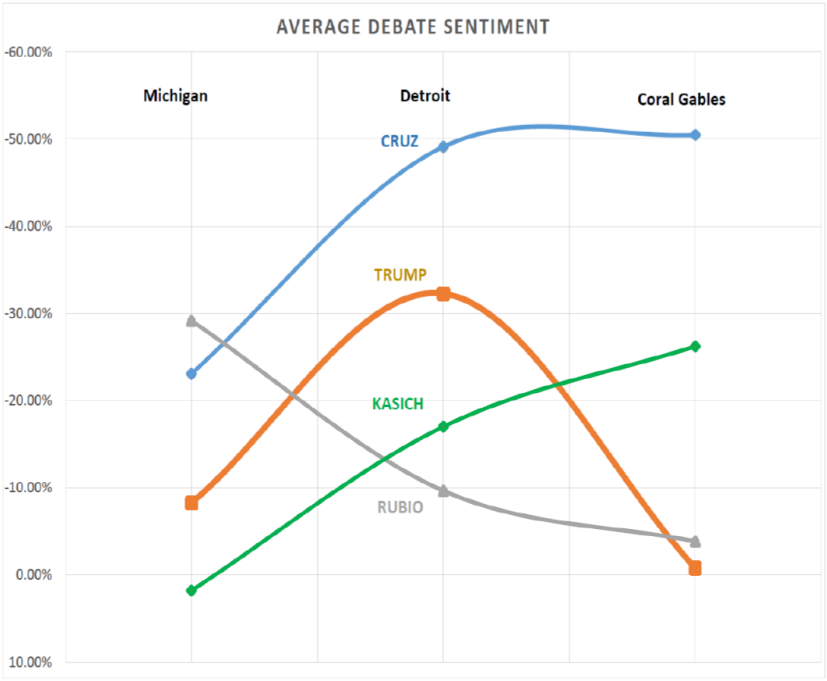
2) The dictionaries were placed in a relational database to find keywords and then sent through Alchemy Sentiment API to get a confidence and sentiment score.

3) We made a Twitter app and used its consumer and access keys to extract tweets using query words and hashtags. Twitter only allowed us to access the last 10 days of data free of charge. So we extracted relevant tweets using Tweepy cursor along with API.search. This data set was also sent through a Sentiment API.

4) After both data sets were collected we assigned a weighting factor and then combined using SQL. From the sets we performed data visualization using pandas (MATPLOTLIB) and Tableau to showcase the trends and our findings.

What did the Debate Data Say?

Figure 2: Debate Sentiment



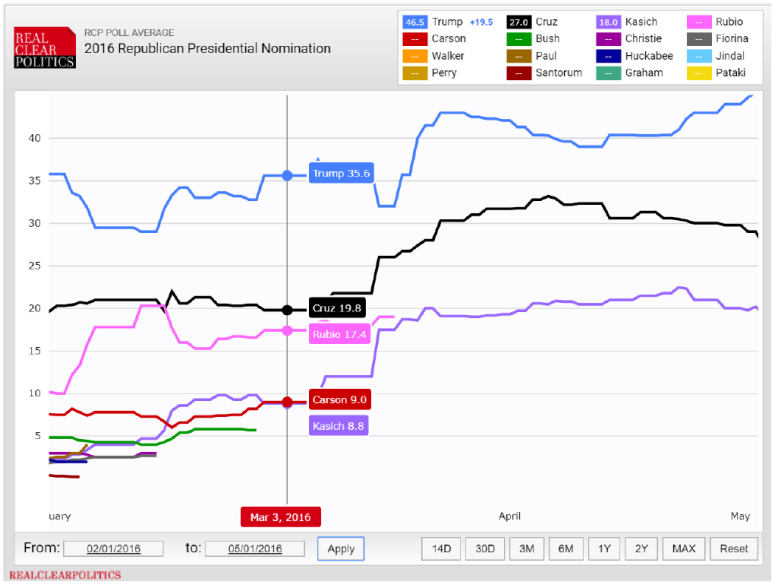
$$Sensitivity = \frac{\sum_{i \to I}(i_{pos} - i_{neg})}{\sum i}$$

where i= relevance

- Cruz and Kasich get more negative as they get desperate.
- Trump has an outlier negative performance in Detroit
- Rubio continues to be neutral

Figure 2 shows the average debate sentiment of all participating GOP candidates through the 3 debates. One thing that immediately is surprising is the fact that Trump was not the most negative in sentiment. We did notice an outlier performance in Detroit, this was due to the fact that he was being attacked by all 3 remaining GOP nominees. So much so that it dominated the headlines the next day (see figure 3).

Figure 3: Where were we before Detroit?



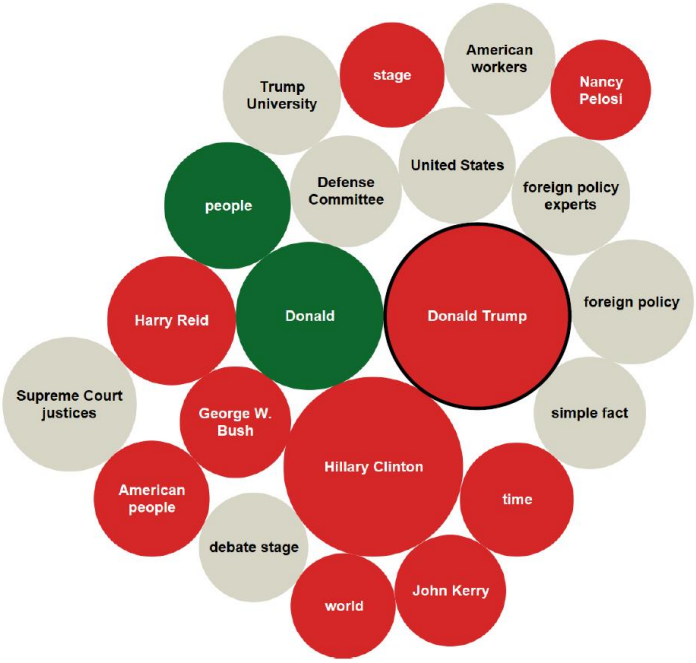
NEWS HEADLINES

- The Washington Post
  "Trump was attention of attacks at GOP debate"
- The New York Times
  "Taking On Donald Trump, at Their Own Peril"
- Rolling Stone
  "Watch every time FOX News attacked Trump at GOP debate."

Due to this increased attack it was no surprise that when we focused on keywords of the three candidates excluding Trump the word "Donald Trump" was the most relevant when aggregated and

had a negative sentiment. The focused attack was so much so that “Hillary Clinton” came in second for relevance and negative sentiment (*Figure 4*).

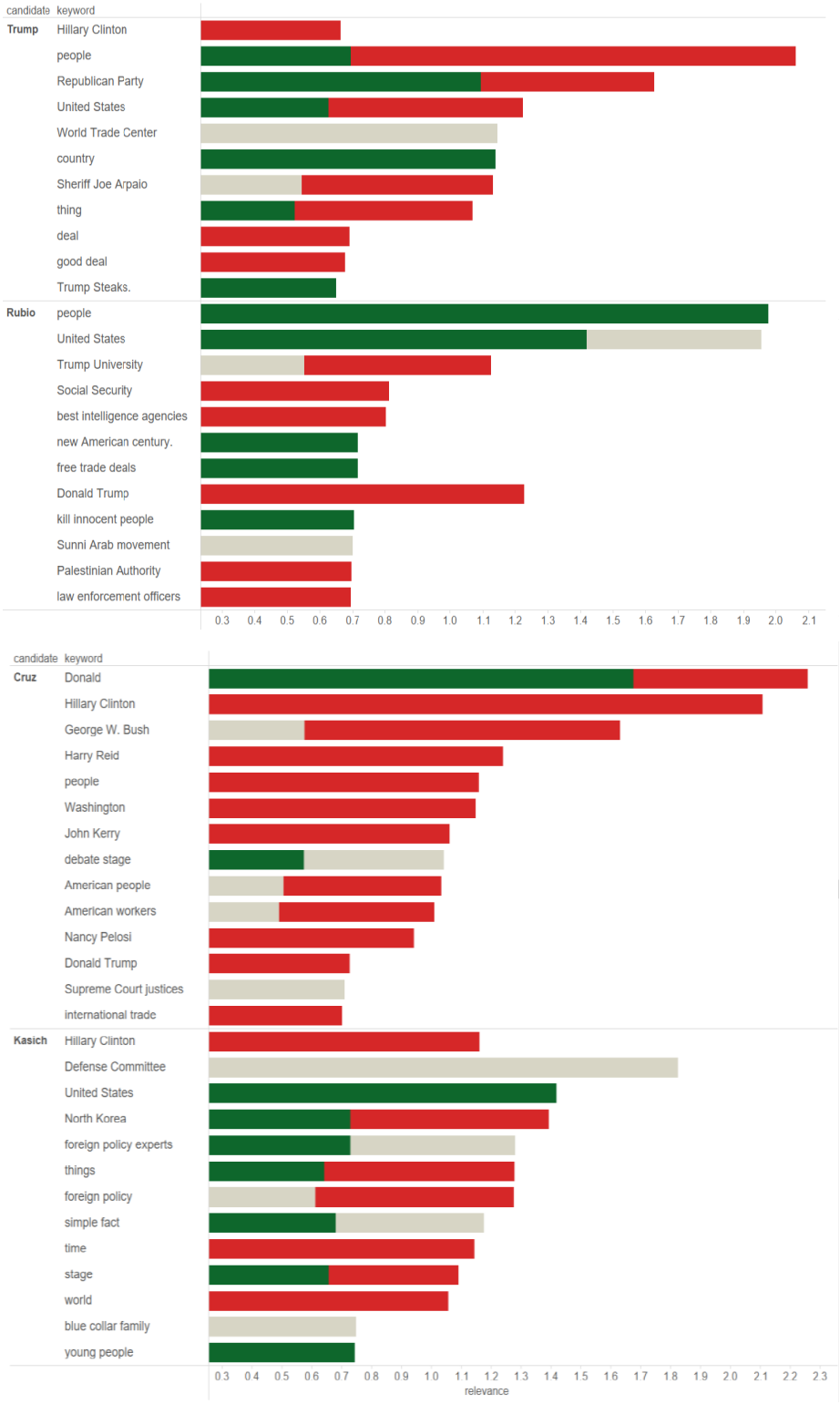
Figure 4: Keywords and sentiment during Detroit debate. (green=positive, red=negative, grey=neutral)



Trump proving he doesn’t have small hands

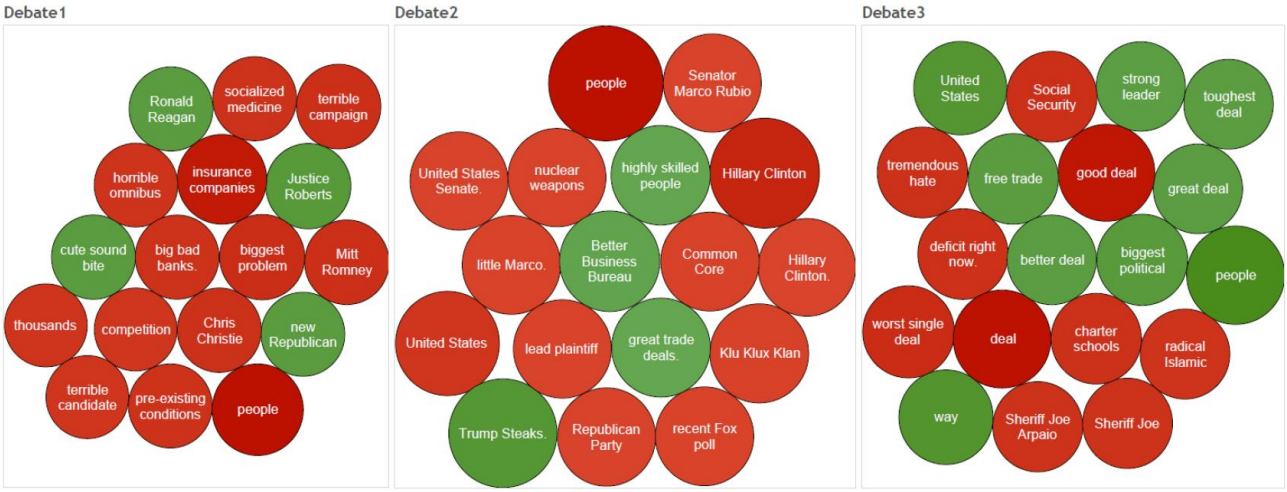
We then looked at keywords the candidates used through all three debates to see if they were continually attacking Trump and what we found further supported our theory that Detroit was an outlier. (*see figure 5 below)*

Figure 5: Keywords of each candidate during all 3 debates.



We finally tried to look through our data set to see what Trump was focusing on that has made him a runaway success. We found nothing astonishing other than the common theme of people not being happy and his general denouncement of the GOP establishment.

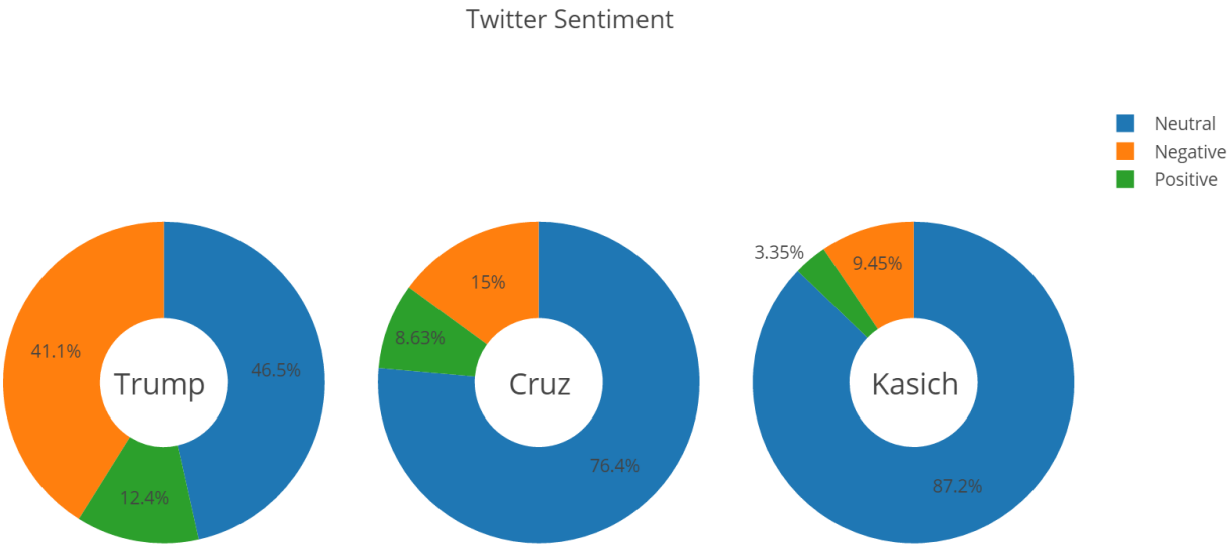
Figure 6: Trump through 3 debates.



What did the Twitter Data Say?

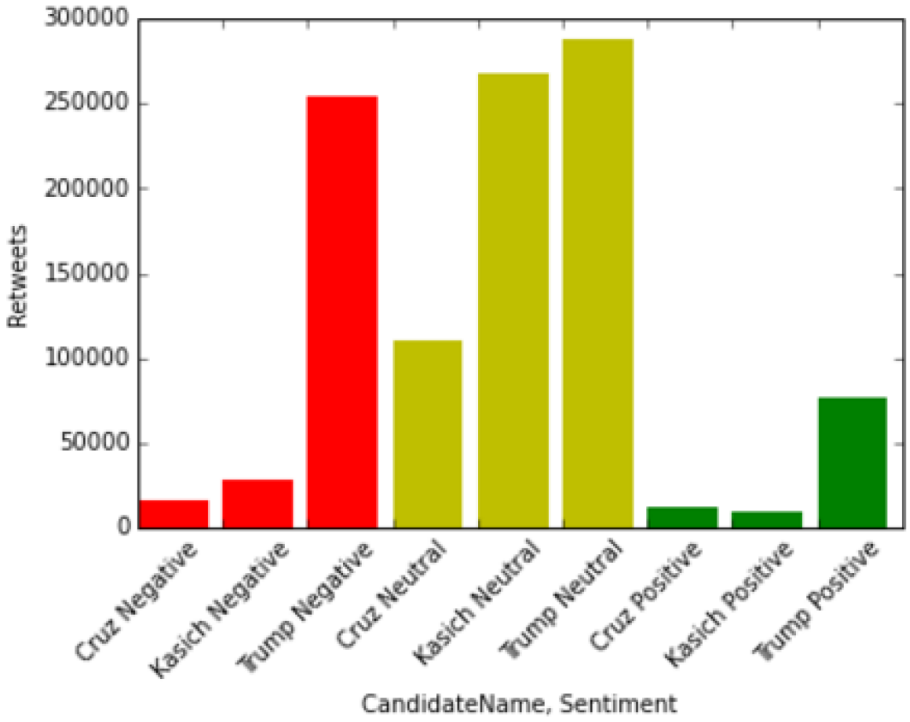
Due to the limitation on Twitter of 10 days to get historical data it was very hard to get a correlation between debate performance and public sentiment. What we were able to see was that Trump contrary to our hypothesis was not being received on Twitter in a positive light. Overall Twitter sentiment was dominated by neutral (46.5%) and negative (41.1%).

Figure 7: Twitter sentiment.



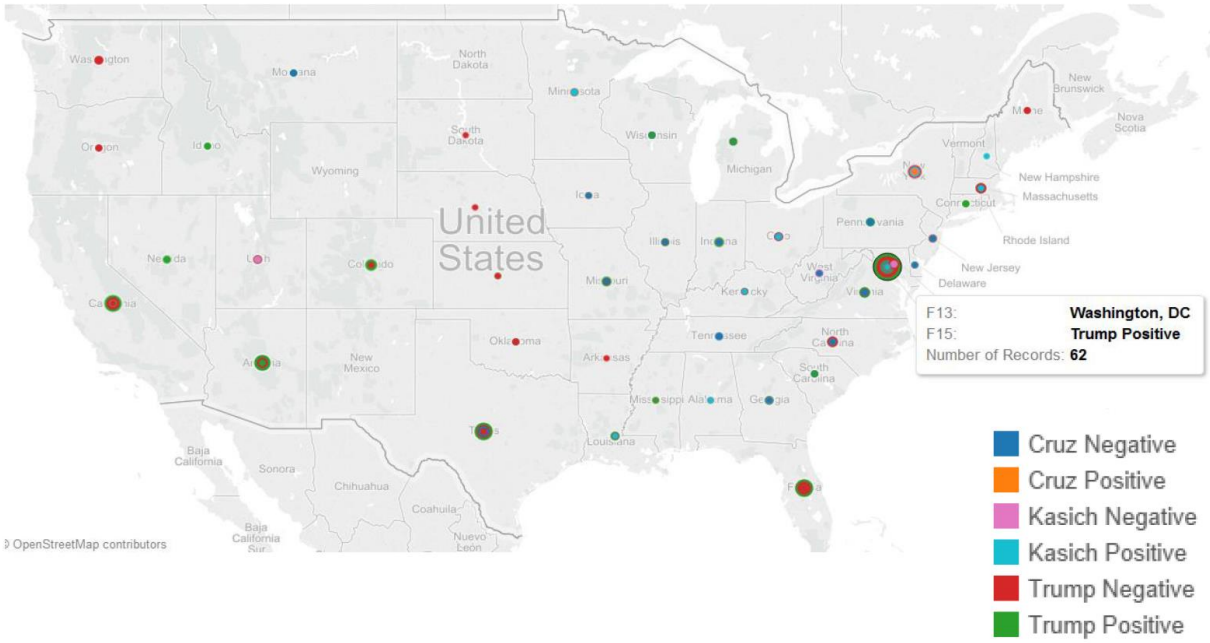
Where Trump did succeed was in the sheer volume of tweets and re-tweets he was getting, regardless of sentiment.

Figure 8: Tweet count by volume and sentiment.



We also looked at the distribution of tweets in the U.S. and found nothing telling.

Figure 9: Distribution of tweets.



Conclusion

Our hypothesis erred on two key aspects:

- Assumption: Trump most negative debate sentiment
- Assumption: Trump most positive social media support

Ultimately, Trump's debate sentiment *improved* over the course of the debates while his primary challenges Cruz and Kasich grew more negative. Trump generated the most social media, but also accounted for the **most negative** social media due to heightened general scrutiny of controversial comments

Future Opportunities

Our approach yielded additional research opportunities:

1. Using Twitter sentiment analysis and location data of tweets, examine correlation between change in tweet sentiment pre/post-debate and whether debate sentiment had an impact on polling data relative to state primary results.
2. For the debates between Donald Trump and the democratic candidate, apply a similar analysis over several debates to chart the impact of debate sentiment for a more constrained data sample.

Appendice - Debate Data Analysis (What do the debates Say?)

May 17, 2016

1 Setting up global variables

```
In [7]: !pip install beautifulsoup4
        from bs4 import BeautifulSoup
        import urllib
        import string
        import collections
        import requests
        import json

        debates= ['https://www.washingtonpost.com/news/the-fix/wp/2016/02/25/the-cnntelemundo-republica
                  'https://www.washingtonpost.com/news/the-fix/wp/2016/03/03/the-fox-news-gop-debate-tr
                  'https://www.washingtonpost.com/news/the-fix/wp/2016/03/10/the-cnn-miami-republican-d
```

```
Requirement already satisfied (use --upgrade to upgrade): beautifulsoup4 in /usr/local/lib/python2.7/di
```

2 Defining global functions.

```
In [8]: def openFile(url):
            #opening files
            file0 = BeautifulSoup(urllib.urlopen(url).read(),"lxml")
            return file0

        def cleanFile(file1):
            #function extracts text lines from html
            textLines = []

            paragraph = file1.find_all("p")
            for lines in paragraph:
                textLines.append(lines.get_text())

            return textLines

        def lines2words(file1):
            #function takes a list of strings and returns it as a string of words
            return_string = ''

            for line in file1:
                return_string += line

            words = return_string.split()

            return words
```

```
def getCandidate(candidate,file2):
    #function gets cadidate lines
    i = 0
    candidateLines = []

    for line in file2:
        if candidate in line:
            candidateLines.append(line)
            i += 1
        elif candidate in file2[i] :
            candidateLines.append(line)
            i += 1
        else:
            i += 1

    return candidateLines
```

3 Function to compile all keywords due to API 50 word limit

```
In [3]: def keywordAnalysis (textFile):
    #Calculating how many passes to send the API through according to dictionaryt length.
        multiple= 0
        multiple= len(textFile)/50
        remainder= len(textFile)%50
        if remainder > 0:
            multiple += 1

        temp=[]


        url = "http://access.alchemyapi.com/calls/text/TextGetRankedKeywords"
        api_key = '1f62c149d1b092011db73353df8215f6a5d3eb9e'
        headers = {"Accept": "application/json"}

        i=0
        while i != multiple:
            text = textFile[i*50:i*50+50]

            parameters = {
                'outputMode': 'json',
                'apikey' : api_key,
                'maxRetrieve' : 200,
                'sentiment': 1,
                'text': text}

            resp = requests.post(url, params=parameters, headers=headers)
            temp.append(json.loads(resp.text))
            i += 1

            #printing out relevenaces
            #j=0
```

```
            #while j != len(data['keywords']):
                #print "Keyword: ", data['keywords'][j]['text'], ", Relevance: ", data['keywords'][
                #j+=1

        return temp
```

4 Final Function.

You choose a candidate name and provide a source of text. The function parses the data and returns back keywords according to relevace.

In [ ]:
```
def getCandidateKeywords(candidateName,URL):

    File1 = openFile(URL)
    textFile = cleanFile(File1)

    candidateLines= getCandidate(candidateName, textFile)

    word_counts = collections.Counter(lines2words(candidateLines))

    return keywordAnalysis(candidateLines)

    #for word, count in word_counts.most_common():
        #print word, count


debate1={}
debate1['Trump']= getCandidateKeywords('TRUMP', debates[0])
debate1['Cruz']= getCandidateKeywords('CRUZ', debates[0])
debate1['Rubio']= getCandidateKeywords('RUBIO', debates[0])

debate2={}
debate2['Trump']= getCandidateKeywords('TRUMP', debates[1])
debate2['Cruz']= getCandidateKeywords('CRUZ', debates[1])
debate2['Rubio']= getCandidateKeywords('RUBIO', debates[1])

debate3={}
debate3['Trump']= getCandidateKeywords('TRUMP', debates[2])
debate3['Cruz']= getCandidateKeywords('CRUZ:', debates[2])
debate3['Rubio']= getCandidateKeywords('RUBIO', debates[2])

#for line in debate1['Cruz']:
#    i = 0
#    while i != len(line['keywords']):
#        print '\n' , "Keyword: ", line['keywords'][i]['text'], ", Relevance: ", line['keywords
#        print "Sentiment: ", line['keywords'][i]['sentiment']
#        i+=1

#i=0
#for line in info:
#    print info[i], '\n'
#    i += 1
```

5 Creating SQL Tables

```
In [11]: import MySQLdb as mdb
         import sys

         con = mdb.connect(host = 'localhost', user = 'root', passwd = 'dwdstudent2015', charset='utf8'

         # Query to create a database
         db_name = 'Final_Project'
         create_db_query = "CREATE DATABASE IF NOT EXISTS {0} DEFAULT CHARACTER SET 'utf8'".format(db_na

         # Create a database
         cursor = con.cursor()
         cursor.execute(create_db_query)
         cursor.close()
```

```
/usr/local/lib/python2.7/dist-packages/ipykernel/__main__.py:12: Warning: Can't create database 'Final_Pr
```

6 Debate1

```
In [ ]: cursor = con.cursor()
        table_name = 'Debate1'
        # Create a table
        # The {0} and {1} are placeholders for the parameters in the format(....) statement
        create_table_query = '''CREATE TABLE IF NOT EXISTS {0}.{1}
                                    (candidate varchar(250),
                                    keyword varchar(250),
                                    relevance varchar(250),
                                    sentiment varchar(250)
                                    )'''.format(db_name, table_name)

        cursor.execute(create_table_query)
        cursor.close()
```

7 Sending information to TABLES

```
In [ ]: def send2SQL (candidateName, DebateSent, DebateDic):
            cursor = con.cursor()

            query_template = 'INSERT INTO Final_Project.'+ DebateSent + '(candidate, keyword, relevance

            for line in DebateDic[candidateName]:

                i = 0
                while i != len(line['keywords']):
                    candidate = candidateName
                    keyword = line['keywords'][i]['text']
                    relevance= line['keywords'][i]['relevance']
                    sentiment= line['keywords'][i]['sentiment']['type']
                    print "Inserting ", candidate, keyword

                    query_parameters = (candidate, keyword, relevance, sentiment)
                    cursor.execute(query_template, query_parameters)
```

```
                    con.commit()
                    i+=1


            cursor.close()
```

```
In [ ]: send2SQL('Trump', 'Debate1', debate1)
        send2SQL('Cruz', 'Debate1', debate1)
        send2SQL('Rubio', 'Debate1', debate1)
```

```
In [ ]: send2SQL('Trump', 'Debate2', debate2)
        send2SQL('Cruz', 'Debate2', debate2)
        send2SQL('Rubio', 'Debate2', debate2)

        send2SQL('Trump', 'Debate3', debate3)
        send2SQL('Cruz', 'Debate3', debate3)
        send2SQL('Rubio', 'Debate3', debate3)
```

```
In [14]: File1 = openFile(debates[1])
         textFile = cleanFile(File1)

         candidateLines= getCandidate('CRUZ', textFile)
         #print candidateLines
         cruzFile= {
           "status": "OK",
           "usage": "By accessing AlchemyAPI or using information generated by AlchemyAPI, you are agre
           "totalTransactions": "2",
           "language": "english",
           "text": "[u'CRUZ: Well, Megyn, you know, at the end of the day for the folks at home, this i
           "keywords": [
             {
               "relevance": "0.973902",
               "sentiment": {
                 "type": "neutral"
               },
               "text": "cruz"
             },
             {
               "relevance": "0.860171",
               "sentiment": {
                 "mixed": "1",
                 "score": "0.0353621",
                 "type": "positive"
               },
               "text": "Donald"
             },
             {
               "relevance": "0.769394",
               "sentiment": {
                 "score": "-0.376831",
                 "type": "negative"
               },
               "text": "Hillary Clinton"
             },
             {
```

```
    "relevance": "0.727488",
    "sentiment": {
      "score": "-0.449751",
      "type": "negative"
    },
    "text": "Donald Trump"
  },
  {
    "relevance": "0.713683",
    "sentiment": {
      "score": "-0.21383",
      "type": "negative"
    },
    "text": "u'CRUZ"
  },
  {
    "relevance": "0.710594",
    "sentiment": {
      "type": "neutral"
    },
    "text": "Supreme Court justices"
  },
  {
    "relevance": "0.693337",
    "sentiment": {
      "type": "neutral"
    },
    "text": "conservative Supreme Court"
  },
  {
    "relevance": "0.651944",
    "sentiment": {
      "score": "-0.579802",
      "type": "negative"
    },
    "text": "Harry Reid"
  },
  {
    "relevance": "0.611968",
    "sentiment": {
      "score": "-0.640245",
      "type": "negative"
    },
    "text": "New York Times"
  },
  {
    "relevance": "0.586689",
    "sentiment": {
      "score": "-0.563325",
      "type": "negative"
    },
    "text": "Jimmy Carter"
  },
  {
```

```
  "relevance": "0.578867",
  "sentiment": {
    "score": "-0.410417",
    "type": "negative"
  },
  "text": "head Donald Trump"
},
{
  "relevance": "0.528143",
  "sentiment": {
    "score": "-0.474125",
    "type": "negative"
  },
  "text": "American people"
},
{
  "relevance": "0.52794",
  "sentiment": {
    "type": "neutral"
  },
  "text": "Senator Cruz"
},
{
  "relevance": "0.499918",
  "sentiment": {
    "score": "-0.490873",
    "type": "negative"
  },
  "text": "simple flat tax"
},
{
  "relevance": "0.493251",
  "sentiment": {
    "score": "-0.563325",
    "type": "negative"
  },
  "text": "Ronald Reagan"
},
{
  "relevance": "0.491057",
  "sentiment": {
    "type": "neutral"
  },
  "text": "American workers"
},
{
  "relevance": "0.487158",
  "sentiment": {
    "score": "-0.437368",
    "type": "negative"
  },
  "text": "John Kerry"
},
{
```

```
    "relevance": "0.482618",
    "sentiment": {
      "score": "-0.437368",
      "type": "negative"
    },
    "text": "George W. Bush"
  },
  {
    "relevance": "0.474669",
    "sentiment": {
      "score": "-0.640245",
      "type": "negative"
    },
    "text": "York Times tape"
  },
  {
    "relevance": "0.472739",
    "sentiment": {
      "score": "-0.594748",
      "type": "negative"
    },
    "text": "United States government"
  },
  {
    "relevance": "0.469633",
    "sentiment": {
      "type": "neutral"
    },
    "text": "debate stage"
  },
  {
    "relevance": "0.467996",
    "sentiment": {
      "score": "-0.563325",
      "type": "negative"
    },
    "text": "Jimmy Carter administration"
  },
  {
    "relevance": "0.462909",
    "sentiment": {
      "type": "neutral"
    },
    "text": "Chuck Schumer sign"
  },
  {
    "relevance": "0.45866",
    "sentiment": {
      "score": "-0.23942",
      "type": "negative"
    },
    "text": "left-wing judicial activist"
  },
  {
```

```
  "relevance": "0.457682",
  "sentiment": {
    "score": "0.736703",
    "type": "positive"
  },
  "text": "South China Sea"
},
{
  "relevance": "0.456996",
  "sentiment": {
    "type": "neutral"
  },
  "text": "support Harry Reid"
},
{
  "relevance": "0.455043",
  "sentiment": {
    "type": "neutral"
  },
  "text": "support John Kerry"
},
{
  "relevance": "0.453574",
  "sentiment": {
    "score": "-0.410417",
    "type": "negative"
  },
  "text": "CNN poll"
},
{
  "relevance": "0.453177",
  "sentiment": {
    "type": "neutral"
  },
  "text": "Senate majority leader."
},
{
  "relevance": "0.452221",
  "sentiment": {
    "type": "neutral"
  },
  "text": "support Jimmy Carter"
},
{
  "relevance": "0.42519",
  "sentiment": {
    "score": "-0.446914",
    "type": "negative"
  },
  "text": "president"
},
{
  "relevance": "0.418681",
  "sentiment": {
```

```
      "type": "neutral"
    },
    "text": "Hillary Clinton."
  },
  {
    "relevance": "0.409659",
    "sentiment": {
      "score": "0.378171",
      "type": "positive"
    },
    "text": "Web site"
  },
  {
    "relevance": "0.408178",
    "sentiment": {
      "score": "-0.473597",
      "type": "negative"
    },
    "text": "Barack Obama."
  },
  {
    "relevance": "0.407642",
    "sentiment": {
      "score": "-0.584961",
      "type": "negative"
    },
    "text": "Mr. Trump"
  },
  {
    "relevance": "0.406459",
    "sentiment": {
      "score": "0.819239",
      "type": "positive"
    },
    "text": "astonishing statement"
  },
  {
    "relevance": "0.405331",
    "sentiment": {
      "score": "-0.293798",
      "type": "negative"
    },
    "text": "liberal Democrats"
  },
  {
    "relevance": "0.402876",
    "sentiment": {
      "type": "neutral"
    },
    "text": "actual record"
  },
  {
    "relevance": "0.400797",
    "sentiment": {
```

```
      "type": "neutral"
    },
    "text": "comprehensive investigation"
  },
  {
    "relevance": "0.400752",
    "sentiment": {
      "type": "neutral"
    },
    "text": "U.S. companies"
  },
  {
    "relevance": "0.40041",
    "sentiment": {
      "score": "-0.605438",
      "type": "negative"
    },
    "text": "troubling development"
  },
  {
    "relevance": "0.39998",
    "sentiment": {
      "score": "-0.281833",
      "type": "negative"
    },
    "text": "foreign workers"
  },
  {
    "relevance": "0.399422",
    "sentiment": {
      "score": "-0.640245",
      "type": "negative"
    },
    "text": "Editorial Board"
  },
  {
    "relevance": "0.399168",
    "sentiment": {
      "type": "neutral"
    },
    "text": "H1B program"
  },
  {
    "relevance": "0.398898",
    "sentiment": {
      "score": "-0.594748",
      "type": "negative"
    },
    "text": "American citizens"
  },
  {
    "relevance": "0.398311",
    "sentiment": {
      "score": "-0.573335",
```

```
        "type": "negative"
      },
      "text": "high-paying jobs"
    },
    {
      "relevance": "0.397868",
      "sentiment": {
        "score": "-0.531953",
        "type": "negative"
      },
      "text": "H1-B program"
    },
    {
      "relevance": "0.397636",
      "sentiment": {
        "score": "-0.66833",
        "type": "negative"
      },
      "text": "H1-B abuse"
    },
    {
      "relevance": "0.397633",
      "sentiment": {
        "score": "-0.574643",
        "type": "negative"
      },
      "text": "Nancy Pelosi"
    },
    {
      "relevance": "0.396027",
      "sentiment": {
        "type": "neutral"
      },
      "text": "Cold War."
    }
  ]
}
```


8 Alchemy API wouldn't work for Cruz Debate 2

In [15]:
```
print len(cruzFile['keywords'])
```

```
#for line in cruzFile['keywords']:
#    print '\n' , "Keyword: ", line['text'], ", Relevance: ", line['relevance']
#    print "Sentiment: ", line['sentiment']



cursor = con.cursor()

query_template = 'INSERT INTO Final_Project.Debate2(candidate, keyword, relevance, sentiment) '
```

```
        for line in cruzFile['keywords']:

            candidate = 'Cruz'
            keyword = line['text']
            relevance= line['relevance']
            sentiment= line['sentiment']['type']
            print "Inserting ", candidate, keyword

            query_parameters = (candidate, keyword, relevance, sentiment)
            cursor.execute(query_template, query_parameters)
            con.commit()



        cursor.close()
```

```
50
Inserting  Cruz cruz
Inserting  Cruz Donald
Inserting  Cruz Hillary Clinton
Inserting  Cruz Donald Trump
Inserting  Cruz u'CRUZ
Inserting  Cruz Supreme Court justices
Inserting  Cruz conservative Supreme Court
Inserting  Cruz Harry Reid
Inserting  Cruz New York Times
Inserting  Cruz Jimmy Carter
Inserting  Cruz head Donald Trump
Inserting  Cruz American people
Inserting  Cruz Senator Cruz
Inserting  Cruz simple flat tax
Inserting  Cruz Ronald Reagan
Inserting  Cruz American workers
Inserting  Cruz John Kerry
Inserting  Cruz George W. Bush
Inserting  Cruz York Times tape
Inserting  Cruz United States government
Inserting  Cruz debate stage
Inserting  Cruz Jimmy Carter administration
Inserting  Cruz Chuck Schumer sign
Inserting  Cruz left-wing judicial activist
Inserting  Cruz South China Sea
Inserting  Cruz support Harry Reid
Inserting  Cruz support John Kerry
Inserting  Cruz CNN poll
Inserting  Cruz Senate majority leader.
Inserting  Cruz support Jimmy Carter
Inserting  Cruz president
Inserting  Cruz Hillary Clinton.
Inserting  Cruz Web site
Inserting  Cruz Barack Obama.
Inserting  Cruz Mr. Trump
Inserting  Cruz astonishing statement
```

```
Inserting  Cruz liberal Democrats
Inserting  Cruz actual record
Inserting  Cruz comprehensive investigation
Inserting  Cruz U.S. companies
Inserting  Cruz troubling development
Inserting  Cruz foreign workers
Inserting  Cruz Editorial Board
Inserting  Cruz H1B program
Inserting  Cruz American citizens
Inserting  Cruz high-paying jobs
Inserting  Cruz H1-B program
Inserting  Cruz H1-B abuse
Inserting  Cruz Nancy Pelosi
Inserting  Cruz Cold War.
```

Appendice - Twitter - Public Sentiment Analysis

May 17, 2016

1 Tweet Data load in CSV for Trump using Tweepy

In [ ]:
```
from twython import Twython # pip install twython
import time # standard lib
import urllib
import requests
import tweepy
from requests_oauthlib import OAuth1
import pandas as pd
import matplotlib.pyplot as plt
import json
from nltk.tokenize import word_tokenize
from collections import defaultdict
from nltk.corpus import stopwords
import string
import operator
import csv
from collections import Counter



CONSUMER_KEY ='SFh6rygaDcA4eSKxD9HFW3Yq4'
CONSUMER_SECRET = '2PyhorQtbW7AQQCcNQqzlJg0Y9GW6XulBobeSz1oNXbVHH0eMW'

ACCESS_KEY = '130253651-gL2gsx1nyy4yi8S2w3gSdCE9rbMazLJ4ptyciN0q'
ACCESS_SECRET = '9DAQi4iDFBDOHtTiMsDZ2E5Xa6HaPVodYj7Eu7NiFUWLm'


auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)

#Trump API

api = tweepy.API(auth)
# Append data to CSV
CSVF = open('/home/ubuntu/data/Trump.csv', 'a')
CSV Writer
CSVW = csv.writer(CSVF)
#most recent tweet id for iteration
lis = [724910184376750080]
x=0
total_retweets=0
```

```
        #print the_page

        for i in range(0, 16):
            time.sleep(40)
            for id,tweet in enumerate (tweepy.Cursor(api.search,
                            q='Trump',
                            since="2016-04-26",
                            until="2016-05-03",
                            lang="en",
                            max_id=lis[-1]
                            ,include_retweets=False
                            ).items(200)):
                #Write a row to the csv file/ I use encode utf-8
                data = urllib.urlencode({"text":tweet.text.encode('utf-8')})
                u = urllib.urlopen("http://text-processing.com/api/sentiment/", data)
                the_page = u.read()

                lis.append(tweet.id)
                total_retweets+=tweet.retweet_count+ 1
                CSVW.writerow(['Trump',tweet.geo,tweet.created_at
                               ,tweet.retweet_count,the_page,tweet.text.encode('utf-8')])
                print x,'Trump',tweet.created_at, tweet.text,tweet.retweet_count,the_page
                print "Retweets Till now :",total_retweets
                x+=1
        CSVF.close()

        print "This is total retweet count"
        print total_retweets
```

2 Tweet Data load in CSV for Cruz using Tweepy

```
In [ ]: #Cruz
        from twython import Twython # pip install twython
        import time # standard lib
        import urllib
        import requests
        import tweepy
        from requests_oauthlib import OAuth1
        import pandas as pd
        import matplotlib.pyplot as plt
        import json
        from nltk.tokenize import word_tokenize
        from collections import defaultdict
        from nltk.corpus import stopwords
        import string
        import operator
        import csv
        from collections import Counter


        CONSUMER_KEY ='SFh6rygaDcA4eSKxD9HFW3Yq4'
        CONSUMER_SECRET = '2PyhorQtbW7AQQCcNQqzlJg0Y9GW6XulBobeSz1oNXbVHH0eMW'
```

```
ACCESS_KEY = '130253651-gL2gsx1nyy4yi8S2w3gSdCE9rbMazLJ4ptyciN0q'
ACCESS_SECRET = '9DAQi4iDFBDOHtTiMsDZ2E5Xa6HaPVodYj7Eu7NiFUWLm'


auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)

api = tweepy.API(auth)

CSVF3 = open('/home/ubuntu/data/Cruz.csv', 'a')
#CSV Writer
CSVW = csv.writer(CSVF3)

#get the lis value for most recent tweet
lis = [724910184376750080]
x=0
total_retweets=0

for i in range(0, 16):
    time.sleep(40)
    for id,tweet in enumerate (tweepy.Cursor(api.search,
                    q='Cruz',
                    since="2016-04-26",
                    until="2016-04-28",
                    lang="en",
                    max_id=lis[-1]
                    ,include_retweets=False
                    ).items(200)):
        #Write a row to the csv file/ I use encode utf-8
        data = urllib.urlencode({"text":tweet.text.encode('utf-8')})
        u = urllib.urlopen("http://text-processing.com/api/sentiment/", data)
        the_page = u.read()
        print x,'Cruz',tweet.created_at,tweet.geo, tweet.text,tweet.retweet_count,the_page
        lis.append(tweet.id)
        total_retweets+=tweet.retweet_count+ 1
        CSVW.writerow(['Cruz',tweet.geo,tweet.created_at
                       ,tweet.retweet_count,the_page,tweet.text.encode('utf-8')])
        print "Retweets Till now :",total_retweets
        x+=1
CSVF.close()
```

3 Tweet Data load in CSV for Kasich using Tweepy

In [ ]:
```
#Kasich
from twython import Twython # pip install twython
import time # standard lib
import urllib
import requests
import tweepy
from requests_oauthlib import OAuth1
import pandas as pd
import matplotlib.pyplot as plt
import json
```

```
from nltk.tokenize import word_tokenize
from collections import defaultdict
from nltk.corpus import stopwords
import string
import operator
import csv
from collections import Counter


CONSUMER_KEY ='SFh6rygaDcA4eSKxD9HFW3Yq4'
CONSUMER_SECRET = '2PyhorQtbW7AQQCcNQqzlJg0Y9GW6XulBobeSz1oNXbVHH0eMW'

ACCESS_KEY = '130253651-gL2gsx1nyy4yi8S2w3gSdCE9rbMazLJ4ptyciN0q'
ACCESS_SECRET = '9DAQi4iDFBDOHtTiMsDZ2E5Xa6HaPVodYj7Eu7NiFUWLm'


auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)

api = tweepy.API(auth)


CSVF4 = open('/home/ubuntu/data/Kasich.csv', 'a')
#CSV Writer
CSVW = csv.writer(CSVF4)

#get the lis value for most recent tweet
lis = [724910184376750080]
x=0
total_retweets=0

for i in range(0, 16):
    time.sleep(40)
    for id,tweet in enumerate (tweepy.Cursor(api.search,
                    q='Kasich',
                    since="2016-04-26",
                    until="2016-05-03",
                    lang="en",
                    max_id=lis[-1]
                    ,include_retweets=False
                    ).items(200)):
        #Write a row to the csv file/ I use encode utf-8
        data = urllib.urlencode({"text":tweet.text.encode('utf-8')})
        u = urllib.urlopen("http://text-processing.com/api/sentiment/", data)
        the_page = u.read()
        print x,'Kasich',tweet.geo,tweet.place, tweet.text,tweet.retweet_count,the_page
        lis.append(tweet.id)
        total_retweets+=tweet.retweet_count+ 1
        CSVW.writerow(['Kasich',tweet.geo,tweet.created_at
                       ,tweet.retweet_count,the_page,tweet.text.encode('utf-8')])
        print "Retweets Till now :",total_retweets
        x+=1
CSVF.close()
```

4 Reading and Transforming CSV data for Trump

In [ ]:
```
!head /home/ubuntu/data/Trump.csv
```

In [ ]:
```
!cut -f1,2,6,7,8,9,10,11 -d',' /home/ubuntu/data/Trump.csv > /home/ubuntu/data/Trump_reduced.csv
```

In [ ]:
```
!head /home/ubuntu/data/Trump_reduced.csv
```

5 Reading and Transforming CSV data for Cruz

In [ ]:
```
!head /home/ubuntu/data/Cruz.csv
```

In [ ]:
```
!cut -f1,2,4,5,6,7,8,9 -d',' /home/ubuntu/data/Cruz.csv > /home/ubuntu/data/Cruz_reduced.csv
```

In [ ]:
```
!head /home/ubuntu/data/Cruz_reduced.csv
```

6 Reading and Transforming CSV data for Kasich

In [ ]:
```
!head /home/ubuntu/data/Kasich.csv
```

In [ ]:
```
!cut -f1,2,4,5,6,7,8,9 -d',' /home/ubuntu/data/Kasich.csv > /home/ubuntu/data/Kasich_reduced.csv
```

In [ ]:
```
!head /home/ubuntu/data/Kasich_reduced.csv
```

7 Making SQL Connection and creating the database

In [181]:
```
#Now SQL Part
import sys
import MySQLdb


connection = MySQLdb.connect(host = 'localhost', user = 'root'
                                    , passwd = 'dwdstudent2015', charset='utf8', use_unicode=True);
```

In [182]:
```
cur = connection.cursor();
```

In [3]:
```
#Creating HomeWork 5 database
database = 'FinalProject'
Q1 = "CREATE DATABASE IF NOT EXISTS {0} DEFAULT CHARACTER SET 'utf8'".format(database)
cur.execute(Q1)
```

```
/usr/local/lib/python2.7/dist-packages/ipykernel/__main__.py:4: Warning: Can't create database 'FinalProj
```

Out[3]:
```
1L
```

In [188]:
```
%load_ext sql
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
```

In [185]:
```
%sql mysql://root:dwdstudent2015@localhost:3306/FinalProject?charset=utf8
```

Out[185]:
```
u'Connected: root@FinalProject'
```

In [187]:
```
%sql USE FinalProject
```

```
0 rows affected.
```

Out[187]:
```
[]
```

In [ ]:
```
#%sql drop table FinalProject.Republican_debate
```

8 Creating Table for storing Candidate data

```
In [ ]: %sql create table Republican_debate(CandidateName varchar(250),Debate varchar(250)
                                            ,Retweets int,Semantics varchar(250),
                                            Tweet varchar(250));
```

9 Loading candidate data from CSVs into the database table

```
In [ ]: !mysql -u root password=dwdstudent2015 --local-infile FinalProject

        # It can be done in Terminal as well

        cur = connection.cursor()
        database = 'FinalProject'
        table = 'debate1'
        Q2 = '''load data local infile '/home/ubuntu/data/Trump_reduced.csv'
        into table FinalProject.Republican_debate
        fields terminated by ','  enclosed by '"'
        lines terminated by '\n'
        ignore 1 rows;'''.format(database, table)
```

```
In [ ]: !mysql -u root password=dwdstudent2015 --local-infile FinalProject

        # Do this in Terminal as well


        cur = connection.cursor()
        database = 'FinalProject'
        table = 'debate1'
        Q2 = '''load data local infile '/home/ubuntu/data/Cruz_debate1_reduced.csv'
        into table FinalProject.Republican_debate
        fields terminated by ','  enclosed by '"'
        lines terminated by '\n'
        ignore 1 rows;'''.format(database, table)
```

```
In [ ]: !mysql -u root password=dwdstudent2015 --local-infile FinalProject

        # Do this in Terminal as well


        cur = connection.cursor()
        database = 'FinalProject'
        table = 'debate1'
        Q2 = '''load data local infile '/home/ubuntu/data/Kasich_reduced.csv'
        into table FinalProject.Republican_debate
        fields terminated by ','  enclosed by '"'
        lines terminated by '\n'
        ignore 1 rows;'''.format(database, table)
```

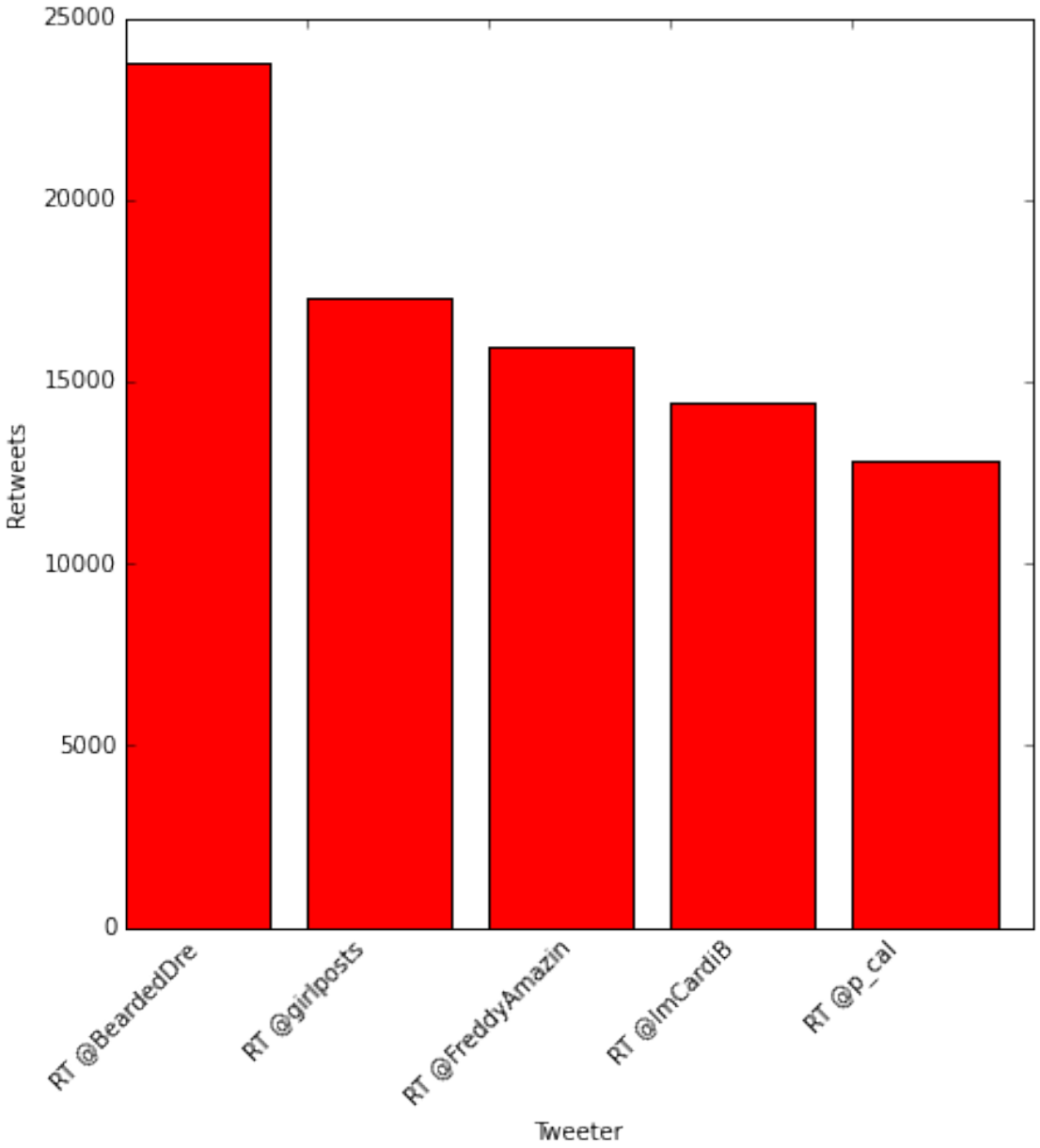
10 Selecting candidate data from database table

```
In [ ]: %%sql select CandidateName,Retweets,Semantics,Tweet from FinalProject.Republican_debate
        where candidatename = 'TRUMP' and Semantics not like '%Throttled, wait%'
```

```
In [ ]: %%sql select CandidateName,Retweets,Semantics,Tweet from FinalProject.Republican_debate
        where candidatename = 'Cruz' and Semantics not like '%Throttled, wait%'

In [ ]: %%sql select CandidateName,Retweets,Semantics,Tweet from FinalProject.Republican_debate
        where candidatename = 'Kasich' and Semantics not like '%Throttled, wait%'
```

11 Displaying Top 5 Negative Tweets for Trump

```
In [179]: tr10=%%sql SELECT distinct Retweets, LEFT(Tweet,INSTR(Tweet,":")-1) as Tweeter
          from FinalProject.Republican_debate
          where candidatename = 'Trump' and Sentiment in ('Negative')
          and Retweets !=14964 order by Retweets desc limit 5
          tr10

5 rows affected.

Out[179]: [(23777L, u'RT @BeardedDre'),
           (17306L, u'RT @girlposts'),
           (16001L, u'RT @FreddyAmazin'),
           (14423L, u'RT @ImCardiB'),
           (12832L, u'RT @p_cal')]

In [24]: tr9=%%sql select distinct Tweet from FinalProject.Republican_debate
         where candidatename = 'Trump' and Retweets !=14964
         and Sentiment in ('Negative') order by Retweets desc limit 5
         tr9

5 rows affected.

Out[24]: [(u"RT @BeardedDre: I pray and I pray trump doesn't get elected VOTE... https://t.co/V3EZHpXJj
          (u'RT @girlposts: me: green is like the ugliest color ever\ndonald trump: i hate the color gr
          (u"RT @FreddyAmazin: I'M SCREAMING THEY DROVE PAST A TRUMP RALLY AND PLAYED THIS https://t.co/
          (u'RT @ImCardiB: Donald trump gona deport all the foreign bitches that yall love fucking  ',)
          (u'RT @p_cal: Donald Trump rally vs Bernie Sanders rally https://t.co/ZFqfly9Zpf\r',)]

In [25]: from matplotlib import pyplot as plt
         plt.figure(figsize=(7,7))
         tr10.bar(color='r')
         tr9

Out[25]: [(u"RT @BeardedDre: I pray and I pray trump doesn't get elected VOTE... https://t.co/V3EZHpXJj
          (u'RT @girlposts: me: green is like the ugliest color ever\ndonald trump: i hate the color gr
          (u"RT @FreddyAmazin: I'M SCREAMING THEY DROVE PAST A TRUMP RALLY AND PLAYED THIS https://t.co/
          (u'RT @ImCardiB: Donald trump gona deport all the foreign bitches that yall love fucking  ',)
          (u'RT @p_cal: Donald Trump rally vs Bernie Sanders rally https://t.co/ZFqfly9Zpf\r',)]
```



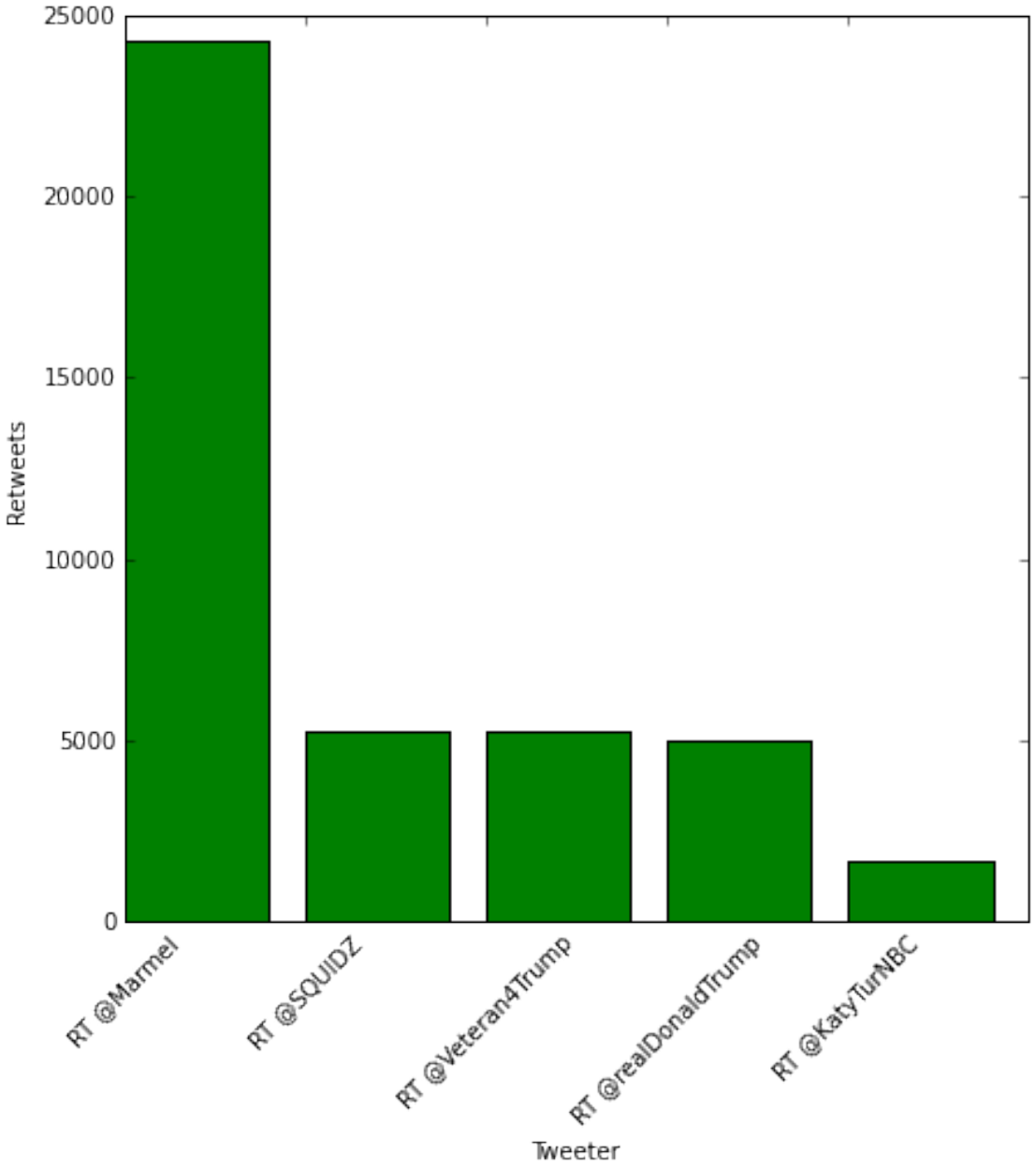
12 Displaying Top 5 Positive Tweets for Trump

In [26]:
```
tr11=%%sql select distinct Tweet from FinalProject.Republican_debate
where candidatename = 'Trump' and Retweets !=14964
and Sentiment in ('Positive') order by Retweets desc limit 5

tr12=%%sql SELECT distinct Retweets, LEFT(Tweet,INSTR(Tweet,":")-1) as Tweeter
from FinalProject.Republican_debate
where candidatename = 'Trump' and Retweets !=14964 and Sentiment in ('Positive')
order by Retweets desc limit 5
plt.figure(figsize=(7,7))
```

```
tr12.bar(color='g')
tr11
```

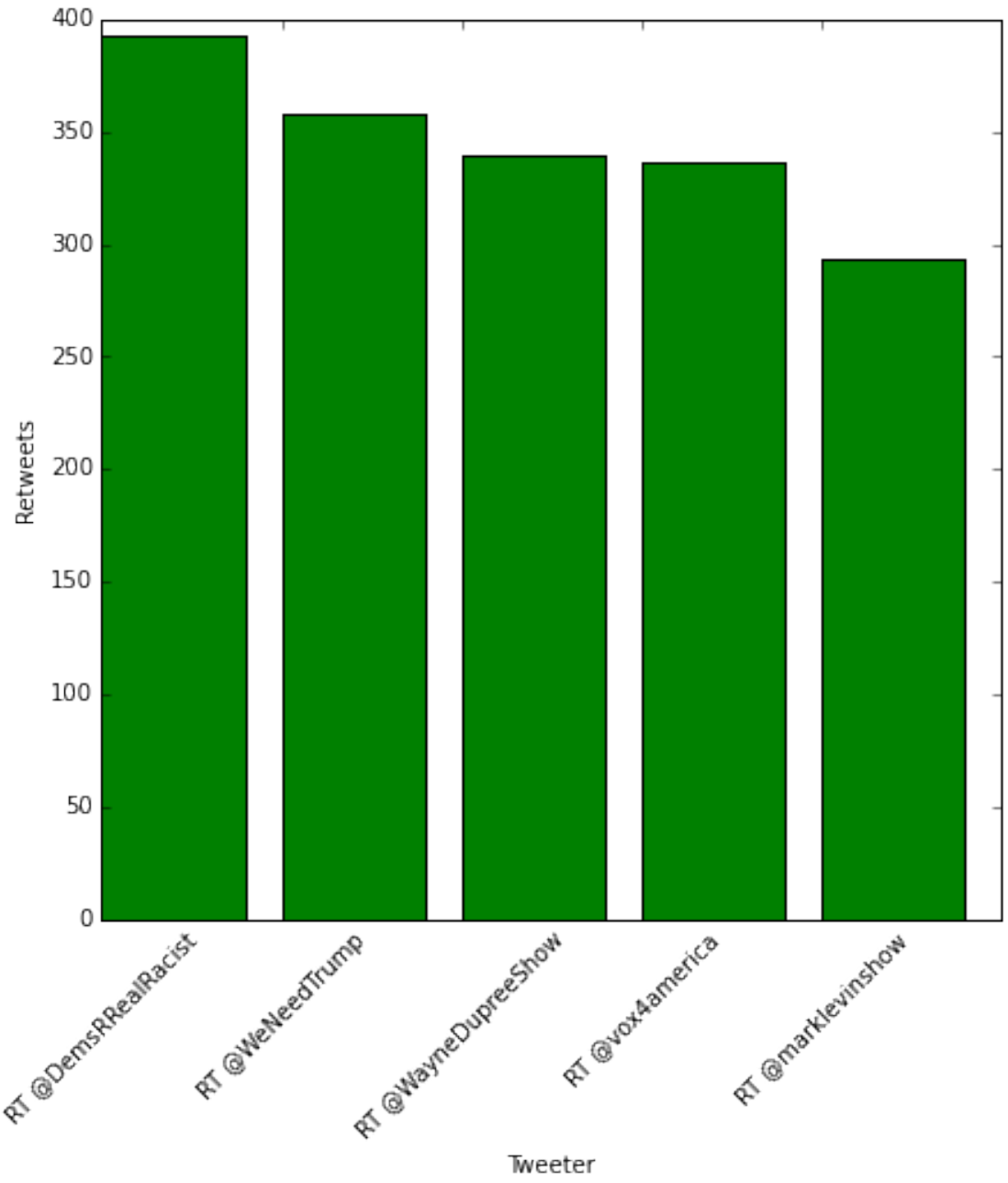
```
5 rows affected.
5 rows affected.
```

Out[26]:
```
[(u'RT @Marmel: Whoever made this\nTrump v. Manson. https://t.co/ztUx36aY1c"\r\n',),
 (u'RT @SQUlDZ: Mexicans and Black People teaming up against Donald Trump BOY LIFE BEAUTIFUL RI
 (u"RT @Veteran4Trump: I'm a Veteran. I was born in Mexico\n",),
 (u'RT @realDonaldTrump: Thank you Wilkes-Barre\n#MakeAmericaGreatAgain #Trump2016 \nhttps://t
 (u'RT @KatyTurNBC: Wilkes Barre\n',)]
```



13 Displaying Top 5 Positive Tweets for Cruz

```
In [120]: tr13=%%sql select distinct Tweet,Retweets,sentiment
          from FinalProject.Republican_debate
          where candidatename = 'Cruz' and Sentiment in ('Positive')
          order by Retweets desc limit 5

          tr14=%%sql SELECT distinct Retweets,
          LEFT(Tweet,INSTR(Tweet,":")-1) as Tweeter
          from FinalProject.Republican_debate
          where candidatename = 'Cruz' and Sentiment in ('Positive')
          order by Retweets desc limit 5
          plt.figure(figsize=(7,7))
          tr14.bar(color='g')
          tr13

5 rows affected.
5 rows affected.

Out[120]: [(u'RT @DemsRRealRacist: Ted Cruz understands that step number one in defeating that nasty Wa
           (u'RT @WeNeedTrump: Cruz\n', 358L, u'Positive'),
           (u'RT @WayneDupreeShow: Now do you believe me that the Cruz campaign is struggling\n\n#Trump
           (u'RT @vox4america: On behalf of Trump supporters\n', 336L, u'Positive'),
           (u'RT @marklevinshow: Sellout? Nah\n', 293L, u'Positive')]
```



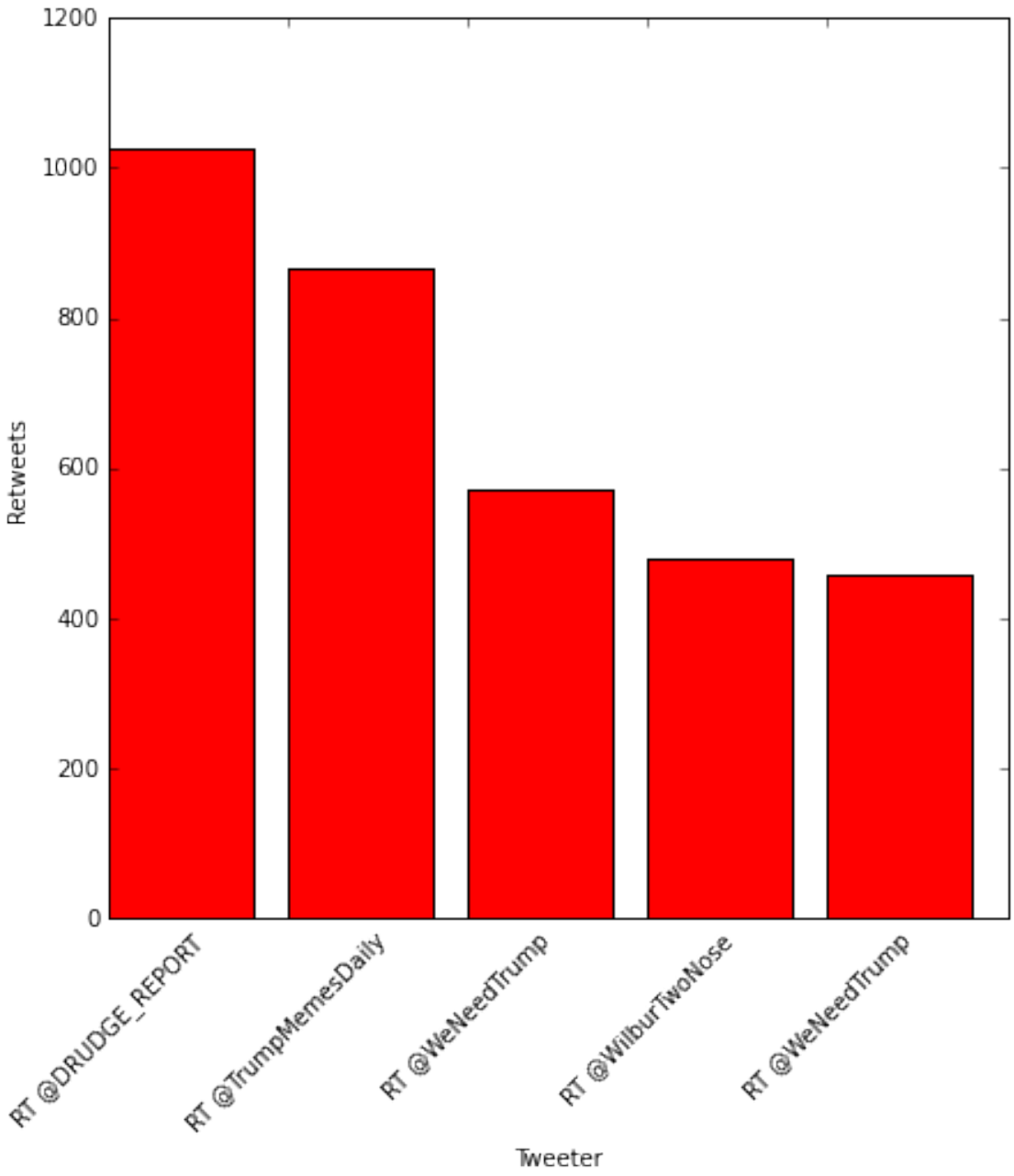
14 Displaying Top 5 Negative Tweets for Cruz

```
In [121]: tr15=%%sql select distinct Tweet
          from FinalProject.Republican_debate
          where candidatename = 'Cruz' and Sentiment in ('Negative')
          order by Retweets desc limit 5

          tr16=%%sql SELECT distinct Retweets,
          LEFT(Tweet,INSTR(Tweet,":")-1) as Tweeter
```

```
          from FinalProject.Republican_debate
          where candidatename = 'Cruz' and Sentiment in ('Negative')
          order by Retweets desc limit 5
          plt.figure(figsize=(7,7))
          tr16.bar(color='r')
          tr15
```

```
5 rows affected.
5 rows affected.
```

```
Out[121]: [(u'RT @DRUDGE_REPORT: CRUZ CONFRONTED: HOW CAN YOU HAVE DELEGATES WITHOUT A VOTE!? https://t.
           (u'RT @TrumpMemesDaily: Cruz and Kasich should just drop out drop out already\n#Trump2016 #V
           (u"RT @WeNeedTrump: The people aren't happy with the dirty politics Cruz is playing. It's ov
           (u'RT @WilburTwoNose: Cruz on CNN just said\n',),
           (u"RT @WeNeedTrump: RETWEET if you think Cruz and Kasich's attempt to subvert the popular vo
```



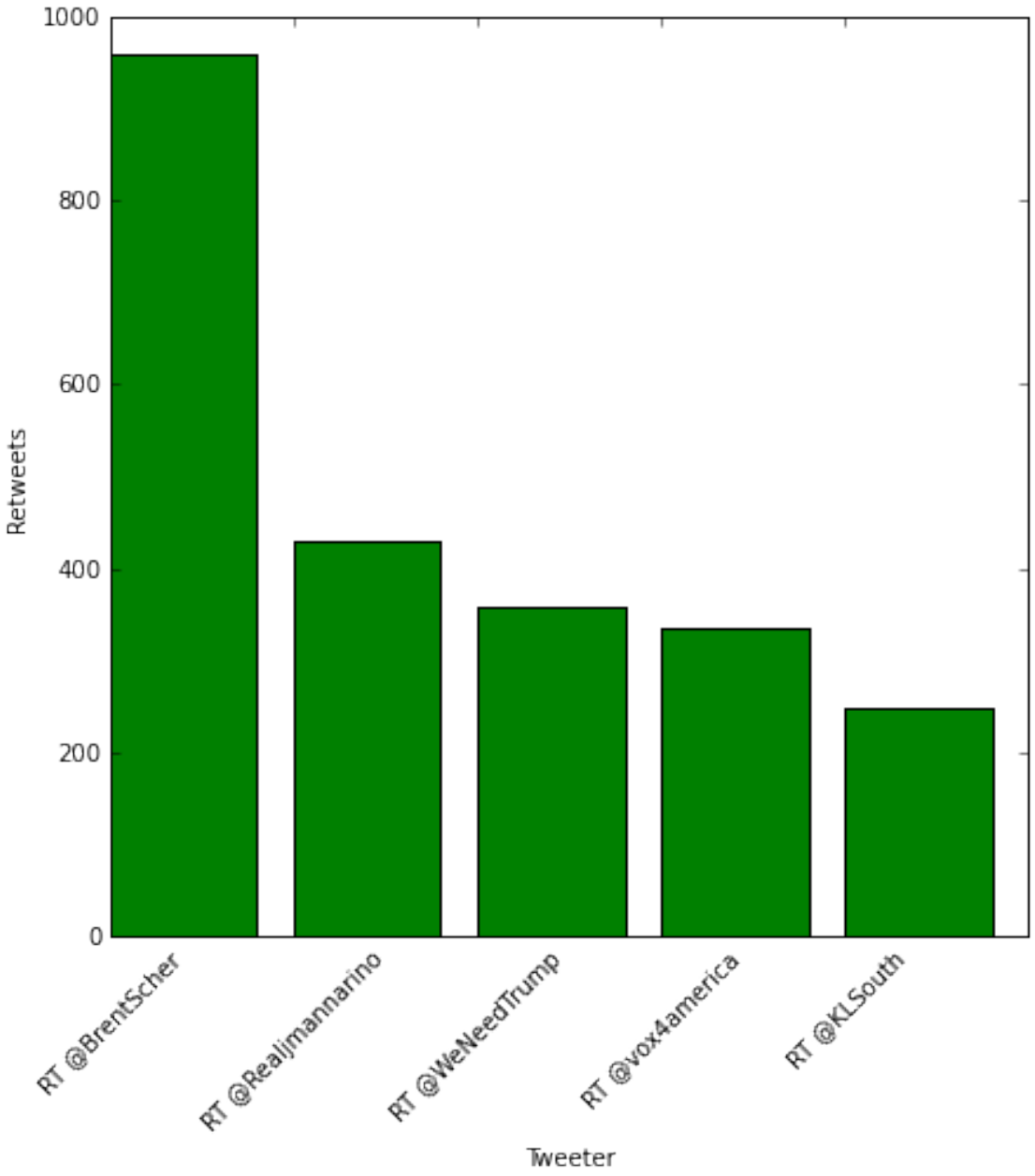
15 Displaying Top 5 Positive Tweets for Kasich

```
In [178]: tr17=%%sql select distinct Tweet,retweets
          from FinalProject.Republican_debate
          where candidatename = 'Kasich' and Sentiment in ('Positive')
          order by Retweets desc limit 5

          tr18=%%sql SELECT distinct Retweets,
          LEFT(Tweet,INSTR(Tweet,":")-1) as Tweeter
```

```
          from FinalProject.Republican_debate
          where candidatename = 'Kasich' and Sentiment in ('Positive')
          order by Retweets desc limit 5
          plt.figure(figsize=(7,7))
          tr18.bar(color='g')
          tr17
```

```
5 rows affected.
5 rows affected.
```

```
Out[178]: [(u'RT @BrentScher: Am I crazy\nKasic', 959L),
           (u'RT @Realjmannarino: RETWEET THIS POLL SO WE CAN GET A BIG PICTURE IDEA! \n\nCruz and Kasi
           (u'RT @WeNeedTrump: Cruz\nKasic', 358L),
           (u'RT @vox4america: On behalf of Trump supporters\nKasich', 336L),
           (u'RT @KLSouth: #Cruz #Kasich are publicly confessing they\u2019re nothing more than spoiler
```



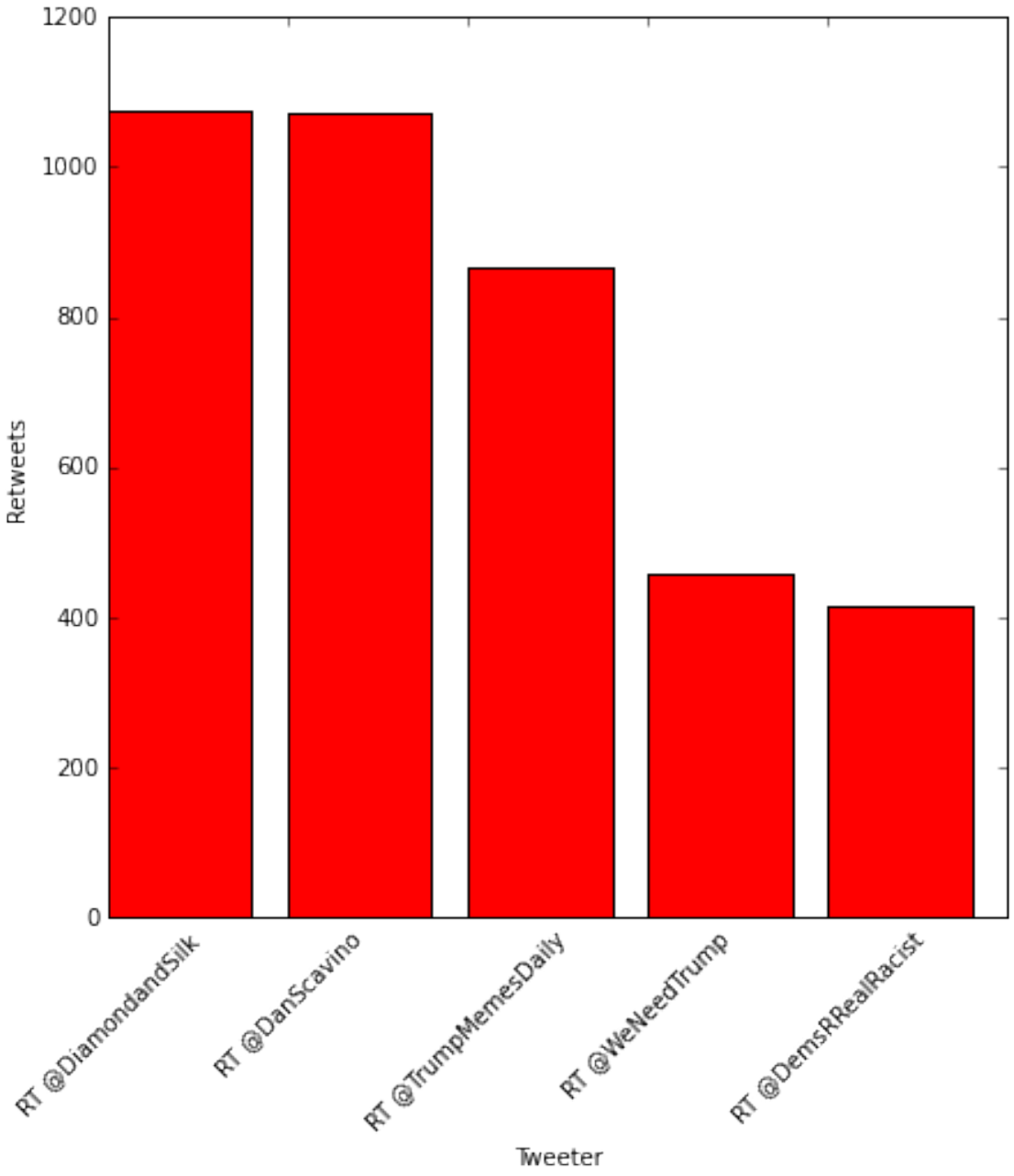
16 Displaying Top 5 Negative Tweets for Kasich

```
In [152]: tr19=%%sql select distinct Tweet
          from FinalProject.Republican_debate
          where candidatename = 'Kasich' and Sentiment in ('Negative')
          order by Retweets desc limit 5

          tr20=%%sql SELECT distinct Retweets,
          LEFT(Tweet,INSTR(Tweet,":")-1) as Tweeter
          from FinalProject.Republican_debate
```

```
          where candidatename = 'Kasich' and Sentiment in ('Negative')
          order by Retweets desc limit 5
          plt.figure(figsize=(7,7))
          tr20.bar(color='r')
          tr19
```

```
5 rows affected.
5 rows affected.
```

```
Out[152]: [(u"RT @DiamondandSilk: Lyin' Ted &amp;amp; Lunatic Kasich It will be a cold day inside of a Volca
           (u"RT @DanScavino: Should say: Follow talking points in Kasich deal. WE ARE LOSING &amp;amp; WE A
           (u'RT @TrumpMemesDaily: Cruz and Kasich should just drop out drop out already\n#Trump2016 #Vo
           (u"RT @WeNeedTrump: RETWEET if you think Cruz and Kasich's attempt to subvert the popular vot
           (u"RT @DemsRRealRacist: Attention Cruz and Kasich supporters: read your e-mail CAREFULLY so y
```



17 Running queries to transform and check table data

In [ ]:
```
%%sql select distinct Tweet,CandidateName,Retweets,Semantics
from FinalProject.Republican_debate
where candidatename = 'Cruz'
and Semantics not like '%Throttled, wait%'
order by Retweets desc limit 10
```

In [ ]:
```
%%sql select distinct Tweet,Retweets,Semantics,CandidateName
from FinalProject.Republican_debate
```

```
        where candidatename = 'Kasich'
        and Semantics not like '%Throttled, wait%'
        order by Retweets desc limit 10
```

```
In [ ]: %%sql alter table FinalProject.Republican_debate
        add Sentiment varchar(250)
```

```
In [ ]: %%sql update FinalProject.Republican_debate
        SET Sentiment = substring(Semantics,-5)
```

```
In [ ]: %%sql select distinct Sentiment
        from FinalProject.Republican_debate
        where candidatename='Trump'
        and Semantics not like '%Throttled, wait%'
```

```
In [ ]: %%sql select distinct candidatename
        from FinalProject.Republican_debate
        where Semantics not like '%Throttled, wait%'
```

18 Counting Positive, Negative and Neutral Tweets (No Retweet)

```
In [ ]: ### Count of  Original Tweets as Neutral/Positive/Negative for candidates

        %%sql select CandidateName,
        SUM(CASE Sentiment WHEN 'ral"}' THEN 1 else 0 END) AS Neutral,
        SUM(CASE Sentiment WHEN 'pos"}' THEN 1 else 0 END) AS Positive,
        SUM(CASE Sentiment WHEN 'neg"}' THEN 1 else 0 END) AS Negative
        from FinalProject.Republican_debate
        where Semantics not like '%Throttled, wait%'
        and Candidatename in ('Trump','Cruz','Kasich')
        Group by CandidateName
```

19 Counting Positive, Negative and Neutral Tweets (with Retweet)

```
In [ ]: ### Count of Tweets + Retweets as Neutral/Positive/Negative for candidates
        %%sql select CandidateName,
        SUM(CASE Sentiment WHEN 'ral"}' THEN (Retweets+1) else 0 END) AS Neutral,
        SUM(CASE Sentiment WHEN 'pos"}' THEN (Retweets+1) else 0 END) AS Positive,
        SUM(CASE Sentiment WHEN 'neg"}' THEN (Retweets+1) else 0 END) AS Negative
        from FinalProject.Republican_debate
        where Semantics not like '%Throttled, wait%'
        and Candidatename in ('Trump','Cruz','Kasich')
        Group by CandidateName
```

```
In [ ]: %sql select distinct candidatename from Republican_debate
```

20 Selecting Positive, Negative and Neutral Tweets (with Retweets) for Trump

```
In [189]: tr= %%sql select CandidateName,
          Sentiment,SUM(Retweets+1) as Retweets
```

```
          from FinalProject.Republican_debate
          where Sentiment in ('Neutral','Positive','Negative')
          and CandidateName='Trump'
          Group by Sentiment,CandidateName
```

```
3 rows affected.
```

21 Selecting Positive, Negative and Neutral Tweets (with Retweets) for Cruz

```
In [190]: tr2= %%sql select CandidateName,
          Sentiment,SUM(Retweets+1) as Retweets
          from FinalProject.Republican_debate
          where Sentiment in ('Neutral','Positive','Negative')
          and CandidateName='Cruz'
          Group by Sentiment,CandidateName
```

```
3 rows affected.
```

22 Selecting Positive, Negative and Neutral Tweets (with Retweets) for Kasich

```
In [191]: tr3= %%sql select CandidateName,
          Sentiment,SUM(Retweets+1) as Retweets
          from FinalProject.Republican_debate
          where Sentiment in ('Neutral','Positive','Negative')
          and CandidateName='Kasich'
          Group by Sentiment,CandidateName
```

```
3 rows affected.
```

```
In [ ]: %sql select * from Republican_debate
```

```
In [ ]: %sql select distinct Sentiment from Republican_debate
```

23 Using Pandas, data frames for data visualization

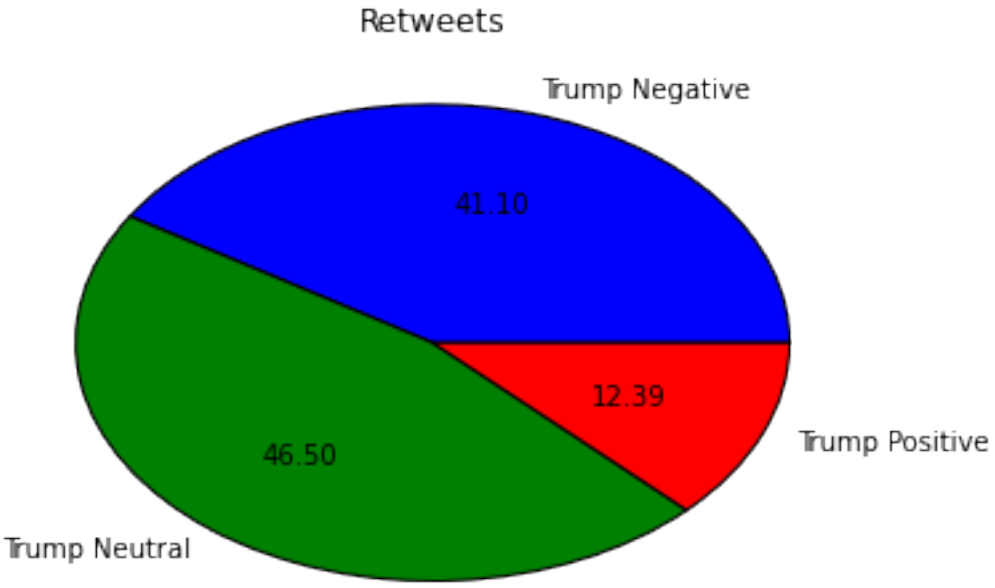
```
In [192]: import pandas as pd
          df_docks = pd.DataFrame(tr, columns=tr.keys)
          df_docks
```

```
Out[192]:   CandidateName Sentiment Retweets
          0         Trump  Negative   254404
          1         Trump   Neutral   287854
          2         Trump  Positive    76722
```

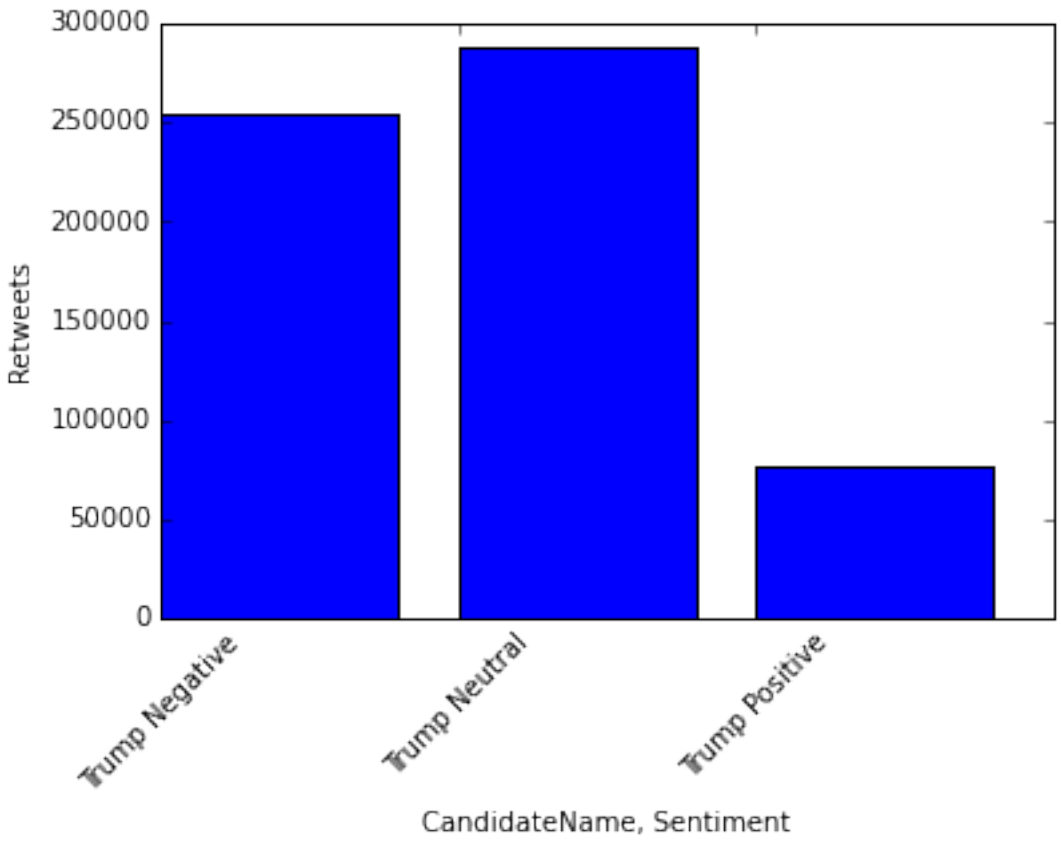
24 Percentage of Positive, Negative and Neutral Tweets + Retweets for trump

```
In [13]: %matplotlib inline
         tr.pie(autopct='%.2f')
```

```
Out[13]: ([<matplotlib.patches.Wedge at 0x7f4e0e5311d0>,
           <matplotlib.patches.Wedge at 0x7f4e0e543050>,
           <matplotlib.patches.Wedge at 0x7f4e0e543e10>],
          [<matplotlib.text.Text at 0x7f4e0e531690>,
           <matplotlib.text.Text at 0x7f4e0e543610>,
           <matplotlib.text.Text at 0x7f4e0e551350>],
          [<matplotlib.text.Text at 0x7f4e0e531bd0>,
           <matplotlib.text.Text at 0x7f4e0e5439d0>,
           <matplotlib.text.Text at 0x7f4e0e551850>])
```




```
In [14]: tr.bar()

Out[14]: <Container object of 3 artists>
```

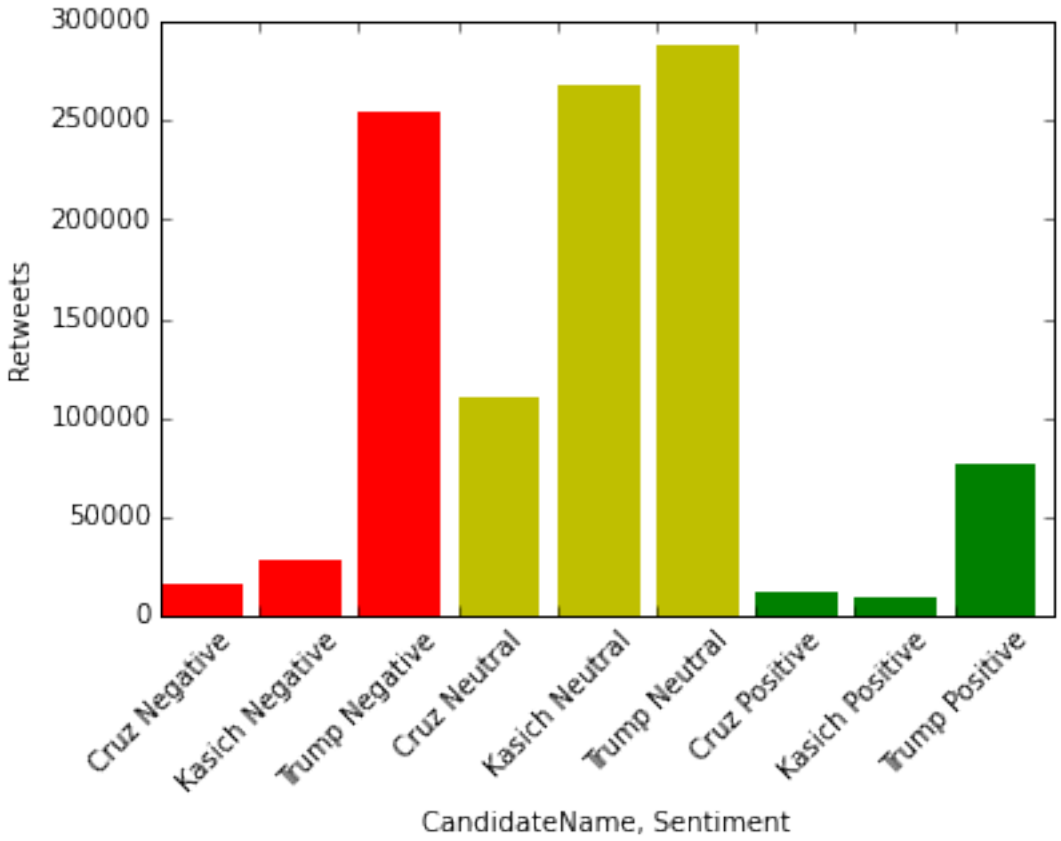



```
In [19]: tr4= %%sql select CandidateName,
         Sentiment,SUM(Retweets+1) as Retweets
         from FinalProject.Republican_debate
         where Sentiment in ('Neutral','Positive','Negative')
         Group by Sentiment,CandidateName

9 rows affected.

In [20]: tr5= %%sql select CandidateName,
         Sentiment,SUM(Retweets+1) as Retweets
         from FinalProject.Republican_debate
         where Sentiment in ('Neutral','Positive','Negative')
         Group by Sentiment,CandidateName

9 rows affected.
```

25 Bar chart showing absolute values of Positive, Negative and Neutral Tweets+ Retweets for Trump, Cruz and Kasich

```
In [21]: import matplotlib.pyplot as plt
         from matplotlib.dates import date2num
         import datetime
         import plotly.plotly as py
```

```
barlist=tr4.bar()
barlist[0].set_color('r')
barlist[1].set_color('r')
barlist[2].set_color('r')
barlist[3].set_color('y')
barlist[4].set_color('y')
barlist[5].set_color('y')
barlist[6].set_color('g')
barlist[7].set_color('g')
barlist[8].set_color('g')
plt.show()
```




```
In [17]: !sudo pip install plotly

The directory '/home/ubuntu/.cache/pip/http' or its parent directory is not owned by the current user ar
The directory '/home/ubuntu/.cache/pip' or its parent directory is not owned by the current user and ca
Collecting plotly
  Downloading plotly-1.9.10.tar.gz (560kB)
ent already satisfied (use --upgrade to upgrade): requests in /usr/local/lib/python2.7/dist-packages (fi
Requirement already satisfied (use --upgrade to upgrade): six in /usr/local/lib/python2.7/dist-packages
Requirement already satisfied (use --upgrade to upgrade): pytz in /usr/local/lib/python2.7/dist-packages
Installing collected packages: plotly
  Running setup.py install for plotly
```

```
Successfully installed plotly-1.9.10
You are using pip version 7.1.2, however version 8.1.1 is available.You should consider upgrading via t
```

```
In [47]: df_docks2 = pd.DataFrame(tr2, columns=tr.keys)
         df_docks2
```

```
Out[47]:   CandidateName Sentiment Retweets
         0          Cruz  Negative    21721
         1          Cruz   Neutral   110934
         2          Cruz  Positive    12535
```

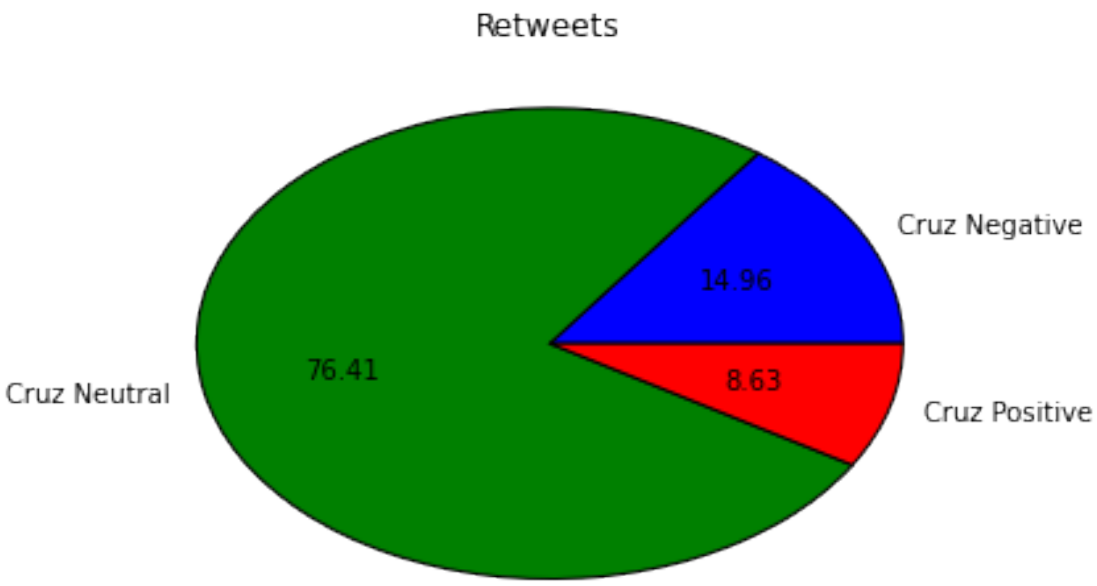
```
In [48]: #df_docks2['Retweets'].hist()
         df_docks2.groupby('Sentiment').Retweets.sum()
         #.plot(kind='bar')
         #df_docks2.plot(kind='bar', stacked=True, color=['red','blue','Yellow'], grid=True)
```

```
Out[48]: Sentiment
         Negative     21721
         Neutral     110934
         Positive     12535
         Name: Retweets, dtype: object
```

26 Percentage of Positive, Negative and Neutral Tweets + Retweets for Cruz

```
In [49]: tr2.pie(autopct='%.2f')
```

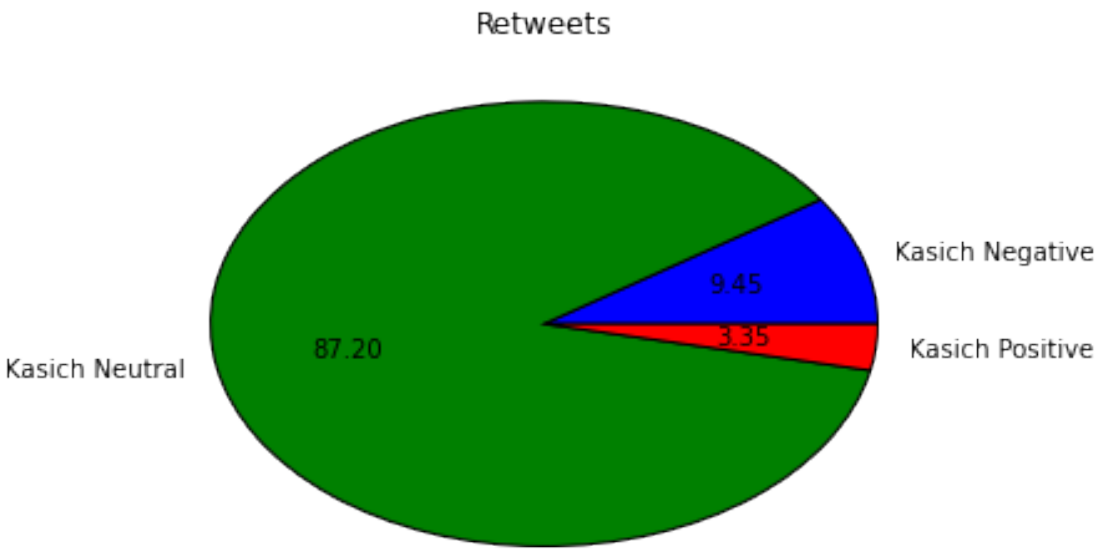
```
Out[49]: ([<matplotlib.patches.Wedge at 0x7fc7b69d0fd0>,
           <matplotlib.patches.Wedge at 0x7fc7b69dbe10>,
           <matplotlib.patches.Wedge at 0x7fc7b69e6c10>],
          [<matplotlib.text.Text at 0x7fc7b69db590>,
           <matplotlib.text.Text at 0x7fc7b69e6410>,
           <matplotlib.text.Text at 0x7fc7b69f2210>],
          [<matplotlib.text.Text at 0x7fc7b69db9d0>,
           <matplotlib.text.Text at 0x7fc7b69e67d0>,
           <matplotlib.text.Text at 0x7fc7b69f25d0>])
```




```
In [50]: df_docks3 = pd.DataFrame(tr3, columns=tr.keys)
         df_docks3

Out[50]:   CandidateName Sentiment Retweets
         0        Kasich  Negative    28981
         1        Kasich   Neutral   267516
         2        Kasich  Positive    10292
```

27 Percentage of Positive, Negative and Neutral Tweets + Retweets for Kasich

```
In [51]: tr3.pie(autopct='%.2f')

Out[51]: ([<matplotlib.patches.Wedge at 0x7fc7b6970750>,
           <matplotlib.patches.Wedge at 0x7fc7b697c610>,
           <matplotlib.patches.Wedge at 0x7fc7b6908490>],
          [<matplotlib.text.Text at 0x7fc7b6970c10>,
           <matplotlib.text.Text at 0x7fc7b697cb10>,
           <matplotlib.text.Text at 0x7fc7b6908990>],
          [<matplotlib.text.Text at 0x7fc7b697c1d0>,
           <matplotlib.text.Text at 0x7fc7b6908050>,
           <matplotlib.text.Text at 0x7fc7b6908e90>])
```




```
In [ ]: x=pd.read_csv("/home/ubuntu/data/Trump_debate1.csv", dtype=unicode,index_col=["Trump"], encodin
```

```
In [ ]: x
```

```
In [ ]: x.describe()
```

28 Sentiment Plot on a U.S map in Tableau by Candidate

```
In [193]: from IPython.display import Image

          Image(filename='Tableau-Map.PNG')
```

```
Out[193]:
```