SEPT. 2019 | VOL. 1

FUTURE TECH MAGAZINE

By Luis Aragon

Russian Twitter Bots

An analysis of over 2 million tweets

The Data:
Over the last couple years, the United States has discovered a company in Russia, the Internet Research Agency, which deployed fake twitter accounts and over 2 million tweets since 2012. Many of these accounts have since been deleted. Researchers at Clemson University, Darren Linvill and Patrick Warren, collected and classified over 2.9 million tweets which have come out of the Internet Research Agency. These tweets have been confirmed to come from Russian bots. The dataset contains information on tweets such as the username, time posted, and and the tweet text itself. (A link to the dataset can be found in the right margin of this page)

These twitter accounts have names such as 'dailysanfran', 'chicagodailynew', or 'finley1598', which seem like generic news or individual usernames. These bots release consistent content and make it difficult to distinguish them from other Twitter users. For example, the user 'dailysanfran' released news related to the Bay Area once or twice an hour. While the content of the tweets vary per user, a large portion are political, specifically referring to the 2016 presidential election. These tweets consistently mention Donald Trump, Hillary Clinton, and Barack Obama.

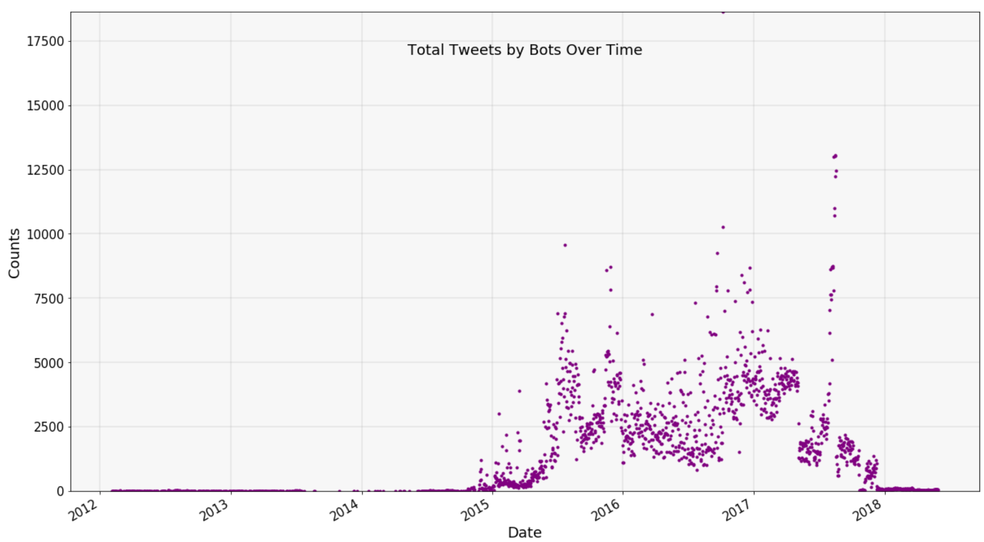
WHAT'S INSIDE THIS ISSUE?

Python
Text Mining
Text Analysis
Data Analysis

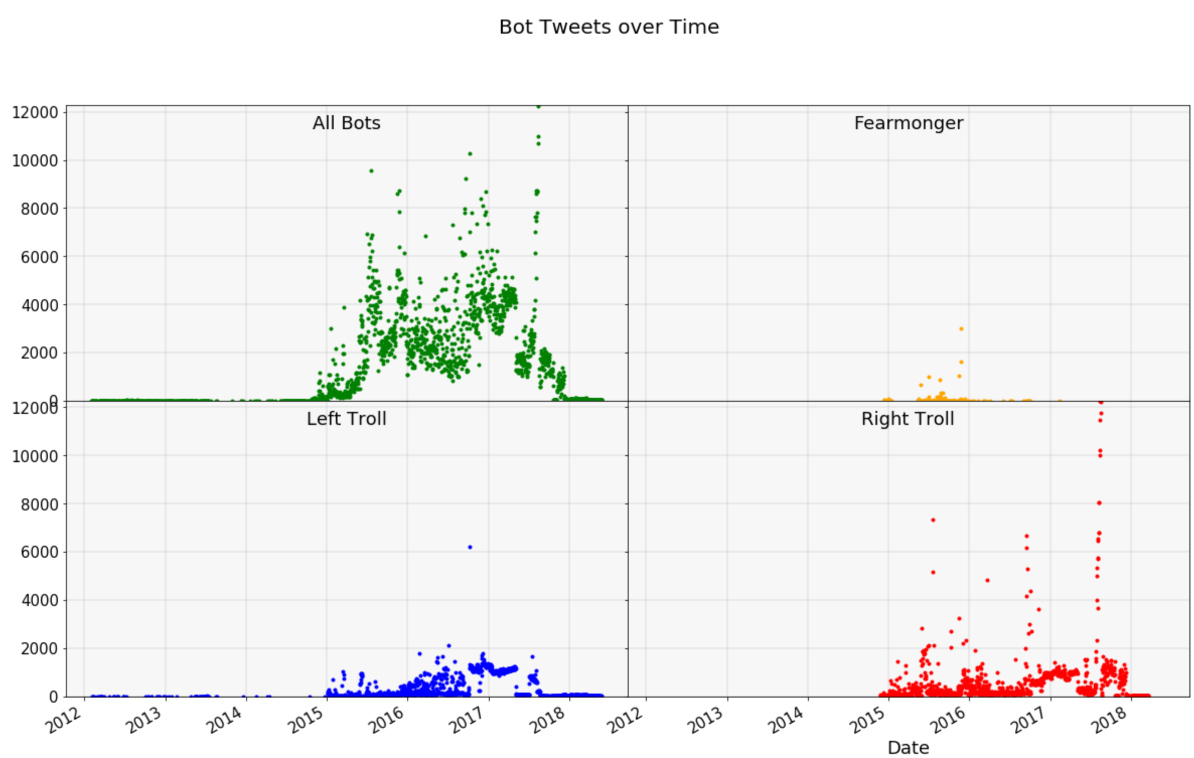
***Link to Russian Bot Tweets Dataset: https://github.com/fivethirtyeight/russian-troll-tweets/**

***Full code can be found at the end of the article**

To the right we have the total number of the tweets posted each day by the bots. We can see that the bots began to tweet heavily around 2015 and 2016, prior to the 2016 presidential election. The largest bot activity spike occurred on October 6, 2016, the day before WikiLeaks released Hillary Clinton's emails. The content of the tweets that day were not unique to the emails, but rather reflected the political tone used throughout the several years of tweets.



Linvill and Warren categorized each tweet into one of several categories. Interesting categories tweets were classified into include Left Trolls, Right Trolls, and Fear mongers. The 'Left' and 'Right' Trolls. Below we can see the distribution of tweet activities per day for each category. Warren and Linvill concluded that the Right Trolls posed as Trump supporters and aimed at talking up Republicans or talking down Democrats. The reverse was true with the Left Trolls. They were very straight forward in political content. We can see that a large number of tweets were created by Right Trolls. It is unclear the motives of the bots, but it is very clear that the tweets are very polarizing.

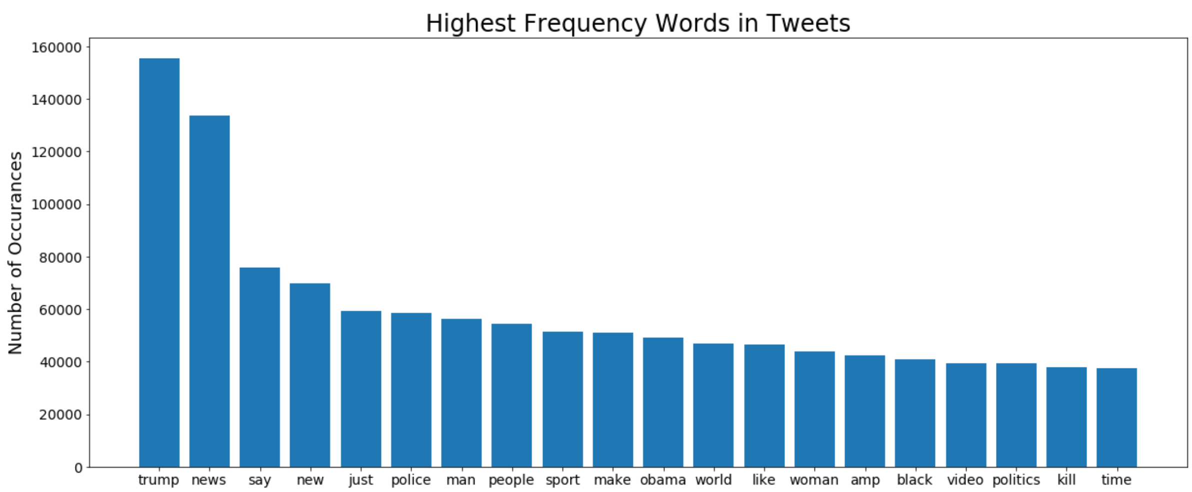


To get a deeper look into the tweets, I first normalized all 2.9 million tweets. This eliminates unnecessary words like 'the', 'at', and 'to' and extracts the core of each sentence. Below is an example of a normalized tweet.

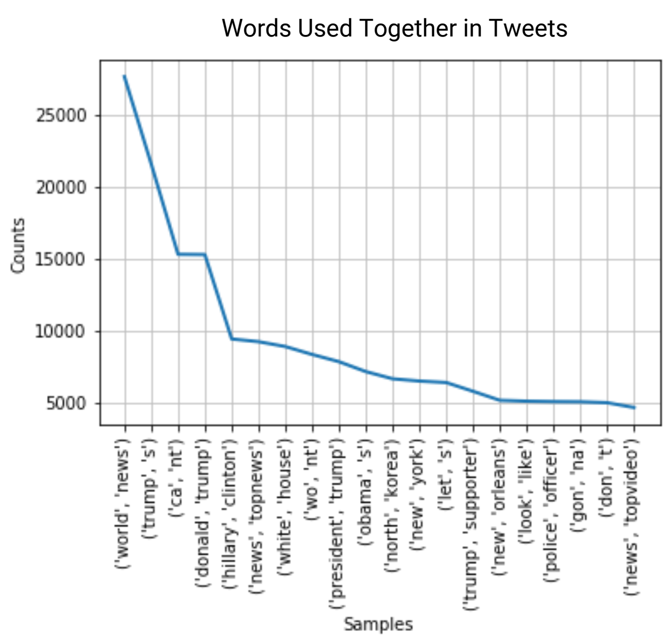
Original Tweet: Spectators flock to D.C. on the day before Donald Trump's inauguration https://t.co/hkhbtA38q3 https://t.co/35Fy7a5bHU

Normalized Tweet: spectator flock dc day donald trump s inaguration

After cleaning each tweet, I then found the 20 words used most throughout all the bot tweets. We can see that the word 'trump' and 'news' are tweeted the most. This clearly indicates that in general, the bots from the Internet Research Agency in Russia were political. Other political terms included in the tweets were 'obama' and 'politics'. The bots also repeatedly said words like 'police', 'sports', and 'news'. By looking at this graph, we can see the the Russian bots were closely linked to news and politics. The tweets ranged from objective (posing as a credible news source) to emotionally charged (outright insulting politicians).



After seeing the word frequencies, I checked the which words were commonly used together. We can see the most common two words used consecutively are 'world news'. Additionally, we can see that Donald Trump, Hillary Clinton, and Barack Obama are consistently mentioned. Again, this shows the political nature of many of the Russian bots on Twitter.



To the left is a wordcloud, another way to visualize the words frequently used by the 2.9 million Russian bots.

Taking a closer look into the words used by tweets, I randomly selected (see code below) tweets that contained "Trump" and "Obama" to see what bots were saying about the former and current president. We see that the Russian bots lean heavily against Barack Obama and are large supporters of Donald Trump

Tweets containing "Trump":

THEFOUNDINGSON, 5/4/2017: May Day protesters calling for the assassination of Trump and Pence=Free Speech. Trump voters saying a word=hate speech #LeftistLogic101

HYDDROX, 12/9/2016: Twitter has suspended Trump supporter @JaredWyand! #FreeJared immediately @Jack!

BARRLAUTRS, 8/18/2017: Idiot Jimmy Kimmel Proposes a New Way to Defeat Trump's Racism

CHICAGODAILYNEW, 1/30/2017: Chicago tech leaders come out swinging on Trump's immigration ban

DARCYYSTR, 8/12/2017 Don't piss us off... again. #2A #LockHerUp #TrumpTrain #MAGA #POTUS #TheResistance

BATONROUGEVOICE, 7/19/2016: RNC 2016: 'Duck Dynasty's' Willie Robertson says Trump will have Americans' back #politics

Tweets containing "Obama":

MONEYFORM, 6/20/2015: Obama blames all these mass shootings on the guns. Why not on the complete lack of care and follow-up for the mentally ill? 12 @u2runfar

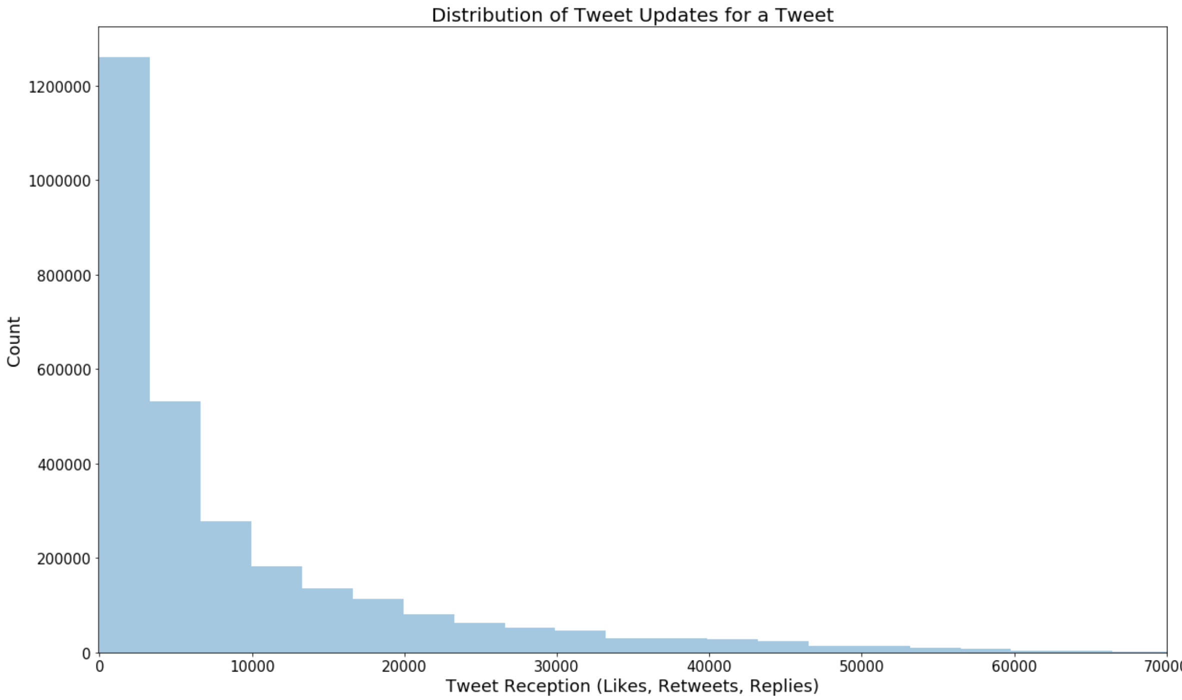
FINLEY1589, 9/17/2016: Our Allies Watch in Horror as Obama Switches Sides, Now Allied with Enemy Iran

EVEWEBSTER373, 12/16/2015: .@HillaryClinton You are giving into the terrorist just like Obama ... you are a disgrace to our nation. #NoHillary2016

BRONICKIJEWEL, 8/13/2015 16:03 : #TrumpBecause It's time for @BarackObama and @HillaryClinton to go quietly into the night #MakeAmericaGreatAgain

ARM_2_ALAN 7/22/2015: Green Beret Saves Lives by Shooting a Terrorist, Here's How Obama's CIA "Thanked" Him

HAPPKENDRAHAPPY, 12/30/2016: Poll: Putin or Obama? I honestly think Putin is a better world leader, smarter, and far more competent leader than Obama! Who do u like more



Many of these tweets are polarizing, so I looked into the influence of the tweets. Above, is a graph showing the amount of distribution of tweet responses including the number of likes, retweets and replies. We can see that the majority of the tweets create very little interaction. Surprisingly, the calculated median of tweet response 4361. This means that half of the bot tweets had less than 4361 likes, retweets, and replies, while the other half had more than 4361. This raises the question if what percentage of the responses were from bots and what percentage of the responses were from the general public.

Not only were the tweets divisive, the tweets lied. Warren and Linvill classified certain tweets as Fearmongers, which aim to lie and cause fear. These are not as prevalent as the political tweets. Below are tweets referring to Turkey poisoning during Thanksgiving, which may have cause fear among the general public.

Fear monger tweets:

TERRYBEVER_LY, 11/27/2015: Happy Thanksgiving @UNMC_DrKhan Hell, maybe I'm poisoned too? #USDA

CALVIN_YEK, 11/27/2015: #ImThankfulFor This is horrible #KochFarms #FSIS #USDA'

TOLLIVERSTEPH , 11/27/2015 0:41 : '@DrLennyDavidman I'm not surprised that #Kochfarms doesnt care about quality of their products... Everything Kochs worried about is #money'

We can see to the right that the tweets were in majority English, but a significant amount were in Russian and German. This indicates that the bots had a larger reach that just the United States and when created with intention, bots can infiltrate most parts of the world with social media.

Tweets by Language (top 5):

English: 2,116,867
Russian: 610,943
German: 86,983
Ukranian: 38,669
Italian: 18,063

The dataset contains rich information of Russian Twitter bots and it is clear that the bots were political and imitated Donald Trump supporters before and after the 2016 election. "Trump" was the most frequent word throughout all 2.9 million tweets. As a whole, the majority of the bots pushed for and supported Trump's presidency. Furthermore, these profiles were created to resemble a person or news site and consistently released tweets associated with the persona created. It is highly likely that the this was done through machine learning. For instance, someone could scrape all the tweets from thousands of soccer fans, learn from the text data, and create a bot which systematically tweets about soccer everyday. Upon collecting this data, Warren and Linvill categorized these bots into one of eight categories: Right Troll, Left Troll, Fearmonger, Commercial, News Feed, Hashtag Gamer, Non-English, and Unknown.

Even though the Internet Research Agency released over 3500 bots, which skewed right on the political spectrum, it is still unclear how much of an influence these bots had on the 2016 election and on the views of Americans. It is important to note that this agency was able to accomplish this feat with around 400 employees. At this moment, we do not know how many more bots are influencing people. Although many might be benign bots, it is still unsettling knowing that our social media is vulnerable and can be taken advantage of to achieve malicious political, social, or economic goals. Twitter and social media are constantly growing and having a greater reach around the globe, so it is important that we protect ourselves from malicious bots and, as individuals, be aware of these issues as we engage with social media moving forward.

Sources:

Roeder, Oliver. “Why We're Sharing 3 Million Russian Troll Tweets.” FiveThirtyEight, FiveThirtyEight, 31 July 2018, https://fivethirtyeight.com/features/why-were-sharing-3-million-russian-troll-tweets/.

Data:

Fivethirtyeight. “Fivethirtyeight/Russian-Troll-Tweets.” GitHub, 28 Aug. 2018, https://github.com/fivethirtyeight/russian-troll-tweets/.

twitter_code

September 1, 2019

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import glob
import seaborn as sns
from datetime import datetiem
```

In [ ]:
```
# Read in all the data
d = {}

for filename in glob.glob('russian-troll-tweets-master/*.csv'):
    d[filename[:-4]] = pd.read_csv(filename, header=0)
```

In [ ]:
```
# Create Dataframe
data = pd.concat(d.values())
```

In [ ]:
```
# View first few rows
data.head()
```

In [ ]:
```
# Size of dataframe
data.shape
```

In [ ]:
```
# Plot tweets function
def plot_tw(d, col, title, ax=None):

    if ax is None:
        ax = plt.gca()

    time_list = []
    for item in d['publish_date']:
        old_date = datetime.strptime(item, '%m/%d/%Y %H:%M')
        new_date = old_date.date()
        time_list.append(new_date)

    df = pd.DataFrame(time_list)

    plotting_df = df.stack().value_counts().reset_index(name='counts')
```

```
    ax.plot_date(plotting_df['index'], plotting_df['counts'], fmt='.', color=col)
    ax.set_ylim(0, plotting_df['counts'].values.max()+1)
    fig.autofmt_xdate()
    plt.xlabel('Date', fontsize=18)
    plt.ylabel('Counts', fontsize=18)
    ax.grid(color='grey', linestyle='-', linewidth=0.25, alpha=0.9)
    ax.set_facecolor("#f7f7f7")
    ax.set_title(title, y=0.9, fontsize=18)
    # We change the fontsize of minor ticks label
    ax.tick_params(axis='both', which='major', labelsize=15)
    ax.tick_params(axis='both', which='minor', labelsize=15)

    return ax

# Plot total tweets over time
fig, ax = plt.subplots(figsize=(20,12))
plot_tw(data, 'purple', 'Total Tweets by Bots Over Time',ax)
```

In [ ]:
```
# packages need for text mining
import nltk
nltk.download('stopwords')
nltk.download('punkt') # tockenizer
nltk.download('averaged_perceptron_tagger')
nltk.download('wordnet')

# import necessary modules
import re
import string
from nltk.stem import WordNetLemmatizer

wnl = WordNetLemmatizer()
```

In [ ]:
```
# Tokenize words
def tokenize_text(text):
    tokens = nltk.word_tokenize(text)
    tokens = [token.strip() for token in tokens]
    return tokens
```

In [ ]:
```
# Remove special characters
def remove_special_characters(text):
    tokens = tokenize_text(text)
    pattern = re.compile('[{}]'.format(re.escape(string.punctuation)))
    filtered_tokens = filter(None, [pattern.sub('', token) for token in tokens])
    filtered_text = ' '.join(filtered_tokens)
    return filtered_text
```

In [ ]:
```
# Remove stopwords
from sklearn.feature_extraction.stop_words import ENGLISH_STOP_WORDS
def remove_stopwords(text,stopword_list):
```

```
    tokens = tokenize_text(text)
    filtered_tokens = [token for token in tokens if token not in stopword_list]
    filtered_text = ' '.join(filtered_tokens)
    return filtered_text
```

In [ ]:
```
# Remove twitter link
def remove_custom(text):
    sep = 'htt'
    rest = text.split(sep, 1)[0]
    return rest
```

In [ ]:
```
from nltk import pos_tag
from nltk.corpus import wordnet as wn

# Annotate text tokens with POS tags
def pos_tag_text(text):

    def penn_to_wn_tags(pos_tag):
        if pos_tag.startswith('J'):
            return wn.ADJ
        elif pos_tag.startswith('V'):
            return wn.VERB
        elif pos_tag.startswith('N'):
            return wn.NOUN
        elif pos_tag.startswith('R'):
            return wn.ADV
        else:
            return None

    tagged_text = pos_tag(text)
    tagged_lower_text = [(word.lower(), penn_to_wn_tags(pos_tag))
                         for word, pos_tag in
                         tagged_text]
    return tagged_lower_text
```

In [ ]:
```
# lemmatize text based on POS tags
def lemmatize_text(text):
    text = tokenize_text(text)
    pos_tagged_text = pos_tag_text(text)
    lemmatized_tokens = [wnl.lemmatize(word, pos_tag) if pos_tag
                         else word
                         for word, pos_tag in pos_tagged_text]
    lemmatized_text = ' '.join(lemmatized_tokens)
    return lemmatized_text
```

In [ ]:
```
# Text normalization pipeline
from sklearn.feature_extraction.stop_words import ENGLISH_STOP_WORDS
import re
```

```
def keep_text_characters(text):
    filtered_tokens = []
    tokens = tokenize_text(text)
    for token in tokens:
        if re.search('[a-zA-Z]', token):
            filtered_tokens.append(token)
    filtered_text = ' '.join(filtered_tokens)
    return filtered_text

# Main normalize text function
def normalize_text(text,tokenize=False):
    text = lemmatize_text(text)
    text = remove_special_characters(text)
    text = text.lower()
    text = remove_stopwords(text,ENGLISH_STOP_WORDS)
    text = keep_text_characters(text)
    text = remove_custom(text)

    return text
```

In [ ]:
```
# Get tweets text
tweets = data['content'].dropna()
```

In [ ]:
```
# Example of normalization
print("Example sentence:", tweets.iloc[101])
print("Normalized sentence:", normalize_text(tweets.iloc[101]))
```

In [ ]:
```
# Normalize all the tweets (takes several hours)
normalized_list = []

for word_entry in tweets:
    normalized_list.append(normalize_text(word_entry))

# Save in file
with open('norm.txt', 'w') as f:
    for item in normalized_list:
        f.write("%s\n" % item)
```

In [ ]:
```
# Get list from file
with open('norm.txt', 'r') as f:
    mylist = f.read().split('\n')
```

In [ ]:
```
from nltk.tokenize import word_tokenize

# Split normalized tweets for single words
single_word_list = []

for entry in mylist:
    tk = word_tokenize(entry)
```

```
    for i in tk:
        single_word_list.append(i)

single_word_list[0:10]
```

In [ ]:
```
# Finds most frequent words
fdist = nltk.FreqDist(single_word_list)

try:
    del fdist["s"]
    del fdist['nt']
    del fdist['rt']
except KeyError:
    print("Key not found")

fdist
```

In [ ]:
```
# bar chart with count of most common words

x, y = zip(*fdist.most_common(n=20)) # Unzip the tuples into lists
plt.figure(figsize=(20,8))
plt.bar(range(len(x)), y)
plt.xticks(range(len(x)), x)
plt.tick_params(axis='both', which='major', labelsize=14)
plt.tick_params(axis='both', which='minor', labelsize=16)
plt.title("Highest Frequency Words in Tweets", fontsize = 24)
plt.ylabel("Number of Occurances", fontsize=18)
plt.show()
```

In [ ]:
```
# Find tweets with word 'Trump'

data_subset = data.sample(n = 100, random_state = 122)

for index, row in data_subset.iterrows():
    if ('trump' in row['content'] or 'Trump' in row['content']):
        print(row['author'], ',', row['publish_date'], ':', row['content'], '\n')
```

In [ ]:
```
# Find tweets with word 'Obama'

data_subset2 = data.sample(n = 500, random_state = 122)

for index, row in data_subset2.iterrows():
    if ('Obama' in row['content'] or 'obama' in row['content']):
        print(row['author'], ',', row['publish_date'], ':', row['content'], '\n')
```

In [ ]:
```
# View different categories
print(data['account_category'].nunique())
data.groupby('account_category').agg('count')
```

```
In [ ]: # Subset data based on three groups
        fearmonger = data.loc[data['account_category'] == 'Fearmonger']
        leftTroll = data.loc[data['account_category'] == 'LeftTroll']
        rightTroll = data.loc[data['account_category'] == 'RightTroll']
```

```
In [ ]: # Plot tweets throughout time per group

        fig, axs = plt.subplots(2, 2, sharex=True, sharey=True,
                                gridspec_kw={'hspace': 0, 'wspace': 0},
                                figsize=(20,12))

        (ax1, ax2), (ax3, ax4) = axs
        fig.suptitle('Bot Tweets over Time', fontsize = 20)
        plot_tw(data, 'green', 'All Bots', ax1)
        plot_tw(fearmonger, 'orange', 'Fearmonger', ax2)
        plot_tw(leftTroll, 'blue', 'Left Troll', ax3)
        plot_tw(rightTroll, 'red', 'Right Troll', ax4)

        for ax in axs.flat:
            ax.label_outer()
```

```
In [ ]: # Plot biplot
        biTradeWords = nltk.bigrams(single_word_list)
        biFdist = nltk.FreqDist(biTradeWords)
        print(biFdist.most_common(10))
        biFdist.plot(20, cumulative=False)
        plt.xticks(fontsize=20)
```

```
In [ ]: # Plot histogram of tweet responses

        fig, ax = plt.subplots(figsize=(20,12))

        update_list = data.groupby('updates')['tweet_id'].count()

        ax = sns.distplot(data['updates'], kde=False)
        ax.set_title("Distribution of Tweet Updates for a Tweet", fontsize = 20)
        plt.xlabel('Tweet Reception (Likes, Retweets, Replies)', fontsize=18)
        plt.ylabel('Count', fontsize=18)
        ax.tick_params(axis='both', which='major', labelsize=15)
        ax.tick_params(axis='both', which='minor', labelsize=15)
        ax.set_xlim(-100, 70000)
```

```
In [ ]: # Find median of tweet responses
        np.median(data['updates'])
```

```
In [ ]: # Check to see if tweets on October 6 were unique

        # Get times
        time_list2 = []
```

```
for item in data['publish_date']:
    old_date = datetime.strptime(item, '%m/%d/%Y %H:%M')
    new_date = old_date.date()
    time_list2.append(new_date)

import datetime

# Get time equal to Oct 6, 2016
indexes = []
for i in range(len(time_list2)):
    if (time_list2[i] == datetime.date(2016, 10, 6)):
        indexes.append(i)

import random
random.seed(122)
idx_subset = random.sample(indexes, 200)

# View tweets on that day
for i in idx_subset:
    print(data['author'].iloc[i], ',', data['publish_date'].iloc[i], ':', data['conten
```

In [ ]:
```
# View tweets from category Hashtag Gamer

data_subset4 = data.sample(n = 2000, random_state = 122)

for index, row in data_subset4.iterrows():
    if (row['account_category'] == 'HashtagGamer'):
        print(row['author'], ',', row['publish_date'], ':', row['content'], '\n')
```

In [ ]:
```
# Plot wordcloud

from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator

wordcloud = WordCloud(background_color="white").generate(" ".join(single_word_list))

plt.figure()
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```

In [ ]:
```
# Find tweets from category fearmonger

data_subset_fear = fearmonger.sample(n = 5, random_state = 1234)

for index, row in data_subset_fear.iterrows():
    print(row['author'], ',', row['publish_date'], ':', row['content'], '\n')
```

In [ ]:
```
# View data by language
data.groupby('language').count()
```