INDEPENDENT STUDY RESEARCH REPORT

Name: Prithvi Shetty

Professor: Benjamin Althouse

Topic: Sentiment analysis of Ebola datasets

IMT 600
Prithvi Shetty

INDEX:

IMT 600
Prithvi Shetty

Goal: It all started with the research project under my Data Science Professor, Benjamin Althouse at the University of Washington where my main goal was to perform sentiment analysis on million records of a dataset of Ebola disease and find the social drivers of Ebola disease.

About the dataset:

This dataset comprised of more than a million records which involved responses to more than 30 survey questions for more than 20,000 subjects in West Africa during the years 2014–2015 for Ebola epidemic.

Phases: There were roughly 5 stages involved in this research project.

1. Data cleaning
2. Analysis/count of words
3. Classifier
4. Results
5. Future scope/Conclusion

I) Data Cleaning

The survey responses had numerous spelling mistakes and the whole data as a whole had many corrupt/null values.

Majority of the efforts (almost 60%) went in cleaning the data in the proper format with the required null handling.

Python code to clean the data:

```
t=pd.read_csv(‘clean_other_trigger.csv’)

t.isnull().sum()

def myfillna(series):
 if series.dtype is pd.np.dtype(float):
 return series.fillna(‘’)
 elif series.dtype is pd.np.dtype(int):
 return series.fillna(‘’)
 else:
 return series.fillna(‘NA’)

t=t.apply(myfillna)

#Remove punctuations
t[‘t_q6’] = t[‘t_q6’].apply(lambda x:’’.join([i for i in x
 if i not in string.punctuation]))
t[‘t_q7’] = t[‘t_q7’].apply(lambda x:’’.join([i for i in x
 if i not in string.punctuation]))
t[‘t_q8’] = t[‘t_q8’].apply(lambda x:’’.join([i for i in x
 if i not in string.punctuation]))
t[‘t_q9’] = t[‘t_q9’].apply(lambda x:’’.join([i for i in x
 if i not in string.punctuation]))
t[‘t_q10’] = t[‘t_q10’].apply(lambda x:’’.join([i for i in x
 if i not in string.punctuation]))
t[‘t_q11’] = t[‘t_q11’].apply(lambda x:’’.join([i for i in x
 if i not in string.punctuation]))
```

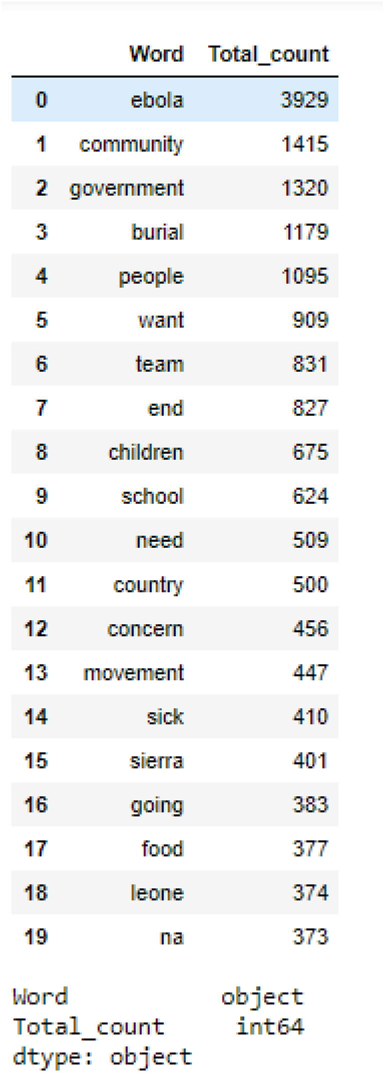
II) Analysis/Count of words and training the model:

This involved creating a function to correct all the spellings of all the survey records and then analysis and training the data using CountVectorizer. I removed all the stop words while analysing the words in the survey responses thus resulting in the reduction of features.

Python Code:

```
def f(spacedfile):
 chkr = SpellChecker(“en_UK”,”en_US”)
 chkr.set_text(spacedfile)
 for err in chkr:
 sug = err.suggest()[0]
 err.replace(sug)
 Spellchecked = chkr.get_text()
 return(Spellchecked)

f(t.t_q6[0])

from sklearn.feature_extraction.text import CountVectorizer
vect=CountVectorizer(stop_words=’english’)
vect.fit(t.t_q6)
vect.get_feature_names()
t_q6_dtm=vect.transform(t.t_q6)
t_q6_dtm.toarray()
pd.DataFrame(t_q6_dtm.toarray(),columns=vect.get_feature_names())
```

The following was the list of most frequent words for the first survey question after training it on CountVectorizer:



	Word	Total_count
0	ebola	3929
1	community	1415
2	government	1320
3	burial	1179
4	people	1095
5	want	909
6	team	831
7	end	827
8	children	675
9	school	624
10	need	509
11	country	500
12	concern	456
13	movement	447
14	sick	410
15	sierra	401
16	going	383
17	food	377
18	leone	374
19	na	373

```
Word            object
Total_count      int64
dtype: object
```

Count of words

III) Classifier

Later, I built a simple classifier which marks sentences negative and positive using TextBlob function.

- 1 polarity indicates the most negative sentence possible while +1 indicates the most positive sentiment possible.
- I analysed some of the most extreme end sentiments and tried to extract meaning out of it.

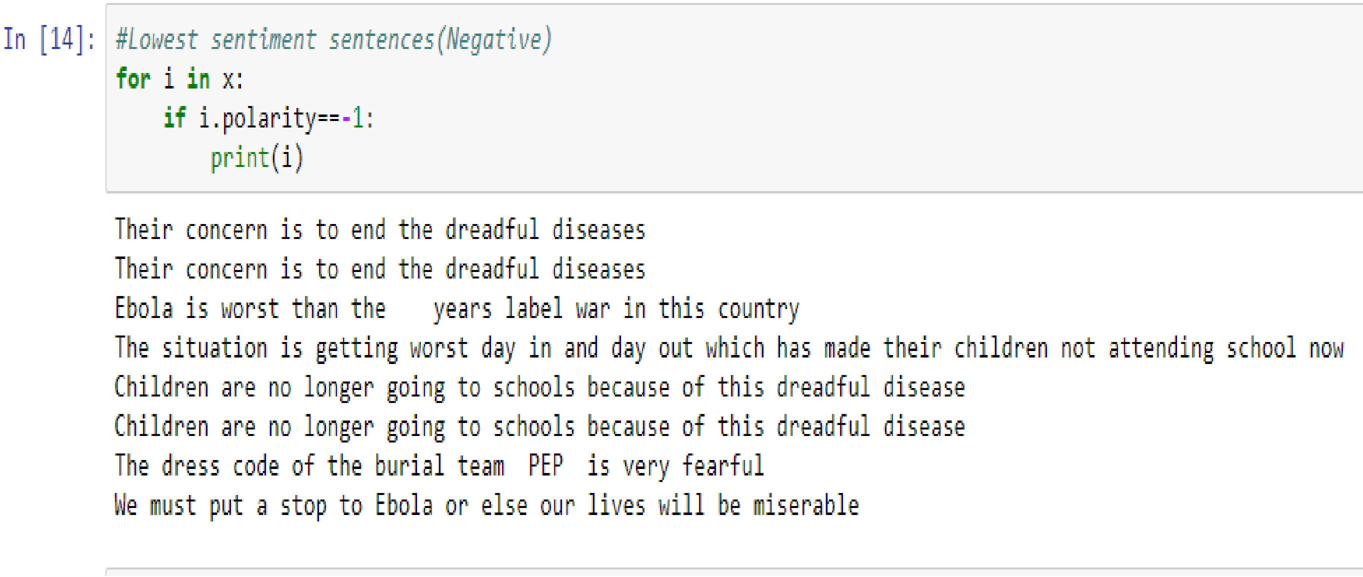
Python code:

```
from textblob import TextBlob

x=[]
for s in a:
 x.append(TextBlob(s))
```

IMT 600
Prithvi Shetty

```
y=[]
for i in x:
 y.append(i.polarity)

np.mean(y)
#This indicates slightly positive sentiment

#Highest sentiment sentences(Positive)
for i in x:
 if i.polarity==1:
 print(i)

#Lowest sentiment sentences(Negative)
for i in x:
 if i.polarity==-1:
 print(i)

#Neutral sentiment count
d=0
for i in x:
 if i.polarity==0:
 d+=1
```

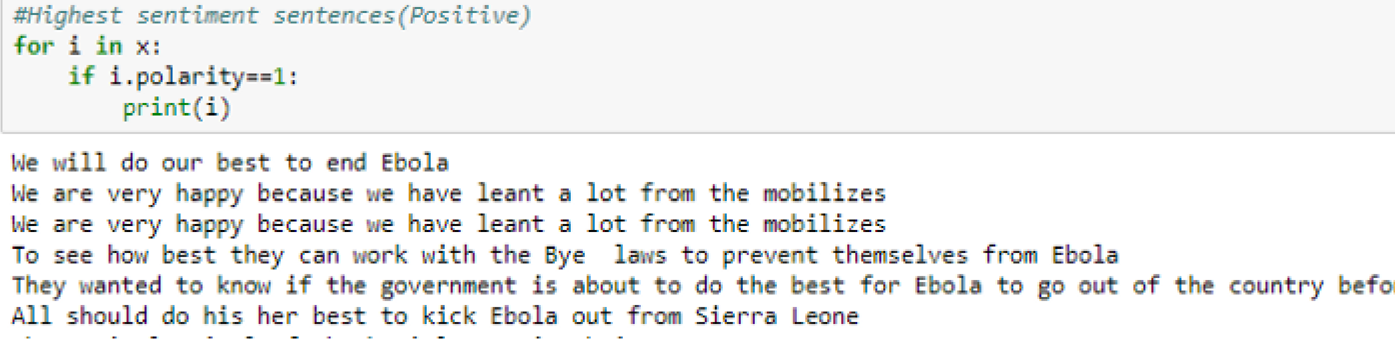
a) -1 polarity negative sentiment sentences



In [14]:
```
#Lowest sentiment sentences(Negative)
for i in x:
    if i.polarity==-1:
        print(i)
```

```
Their concern is to end the dreadful diseases
Their concern is to end the dreadful diseases
Ebola is worst than the    years label war in this country
The situation is getting worst day in and day out which has made their children not attending school now
Children are no longer going to schools because of this dreadful disease
Children are no longer going to schools because of this dreadful disease
The dress code of the burial team  PEP  is very fearful
We must put a stop to Ebola or else our lives will be miserable
```

b) +1 polarity Positive sentiment sentences



```
#Highest sentiment sentences(Positive)
for i in x:
    if i.polarity==1:
        print(i)
```

```
We will do our best to end Ebola
We are very happy because we have leant a lot from the mobilizes
We are very happy because we have leant a lot from the mobilizes
To see how best they can work with the Bye  laws to prevent themselves from Ebola
They wanted to know if the government is about to do the best for Ebola to go out of the country befo
All should do his her best to kick Ebola out from Sierra Leone
```

The negative sentiment sentences indicate that the education has reduced to a great extent due to Ebola as a lot of children have stopped going to school and the situation is getting worse. On the other hand, the positive sentiment sentences indicate that there is still hope among the people in the Ebola affected areas and they are trying their best to eradicate the disease.

IV) Results

A) Sentiment analysis over time

I plotted the sentiment grouped by month of the year and analysed the trends of the same.

Code:

t.groupby(t.Trig_date.dt.month).t_q6v.mean()

t.groupby(t.Trig_date.dt.month).t_q6v.mean().plot()

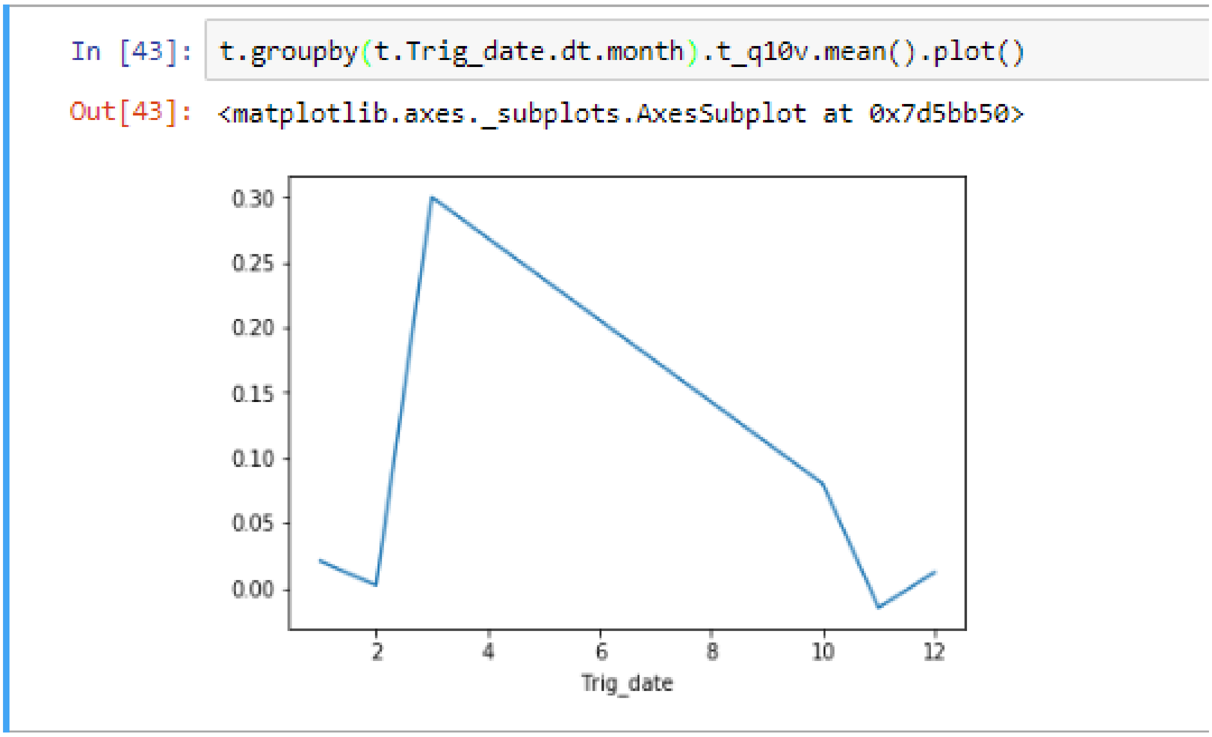
t.groupby(t.Trig_date.dt.month).t_q7v.mean()

t.groupby(t.Trig_date.dt.month).t_q7v.mean().plot()

t.groupby(t.Trig_date.dt.month).t_q10v.mean().plot()

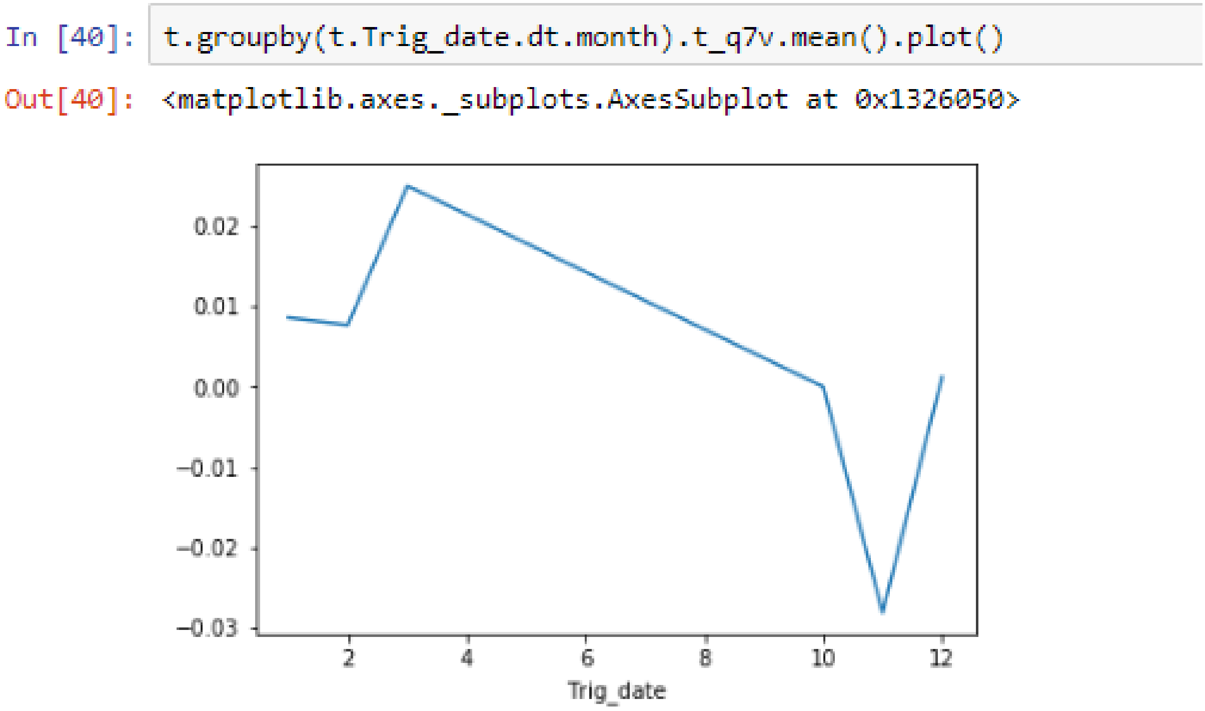
IMT 600
Prithvi Shetty

1) **Analysis of sentiment by time for Survey question: What else did you hear in the community discussions that you think is important to note?**



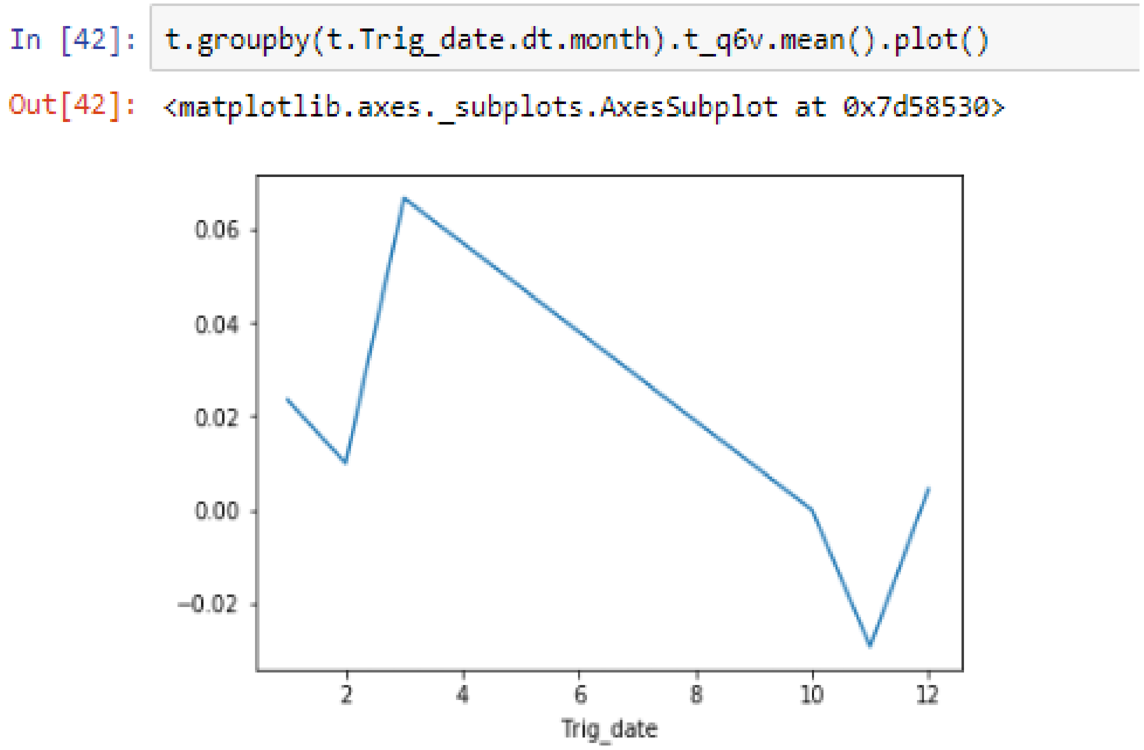
```
In [43]: t.groupby(t.Trig_date.dt.month).t_q10v.mean().plot()

Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x7d5bb50>
```

2) **Analysis of sentiment by time for Survey question: Most commonly asked questions about Ebola?**



```
In [40]: t.groupby(t.Trig_date.dt.month).t_q7v.mean().plot()

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x1326050>
```

3) **Analysis of sentiment by time for Survey question: Most common concerns about Ebola?**



```
In [42]: t.groupby(t.Trig_date.dt.month).t_q6v.mean().plot()

Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x7d58530>
```

The plots showed a similar for the sentiments vs month of the year showing the most negative sentiment in the month of November and the most positive sentiments in the month of March.

B) Sentiment analysis over district

Next, what I did was I plotted the sentiment grouped by District and analysed the trends of the same.

Code:

#Sentiment analysis by responses to 'Most commonly asked questions' by District

t.groupby('District').t_q7v.mean()

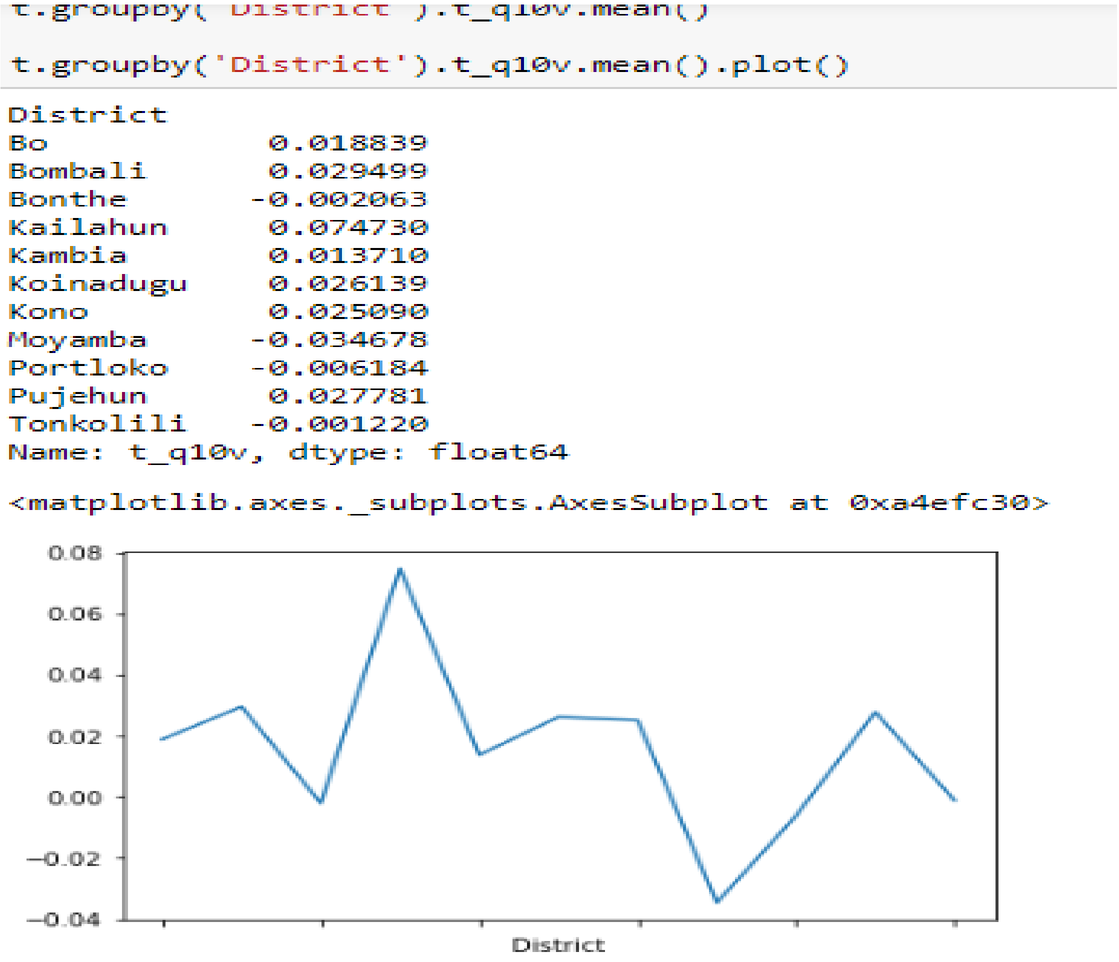
t.groupby('District').t_q7v.mean().plot()

# In[92]:

#Sentiment analysis by responses to 'What else did you hear in the community discussions that you think is important to note?'
#by District

```
t.groupby('District').t_q10v.mean()

t.groupby('District').t_q10v.mean().plot()
```

1) Analysis of sentiment for Survey question: What else did you hear in the community discussions that you think is important to note?

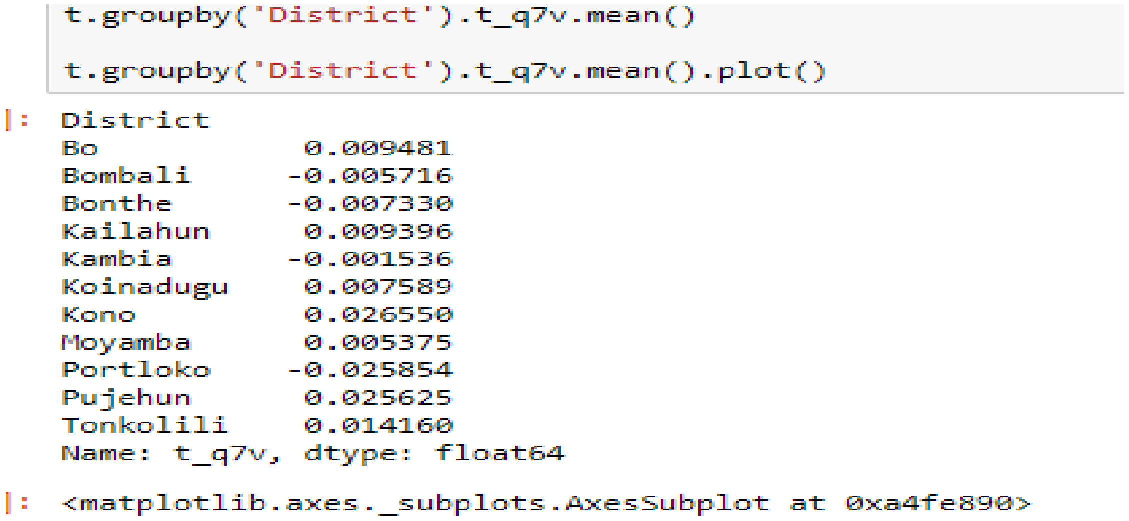


```
t.groupby('District').t_q10v.mean()

t.groupby('District').t_q10v.mean().plot()
```

```
District
Bo             0.018839
Bombali        0.029499
Bonthe        -0.002063
Kailahun       0.074730
Kambia         0.013710
Koinadugu      0.026139
Kono           0.025090
Moyamba       -0.034678
Portloko      -0.006184
Pujehun        0.027781
Tonkolili     -0.001220
Name: t_q10v, dtype: float64
```

```
<matplotlib.axes._subplots.AxesSubplot at 0xa4efc30>
```

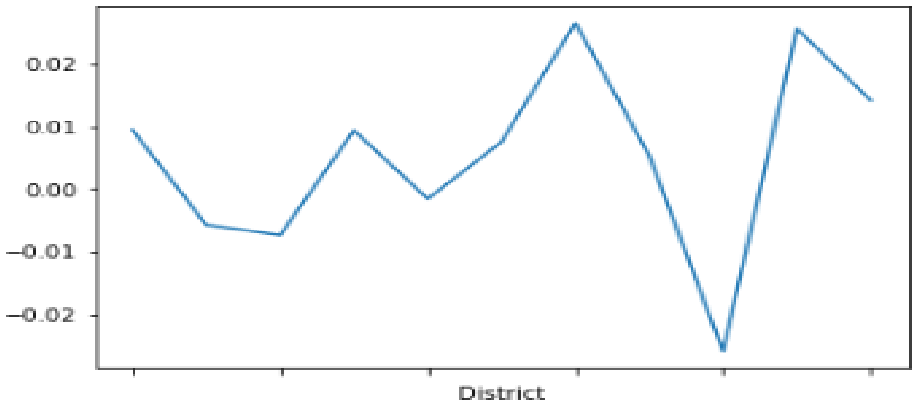
2) Analysis of sentiment for Survey question: Most commonly asked questions about Ebola disease?



```
t.groupby('District').t_q7v.mean()

t.groupby('District').t_q7v.mean().plot()
```

```
]:  District
    Bo            0.009481
    Bombali      -0.005716
    Bonthe       -0.007330
    Kailahun      0.009396
    Kambia       -0.001536
    Koinadugu     0.007589
    Kono          0.026550
    Moyamba       0.005375
    Portloko     -0.025854
    Pujehun       0.025625
    Tonkolili     0.014160
    Name: t_q7v, dtype: float64
```

```
]:  <matplotlib.axes._subplots.AxesSubplot at 0xa4fe890>
```



The plots showed pretty much the same results throughout with District of Port Loko showing the least amount of sentiment. This hold true as well as the mortality rate of Ebola was the highest in the state of Port Loko.

C) Sentiment analysis mean distribution.

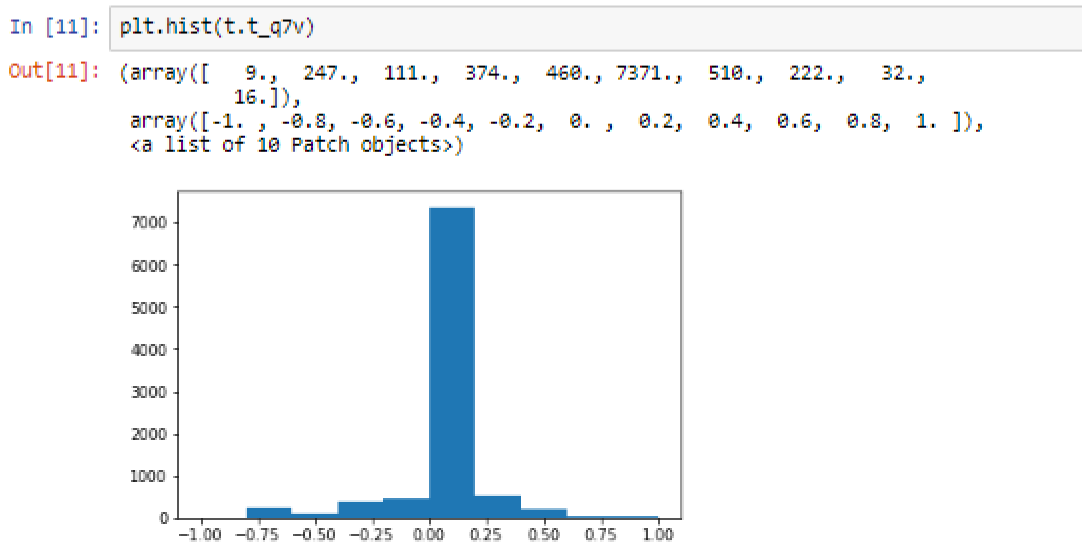
The mean distribution show that most of the sentiments are concentrated in the neutral region or slightly positive sentiment region. This indicates the positive/negative sentiment is not strong enough to classify the sentences and the need for an emotional classifier is felt. The following is the histogram plot for the sentiments of the three questions:

1)Most commonly asked questions about Ebola?

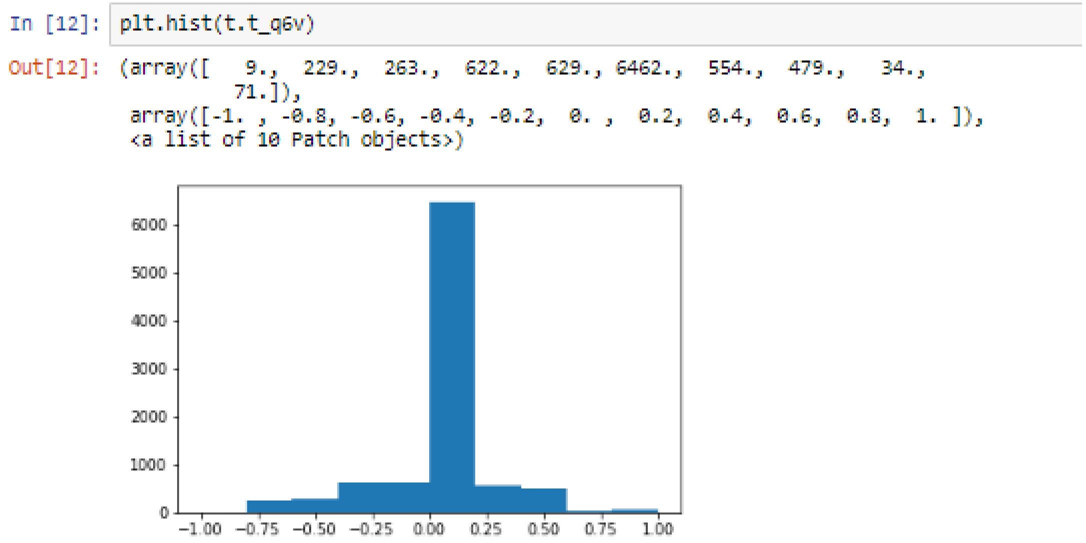
2)Most common concerns about the Ebola disease?

3)What else did you hear in the community discussions about Ebola that you think is important to note?

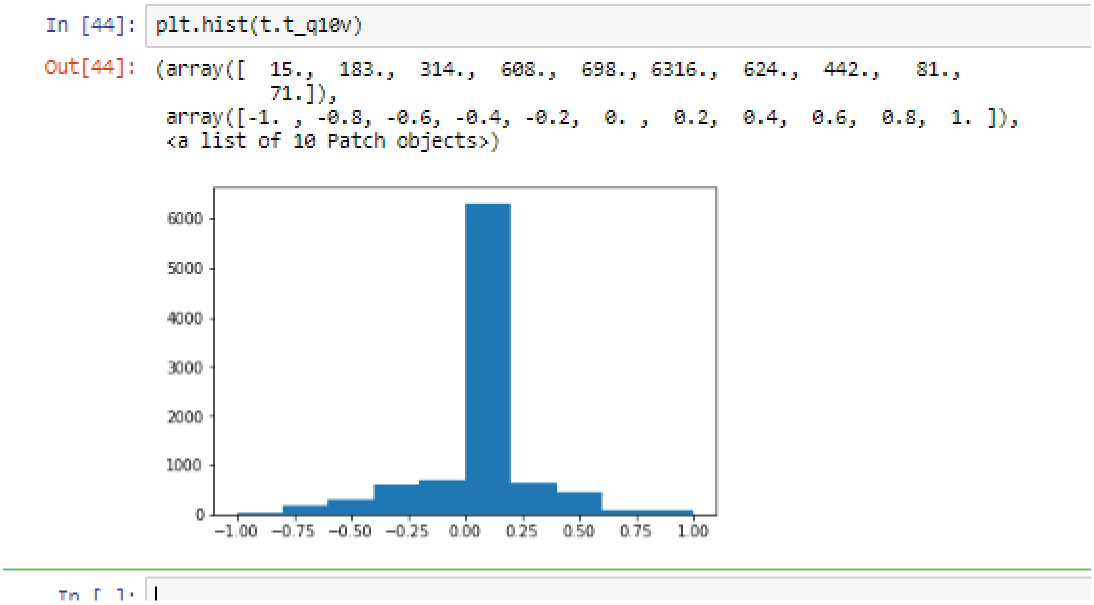
IMT 600
Prithvi Shetty



```
In [11]: plt.hist(t.t_q7v)

Out[11]: (array([   9.,  247.,  111.,  374.,  460., 7371.,  510.,  222.,   32.,
           16.]),
          array([-1. , -0.8, -0.6, -0.4, -0.2,  0. ,  0.2,  0.4,  0.6,  0.8,  1. ]),
          <a list of 10 Patch objects>)
```



```
In [12]: plt.hist(t.t_q6v)

Out[12]: (array([   9.,  229.,  263.,  622.,  629., 6462.,  554.,  479.,   34.,
           71.]),
          array([-1. , -0.8, -0.6, -0.4, -0.2,  0. ,  0.2,  0.4,  0.6,  0.8,  1. ]),
          <a list of 10 Patch objects>)
```



```
In [44]: plt.hist(t.t_q10v)

Out[44]: (array([  15.,  183.,  314.,  608.,  698., 6316.,  624.,  442.,   81.,
           71.]),
          array([-1. , -0.8, -0.6, -0.4, -0.2,  0. ,  0.2,  0.4,  0.6,  0.8,  1. ]),
          <a list of 10 Patch objects>)

In [ ]:
```

V) Future Scope/ Conclusion:

Even though there is a lot of future scope involved in this analysis, the key takeaways from this project are:

1. Data cleaning though requires a lot of effort is really critical for the correct analysis of the dataset.
2. The highest count of words can be used to portray a lot of meaningful information with respect to the specific feature.
3. The extremely positive and negative sentiment sentences gives us a lot of information about the problems in the Ebola affected areas and also, the progressive measures in the eradication of the disease.
4. Sentiments can be easily mapped with region and time and great meaningful insights can be plotted from it.
5. The histogram distribution of the sentiments give us an idea of the mean of the sentiments along with the variance of sentiments in the same.

The main limitations of this research project were:

a) Even though, the data was cleaned properly, there were some words such 'stanger' which were corrected to 'stringer' instead of stranger.

b) The positive and the negative sentiments do give us a lot of information but the project could be widely scopened if the scope of the classifiers is increased to an emotional classifier mainly anger, hate, jealousy, happiness, excitement and many others.

c)Sentences such as 'It is too good to be bad' is classified incorrectly due to clash of the sentiment words. This could be potential scope for improvement in the research project.

c)Sarcasm detection: Sarcastic sentences have a hard time finding a correct position in the sentiment analysis as they can be misleading for a classifier to analyse.