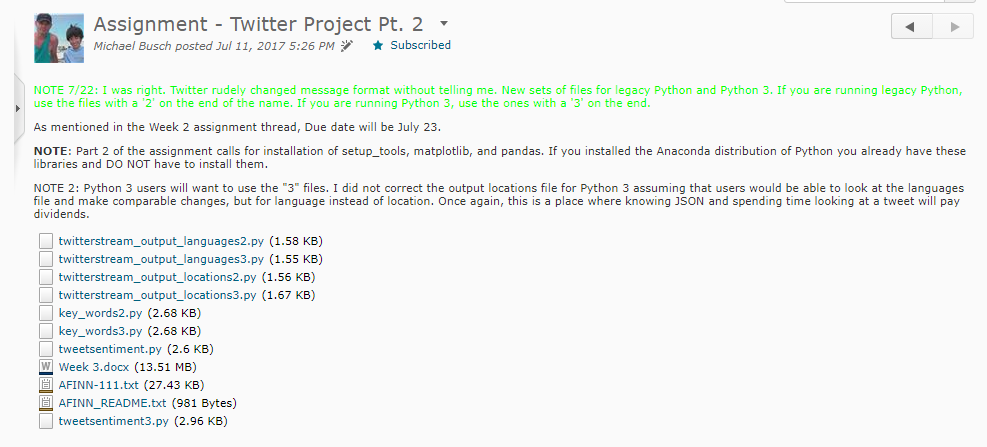
**Week 3 Assignment:**

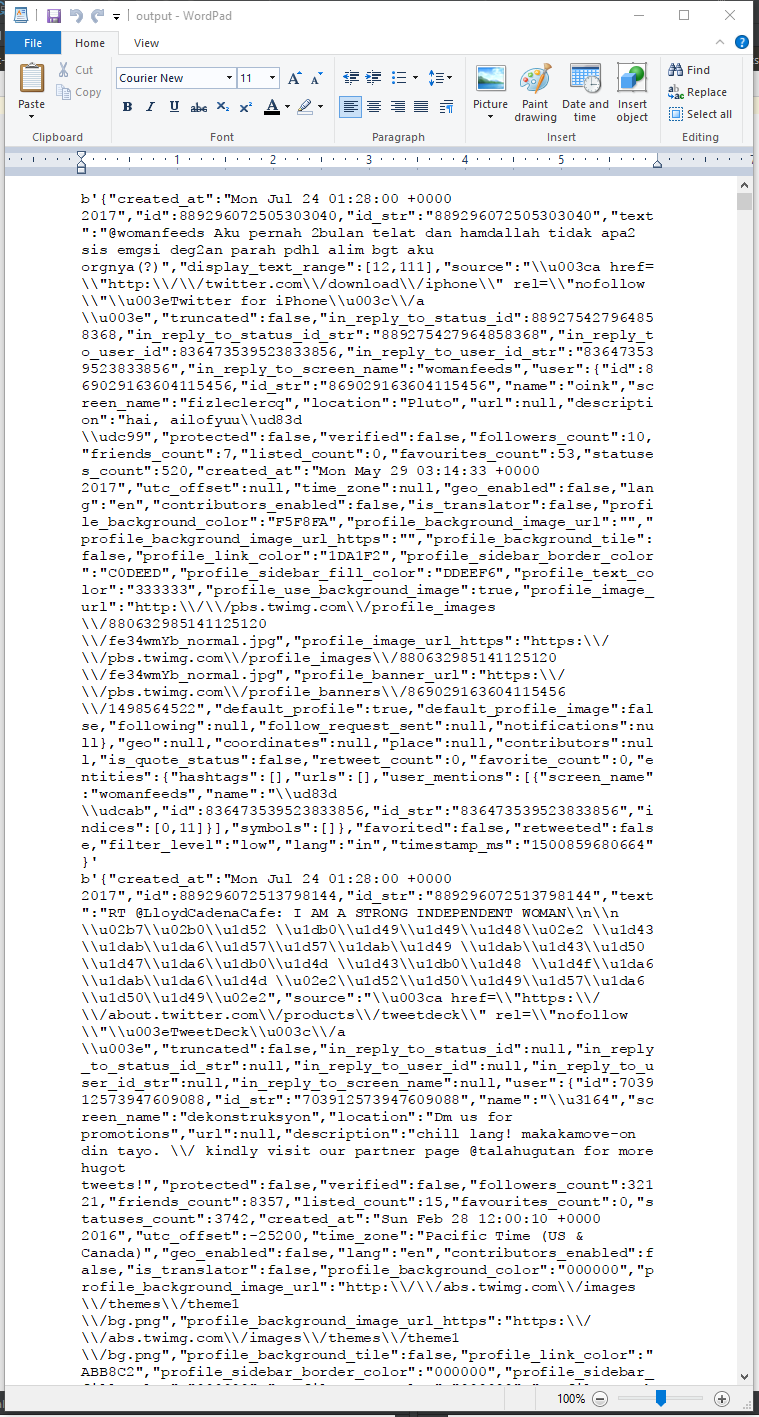
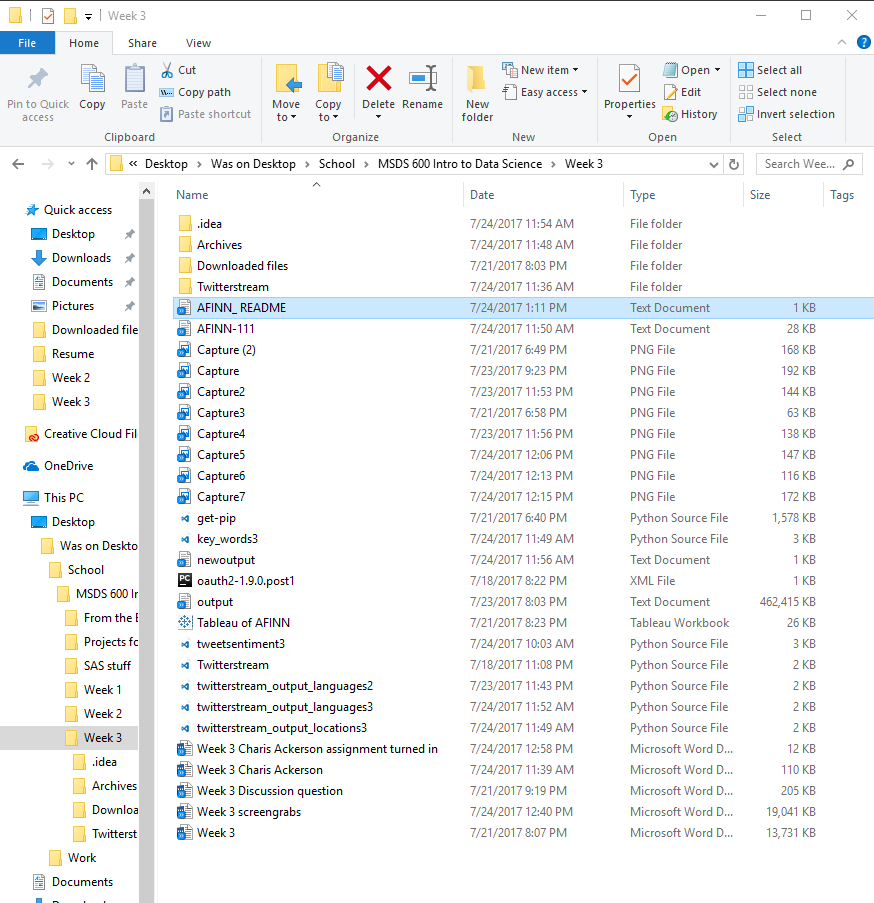
**Twitterstream Sentiment Analysis**



**Assignment I** (Drop box *Sentiment Analysis using Python*)

*Attachments – tweetsentiment.py, AFINN-111.txt & AFINN\_Readme.txt*

*Other files needed – output.txt (created last week)*

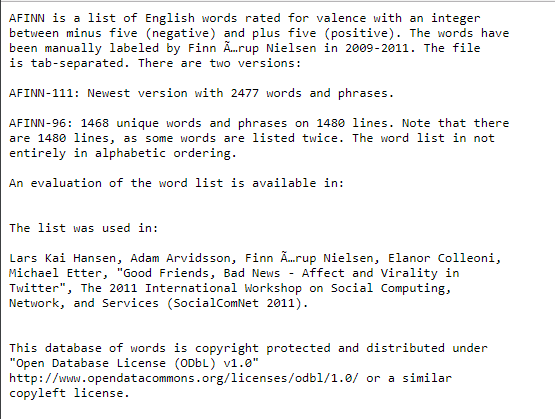
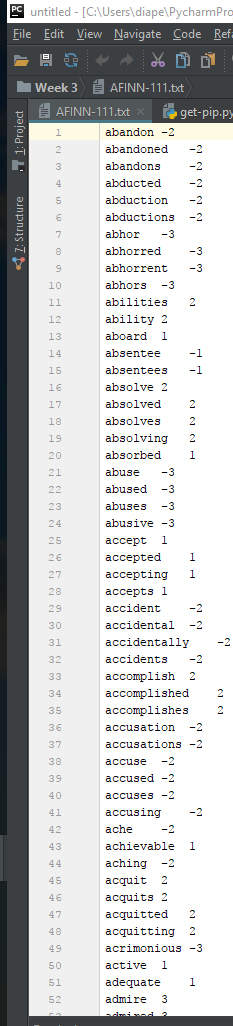


**Figure 1 – Screenshot showing Output.txt file and file path for assignment**

**Part 2 – Sentiment Analysis of Tweets** (Part 1A & 1B were completed last week)

* Take a moment to look through the AFINN-111.txt and the AFINN-Readme.txt (attached). The AFINN-111 list is a document of English words rated for valence with an integer between negative 5 and positive 5. The words were rated by Finn Arup Nielsen. This file contains 2477 words and phrases.

**AFINN\_README.txt: AFINN-111.txt:**

* For video tutorial visit: <https://youtu.be/hS60jDshOyw>
* In this assignment the **tweetsentiment.py** file (attached) will read the sentiment file into a python dictionary. The code will then read your collected tweets using your **output.txt** file that you created last week and compare the words in the tweets against the sentiments. The code will increment a counter for each sentiment value and prints the sentiment value count.
* Finally, it will provide you with a list of results, see the example below.

-5 sentiments 2

-4 sentiments 24

-3 sentiments 53

-2 sentiments 91

-1 sentiments 112

0 sentiments 0

1 sentiments 101

2 sentiments 134

3 sentiments 116

4 sentiments 31

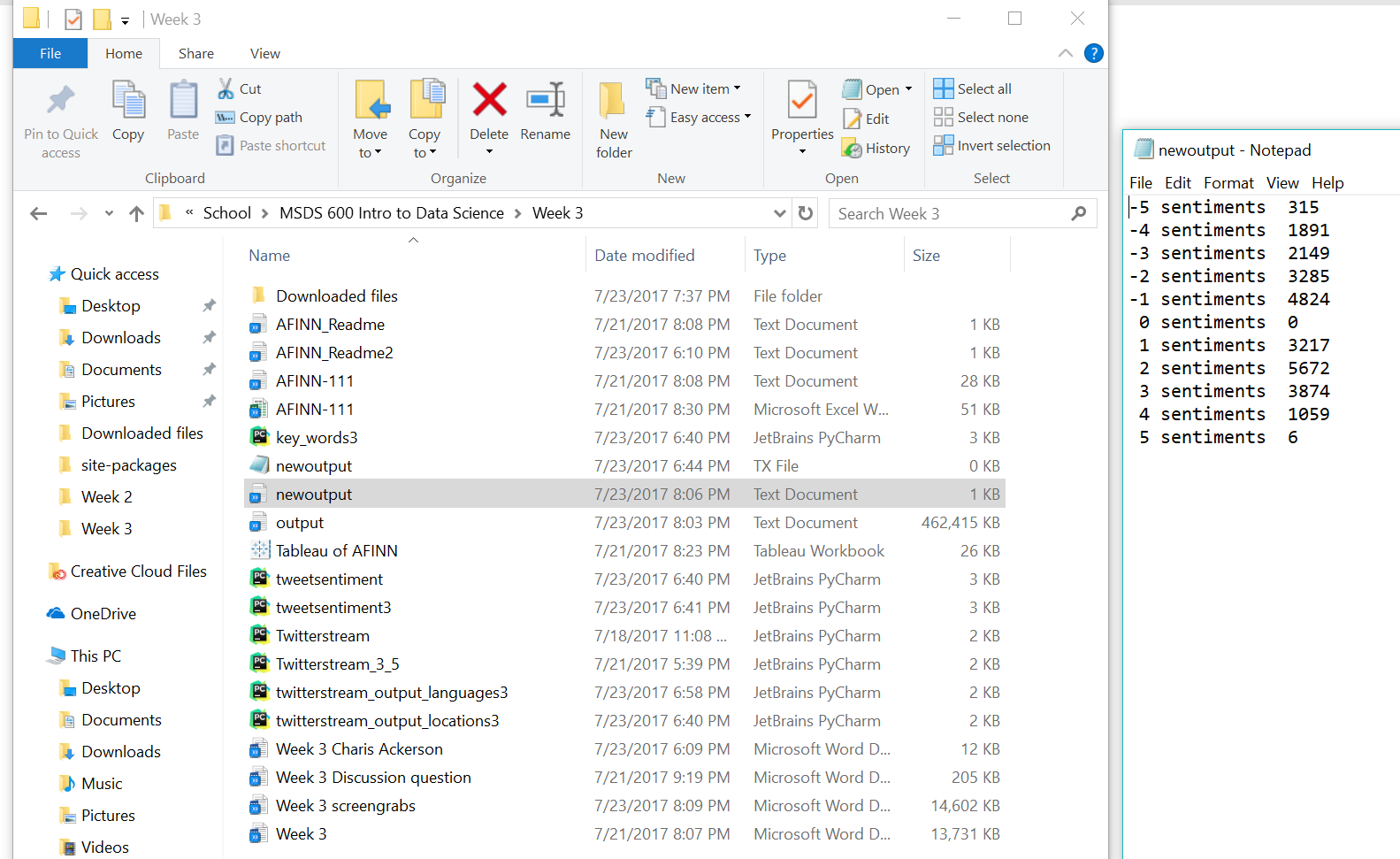
5 sentiments 0

* To begin this assignment, download all attached files. Next run the following command in your Command Prompt or Terminal window:

***python tweetsentiment.py AFINN-111.txt output.txt > newoutput.txt***

* + - **Note** – if you renamed your **output.txt** file that you created last week, be sure to reflect that in the above command.

**Newoutput.txt:**

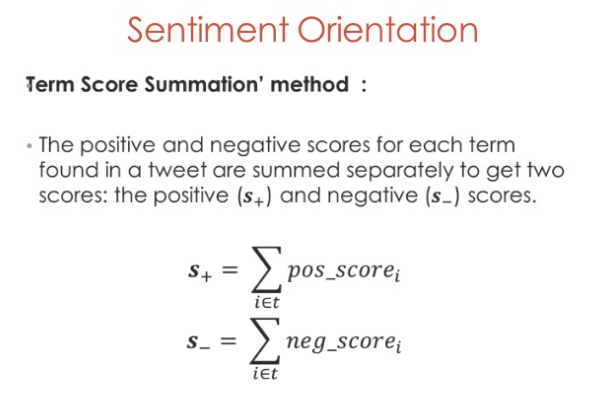
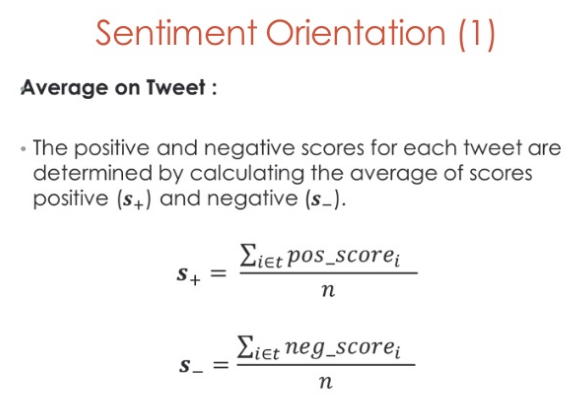
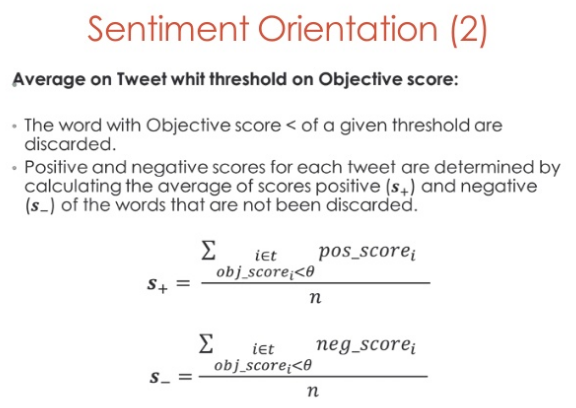
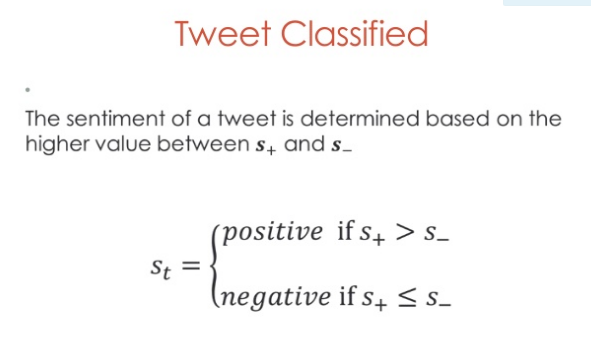


* After running the above command you will have generated a new file – **newoutput.txt**
  + Use the values in this value to answer the questions below and include them in your write up:
    - What type of data are the sentiment values i.e., {-5, -4, …, 4, 5}?
      * Interval data because the interval is not meaningful (i.e. a score of 4 is not twice as much as 2).
    - Compute the mean for the sentiments. (Note that the equation below is wrong, please correct it for your assignment)



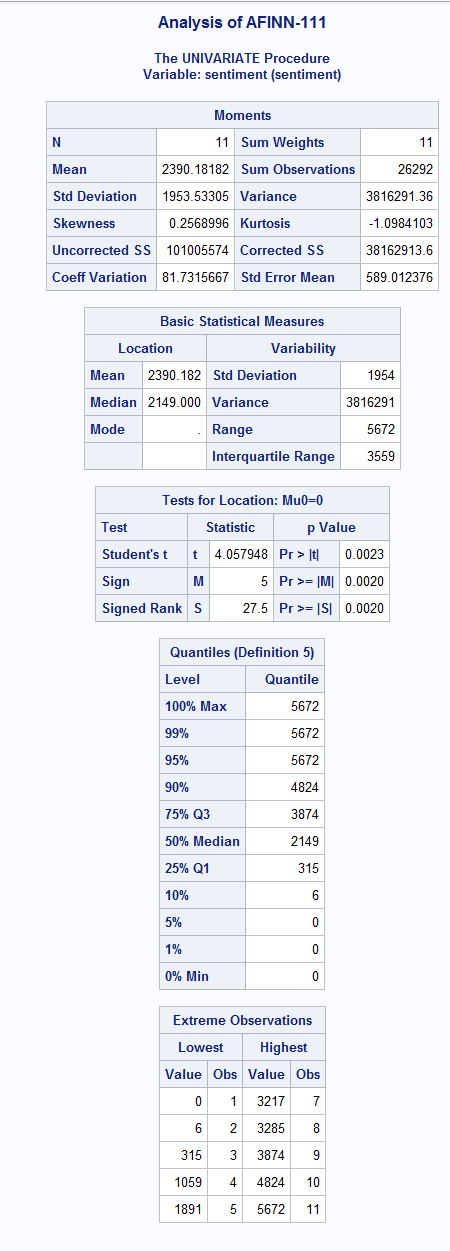
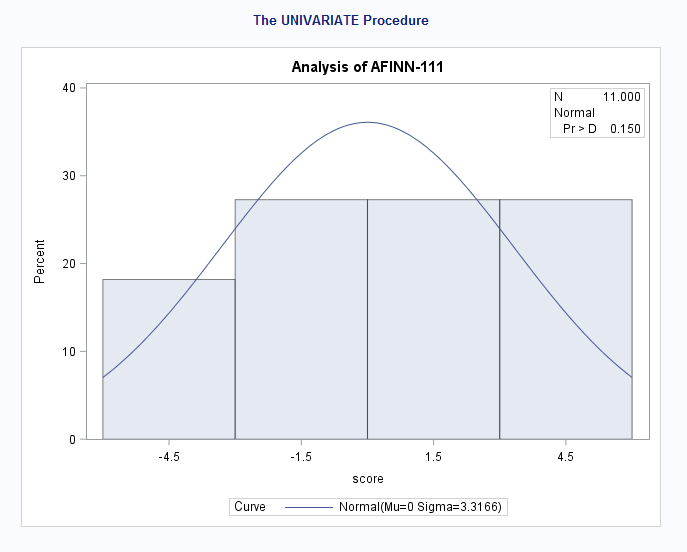
Corrected:

x̅ = n-5(-5) + n-4(-4) + n-3(-3) + n-2(-2) + n-1(-1) + n0(0) + n1(1) + n2(2) + n3(3) + n4(4) + n5(5) = ii

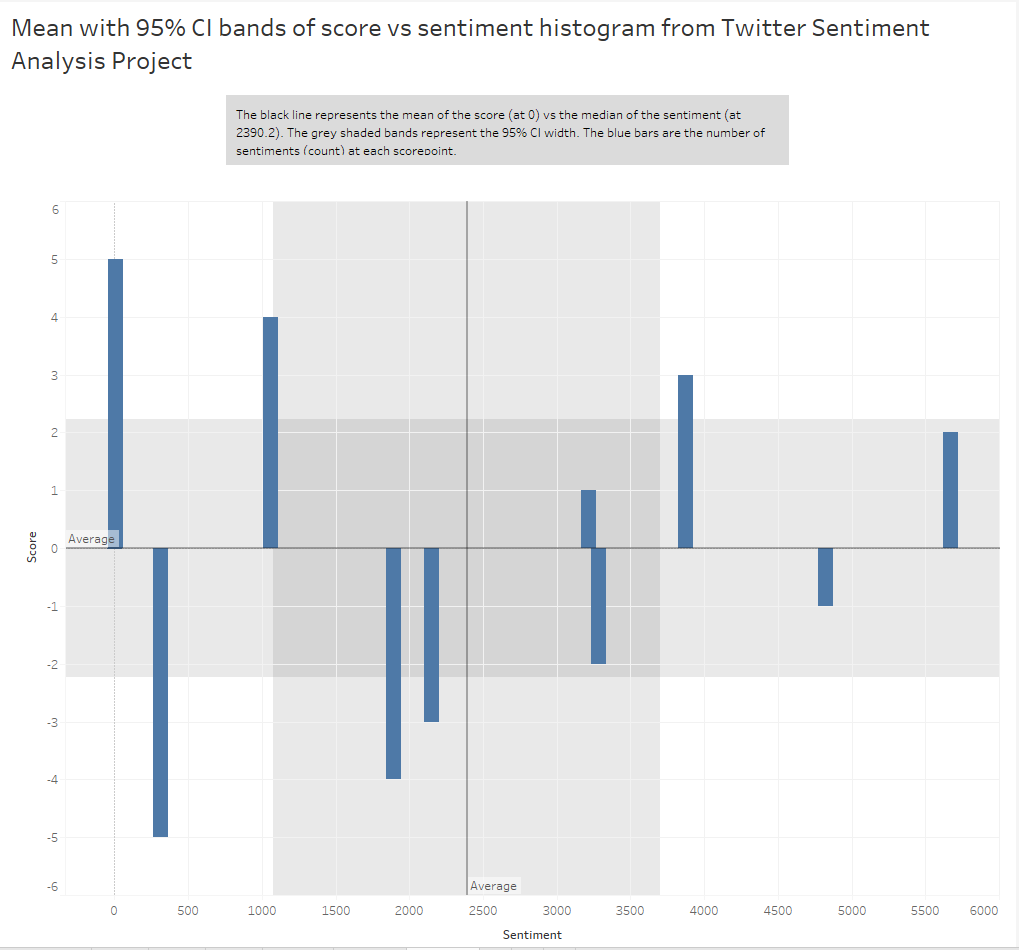
   

**Figure 2 – Correct mean calculation for Twitter Sentiment analysis (Tutorial of Sentiment Analysis, 2017)**

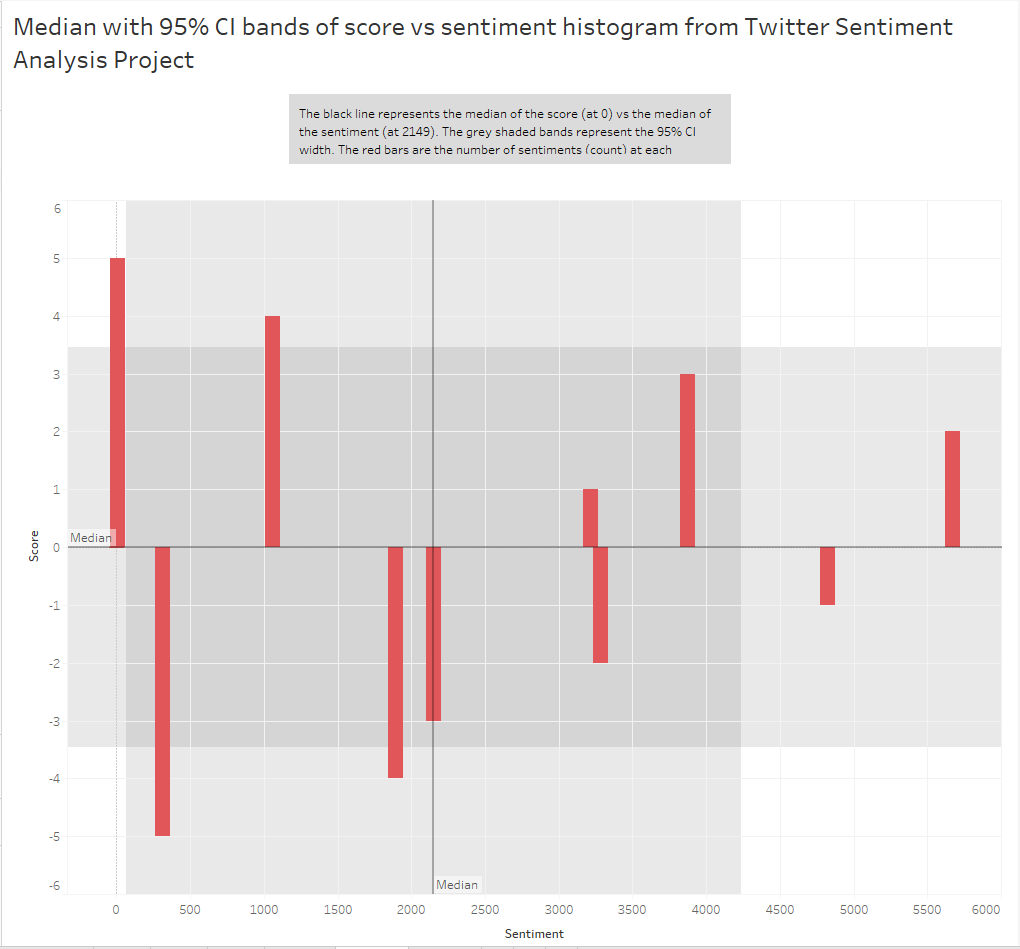
I did some research on this, because this isn’t a simple mean calculation. Since it is interval data, the better way to calculate mean would be by weighting the positive and negative scores. So in this case, I would calculate as shown in Figure 2 (Tutorial of Sentiment Analysis, 2017). If you really dive into the calculations for mean, the mean of score = 0, and the mean of sentiment analysis = 2390.2 as shown in Figures 3 and 4. I calculated median as well, just because if the data does not meet the mean assumptions, calculating median would be a better measure of central tendency. Here, the median of score = 0 and the median of sentiment = 2149.0 as seen in Figures 3 and 5. I also plotted out a histogram to test for normality. If I wanted to get really fancy, I’d do a QQplot to test for normality. Wide confidence interval bands mean that there is a loss of precision in our sampling. I could increase the sample size to account for this.

**Figure 3 – Calculation of mean & median of sentiment (in SAS)**



**Figure 4 – Calculation of mean of score (in Tableau)**



**Figure 5 – Calculation of median of score (in Tableau)**

* What does this mean value mean?

If you calculate the mean of the sentiments from newoutput.txt, then that gives you the average (measure of central tendency) of the frequency of the tweets that were made by AFINN-111 scoring at: -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5. In essence, since -5 is the most negative Tweet and +5 is the most positive tweet, this is telling us that the average of the Tweets that I collected over 3 minutes is neutral (0) in sentiment.

* + - In calculating the mean, what additional assumptions about the type of the sentiment data have been made?
      * The calculation of mean assumes homogeneity of variance, linearity (standard normal distribution in this case), and that the data are independent (Real Statistics, 2017).
    - Suppose that in **AFINN-111.txt** that there were different numbers of words that have been rated. What effect would this have on your analysis? What might be a way to compensate for this problem?
  + That would entirely change the results of my analysis. If the number of words were to increase, say from n=1,000 to n=10,000, the 95% CI width would decrease, resulting in more precision, meaning that we are more confident that the true mean lies within our confidence limits and that the “sampling distributions are more clustered around the mean” (Sample Size and CI, 2017). The converse is true with smaller sample sizes, n=100.
  + A way to compensate for this is to increase my sample size. I would do this by running the twitter collection longer in program “twitterstream.py”. If I ran it overnight or longer, that would decrease 95% CI width.
    - Visit this site and discuss how sarcasm and other factors will effect your above results. [**http://darenr.github.io/afinn/**](http://darenr.github.io/afinn/)
  + Sarcasm, irony, and metaphors can influence the mean of Twitter Sentiment Analysis (Sulis, 2017). Sulis et al devised a method for trying to account for this in their ValenTo System (Sulis, 2017). Without adding in a layer of Bayesian, Decision Tree, or Support Vector Machines, the data could be biased (Information Bias or Selection Bias).

**Assignment II** (Drop box *Sentiment Analysis using Python*)

*Attachments – twitterstream\_output\_languages.py, twitterstream\_output\_locations.py & key\_words.py*

*Other files needed – output.txt (created last week)*

**Part 3 – Further Analysis of Tweets**

* For this assignment you will work through the exercise below. You will learn how to make changes to Python files and produce results tailored to your interests. Put together a word document and explain your process and results.
* For video tutorial visit: https://youtu.be/qiuEZFzrljU

\*Please note that this exercise uses Matplotlib. If you run into excessive errors downloading and installing Matplotlib, feel free to skip it. It produces basic graphs, but what I am really interested in here is your results and analysis of your data. *You will probably get an error message if you do not install Matplotlib, and you should be able to overlook it – your results should still print. If not please let me know and I will edit the files.*

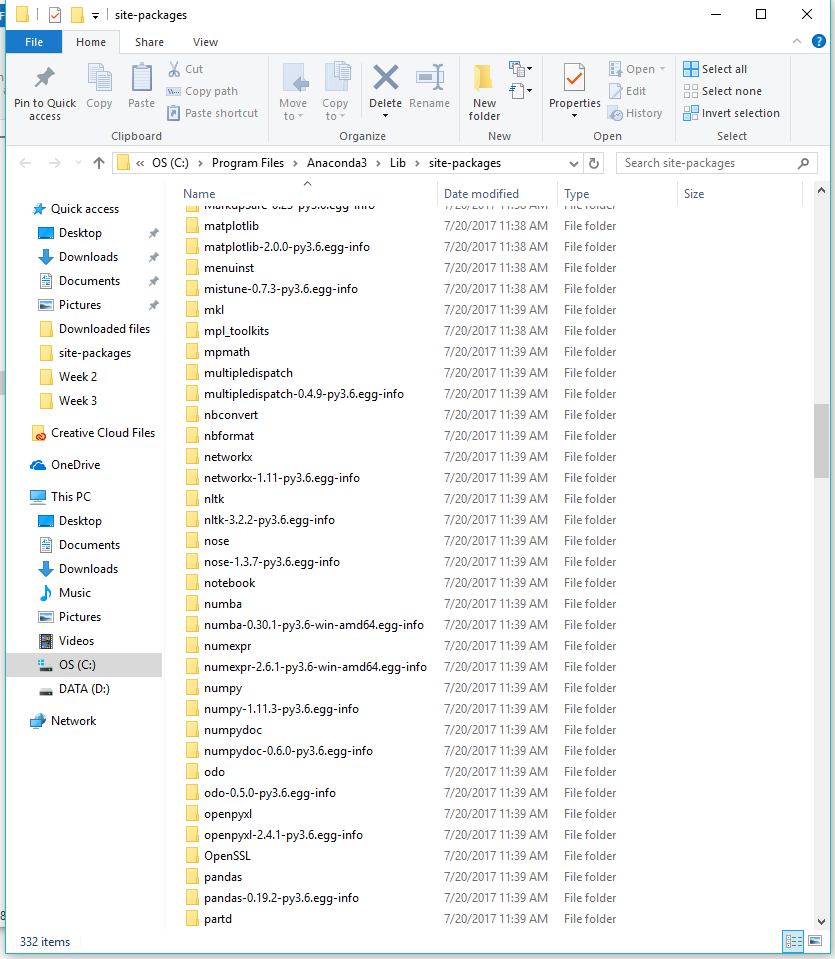
**Exercise** (commands in blue, comments in green)

**Step 1:** Install dependent packages. Use your Command Prompt/Terminal to do this

pip install -U pip setuptools

pip install matplotlib

pip install pandas



**Figure 2 – Pandas, Matplotlib already installed in Anaconda3**

C :\Users\diape\Desktop\Was on Desktop\School\MSDS 600 Int

ro to Data Science\Week 3>dir

Volume in drive C is OS

Volume Serial Number is 4845-CAAD

Directory of C:\Users\diape\Desktop\Was on Desktop\Schoo

l\MSDS 600 Intro to Data Science\Week 3

07/24/2017 11:54 AM <DIR> .

07/24/2017 11:54 AM <DIR> ..

07/24/2017 11:54 AM <DIR> .idea

07/24/2017 11:50 AM 28,093 AFINN-111.txt

07/24/2017 11:48 AM <DIR> Archives

07/21/2017 06:49 PM 171,133 Capture (2).PNG

07/23/2017 09:23 PM 196,264 Capture.PNG

07/23/2017 11:53 PM 147,310 Capture2.PNG

07/21/2017 06:58 PM 63,927 Capture3.PNG

07/23/2017 11:56 PM 140,355 Capture4.PNG

07/23/2017 08:39 PM <DIR> Downloaded files

07/21/2017 06:40 PM 1,615,420 get-pip.py

07/24/2017 11:49 AM 2,748 key\_words3.py

07/24/2017 11:56 AM 224 newoutput.txt

07/18/2017 08:22 PM 590 oauth2-1.9.0.post1

.xml

07/23/2017 08:03 PM 473,512,937 output.txt

07/21/2017 08:23 PM 25,931 Tableau of AFINN.t

wb

07/24/2017 11:52 AM 3,031 tweetsentiment3.py

07/24/2017 11:36 AM <DIR> Twitterstream

07/24/2017 11:52 AM 1,592 twitterstream\_outp

ut\_languages3.py

07/24/2017 11:49 AM 1,709 twitterstream\_outp

ut\_locations3.py

07/24/2017 11:39 AM 111,966 Week 3 Charis Acke

rson.docx

07/21/2017 09:19 PM 209,708 Week 3 Discussion

question.docx

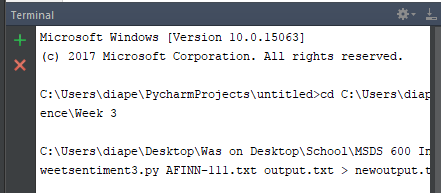
07/23/2017 08:13 PM 15,364,573 Week 3 screengrabs

.docx

07/21/2017 08:07 PM 14,059,553 Week 3.docx

19 File(s) 505,657,064 bytes

6 Dir(s) 116,505,423,872 b



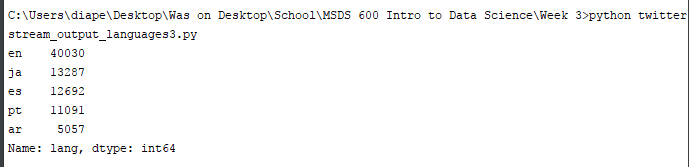
**Figure 3 – PyCharm Terminal demonstrating proper file path**

**Step 2:** Open up PyCharm or other.

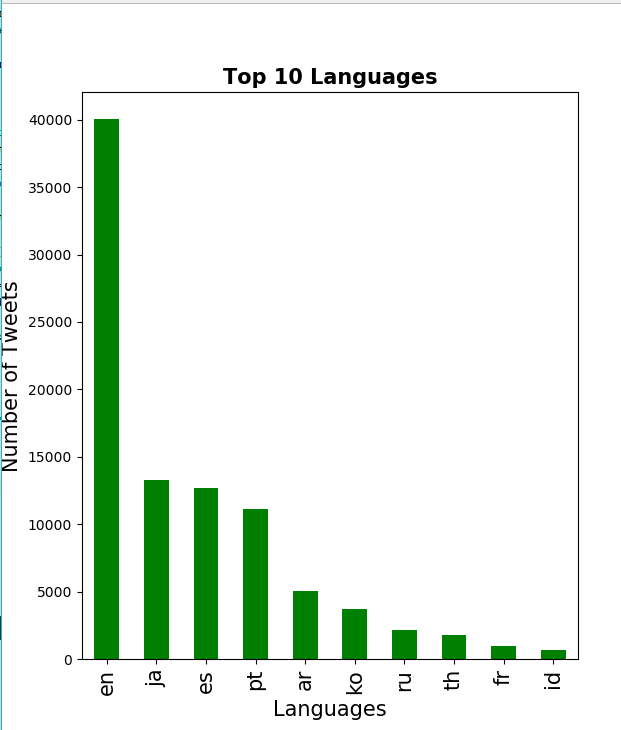
* Start with **twitterstream\_output\_languages3.py** file. Open it in your editor and take a look.
  + This file uses your **output.txt** file that you created last week, so be sure that you have this file in your environment.
  + What we are doing here is instructing Python to output a list of the top 10 languages of the tweets we collected.
  + You can modify this file to print the top 5, or to view the top 50
    - To get in good habit, when you make a change to a file, be sure to “save as” something else, this way you always have the original file to go back to/reference for future use.
* Open your Terminal and run the following command:

***python twitterstream\_output\_languages3.py***

* + You will produce a list of the top 10 languages found in your **output.txt** file.
  + If you have matplotlib installed, a graph should pop up



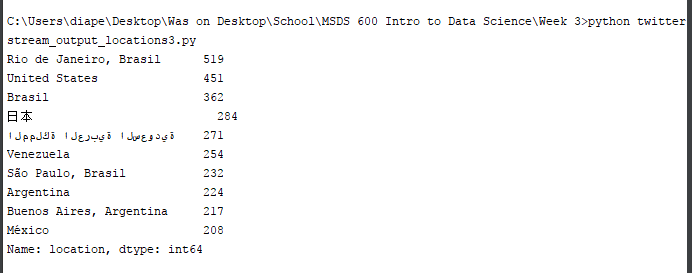
**Figure 4 – PyCharm Terminal screen output from twitterstream\_output\_languages3.py**



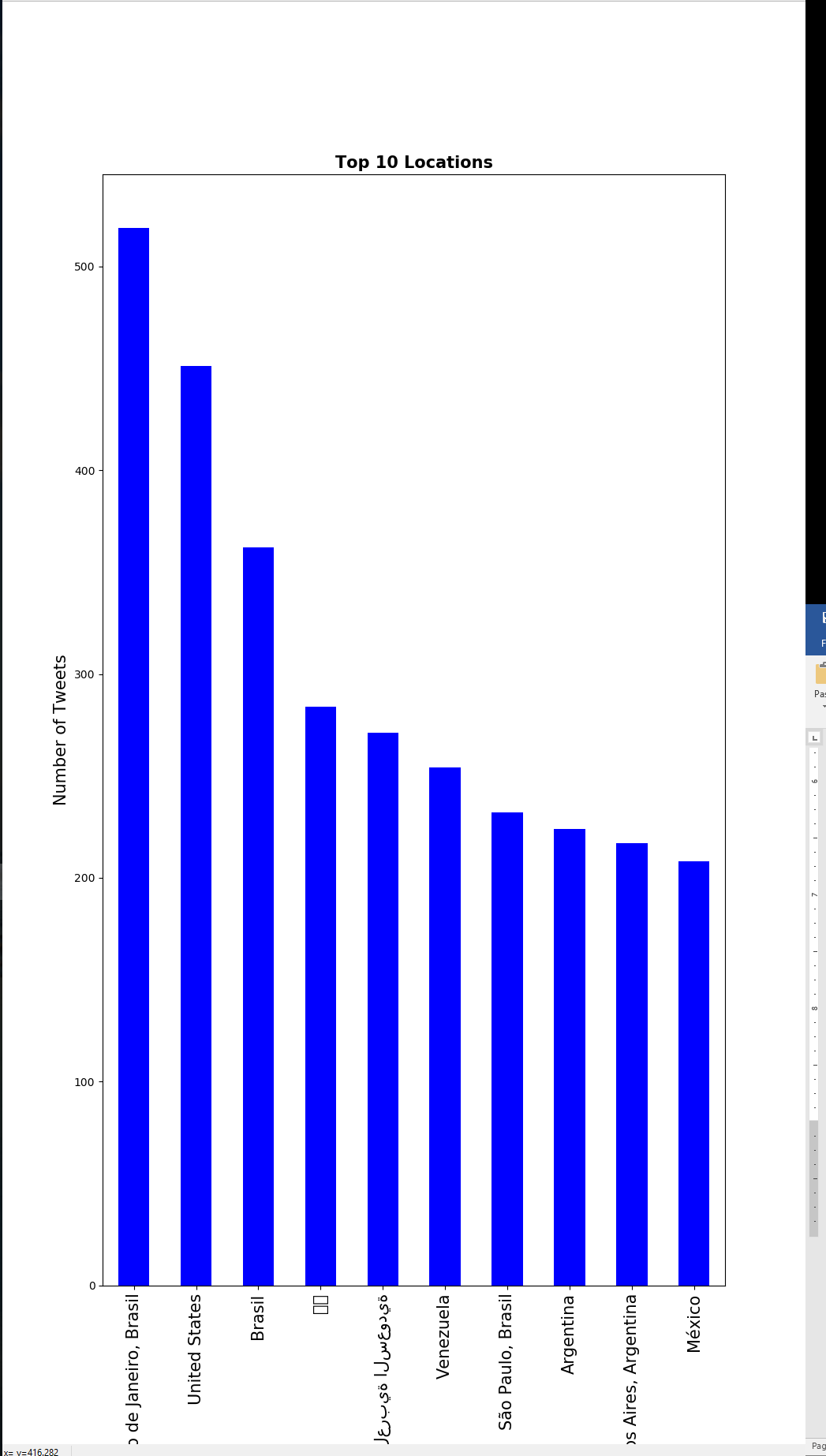
**Figure 5 – MatPlotLib graph comparing number of tweets to languages**

* Now lets open **twitterstream\_output\_locations.py** and take a look at this
  + When run, this file will produce a list of the top 10 locations from where your tweets originated.
  + Again, feel free to change the number of locations, the color of the bar chart, etc.
    - Note that you need to close the chart in order to continue running commands in the Terminal
  + Run the command:

***python twitterstream\_output\_locations3.py***



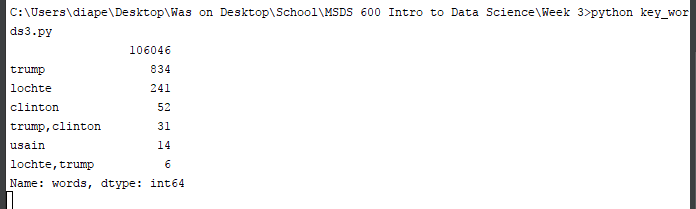
**Figure 6 – PyCharm Terminal output from twitterstream\_output\_locations3.py**



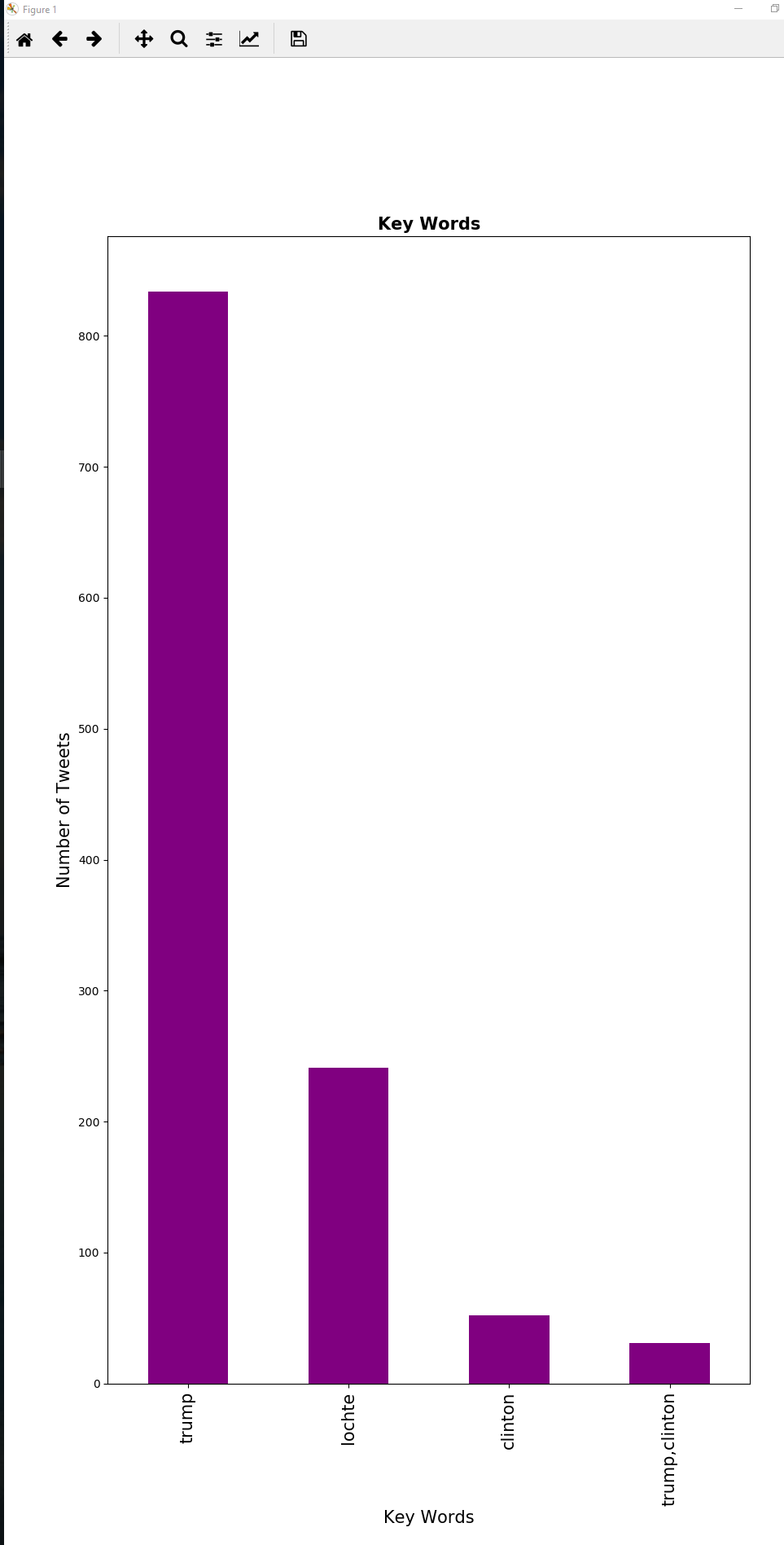
**Figure 7 – MatPlotLib graph comparing number of tweets vs top 10 languages created by twittersttream\_output\_locations3.py program**

* Now take a look at the **key\_words3.py** file.
  + This file should be customized by you. As it is currently, it instructs Python to count the number of occurrences for the designated words.
  + Modify this file and enter in any phrase, name, etc. that you are interested in.
  + You will notice that the graph is instructed not to output the top result – this includes all words not found in our designated key words – our graph would be thrown off if we included them.
  + Close any open graphs and run the following:

***python key\_words3.py***



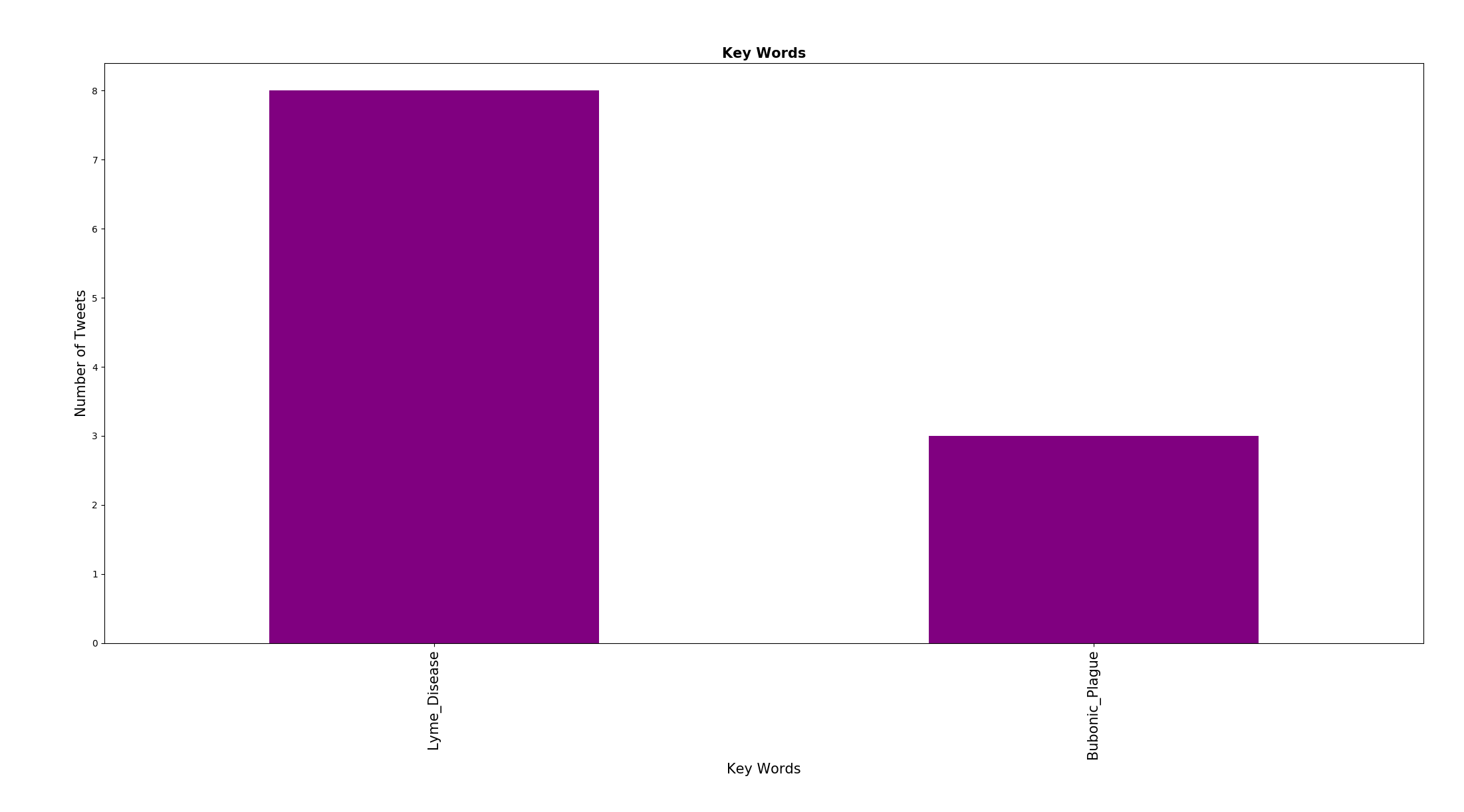
**Figure 8 - PyCharm Terminal output from key\_words3.py**



**Figure 9 - MatPlotLib graph comparing number of tweets vs key words, created by key\_words3.py program**

* For your deliverable, work through the exercise and modify the last code (**key\_words3.py**) to include words that interest you, current events or other\*. Talk about how this may have affected your mean sentiment results in Assignment I Part1B. Be sure to include screen shots.
  + \*Feel free to create a new output file (perhaps output2.txt) and compare the key word counts from your initial output.txt file to your new output2.txt file. As you can see in my file, Ryan Lochte was tweeted quite a few times, are the counts different a few weeks later?

***python key\_words4.py***

**import** json  
**import** pandas **as** pd  
**import** matplotlib.pyplot **as** plt  
**import** pylab  
*# import re***def** readTwitterData(twitterDataFile):  
 tweets = [] *# an empty list* **with** open(twitterDataFile, **"r"**) **as** tweetfile: *# open tweet file* **for** line **in** tweetfile: *# loop over each tweet* line = line[0:-2]  
 line = line.replace(**"b'"**,**''**)  
 line = line.replace(**"\\\\"**,**"\\"**)  
 line = line.replace(**"\\'"**,**"'"**)  
 line = json.loads(line) *# change json to dictionary* tweets.append(line) *# append tweet string to list* **return** tweets *# return the list  
  
# tweets\_data = readTwitterData('output.txt')  
  
# tweets = pd.DataFrame()  
  
# tweets['tweet'] = list(map(lambda tweet: tweet['text'] if 'text' in tweet else None, tweets\_data))***def** word\_in\_text(word, text):  
 word = word.lower()  
 text = text.lower()  
 match = re.search(word,text)  
 **if** match:  
 **return True  
 return False  
  
def** key\_words(row):  
 words = []  
 **if** row[**'text'**]:  
 text = row[**"text"**].lower()  
 **if "lyme" in** text **or "lyme disease" in** text:  
 words.append(**"Lyme\_Disease"**)  
 **if "CDC" in** text **or "Centers for Disease Control" in** text:  
 words.append(**"CDC"**)  
 **if "tularemia" in** text **or "rabbit fever" in** text:  
 words.append(**"Tularemia"**)  
 **if "plague" in** text **or "black death" in** text:  
 words.append(**"Bubonic\_Plague"**)  
 **return ","**.join(words)  
  
*# def extract\_link(text):  
# regex = r'https?://[^\s<>"]+|www\.[^\s<>"]+'  
# match = re.search(regex, text)  
# if match:  
# return ''***if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 tweets\_data\_path = **'output.txt'** tweets\_data = []  
  
 tweets\_data = readTwitterData(tweets\_data\_path)  
 tweets = pd.DataFrame()  
 texts = []  
 tweets[**'text'**] = list(map(**lambda** tweet: tweet[**'text'**] **if 'text' in** tweet **else None**, tweets\_data))  
 *# for tweet in tweets['tweet']:  
 # texts.append(tweet['text'])  
  
 # tweets['text'] = texts* tweets[**"words"**] = tweets.apply(key\_words,axis=1)  
 counts = tweets[**"words"**].value\_counts()  
 print(counts)  
  
 fig, ax = plt.subplots()  
 ax.tick\_params(axis=**'x'**, labelsize=15)  
 ax.tick\_params(axis=**'y'**, labelsize=10)  
 ax.set\_xlabel(**'Key Words'**, fontsize=15)  
 ax.set\_ylabel(**'Number of Tweets'** , fontsize=15)  
 ax.set\_title(**'Key Words'**, fontsize=15, fontweight=**'bold'**)  
 counts[1:5].plot(ax=ax, kind=**'bar'**, color=**'purple'**)  
  
 pylab.show()

**Figure 10 – MatPlotLib Graph comparing number of tweets to key words in Twitter Sentiment Analysis, using key\_words4.py program**

Talk about how this may have affected your mean sentiment results in Assignment I Part1B

**References:**

Bickel, D. R. (2003). Robust and efficient estimation of the mode of continuous data: The mode as a viable measure of central tendency. *Journal of statistical computation and simulation*, *73*(12), 899-912.

Büchner, M. H. M. P. M., & Stein, B. (2015). Webis: An ensemble for twitter sentiment detection. *SemEval-2015*, *582*.

Lars Kai Hansen, Adam Arvidsson, Finn Årup Nielsen, Elanor Colleoni,  
Michael Etter, "Good Friends, Bad News - Affect and Virality in  
Twitter", The 2011 International Workshop on Social Computing,  
Network, and Services (SocialComNet 2011).

Farías, D. I. H., Sulis, E., Patti, V., Ruffo, G., & Bosco, C. (2015). ValenTo: Sentiment Analysis of Figurative Language Tweets with Irony and Sarcasm. In *SemEval@ NAACL-HLT* (pp. 694-698).

Finn Årup Nielsen  
"A new ANEW: Evaluation of a word list for sentiment analysis in microblogs",  
Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts':  
Big things come in small packages 718 in CEUR Workshop Proceedings : 93-98. 2011 May.   
<http://arxiv.org/abs/1103.2903>

Lars Kai Hansen, Adam Arvidsson, Finn Årup Nielsen, Elanor Colleoni,  
Michael Etter, "Good Friends, Bad News - Affect and Virality in  
Twitter", The 2011 International Workshop on Social Computing,  
Network, and Services (SocialComNet 2011).

Wilcox, R. R., & Keselman, H. J. (2003). Modern robust data analysis methods: measures of central tendency. *Psychological methods*, *8*(3), 254.

Nielsen, F. Å. (2015). Data Mining using Python—code comments.

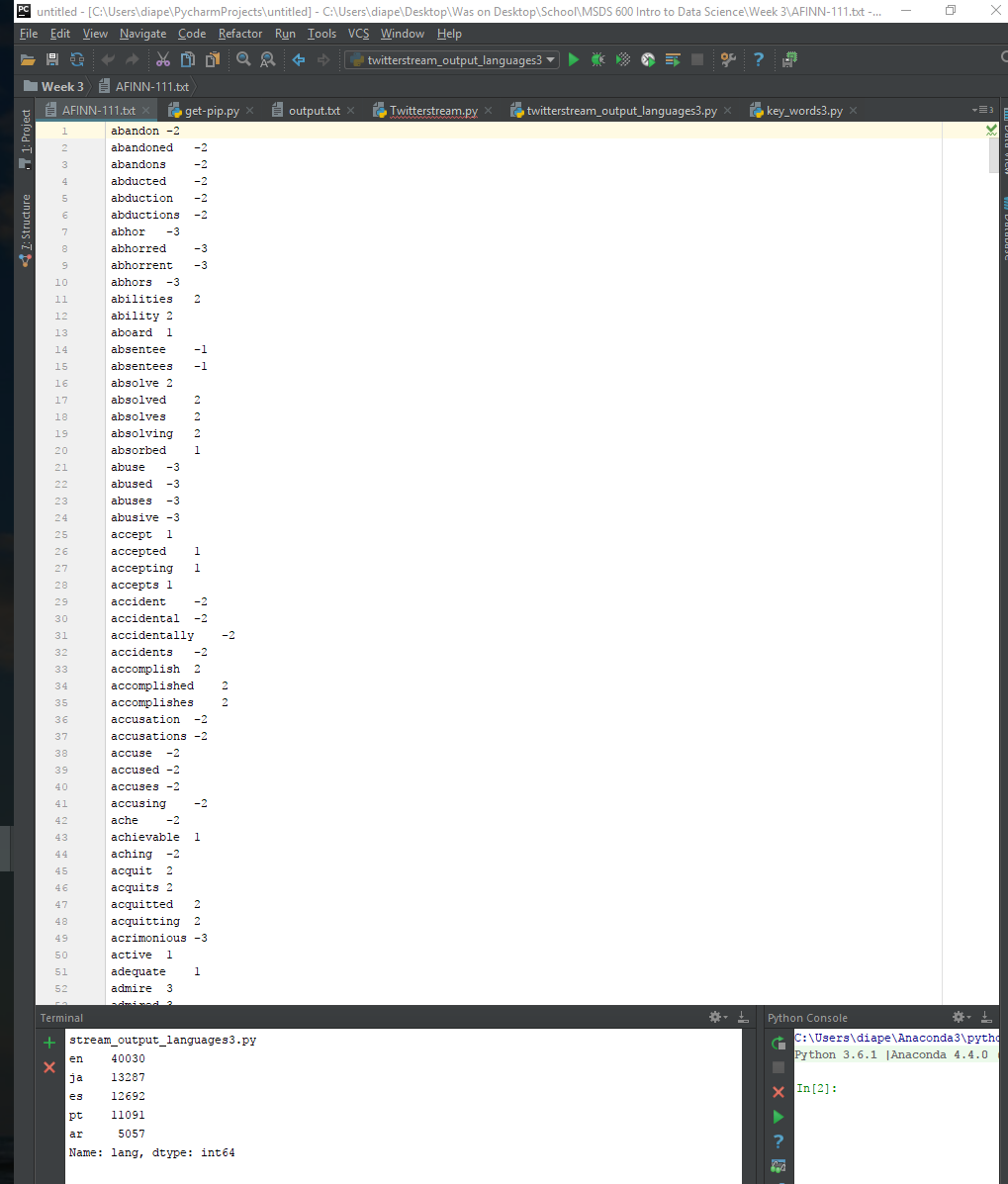
[Real Statistics Using Excel](http://www.real-statistics.com/). Retrieved July 23, 2017 from: <http://www.real-statistics.com/descriptive-statistics/assumptions-statistical-test/>

Sample Size and CI. Retrieved July 23, 2017 from: https://learn.bu.edu/bbcswebdav/pid-826911-dt-content-rid-2073768\_1/courses/13sprgmetcj702\_ol/week03/metcj702\_W03S02T08d\_sample.html

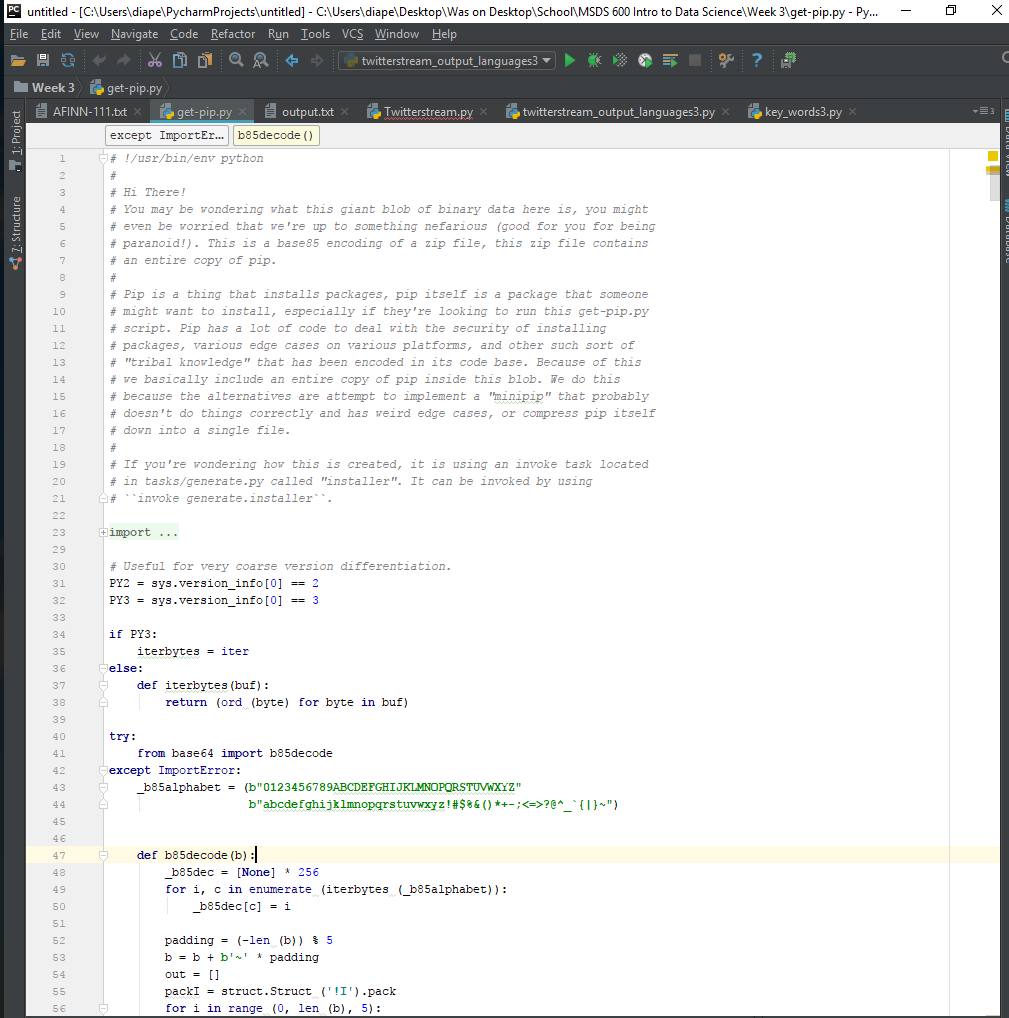
Tutorial of Sentiment Analysis. Retrieved July 23, 2017 from: https://www.slideshare.net/faigg/tutotial-of-sentiment-analysis

**Appendix A- Python Code used in assignment:**

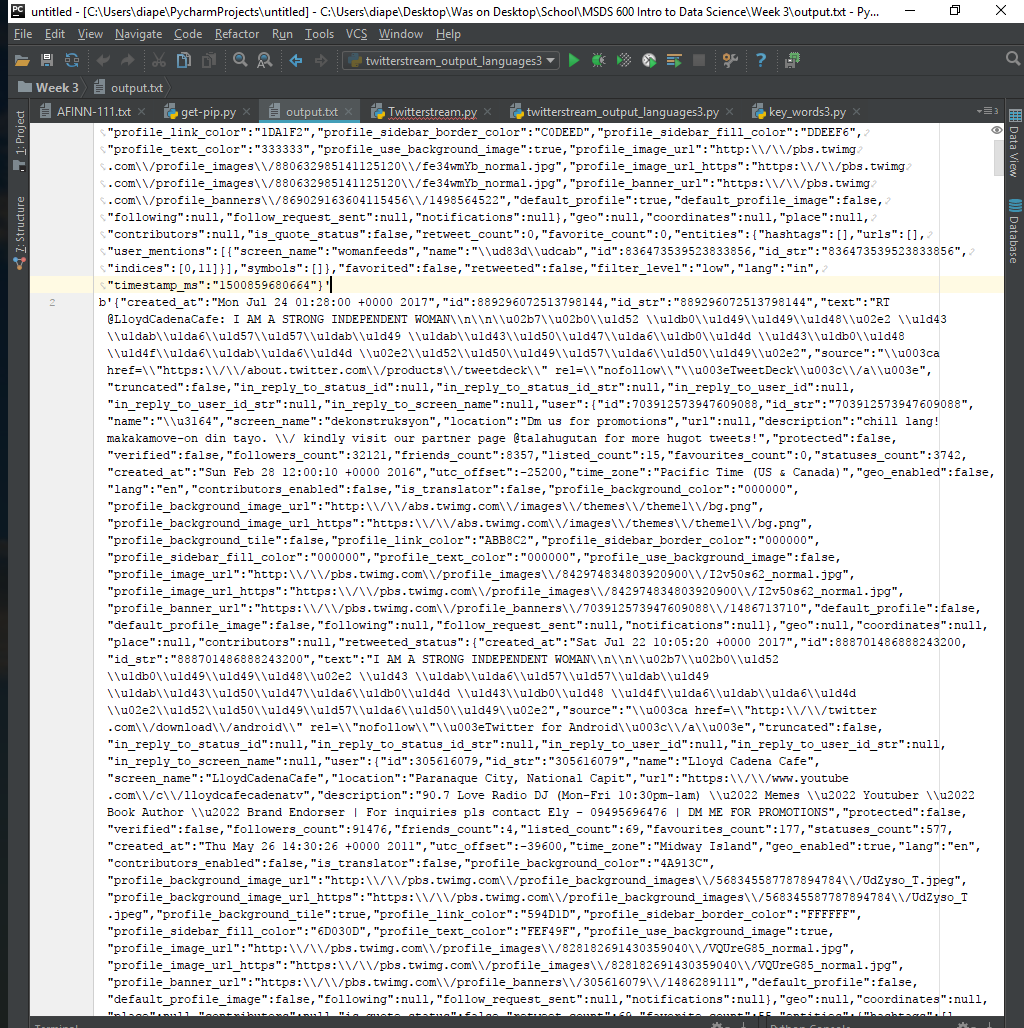
**AFINN-111.txt:**



**Get – pip.py:**



**Output.txt:**



**Twitterstream\_3\_5.py:**

**import** oauth2 **as** oauth  
**import** urllib2 **as** urllib  
  
*# See assignment1.html instructions or README for how to get these credentials*api\_key = **"awq9HpreWL0p3OSHu5kdj2Ivn"**api\_secret = **"DTsKYNvOkcBmJ8lQVYBUYJ42NIw0MK5YbJule1P6W3B4waW3fM"**access\_token\_key = **"887531396771844098-F9MmTXrqDFMevdfYyCd2lysY8x0KhAR"**access\_token\_secret = **"eiaOCBP548r7P5AZpay5LRDHBS7uJK5n8BafewuSiVd0R"**\_debug = 0  
  
oauth\_token = oauth.Token(key=access\_token\_key, secret=access\_token\_secret)  
oauth\_consumer = oauth.Consumer(key=api\_key, secret=api\_secret)  
  
signature\_method\_hmac\_sha1 = oauth.SignatureMethod\_HMAC\_SHA1()  
  
http\_method = **"GET"**http\_handler = urllib.HTTPHandler(debuglevel=\_debug)  
https\_handler = urllib.HTTPSHandler(debuglevel=\_debug)  
  
**'''  
Construct, sign, and open a twitter request  
using the hard-coded credentials above.  
'''  
def** twitterreq(url, method, parameters):  
 req = oauth.Request.from\_consumer\_and\_token(oauth\_consumer,  
 token=oauth\_token,  
 http\_method=http\_method,  
 http\_url=url,   
 parameters=parameters)  
  
 req.sign\_request(signature\_method\_hmac\_sha1, oauth\_consumer, oauth\_token)  
  
 headers = req.to\_header()  
  
 **if** http\_method == **"POST"**:  
 encoded\_post\_data = req.to\_postdata()  
 **else**:  
 encoded\_post\_data = **None** url = req.to\_url()  
  
 opener = urllib.OpenerDirector()  
 opener.add\_handler(http\_handler)  
 opener.add\_handler(https\_handler)  
  
 response = opener.open(url, encoded\_post\_data)  
  
 **return** response  
  
**def** fetchsamples():  
 url = **"https://stream.twitter.com/1.1/statuses/sample.json"** parameters = []  
 response = twitterreq(url, **"GET"**, parameters)  
 **for** line **in** response:  
 print line.strip()  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 fetchsamples()

**Twitterstream\_output\_languages3.py:**

*# \_\_author\_\_ = 'homecomputer'***import** json  
**import** pandas **as** pd  
**import** matplotlib.pyplot **as** plt  
**import** pylab  
**import** re  
  
**def** readTwitterData(twitterDataFile):  
 tweetfile = open(twitterDataFile, **"r"**) *# open tweet file* tweets = [] *# an empty list* **for** line **in** tweetfile: *# loop over each tweet* line = line[0:-2]  
 line = line.replace(**"b'"**,**''**)  
 line = line.replace(**"\\\\"**,**"\\"**)  
 line = line.replace(**"\\'"**,**"'"**)  
 **try**:  
 line = json.loads(line) *# change json to dictionary* tweets.append(line) *# append tweet string to list* **except**:  
 **continue** tweetfile.close() *# close the file* **return** tweets *# return the list*tweets\_data = readTwitterData(**'output.txt'**)  
  
tweets = pd.DataFrame()  
  
tweets[**'tweet'**] = list(map(**lambda** tweet: tweet[**'text'**] **if 'text' in** tweet **else None**, tweets\_data))  
tweets[**'lang'**] = list(map(**lambda** tweet: tweet[**'user'**][**'lang'**] **if 'user' in** tweet **else None**, tweets\_data))  
tweets\_by\_lang = tweets[**'lang'**].value\_counts()  
print (tweets\_by\_lang[:5]) *#outputs top 5 languages*fig, ax = plt.subplots()  
ax.tick\_params(axis=**'x'**, labelsize=15)  
ax.tick\_params(axis=**'y'**, labelsize=10)  
ax.set\_xlabel(**'Languages'**, fontsize=15)  
ax.set\_ylabel(**'Number of Tweets'** , fontsize=15)  
ax.set\_title(**'Top 10 Languages'**, fontsize=15, fontweight=**'bold'**)  
tweets\_by\_lang[:10].plot(ax=ax, kind=**'bar'**, color=**'green'**)  
  
pylab.show()

**Key\_words3.py:**

**import** json  
**import** pandas **as** pd  
**import** matplotlib.pyplot **as** plt  
**import** pylab  
*# import re***def** readTwitterData(twitterDataFile):  
 tweets = [] *# an empty list* **with** open(twitterDataFile, **"r"**) **as** tweetfile: *# open tweet file* **for** line **in** tweetfile: *# loop over each tweet* line = line[0:-2]  
 line = line.replace(**"b'"**,**''**)  
 line = line.replace(**"\\\\"**,**"\\"**)  
 line = line.replace(**"\\'"**,**"'"**)  
 line = json.loads(line) *# change json to dictionary* tweets.append(line) *# append tweet string to list* **return** tweets *# return the list  
  
# tweets\_data = readTwitterData('output.txt')  
  
# tweets = pd.DataFrame()  
  
# tweets['tweet'] = list(map(lambda tweet: tweet['text'] if 'text' in tweet else None, tweets\_data))***def** word\_in\_text(word, text):  
 word = word.lower()  
 text = text.lower()  
 match = re.search(word,text)  
 **if** match:  
 **return True  
 return False  
  
def** key\_words(row):  
 words = []  
 **if** row[**'text'**]:  
 text = row[**"text"**].lower()  
 **if "usain" in** text **or "bolt" in** text:  
 words.append(**"usain"**)  
 **if "lochte" in** text **or "ryan" in** text:  
 words.append(**"lochte"**)  
 **if "trump" in** text **or "donald" in** text:  
 words.append(**"trump"**)  
 **if "clinton" in** text **or "hillary" in** text:  
 words.append(**"clinton"**)  
 **return ","**.join(words)  
  
*# def extract\_link(text):  
# regex = r'https?://[^\s<>"]+|www\.[^\s<>"]+'  
# match = re.search(regex, text)  
# if match:  
# return ''***if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 tweets\_data\_path = **'output.txt'** tweets\_data = []  
  
 tweets\_data = readTwitterData(tweets\_data\_path)  
 tweets = pd.DataFrame()  
 texts = []  
 tweets[**'text'**] = list(map(**lambda** tweet: tweet[**'text'**] **if 'text' in** tweet **else None**, tweets\_data))  
 *# for tweet in tweets['tweet']:  
 # texts.append(tweet['text'])  
  
 # tweets['text'] = texts* tweets[**"words"**] = tweets.apply(key\_words,axis=1)  
 counts = tweets[**"words"**].value\_counts()  
 print(counts)  
  
 fig, ax = plt.subplots()  
 ax.tick\_params(axis=**'x'**, labelsize=15)  
 ax.tick\_params(axis=**'y'**, labelsize=10)  
 ax.set\_xlabel(**'Key Words'**, fontsize=15)  
 ax.set\_ylabel(**'Number of Tweets'** , fontsize=15)  
 ax.set\_title(**'Key Words'**, fontsize=15, fontweight=**'bold'**)  
 counts[1:5].plot(ax=ax, kind=**'bar'**, color=**'purple'**)  
  
 pylab.show()

**Key\_words4.py:**

**import** json  
**import** pandas **as** pd  
**import** matplotlib.pyplot **as** plt  
**import** pylab  
*# import re***def** readTwitterData(twitterDataFile):  
 tweets = [] *# an empty list* **with** open(twitterDataFile, **"r"**) **as** tweetfile: *# open tweet file* **for** line **in** tweetfile: *# loop over each tweet* line = line[0:-2]  
 line = line.replace(**"b'"**,**''**)  
 line = line.replace(**"\\\\"**,**"\\"**)  
 line = line.replace(**"\\'"**,**"'"**)  
 line = json.loads(line) *# change json to dictionary* tweets.append(line) *# append tweet string to list* **return** tweets *# return the list  
  
# tweets\_data = readTwitterData('output.txt')  
  
# tweets = pd.DataFrame()  
  
# tweets['tweet'] = list(map(lambda tweet: tweet['text'] if 'text' in tweet else None, tweets\_data))***def** word\_in\_text(word, text):  
 word = word.lower()  
 text = text.lower()  
 match = re.search(word,text)  
 **if** match:  
 **return True  
 return False  
  
def** key\_words(row):  
 words = []  
 **if** row[**'text'**]:  
 text = row[**"text"**].lower()  
 **if "lyme" in** text **or "lyme disease" in** text:  
 words.append(**"Lyme\_Disease"**)  
 **if "CDC" in** text **or "Centers for Disease Control" in** text:  
 words.append(**"CDC"**)  
 **if "tularemia" in** text **or "rabbit fever" in** text:  
 words.append(**"Tularemia"**)  
 **if "plague" in** text **or "black death" in** text:  
 words.append(**"Bubonic\_Plague"**)  
 **return ","**.join(words)  
  
*# def extract\_link(text):  
# regex = r'https?://[^\s<>"]+|www\.[^\s<>"]+'  
# match = re.search(regex, text)  
# if match:  
# return ''***if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 tweets\_data\_path = **'output.txt'** tweets\_data = []  
  
 tweets\_data = readTwitterData(tweets\_data\_path)  
 tweets = pd.DataFrame()  
 texts = []  
 tweets[**'text'**] = list(map(**lambda** tweet: tweet[**'text'**] **if 'text' in** tweet **else None**, tweets\_data))  
 *# for tweet in tweets['tweet']:  
 # texts.append(tweet['text'])  
  
 # tweets['text'] = texts* tweets[**"words"**] = tweets.apply(key\_words,axis=1)  
 counts = tweets[**"words"**].value\_counts()  
 print(counts)  
  
 fig, ax = plt.subplots()  
 ax.tick\_params(axis=**'x'**, labelsize=15)  
 ax.tick\_params(axis=**'y'**, labelsize=10)  
 ax.set\_xlabel(**'Key Words'**, fontsize=15)  
 ax.set\_ylabel(**'Number of Tweets'** , fontsize=15)  
 ax.set\_title(**'Key Words'**, fontsize=15, fontweight=**'bold'**)  
 counts[1:5].plot(ax=ax, kind=**'bar'**, color=**'purple'**)  
  
 pylab.show()

**PyCharm Terminal Input/Output:**

Microsoft Windows [Version 10.0.15063]

(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\diape\PycharmProjects\untitled>cd C:\Users\diape\Desktop\Was on Desktop\School\MSDS 600 Intro to Data Sci

ence\Week 3

C:\Users\diape\Desktop\Was on Desktop\School\MSDS 600 Intro to Data Science\Week 3>oauth2-1.9.0.post1.xml python t

weetsentiment3.py AFINN-111.txt output.txt > newoutput.txt

C:\Users\diape\Desktop\Was on Desktop\School\MSDS 600 Intro to Data Science\Week 3>dir

Volume in drive C is OS

Volume Serial Number is 4845-CAAD

Directory of C:\Users\diape\Desktop\Was on Desktop\School\MSDS 600 Intro to Data Science\Week 3

07/24/2017 01:52 PM <DIR> .

07/24/2017 01:52 PM <DIR> ..

07/24/2017 11:54 AM <DIR> .idea

07/24/2017 11:50 AM 28,093 AFINN-111.txt

07/24/2017 01:11 PM 981 AFINN\_ README.txt

07/24/2017 11:48 AM <DIR> Archives

07/21/2017 06:49 PM 171,133 Capture (2).PNG

07/23/2017 09:23 PM 196,264 Capture.PNG

07/23/2017 11:53 PM 147,310 Capture2.PNG

07/21/2017 06:58 PM 63,927 Capture3.PNG

07/23/2017 11:56 PM 140,355 Capture4.PNG

07/24/2017 12:06 PM 150,075 Capture5.PNG

07/24/2017 12:13 PM 118,691 Capture6.PNG

07/24/2017 12:15 PM 175,634 Capture7.PNG

07/23/2017 08:39 PM <DIR> Downloaded files

07/21/2017 06:40 PM 1,615,420 get-pip.py

07/24/2017 11:49 AM 2,748 key\_words3.py

07/24/2017 01:45 PM 0 newoutput.txt

07/18/2017 08:22 PM 590 oauth2-1.9.0.post1.xml

07/23/2017 08:03 PM 473,512,937 output.txt

07/21/2017 08:23 PM 25,931 Tableau of AFINN.twb

07/24/2017 01:52 PM 3,361 tweetsentiment3.py

07/24/2017 12:27 PM <DIR> Twitterstream

07/18/2017 11:08 PM 1,922 Twitterstream.py

07/23/2017 11:43 PM 1,616 twitterstream\_output\_languages2.py

07/24/2017 12:21 PM 1,591 twitterstream\_output\_languages3.py

07/24/2017 11:49 AM 1,709 twitterstream\_output\_locations3.py

07/24/2017 12:58 PM 11,825 Week 3 Charis Ackerson assignment turned in.docx

07/24/2017 11:39 AM 111,966 Week 3 Charis Ackerson.docx

07/21/2017 09:19 PM 209,708 Week 3 Discussion question.docx

07/24/2017 12:40 PM 19,497,471 Week 3 screengrabs.docx

07/21/2017 08:07 PM 14,059,553 Week 3.docx

26 File(s) 510,250,811 bytes

6 Dir(s) 116,400,312,320 bytes free

C:\Users\diape\Desktop\Was on Desktop\School\MSDS 600 Intro to Data Science\Week 3>python tweetsentiment3.py AFINN

-111.txt output.txt

Traceback (most recent call last):

File "tweetsentiment3.py", line 67, in <module>

main()

File "tweetsentiment3.py", line 40, in main

tweets = readTwitterData(tweets\_file)

File "tweetsentiment3.py", line 22, in readTwitterData

line = json.loads(line) # change json to dictionary

File "C:\Program Files\Anaconda3\lib\json\\_\_init\_\_.py", line 354, in loads

return \_default\_decoder.decode(s)

File "C:\Program Files\Anaconda3\lib\json\decoder.py", line 342, in decode

raise JSONDecodeError("Extra data", s, end)

json.decoder.JSONDecodeError: Extra data: line 1 column 2243 (char 2242)

C:\Users\diape\Desktop\Was on Desktop\School\MSDS 600 Intro to Data Science\Week 3>python twitter

stream\_output\_locations3.py

Rio de Janeiro, Brasil 519

United States 451

Brasil 362

日本 284

المملكة العربية السعودية 271

Venezuela 254

São Paulo, Brasil 232

Argentina 224

Buenos Aires, Argentina 217

México 208

Name: location, dtype: int64

C:\Users\diape\Desktop\Was on Desktop\School\MSDS 600 Intro to Data Science\Week 3>python key\_wor

ds3.py

106046

trump 834

lochte 241

clinton 52

trump,clinton 31

usain 14

lochte,trump 6

Name: words, dtype: int64

C:\Users\diape\Desktop\Was on Desktop\School\MSDS 600 Intro to Data Science\Week 3>python python

key\_words4.py

python: can't open file 'python': [Errno 2] No such file or directory

C:\Users\diape\Desktop\Was on Desktop\School\MSDS 600 Intro to Data Science\Week 3>python key\_wor

ds4.py

107213

Lyme\_Disease 8

Bubonic\_Plague 3

Name: words, dtype: int64