**Connecting to Azure:** <https://docs.microsoft.com/en-us/azure/sql-database/sql-database-connect-query-python> and countless other articles on google.

**Help with database actions and pyodbc:** <https://stackoverflow.com/questions/7744742/cant-create-tables-in-access-with-pyodbc>

**Grabbing database table names:** <https://stackoverflow.com/questions/8334493/get-table-names-using-select-statement-in-mysql?rq=1>

**Converting tweepy object into JSON dictionary:** <https://stackoverflow.com/questions/27900451/convert-tweepy-status-object-into-json>

“The Status object of tweepy itself is not JSON serializable, but it has a \_json property which contains JSON serializable response data.”

**Introduction to Qt5 and PyQt5:** <https://www.youtube.com/watch?v=UjxQRdmvC1k>

I also used several of stackoverflow articles to figure out how to use and access various widgets

**These regular expressions I found on stackoverflow. I forgot to save the page and can no longer find the exact articles.**

hashtags = re.findall(r"#(\w+)", dataset[1])

* This one is simple. Find hashtag # followed by any word character \w repeating +

mentions = re.findall(r'(?<!RT\s)@\S+', dataset[1])

(?<!RT\s) - finds the beginning of a string containing ‘RT’ followed by a space.

@\S+ then an @ symbol followed by any amount of non-whitespace characters

I also used <https://pythex.org/> for testing my own regex,

and <https://docs.python.org/3/library/re.html> for operator information

**Adding stop words:** <https://stackoverflow.com/questions/5511708/adding-words-to-nltk-stoplist>

Simply adding words that we want to omit from counting words, my book introduced stop words as part of the natural language translation kit

**“Python for Programmers” by Paul and Harvey Deitel References (Book):**

99% of the code in the book is in snippets designed to be ran through the interpreter. I used these snippets and modified them as needed to get them to work with the program.

**from** textblob **import** TextBlob  
**from** nltk.corpus **import** stopwords  
**from** operator **import** itemgetter

TextBlob and nltk.corpus is for natural language processing, operator is used for sorting 2d lists.

**This is a list comprehension that my book introduced:**

items = [item **for** item **in** items **if** item[0] **not in** self.stop\_words]  
*# equivalent code  
# items2 = []  
# for item in items:  
# if item[0] not in self.stop\_words:  
# items2.append(item)*

**These are some tweepy methods that my book introduces to gather stream and timeline data, edited for my program.**

1: twitter\_stream = tweepy.Stream(api.auth, TweetListener(api, filename, tweet\_limit))

* initializes twitter stream with Tweet Listener Object

2: twitter\_stream.filter(track=userhandle.TwitterHandle.tolist(), follow=userhandle.TwitterID.tolist())

* filters through the stream, calling TweetListener’s on\_data method

3: for status in tweepy.Cursor(api.user\_timeline, screen\_name=name,tweet\_mode="extended").items():

* iterates through timeline tweets

4: tweets = api.search(q=search\_criteria, count=10000)

* creates a list of tweet objects

All these methods return an object of tweepy that I converted to json format

**The book introduces folium for geo maps using coordinates data I took from tweets with a “place” element.**

self.usmap = folium.Map(location=[39.8283, -98.5795],  
 tiles=**'Stamen Terrain'**,  
 zoom\_start=5,  
 detect\_retina=**True**)

**This p.clean method removes hashtags, mentions, emojis, links, and other binary data harmful to database entry and is part of the tweepy preprocessor library.**

dataset[1] = p.clean(dataset[1])

I also intend to bring my book to class during the presentation to give credit.