CS4624: Multimedia, Hypertext, and Information Access
Visual Displays of School Shooting Data

Gabriel Simmons, Tomy Doan, Peter Park, Evan Keys, Tianna Woodson
Client: Donald Shoemaker
Virginia Tech, Blacksburg, Virginia 24061
5/2/2018

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

EXECUTIVE SUMMARY/ABSTRACT

In order to understand and track emerging trends in school violence, there is no better resource than our current population. Sixty-eight million Americans have a Twitter account, and with the help of the GETAR (Global Events and Trend Archive Research) project, we were able to create datasets of tweets related to 10 school shooting events. Also, we have retrieved several webpages relating to the same shootings. Our job is to use both datasets to develop visualizations that may depict emerging trends.

Based on our data, we created word clouds, maps, and timelines. Our goal was to choose appropriate representations that would provide insight into the changing conversation America was having about gun violence.

To begin, we collected relevant tweets from Twitter. From there, we went through several processes to clean the tweet data to ensure that all of the data was relevant to our project. Using another source, Hydrator, we obtained more detailed information from our datasets, like user profile locations. Additionally, we were tasked with analyzing webpages in a similar fashion. The URLs for these webpages were collected by GETAR and provided to us for this project. We used this URL data by creating word clouds of commonly used words on the webpages and running a sentiment analysis on the words within the webpages. Using this data, as well as our collected tweet data, we created visuals that we believe show the statistical and emotional response to school shootings in the United States.

Included is the final project report for the Visualizations of School Shootings capstone project, given by Dr. Edward Fox of Virginia Tech. Motivation behind this project and its impact can be found in the Abstract and Introduction. Methodologies for collecting, manipulating and visualizing data can be found in the Implementation and Developer's Manual. The User's Manual can be used as a guide to recreate the visuals developed for this project using the data provided in the Supplemental Materials included with the report. An outline of lessons learned while completing this project can be found at the end of the report in the Lessons Learned section.

Included with the report are Supplemental Materials including the suite of visuals that were created for the project, data files derived to create those visuals, and scripting files also used to derive new data.

Those wishing to create visuals similar to those included in the report, as well as our client Dr. Donald Shoemaker and professor Dr. Edward Fox, may find the report useful for understanding the process our team developed to satisfy this project. Those interested in continuing research in this area may also find technologies mentioned in the report useful.

INTRODUCTION

Twitter has become a very popular platform for social media that has given its users the ability to voice their opinions. Any person around the world can share their thoughts on events that take place and Twitter will also then support a community for users to have a discussion. With most tweets being publicly available for anyone to view, it allows researchers to collect and analyze the tweet data. This data is very valuable because tweets expose raw and immediate emotional reactions.

For our project we will take on the task of collecting, analyzing, and organizing Twitter data related to recent school shootings. More specifically we are comparing data against each other, which is done via visualization of data for easier comparison than just raw data. This was developed in an effort to understand and track emerging trends among incidents of school violence in the United States. We used a combination of data mined from the GETAR project and data obtained by our team to develop visual elements to help understand emerging trends

Based on the number of shootings that have occurred, we have agreed to limit our research to 10 shootings from 2007-2017 that had more than four victims. For each of our 10 shootings we will have two separate datasets of tweets and webpages. The tweet dataset contains information regarding the user, tweet, location of tweet, and the date the tweet was made. The URL dataset information is comprised of the URLs for the story, and a short description of the story. Our analysis will be done by creating different visualizations based on this data.

With visualizations, we can compare shootings to find correlations, trends, and common themes. For example, we are able to track if the mentioning of politics increases or decreases with each shooting over time by using word clouds. Our interest lies in seeing if reactions to school shootings are changing and if so how. With the aid of the Virginia Tech's Visualization Studio, we were recommended Tableau[5] which would allow us to demonstrate this.

The programs Tableau and Microsoft Excel were critical to our effort. Tableau was especially useful for finding commonalities in the data. For example, we were able to find where most tweets were coming from in the United States. Tableau allowed us to think of more creative ways to visualize the data we were given. Excel was primarily used for tracking commonly used words within tweets. For example, we often found references to the current US President. Finding commonly used words enables us to compare shootings based on any criteria of our choosing.

The impact of this project should be an improved understanding of sociology and psychology related to school shootings. Hopefully it aids in reducing their frequency and severity.

School Shooting	Year	Location	Details
Virginia Tech	2007	Blacksburg, VA	32 fatalities, 23 injuries
NIU	2008	DeKalb, IL	16 fatalities, 21 injuries
Dunbar HS	2009	Chicago, IL	0 fatalities, 5 injuries
University of Alabama	2010	Tuscaloosa, AL	3 fatalities, 3 injuries
Worthing HS	2011	Houston, TX	1 fatality, 5 injuries
Sandy Hook ES	2012	Newtown, CT	27 fatalities, 2 injuries
Sparks MS	2013	Sparks, NV	2 fatalities, 2 injuries
Reynolds HS	2014	Troutdale, OR	2 fatalities, 1 injury
Umpqua CC	2015	Roseburg, OR	10 fatalities, 9 injuries
Townville ES	2016	Townville, SC	2 fatalities, 2 injuries

Table 1: The 10 school shootings being studied for this project, Dunbar HS was eventually excluded from our tweet data, see Lessons Learned for more info

REQUIREMENTS

PROJECT DESCRIPTION

Collecting, analyzing, and organizing information related to recent school shooting events. Identifying trends and documenting them from the data. Preparing visualizations and displays from available school shooting collections, especially the sample of shootings from 2007-2017.

PROJECT DELIVERABLES

- Database of key information about each event
- Improved webpage archive for each event
- Timelines, word clouds, and other visual displays

VISUAL THEMES

- Themes of tweets and webpages
- Location of tweets: Geotags or Home location of person who tweeted
- Emotional response tones/themes: Measured in volume/frequency
- Clustering of terms/related incidents

VISUAL TYPES

- Word Clouds
- Geographical heat maps of tweet locations
- Point maps of locations of tweet locations
- Tweet volume over time
- Sentiment evaluations over time
- Bar and pie charts

MINIMUM DELIVERABLES

10 static visuals that help to visualize the data gained from the GETAR project, and any correlation from that data.

STRETCH GOALS

15-20 visuals, and an improved database containing the data for school shootings.

TEAM MEMBERS AND ROLES

- Gabriel Simmons, team lead and creation of visuals
- Tianna Woodson, data collection and creation of visuals
- Evan Keys, data cleaning and creation of visuals
- Peter Park, sentiment analysis and data collection
- Tomy Doan, data collection

DESIGN

The main focus of our project, as stated before, is to create visuals. The design aspect of our project comes from choosing the appropriate designs to represent our data. Based on the data that we received we were able to create visuals that used maps, timelines, bar charts, pie charts, and word clouds. Each of these visual types are able to display information in unique ways that allow us to analyze these shootings from multiple angles.

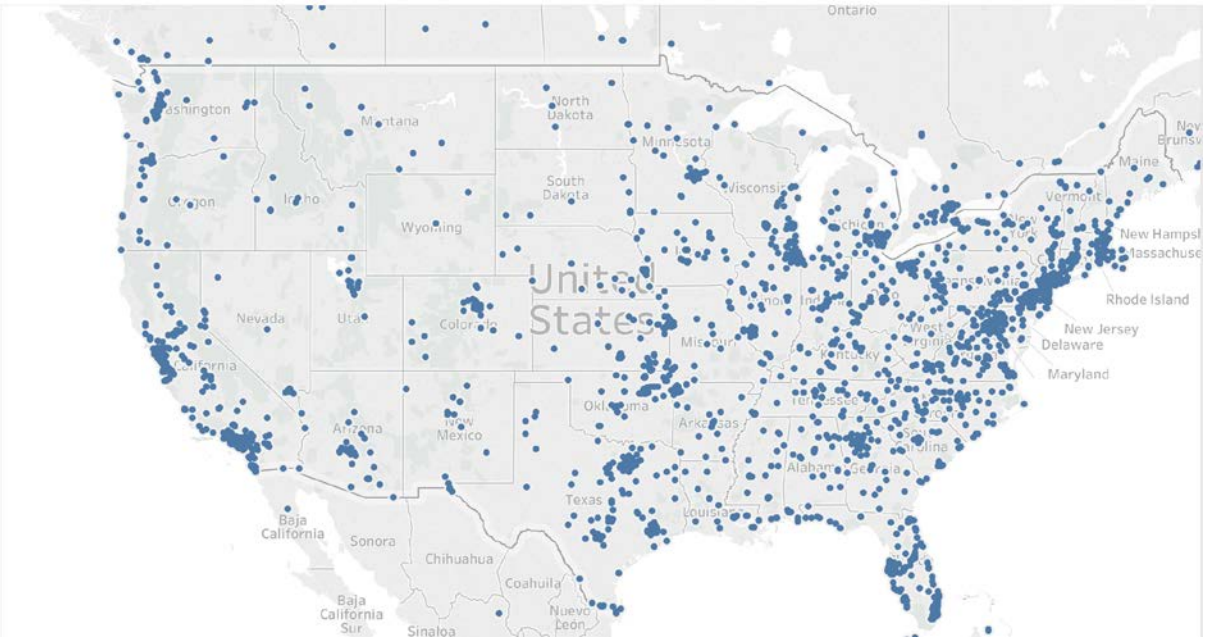


Figure 1: Virginia Tech shooting response by location

These types of maps (see Figure 1) allow us to track where the shootings are taking place and where the tweets are coming from. There are a couple of observations that can be made from a map plot. We are able to see if the location of a shooting effects where tweets come from. It is possible that there would be a greater country wide discussion if the shooting took place in a more populous area and this would be evident if the plot of tweet locations was evenly spread throughout the country. Furthermore, we can see which states have a larger discussion on shootings. Along with the gun laws in those respective states, we may be able to provide further analysis.

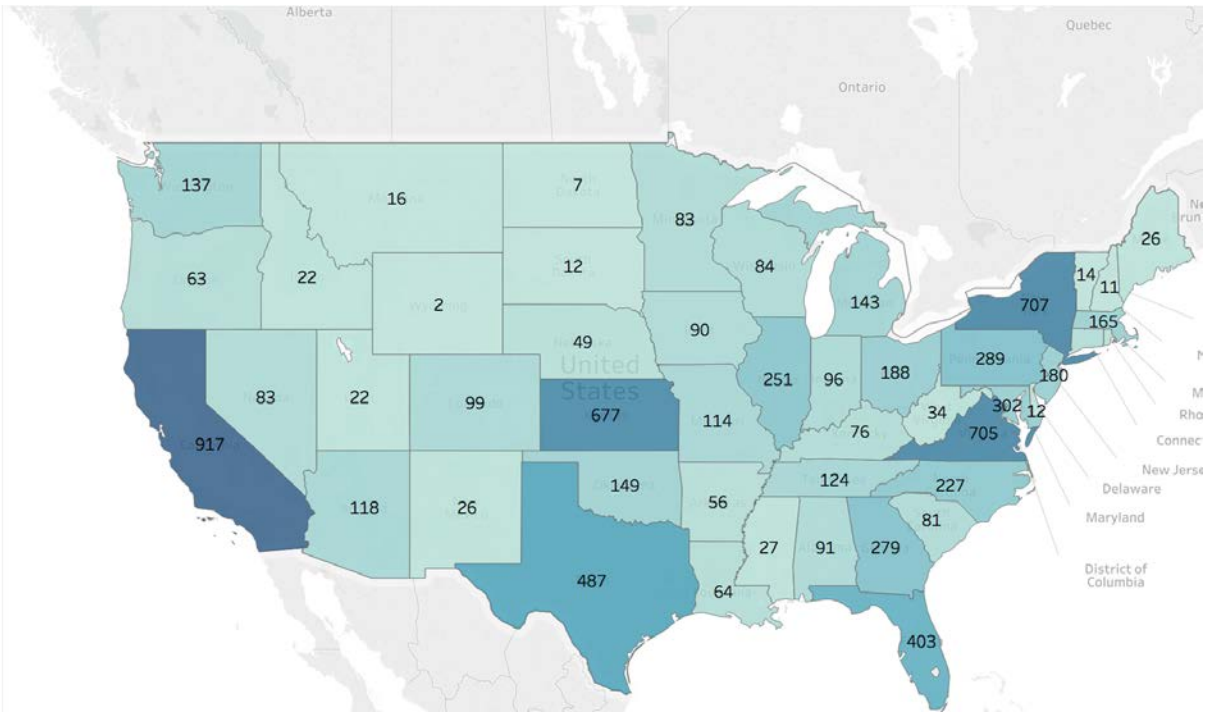


Figure 2: Virginia Tech shooting response heat map

This type of visual (see Figure 2) was made in response to some feedback that we got back at one of the GETAR meetings we attended. It essentially shows the same thing as the point map above, but in a potentially clearer way. If you are more concerned with tweet response by state, then this visual is easier to understand. If you are concerned with tweet response by region, then the point map is a good choice.

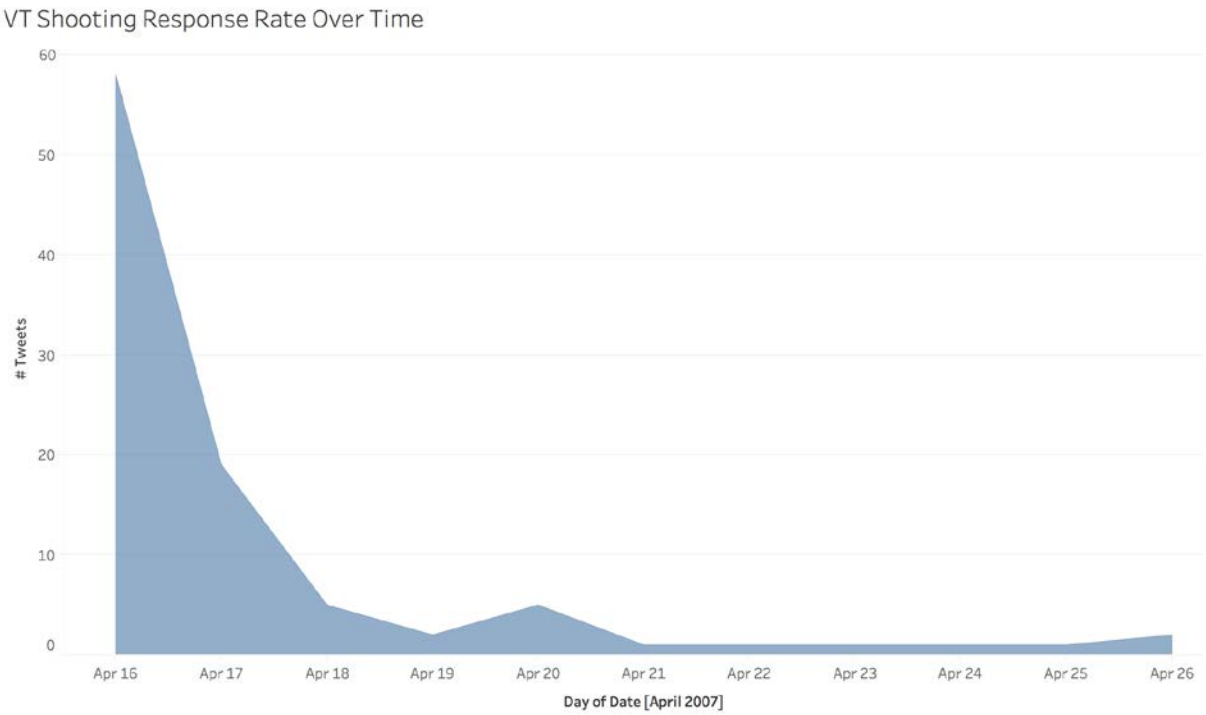


Figure 3: Virginia Tech Tweets over Time

Timelines like these (see Figure 3) are especially useful for tracking when tweets are made over a certain time period. With these timelines we are able to observe how long the conversation about a shooting goes on. An interesting correlation that can come from timelines is if the conversation period of time is smaller or larger depending on the number of victims in the shooting. We also have the ability to expand the period of time. We can check to see if shootings are discussed again in the future and this may be due to the anniversary of the shooting or maybe other shootings occurring.

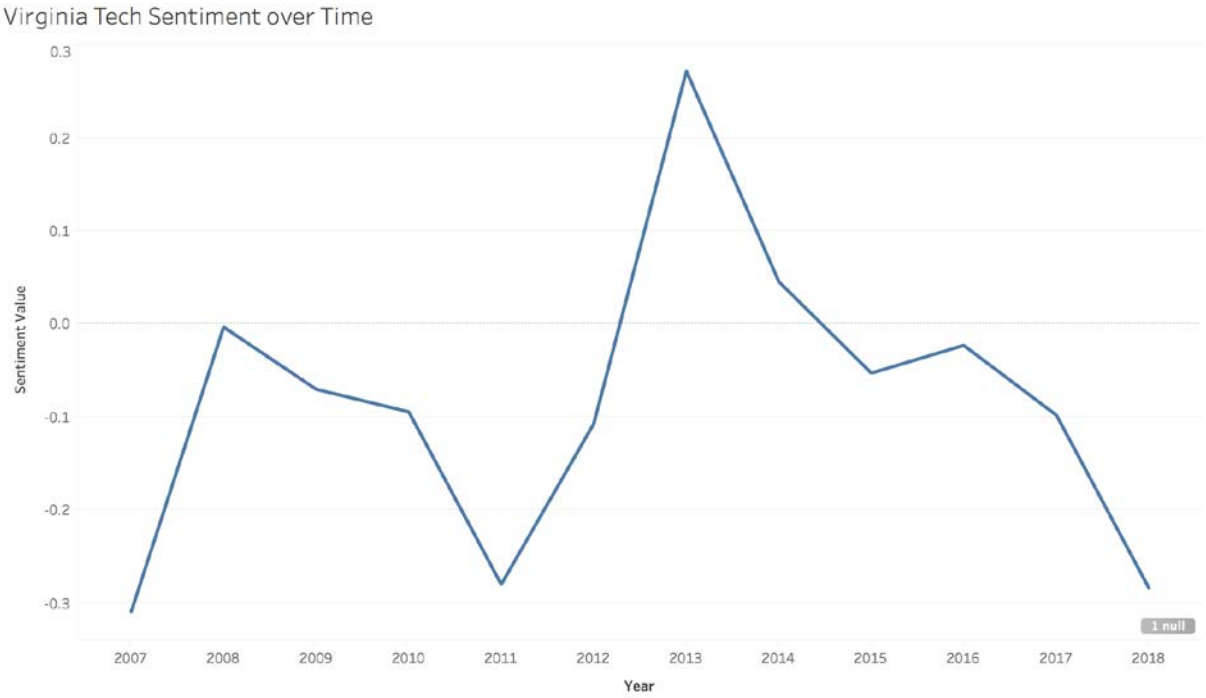


Figure 4: Virginia Tech Tweets over Time

Another goal of this project was to assess how public reactions are changing over time. To do this we used a sentiment analyzer on the tweets which returned a value between negative one and one in which negative one meant very negative sentiment and one meant very positive sentiment[2] (see Sentiment Analysis section for more details). We then plotted the sentiment values of the tweets over the time of the tweet. From these kinds of visuals (see Figure 4), you can see how sentiment has changed over time.

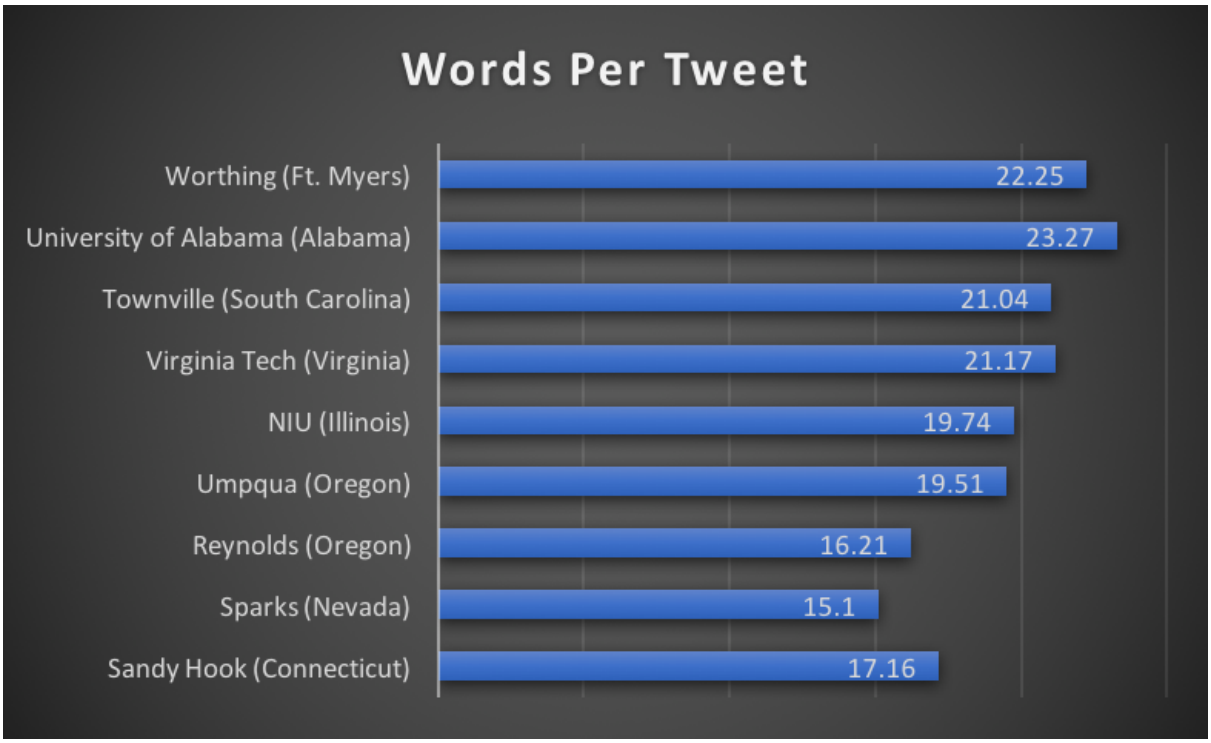


Figure 5: Average number of words per tweet for each shooting

Bar graphs help us immensely when comparing statistics, and can shed some light on certain aspects being more important than others. Specifically we used bar graphs to compare the average word count of tweets (see Figure 5). We decided not to use this visual anymore because we felt that it would not add any analysis for the project. Getting the number of words per tweet doesn't tell us anything special about the event. After discussing with the client we came to an agreement that it was best to not continue making this visual.

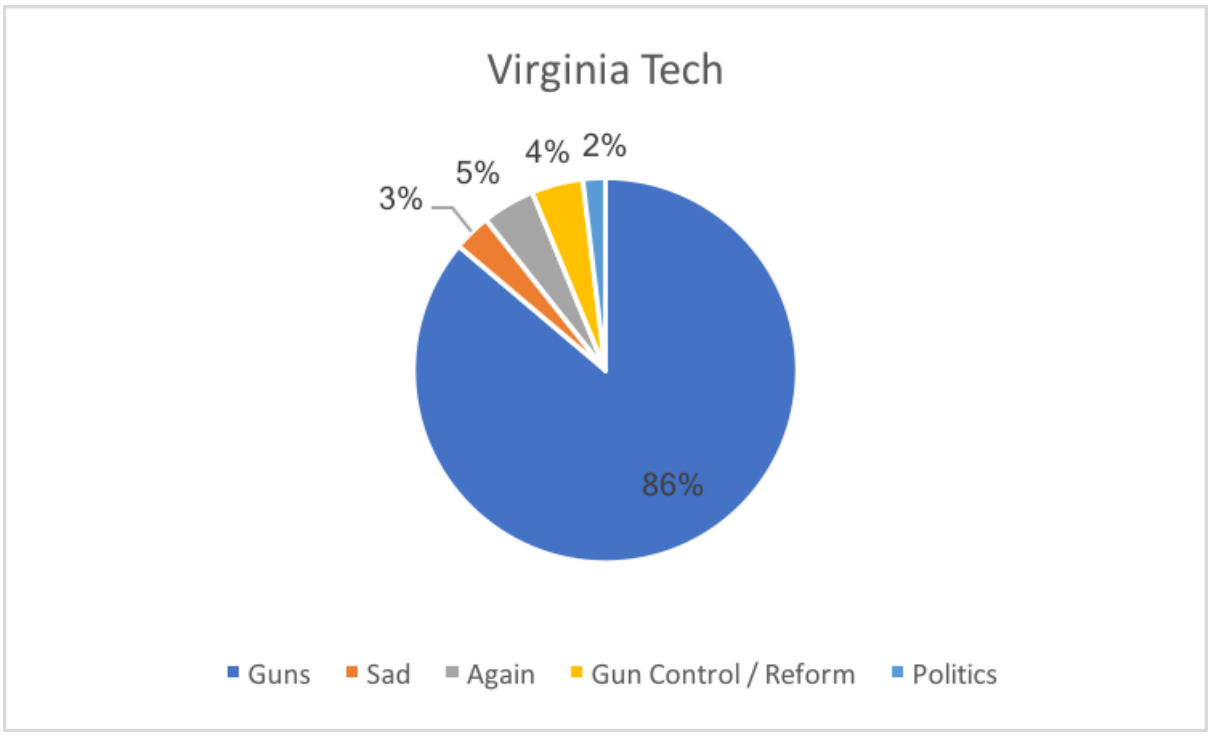


Figure 6: Virginia Tech keywords pie chart

Pie charts are used mostly for comparing shootings based on keywords of our choosing. We are able to create any criteria we need via keywords and displaying the results in a pie chart clearly shows the differences between the events. The main benefit is that these charts (see Figure 6) are visually simpler than other types of graphs and makes it easier to understand the information that is being displayed. These charts are very versatile and allow us to compare data in multiple ways.

Figure 7: Virginia Tech shooting webpage world cloud

Word clouds, at first glance, are good for seeing what words stand out. For our project, its useful for seeing what aspects of a particular shooting people find important (see Figure 7). One feature of word clouds is that the size of a word is tied to the number of times the word is mentioned. This can show if webpages focus more on the victims, shooter, or possibly politics. Comparing word clouds of various shootings can easily show a change in the topic of discussions.

IMPLEMENTATION

DATA COLLECTION

After retrieving tweet data from GETAR, we learned that the datasets contained inconsistencies, along with irrelevant data. Due to these inconsistencies, the decision was made to collect new tweet data. We utilized a Python script named get-old-tweets that simulates a Twitter user manually searching for tweets using a search bar on Twitter's website[3]. This method is not limited by Twitter as there is no limitation to how many tweets a user may view when searching on Twitter's website.

The way the Python script works is that it mimics the way a user interacts with the website. When a query search is made, the exporter will attempt to continuously scroll down to grab tweets that are relevant to the query search through JSON objects. After, it is encoded into a CSV with the tweet's properties such as ID, permalink, username, text, etc.

Shooting	Tweet Count
Virginia Tech	26045
NIU	7524
University of Alabama	7062
Worthing HS	146
Sandy Hook ES	16081
Sparks MS	1045
Reynolds HS	9301
Umpqua CC	159
Townville ES	6667

Table 2: Tweet counts for each shooting, before cleaning

Shooting	URL Count
Virginia Tech	1500
NIU	19
Dunbar	258
University of Alabama	326
Worthing HS	1500
Sandy Hook ES	2000
Sparks MS	1501
Reynolds HS	522
Umpqua CC	1501
Townville ES	1501

Table 3: URL counts for each shooting, before cleaning

DATA CLEANING

To facilitate cleaning our datasets, we used Excel's built-in functions. For example, to remove duplicate tweets from the same user, the Remove Duplicate button was used. Several equations outlined in the Developer's Manual were used to filter out data that we determined to be irrelevant. Finally, as a last precaution, we manually inspected each data file in order to ensure that our data was relevant, manually deleting data when necessary.

Shooting	Tweet Count
Virginia Tech	24445
NIU	4701
University of Alabama	1428
Worthing HS	146
Sandy Hook ES	16015
Sparks MS	1305
Reynolds HS	3326
Umpqua CC	157
Townville ES	6598

Table 4: Tweet counts for each shootings, after cleaning

Shooting	URL Count
Virginia Tech	140
NIU	9
Dunbar	10
University of Alabama	71
Worthing HS	40
Sandy Hook ES	160
Sparks MS	3
Reynolds HS	51
Umpqua CC	32
Townville ES	15

Table 5: URL count for each shooting, after cleaning

HYDRATING TWEETS

HYDRATOR

After showing our client a visual where we plotted user locations corresponding to where they were when they tweeted on a US map, our client became very interested in creating similar visuals for all of our data sets. Unfortunately, very few tweets are tagged with the geolocation of the user, as this is a feature Twitter users can turn on and off. For a dataset with over 500 tweets, we were left with roughly 10-15 locations. Since this was not enough data to make a meaningful visual, we were advised by a GETAR liaison that hydrating our tweets would provide more information about the tweet itself, even the location the user has put on their profile. Our team decided to use this information to make our visuals because we believed that profile locations, if they are valid locations, would be a good representation of where users are when they tweet.

Hydrator is a free, open source tool developed by Ed Summers, a software developer at the University of Maryland[1]. Hydrator is a desktop tool that allows you to extract more meaningful information about a tweet from only the tweet's ID[1]. Data collected from hydration includes user profile information including location, tweet information including interactions and content, as well as much more. This method of hydrating tweets provides us a way to gather more information about tweets and does not interfere with any of Twitter's policies on distributing tweet information. It also allows us to gather information without incurring a cost of using Twitter's API. Detailed steps highlighting our use of Hydrator are given in the Developer's Manual.

We were able to use hydrator to hydrate all of the remaining tweets after cleaning our datasets, since our datasets included tweet IDs for each tweet.

One of the issues we found with continuing our use of hydrator was that our old datasets, provided by GETAR, did not include a tweet ID. This was one of the main factors behind our resolution to collect new tweet data. Details about how that was implemented are given in the Developer's Manual.

EXTRACTING AND VERIFYING LOCATION

The result of our data collection was ten Comma Separated Value (CSV) files containing our tweet information. From there we used several Python scripts to extract necessary information from these files and generate new data.

First, we used the file titled extract_tweet_ids.py to parse through our datasets and extract the tweet's ID using the hyperlink for the tweet. Since all hyperlinks that lead to a tweet are ended by that tweet's ID, we simply split the hyperlink by a / delimiter and collected the last element for each tweet. For each tweet, we simply appended the tweet ID to the end of the schoolshootingname_tweetids.txt file that was created.

From here, we used the desktop Hydrator application with the tweet ID file and created the schoolshootingname_hydrated_tweets files, both in (JavaScript Object Notation) JSON and CSV format. The next step for collecting user profile locations was parsing through the hydrated tweets and extracting that information. To do this we used the file titled extract_user_locations.py and the CSV version of our hydrated tweets. This data collection resulted in the schoolshootingname_user_locations.csv files.

Once we were able to generate the files containing the profile location for our users we noticed a small issue. The Twitter app does not require that users enter a valid location for their profile. After looking through our new datasets we discovered locations such as #DMV, Worldwide, and other non-valid locations. Since these datasets were as large as 24000 data points we needed a way to systematically verify that the user location corresponded to an actual location. To do this we created another Python script named verify_user_locations.py that would parse through each location and send it to the Google Maps Geocoding API for verification. Since the API limits users to 2500 requests per day, before we send the location to the API we went through several steps to eliminate bad requests. For example, if a location started with the # character, the character was removed and then the request was made. We made this decision after seeing that many people will put locations such as #Virginia in their profile and we do not want to throw away these kinds of data points. Next, if there were any non-ASCII characters in the location, those were removed. Finally, if a user's location was longer than 25 characters the entry was never sent to the API. We made this decision after noting that locations longer than 25 characters often have full sentences or multiple locations in them, neither of which would have resulted in a good request. After cleaning the location, we sent it to the API which returned the GPS coordinates of the street address if it was a legitimate location on Earth. With this data set, named schoolshootingname_verified_user_locations.csv, we created several visuals using Tableau which plot these points on a map.

The final Python script titled extract_states_googlemaps.py was used in a similar way as the verify_user_locations.py file. One of our visuals is essentially a heat map of tweet density in the United States, and to create the visual we needed tweet IDs matched up with the state they came from. The extract_states_googlemaps.py file simply takes the GPS coordinates we have and sends them to Google Maps' Reverse Geocoding API, which returns a JSON object including the state (if the location is in the US) the coordinates correspond to. The files titled schoolshootingname_user_states.csv were the result of these executions.

DISCLAIMERS

We understood that it was possible that users did not currently live or were not currently at the locations specified in their user profile when they created the tweets that we mined. After consideration, we believed that the data was still pertinent. Also, because Twitter makes no effort to verify locations that users put into their profiles, there were several instances of users putting GPS coordinates into their location. On several instances, we

noticed that these coordinates correlated to locations that were in impossible places, such as the middle of the Atlantic Ocean. Since there is no way to systematically eliminate these locations, as they are valid locations, we've chosen to leave them within the dataset.

All of the Python scripts mentioned in this section have been included as supplemental material with this report. Information about Hydrator can be found at https://github.com/DocNow/hydrator.

SENTIMENT ANALYSIS

NLTK

After that, we were told to look at alternatives for sentiment analysis. We were then recommended the Natural Language Toolkit[2]. NLTK has multiple functions built in, ranging from tokenization of words to Twitter tweet collection. The reason we did not use NLTK for tweet collection was due to limits of free API access. Since NLTK's sentiment analysis uses Python and takes in an input of string, the script we wrote loads the JSON array of string directly to be analyzed.

VISUAL CREATION

To create the visuals in our project, we decided to use the software called Tableau for two main reasons:

1. Tableau is very user friendly and it does not require any coding experience to create high quality visuals. Our client for this project is a sociologist, and we wanted to make sure that our client could make similar visuals with different data in the future if need be.
2. Tableau is free for all Virginia Tech students and teachers with valid email addresses. Other options cost money and to limit any constraints on future research, we decided to go with Tableau because of its high quality results and free price tag.

EVALUATION/ASSESSMENT

Our project focused on using many software analysis tools to produce visuals. Thus, in terms of testing code, this was not a concern. Our focus was on a feedback loop with Dr. Shoemaker in making sure the visuals we generated were exactly what he needs. Before we could make most of our visuals, we had a few things to complete: clean the data for noise and supplement sparse data where possible.

NOISE

Upon creating some of the visuals we noticed some trends that suggested that we needed to clean our data. For example, in the tweets over time visual (Figure 3) we found that there were two spikes in the data that were mostly unrelated to the event. The other two spikes dealt with other shootings in Nevada that were not related. After speaking with Dr. Shoemaker, he decided to focus on the event in a more local span of time. So, we assessed our time scope to just a couple of weeks around the incident.

We noticed that we were finding a good amount of irrelevant tweets and webpages in our datasets, so we used Microsoft Excel to clean the data to the best of our abilities. To make sure that the data was about the correct subject matter, we used keywords to filter the dataset. With this, we removed tweets that were unrelated to the shooting location. After, we had to make sure that the tweets were about the actual shooting event. For example, the Virginia Tech dataset contained tweets referencing the school's basketball team's shooting performance. For this we used keywords again to remove tweets that would contain basketball references such as 'percentage' or 'halftime'. This same methodology was used to filter the datasets for the given webpages.

We also ran into an issue with Twitter bots. These bots are "dummy" Twitter accounts that repeatedly tweet the same thing over and over in order to spam users' Twitter feeds. To combat this, we removed instances where the same user tweeted the same thing multiple times.

Along with our client, we decided to remove retweets from our dataset as well. We only kept retweets where a user added a comment about the retweet. Simple retweets did not add anything of concern to our datasets.

SPARSE DATA

In a meeting with our client, we discussed mapping locations of tweets and visualizing initial reactions to incidents. To do the mapping we first intended to use the geo coordinates of tweets. However we found that this data was very sparse and a better way to track tweet locations was to use the users' home locations specified on their user profile. More people tend to specify where they live. The data from the GETAR database did not contain this information which led to us having to hydrate our data. The hydration is explained more in Problems under Lessons Learned. Once we had the user locations, we had to clean this data for noise and create a system for plotting points that could show the different degrees of accurateness of the points. For example, if someone used the geotag while tweeting, we can be sure that they were actually in that location when they tweeted so that information was prioritized over home location. If the home location for user was specific in that it contained a state and a city, it was prioritized second and was also included as an actual dot on the map, but in a different color. Finally, if the user location only contained what state they live in, we counted that towards the shaded statistic of the state (the darker the state is shaded the more responses). We agreed with our client in that the resulting visual might be hard to understand, thus we only created visuals based on user profile locations. This was done in order to visualize initial reactions to incidents, which required us

to have data from the day the event occurred. The data we had for the 10 shootings did not have all the tweet data that surrounding the incident. For example, our initial data set for Virginia Tech only had tweets that went as far back as 2012. When our client became aware of this, we agreed to look into getting the older tweets even though collecting data was not in our specified contract. More on how the collection was made is in the Lessons Learned under Problems.

RESULTS

The results generated for this project include a suite of visuals as well as multiple datasets used to create those visuals. Details about why and how we chose to design these visuals can be found in the Design section of this report. The visuals themselves, as well as the datasets, will be included with this report as Supplemental Material.

DEVELOPER'S MANUAL

In order to provide guidance in making visuals similar to the ones created for this project, we have included this Developer's Manual. Step-by-step instructions for how we collected our data, cleaned our data, hydrated our data, and created visuals can be found below. Tools like Excel, Tableau[5], Hydrator[1], and others used in this project are free and available to the public.

DATA COLLECTION USING GET-OLD-TWEETS

The get-old-tweet[3] Python script requires Python 2.7 to be installed on the computer being used. The script can be downloaded from this github page: https://github.com/Jefferson-Henrique/GetOldTweets-python. After downloading the script, you will need to open up a command line terminal and move to the directory that the script is in. From there, to ensure that the script will run correctly, run the command "pip install -r requirements.txt". After the requirements have installed, the capabilities of the program can be seen by running "python Exporter.py -h". For this project we used "python Exporter.py --querysearch "your query search here" --maxtweets 1". You can replace the text in "your query search here" with a specific query, and you can change the number after max tweets to try to grab that number of tweets.

DATA CLEANING EXCEL

Excel was essential to our process of cleaning data. It has simple tools that we used. To remove duplicate entries you can click *Remove Duplicates* within the *Data* tab (see Figure 8). There you will be prompted to select columns in which you want duplicates removed.

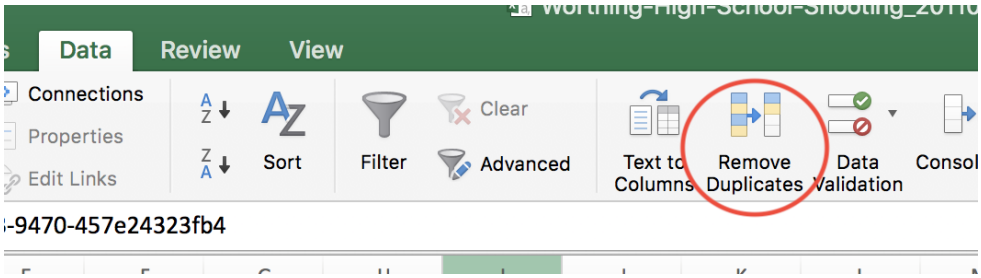


Figure 8: Remove Duplicates within the Data tab.

To filter data, select a column of choice then click *Filter* within the Data tab (see Figure 9).

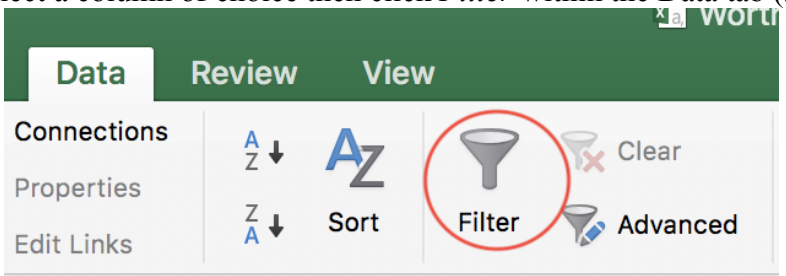


Figure 9: Filter within Data tab.

You will then be asked to enter a keyword. The *Choose One* drop-down menu gives a list of choices for methods of filtering. For example, you can choose to filter for entries that begin with the word "basketball" (see Figure 10). For tweets we filtered the actual tweets and for

URLs we filtered the webpage information to make sure it contained keywords pertaining to the event.

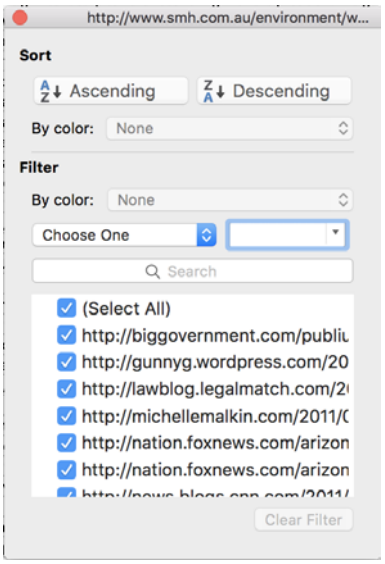


Figure 10: Enter filter keyword.

Excel was also used for word counting keywords. This involved using simple formulas.

To count the total number of words in the dataset:

=SUMPRODUCT(LEN(TRIM(Range))-LEN(SUBSTITUTE(Range," ",""))+1)

- "Range" represents range of cells in the dataset

To count the number of times a certain word was used:

=SUMPRODUCT((LEN(Range)-LEN(SUBSTITUTE(Range,"Word","")))/LEN("Word"))

- "Range" represents range of cells in the dataset and "Word" is the keyword you are

TWEET HYDRATION USING HYDRATOR

Hydrator was used to gather more information about our tweet data. After generating a plain text file containing only the tweet IDs the file can be uploaded to Hydrator from the desktop app, which can be found here: https://github.com/DocNow/hydrator. To begin using Hydrator, sign into the desktop app, and under the Add tab, upload your data files containing a list of tweet IDs in plaintext (txt) (see Figure 11).

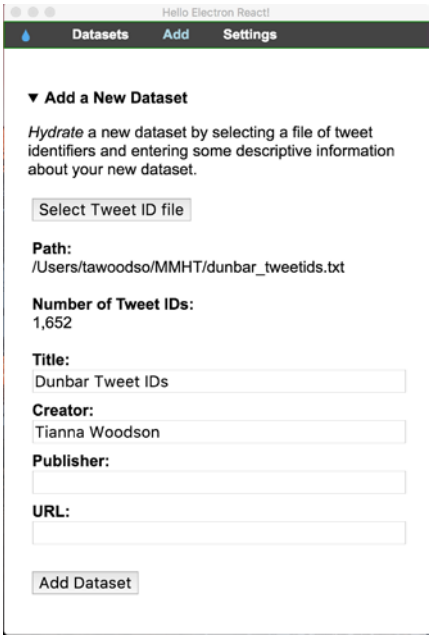


Figure 11: Tweet Hydration

From within the app, navigate to the Datasets tab and from here CSV or JSON files can be created from the gathered data.

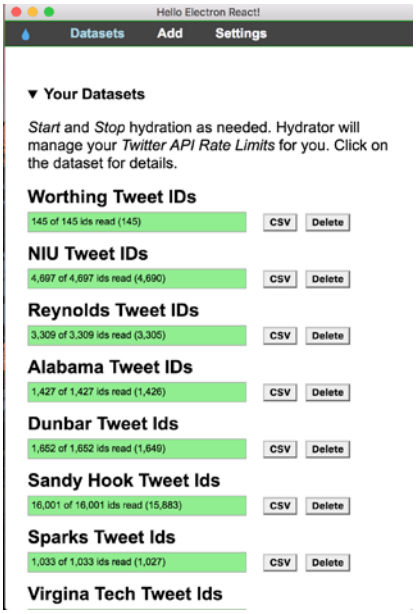


Figure 12: Tweet Hydration B

NATURAL LANGUAGE TOOLKIT - SENTIMENT ANALYSIS

We used the Natural Language Toolkit for our sentiment analysis.

INSTALLATION

NLTK requires Python versions 2.7, 3.4, or 3.5

Mac/Unix

1. Install NLTK: run sudo pip install -U nltk

2. Install Numpy (optional): run sudo pip install -U numpy
3. Test installation: run python then type import nltk
   For older versions of Python it might be necessary to install setuptools (see http://pypi.python.org/pypi/setuptools) and to install pip (sudo easy_install pip).

Windows

These instructions assume that you do not already have Python installed on your machine.

32-bit binary installation

1. Install Python 3.5: http://www.python.org/downloads/ (avoid the 64-bit versions)
2. Install Numpy (optional): http://sourceforge.net/projects/numpy/files/NumPy/ (the version that specifies python3.5)
3. Install NLTK: http://pypi.python.org/pypi/nltk
4. Test installation: Start>Python35, then type import nltk

SENTIMENT ANALYSIS

```
>>> from nltk.classify import NaiveBayesClassifier
>>> from nltk.corpus import subjectivity
>>> from nltk.sentiment import SentimentAnalyzer
>>> from nltk.sentiment.util import *

>>> n_instances = 100
>>> subj_docs = [(sent, 'subj') for sent in
subjectivity.sents(categories='subj')[:n_instances]]
>>> obj_docs = [(sent, 'obj') for sent in
subjectivity.sents(categories='obj')[:n_instances]]
>>> len(subj_docs), len(obj_docs)
(100, 100)
```

Each document is represented by a tuple (sentence, label). The sentence is tokenized, so it is represented by a list of strings:

```
>>> subj_docs[0]
(['smart', 'and', 'alert', ',', 'thirteen', 'conversations', 'about', 'one',
'thing', 'is', 'a', 'small', 'gem', '.'], 'subj')
```

We separately split subjective and objective instances to keep a balanced uniform class distribution in both train and test sets.

```
>>> train_subj_docs = subj_docs[:80]
>>> test_subj_docs = subj_docs[80:100]
>>> train_obj_docs = obj_docs[:80]
>>> test_obj_docs = obj_docs[80:100]
>>> training_docs = train_subj_docs+train_obj_docs
>>> testing_docs = test_subj_docs+test_obj_docs

>>> sentim_analyzer = SentimentAnalyzer()
>>> all_words_neg = sentim_analyzer.all_words([mark_negation(doc) for doc in
training_docs])
```

We use simple unigram word features, handling negation:

```
>>> unigram_feats = sentim_analyzer.unigram_word_feats(all_words_neg,
min_freq=4)
>>> len(unigram_feats)
83
>>> sentim_analyzer.add_feat_extractor(extract_unigram_feats,
unigrams=unigram_feats)
```

We apply features to obtain a feature-value representation of our datasets:

```
>>> training_set = sentim_analyzer.apply_features(training_docs)
>>> test_set = sentim_analyzer.apply_features(testing_docs)
```

We can now train our classifier on the training set, and subsequently output the evaluation results:

```
>>> trainer = NaiveBayesClassifier.train
>>> classifier = sentim_analyzer.train(trainer, training_set)
Training classifier
>>> for key,value in sorted(sentim_analyzer.evaluate(test_set).items()):
...     print('{0}: {1}'.format(key, value))
Evaluating NaiveBayesClassifier results...
Accuracy: 0.8
F-measure [obj]: 0.8
F-measure [subj]: 0.8
Precision [obj]: 0.8
Precision [subj]: 0.8
Recall [obj]: 0.8
Recall [subj]: 0.8
```

SIMPLE EXAMPLE OF PYTHON SCRIPT

Run it using <name of script>

```
import json
import sys
from nltk.sentiment.vader import SentimentIntensityAnalyzer

filename = sys.argv[-1]
with open(filename, encoding="utf8") as data_file:
        data = json.load(data_file)
        sid = SentimentIntensityAnalyzer()
        with open('file.txt', 'w') as f:
                for sentence in data:
                        #print(sentence.encode("utf8"), file=f)
                        ss = sid.polarity_scores(sentence)
                        for k in sorted(ss):
                        # do '{0}: {1}, ' if details are needed
                                print('{1}, '.format(k, ss[k]), end='', file=f)
                        print(file=f)
        print("--------------------------------Done!")
```

This code takes in a JSON array of strings, and runs the sentiment analyzer over each index in the array. Therefore we had to format the text of the tweet and webpage collections into JSON arrays. These JSON arrays can be found in the Supplemental Materials under Datasets.

It outputs a file.txt containing all the sentiments in the following order: compound, negative, neutral, and positive.

SENTIMENT ANALYSIS RESULTS

The output for the sentiment analysis can be found in the Supplemental Materials under the Datasets folder under the Sentiment_Analysis_Results folder. In that folder are the individual results for each webpage and tweet for a given shooting. There is also a summary of the results that contains an average sentiment value and a min and max value for each shooting. Here are some examples of the most negative and positive results found for the Virginia Tech shooting for both webpages and tweets:

Tweets

Most Positive: 0.9622

“Congratulations, @chronicle reporter @erichoov ! His incredibly thoughtful piece, The Arc of Her Survival, has won the 2018 Eugene S. Pulliam National Journalism Writing Award. Read the thoughtful profile of a Virginia Tech shooting survivor here: https://www. chronicle.com/article/The-Ar c-of-Her-Survival/239744/#.WqfJzc1rUQM.twitter”

Most Negative: -0.9877

“Texas church shooting: 26 killed Las Vegas shooting: 58 killed Virginia Tech shooting :32 killed Newton shooting:27 killed Sandy Hook: 27 killed Orlando Nightclub:49 killed Stoneman Douglas High: more than 17 killed All and many more shootings have one thing in common: The AR-15 pic.twitter.com/pc8R65Tvs7”

Webpages

Most Positive: 0.9723

URL: http://www.campussafetymagazine.com/Channel/University-Security/News/2010/11/23/Court-Ruling-Allows-Virginia-Tech-Officials-to-Be-Sued-Over-Shooting.aspx

“...aspects of safety measures including access control video surveillance mass notifications and security staff practices. Take advantage of a free subscription to Campus Safety today and add its practical insights product updates and know-how to your toolkit. Lawsuits School Shooting Virginia Tech Comments…”

Most Negative: -0.9999

URL: https://thelede.blogs.nytimes.com/2007/04/16/shooting-at-virginia-tech/comment-page-72

“...This does nothing of course to ease the pain and suffering of all those in Virginia. But gun violence of a certain type has come about and that is largely copy catting from the movies. Add to that the immediate celebrity of any shooter adds glamour to the deranged minds…”

USER'S MANUAL

TABLEAU VISUAL CREATIONS

In this section we will be detailing how someone could edit some of the visuals without any coding knowledge. We used a software called Tableau to create the following kinds of visuals:

- Tweets over Time
- Sentiment over Time
- Point Maps
- Heat Maps

The goal of this section is to show how visuals can be recreated with new or updated data in the future if need be.

First you will need to download the software. Tableau can be downloaded for free with a Virginia Tech email. The software is free for students and teachers. Just simply Google "Tableau for Students" and it should be one of the first options with a logo (see Figure 13).

Figure 13: Tableau logo for easy reference

Then there should be an option to get Tableau for free (see Figure 14).

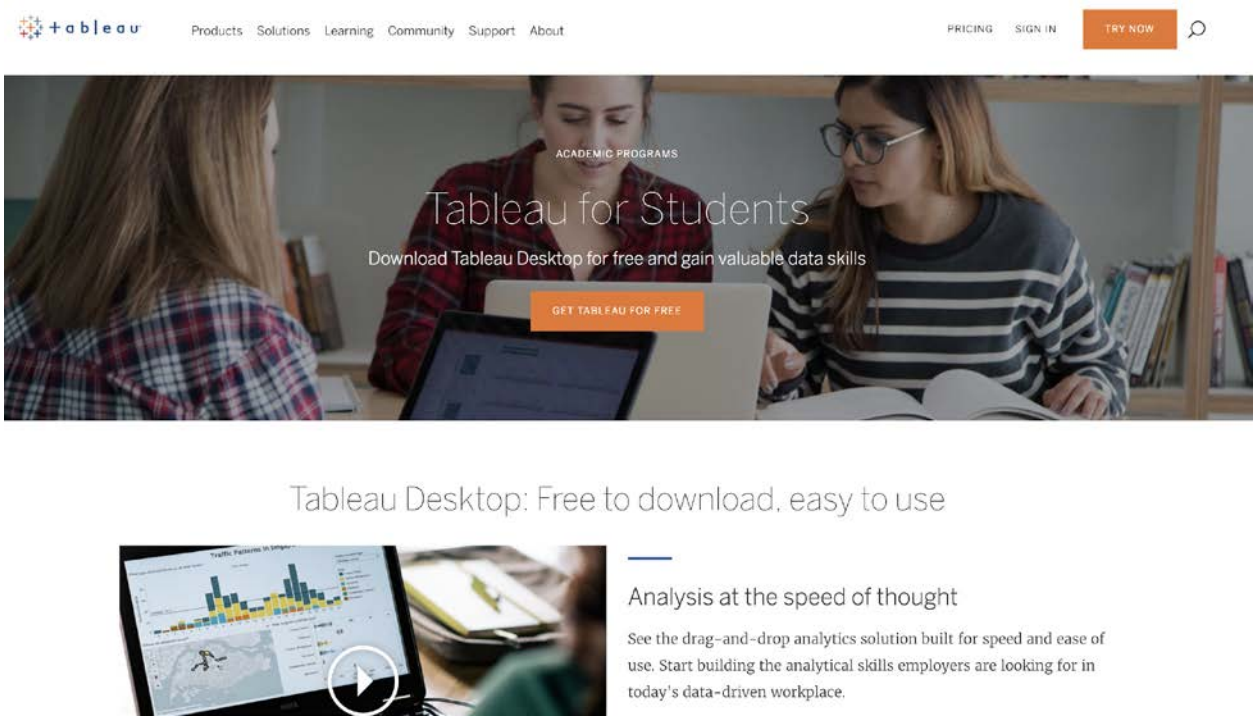


Figure 14: Tableau for Students webpage

Then you'll just need to provide some basic information to confirm that you are a student or teacher (see Figure 15).

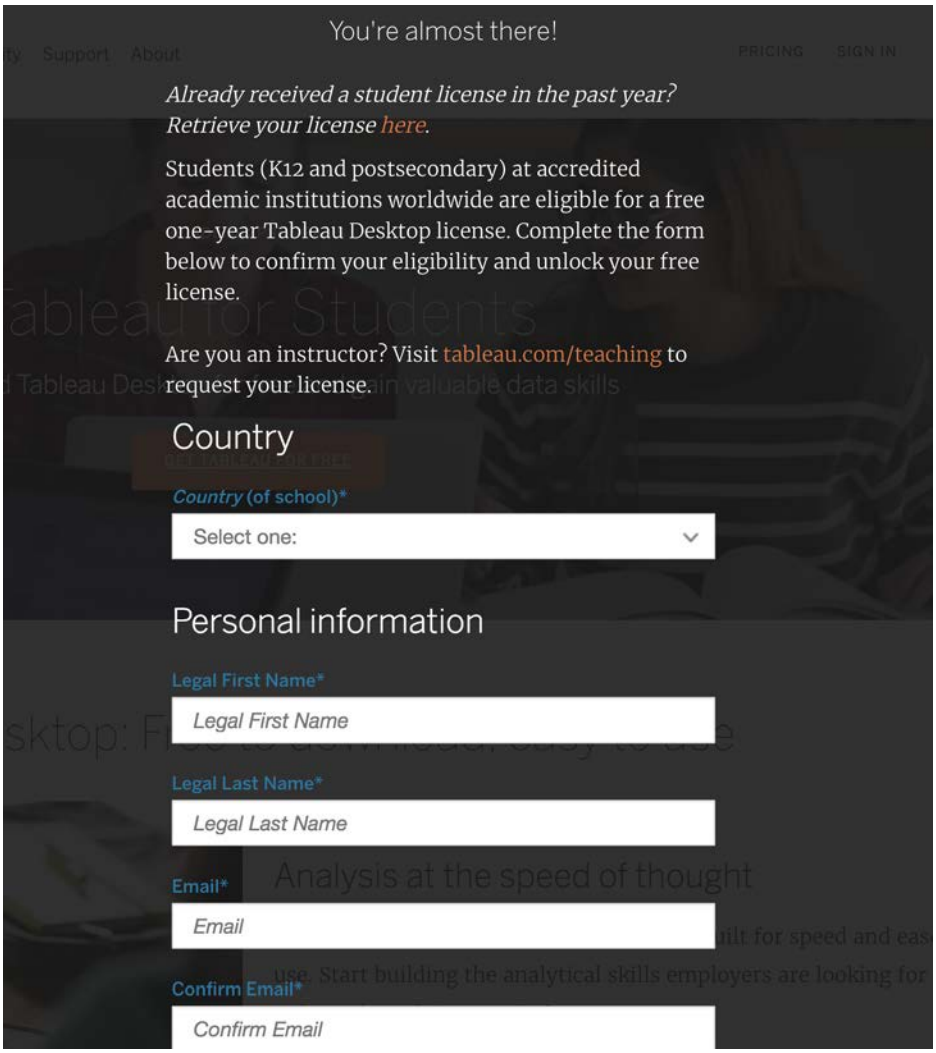
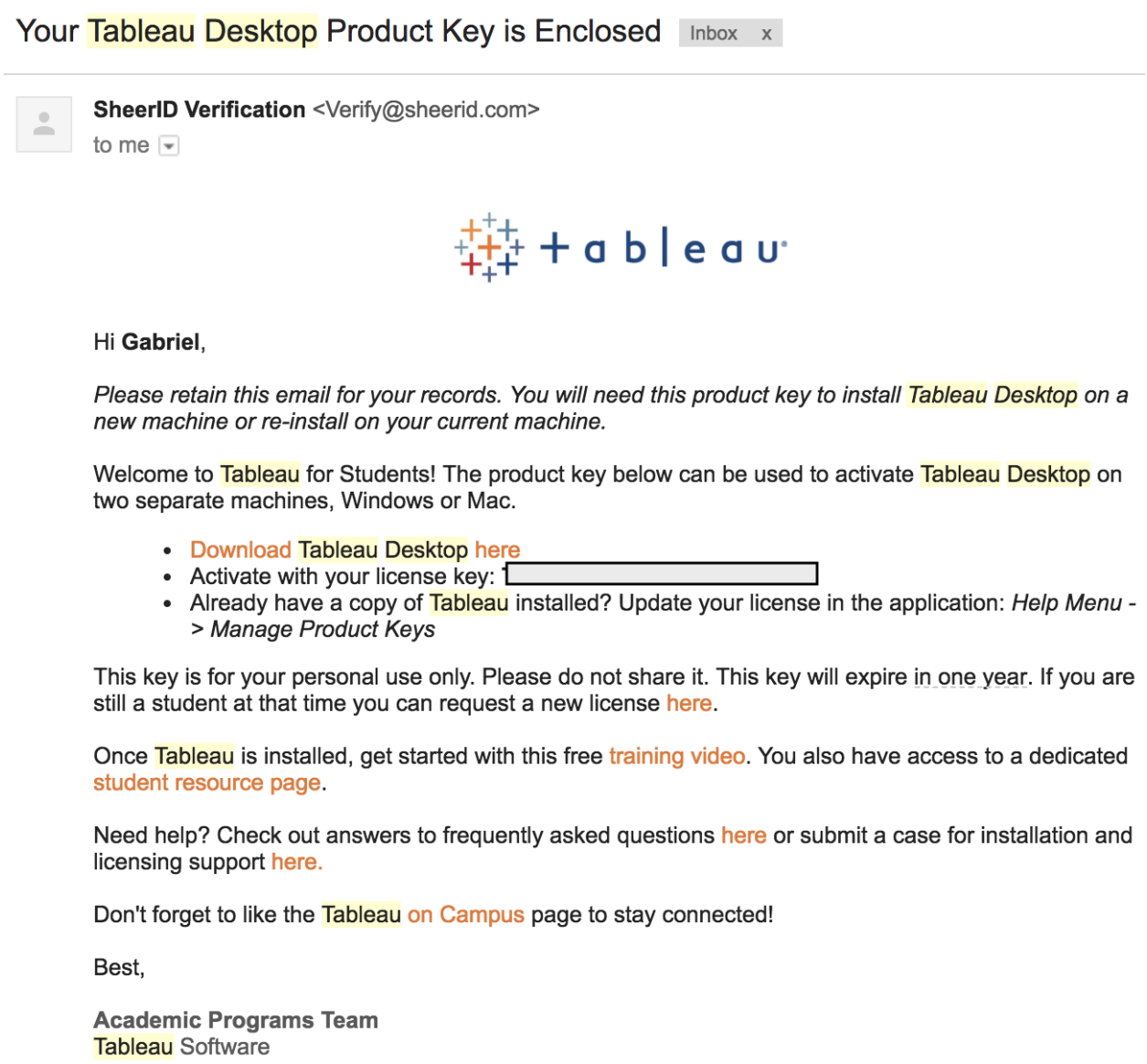


Figure 15: Basic info for Tableau account creation.

After you have filled out the basic information you will be sent a product key in an email (see Figure 16).


Your Tableau Desktop Product Key is Enclosed Inbox x

SheerID Verification <Verify@sheerid.com>
to me

Hi **Gabriel**,

Please retain this email for your records. You will need this product key to install Tableau Desktop on a new machine or re-install on your current machine.

Welcome to Tableau for Students! The product key below can be used to activate Tableau Desktop on two separate machines, Windows or Mac.

- Download Tableau Desktop here
- Activate with your license key: 1
- Already have a copy of Tableau installed? Update your license in the application: *Help Menu -> Manage Product Keys*

This key is for your personal use only. Please do not share it. This key will expire in one year. If you are still a student at that time you can request a new license here.

Once Tableau is installed, get started with this free training video. You also have access to a dedicated student resource page.

Need help? Check out answers to frequently asked questions here or submit a case for installation and licensing support here.

Don't forget to like the Tableau on Campus page to stay connected!

Best,

Academic Programs Team
Tableau Software

Figure 16: Tableau confirmation email.

You can then use the download link in the email to download Tableau. After installing Tableau, you can open it and the Tableau registration window appears from which you should just click the Activate Tableau link because you already have a product key (see Figure 17).

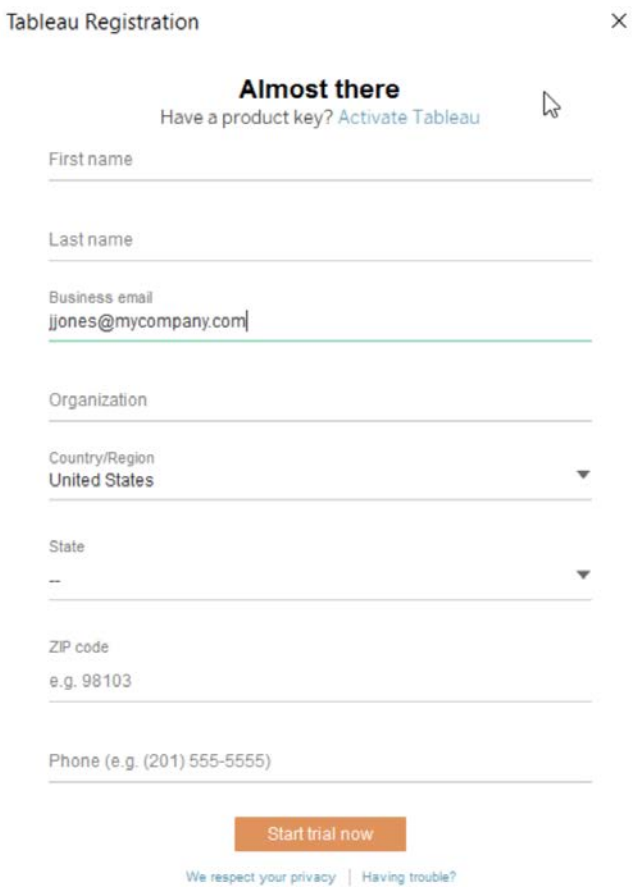


Figure 17: Tableau registration window

Then you can enter the product key from the email in the activation window (see Figure 18).

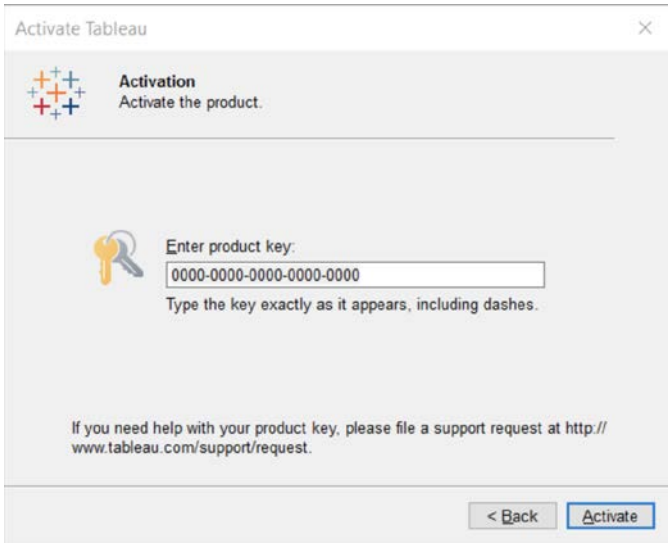


Figure 18: Key Activation

To make visuals of tweets we imported the CSV files we had into Tableau using the connect dialogue box (see Figure 19).

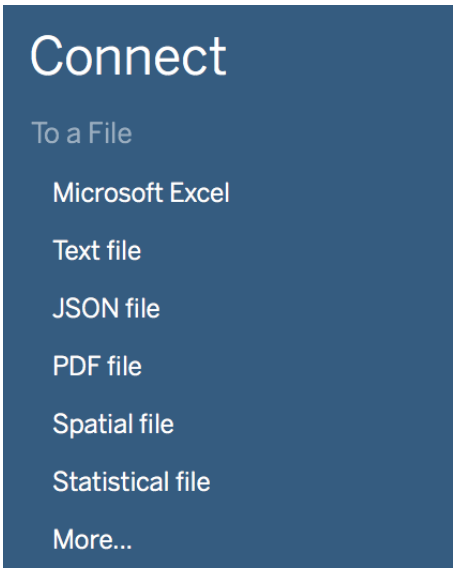


Figure 19: Tableau connect dialogue box

You can connect different types of data to make different visuals. We will detail the following kinds of visuals you can create from the different kinds of data below.

TWEETS OVER TIME

If you would like to graph the number of tweets over time for an event, you will need to use the tweet data called Clean_Tweet_Data in the Supplemental Materials found in Datasets. Pick the school you are interested in and upload the data to Tableau and it will look like Figure 20. First format the column containing the time of the tweet by clicking the small ABC icon and choosing the "Date and Time" option so the Tableau will not have any problems recognizing the current Date data form (see Figure 20).

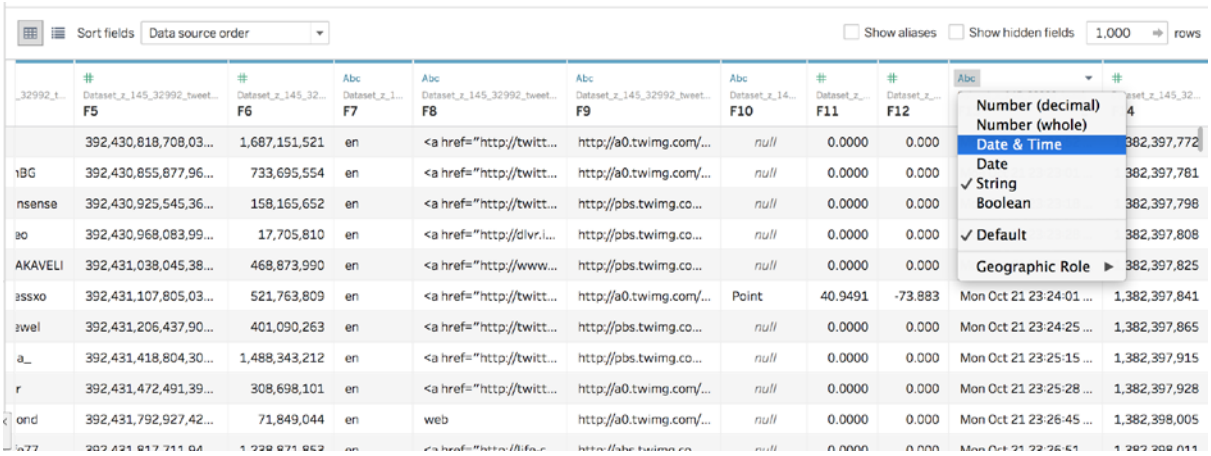


Figure 20: Format the time of tweet.

Navigate to the active sheet. To create the tweets over time visual (see Figure 3), drag the Date data we formatted earlier and the tweet data that contains the text of the tweets to the row and column slots (see Figure 21).

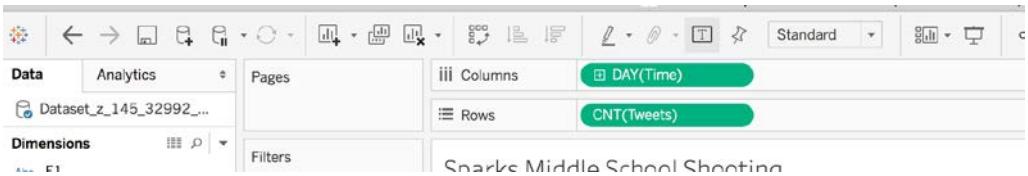


Figure 21: Drag Tweet and Date data to rows and columns.

To count the number of tweets, you click the small arrow on the tweets data and choose measure count. This will count the number of responses. Figure 22 shows changing the rows to count.

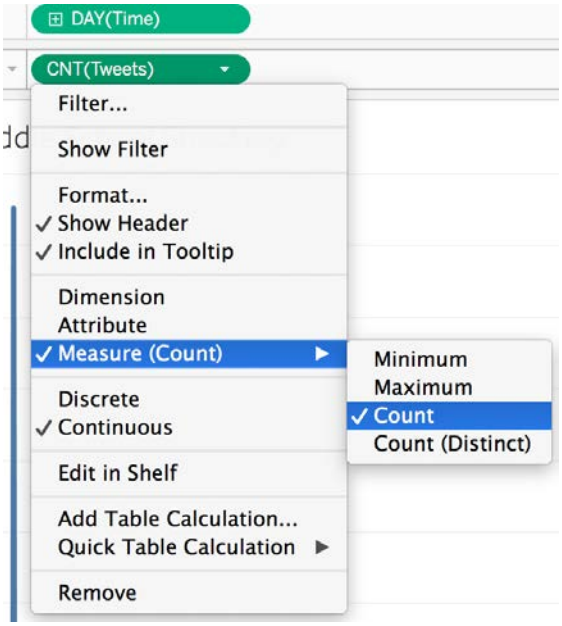


Figure 22: Change type of the tweets to count.

Then Tableau usually automatically applies a line graph for the visual. If it does not, simply just choose a line graph or whatever graph you desire. In Figure 21, notice that there is an option to filter time. You can apply filters if you wish to show a shorter time span (see Figure 23).

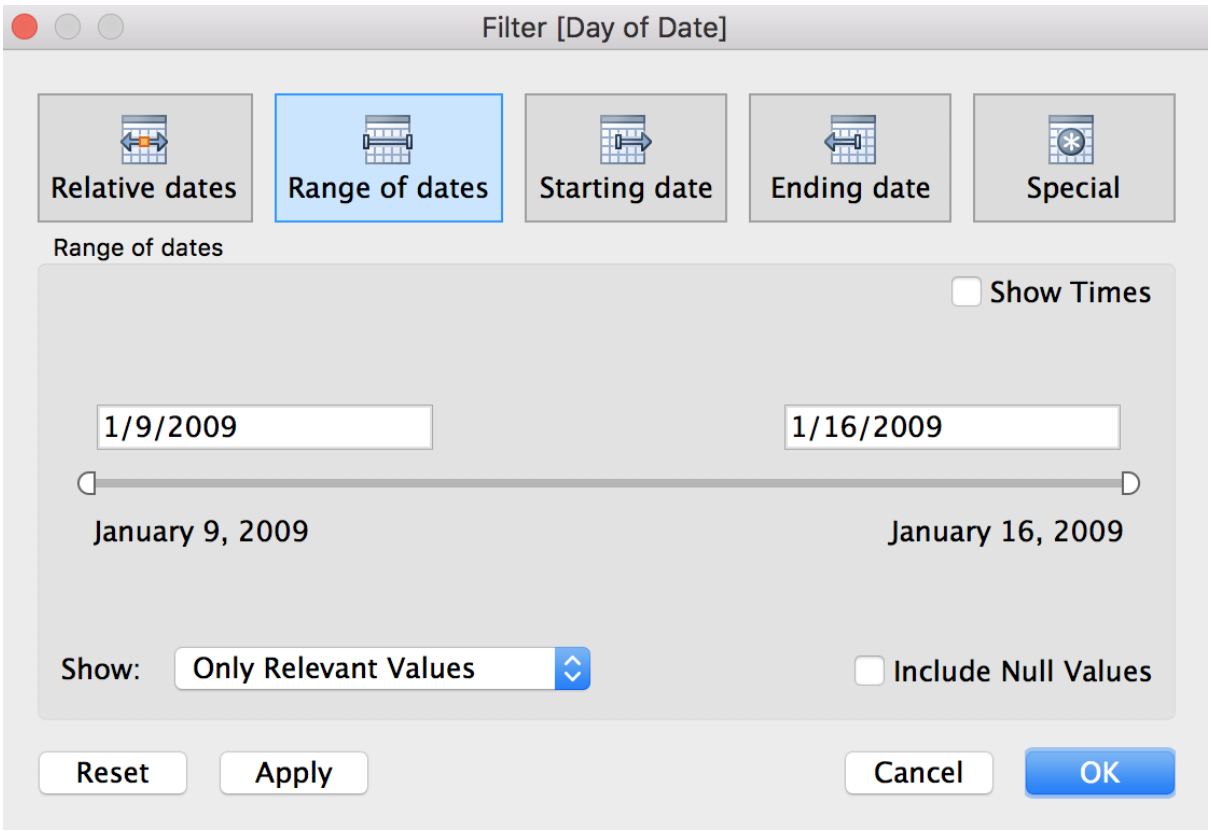


Figure 23: Dialog box for filtering time.

POINT MAPS

The second type of visual can be plotted by doing the same steps as above except you will need data that contains GPS information. This data can be found in the Supplemental Materials in the Datasets file under the Hydration_Data file. Once you are in the Hydration_Data file, select the school you are interested in and then select the file with the form school_verified_user_locations and upload it into Tableau. Similar to Tweets over Time, you must format the GPS data by assigning the latitude and longitude to their respective columns so that Tableau can recognize it as a GPS format (see Figure 24).

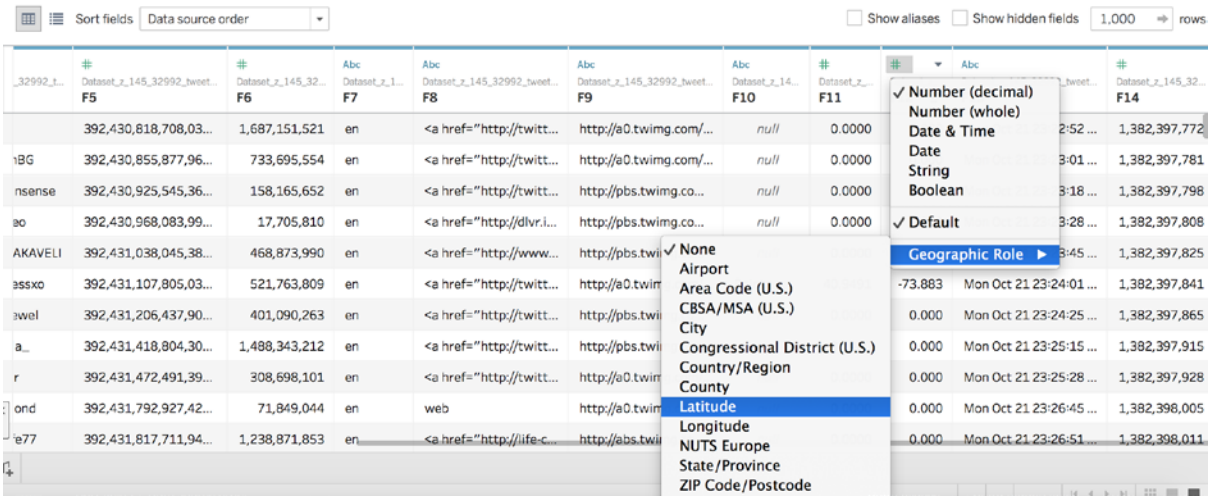


Figure 24: Format the geolocation of tweet.

You can do the same with the latitude and longitude data by dragging the data sets to the rows and columns and changing the type of the row and column to dimension (see Figure 25).

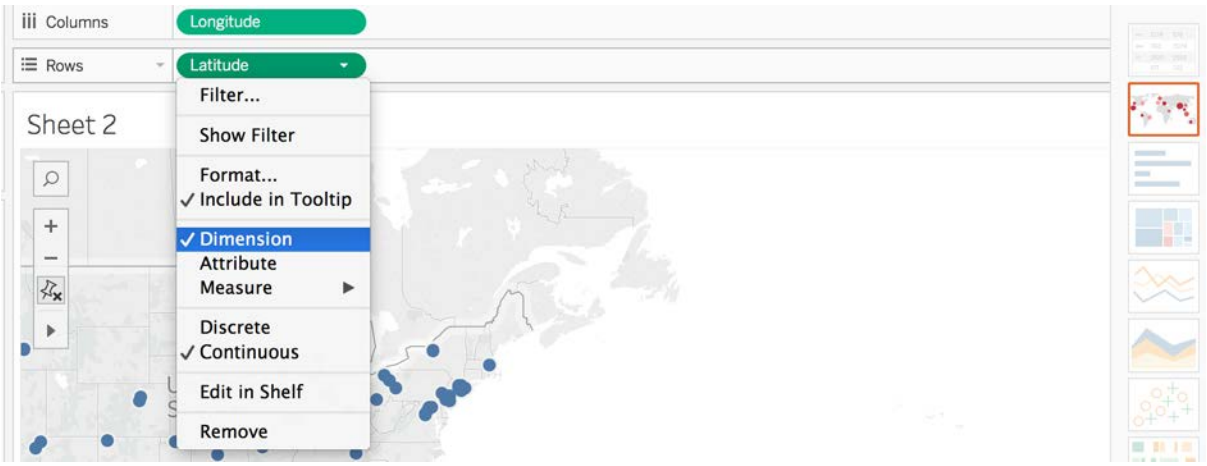


Figure 25: Change the type of the latitude and longitude to dimension.

HEAT MAPS

The third type of visual can be plotted by doing the same steps as the point map except the data must know contain state location information. This data can be found in the Supplemental Materials in the Datasets file under the Hydration_Data file. Once you are in the Hydration_Data file, select the school you are interested in and then select the file with the form school_user_states and upload it into Tableau. Similar to Tweets over Time, you must format the state data by assigning the State/Province data type so that Tableau can recognize it as a state format (see Figure 26).

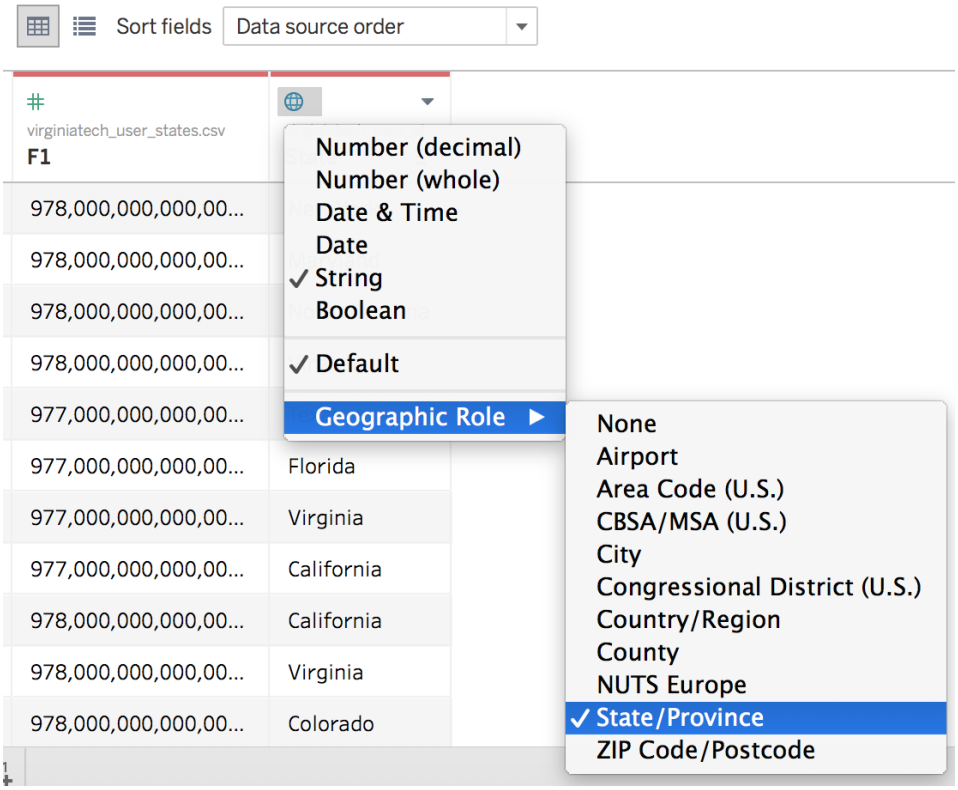


Figure 26: Change the data type for the state information to State/Province

This is one of the most tricky visuals to create but basically you do four things:

1. Drag the state data to the column slot (see Figure 27).

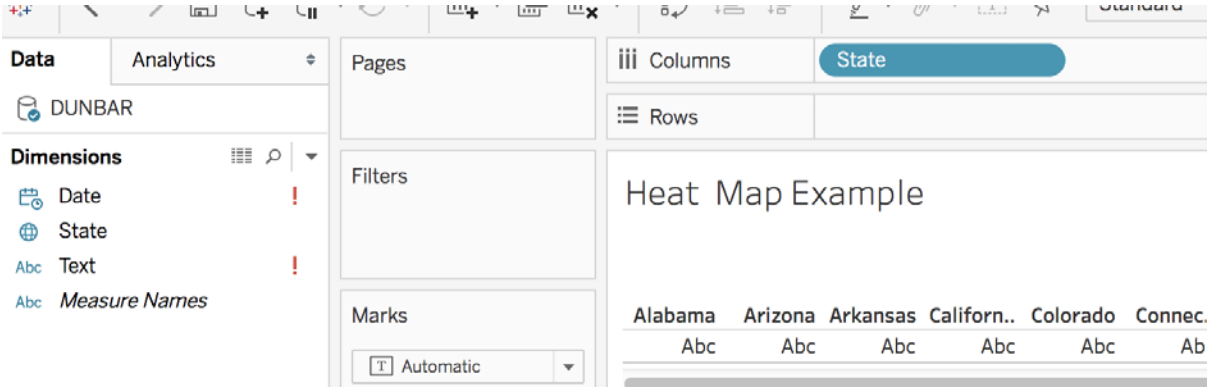


Figure 27: Drag state data to column slot.

2. To the right there will be a show me tab. Click the tile that looks like a map (see Figure 28).

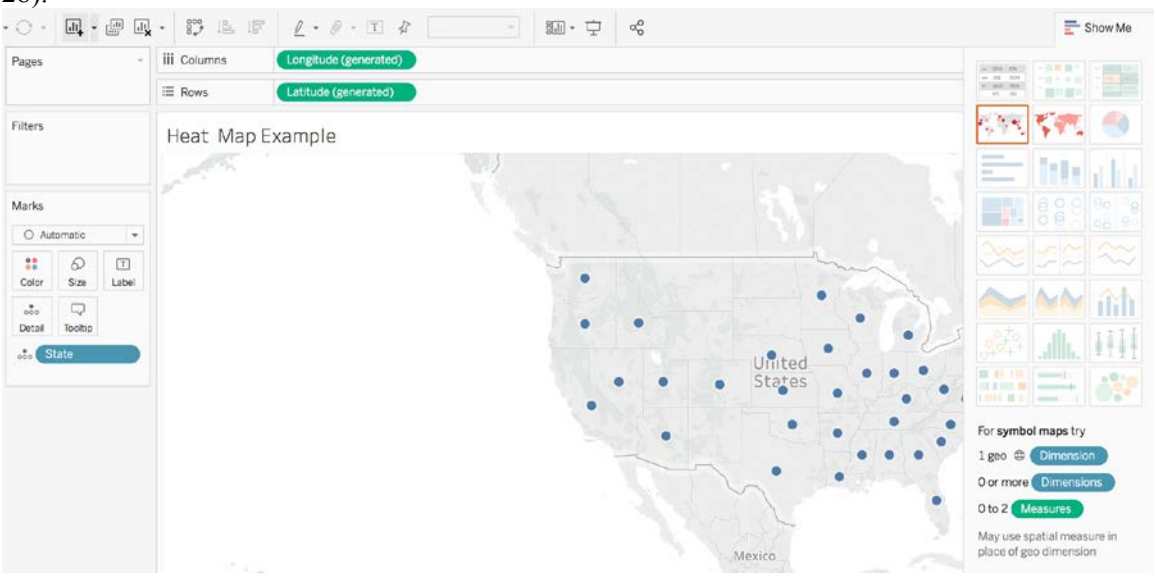


Figure 28: Choose the Map tile in the Show Me tab.

3. Then drag another State data to the color tile in the Marks dialogue box and choose count to get the shading (see Figure 29).

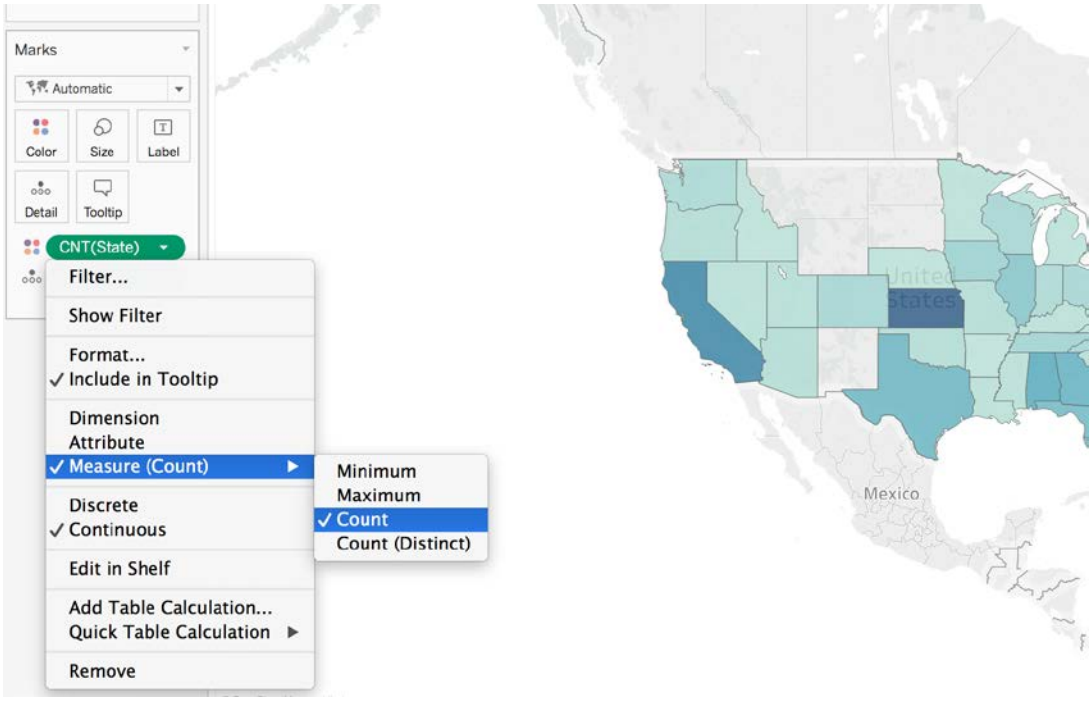


Figure 29: Drag State data to the color in Marks and select measure by count.

4. Then if you wish to add number of tweets in each state on the states, you drag one more State data to the label tile in the Marks box and choose count as in step 3 (see Figure 30).

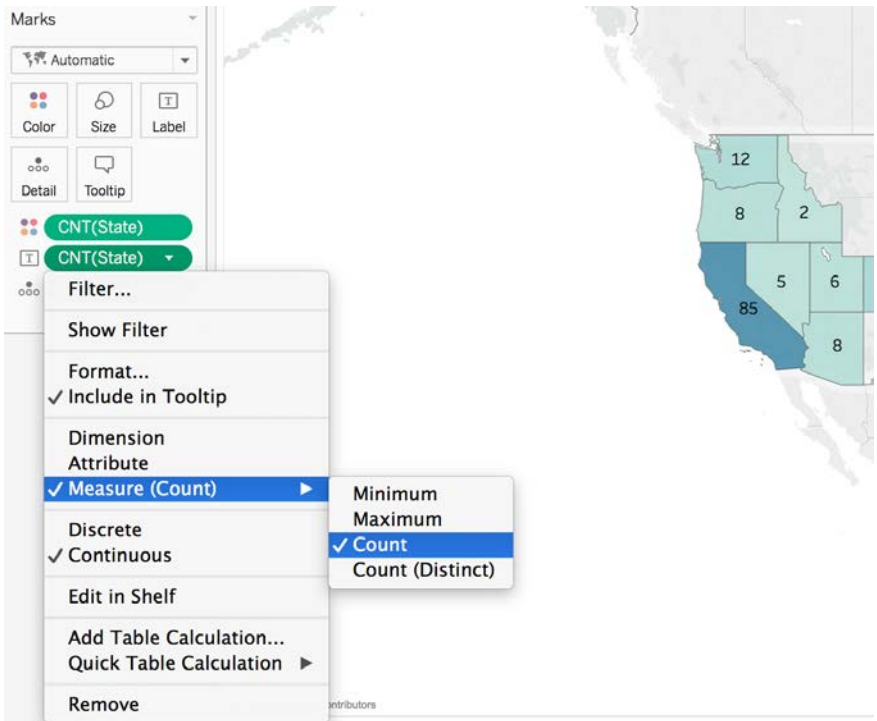


Figure 30: Drag State data to the label in Marks and select measure by count.

SENTIMENT OVER TIME

The fourth type of visual can be plotted by doing the same steps as above except you will now need to use data that contains both time and sentiment information. This data can be found in the Supplemental Materials in the Datasets file under the Sentiment_And_Time_Data_Tweets file. Just pick the school you are interested in and upload it to Tableau. Similar to the other data types, you must format the type so that Tableau can recognize it as a date and decimal format (see Figure 31).

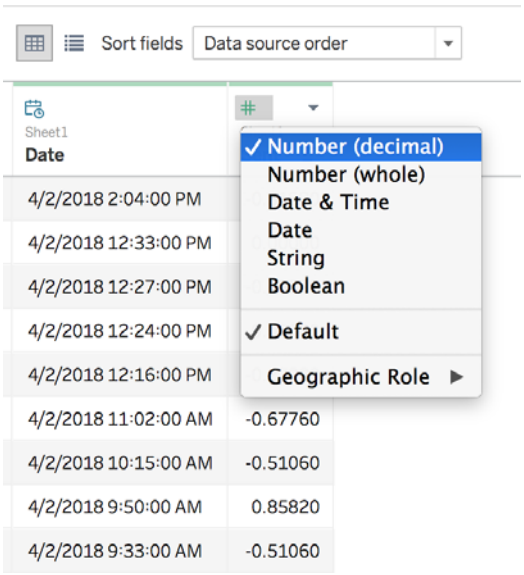


Figure 31: Change format to date and decimal.

Then, in a very similar way to Tweets over Time, drag the date to columns and sentiment data to rows. Make sure to view sentiment as a dimension (see Figure 32).

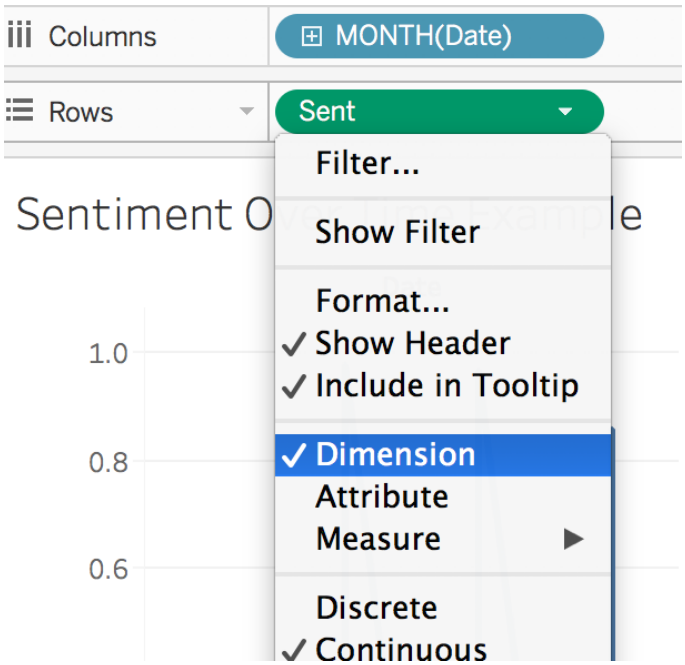


Figure 32: Change sentiment to a dimension

You can also filter the time however you like as shown in Tweets over Time.

PIE CHARTS

To make the pie charts Microsoft Excel was used. It is a fairly simple process. Using the formulas given above we calculated how often certain keywords were used (politics, sad, guns, gun reform, and again). These five numbers were placed into another Excel spreadsheet. With the values selected you can create pie charts by clicking the pie chart icon within the *Insert* tab as shown below.

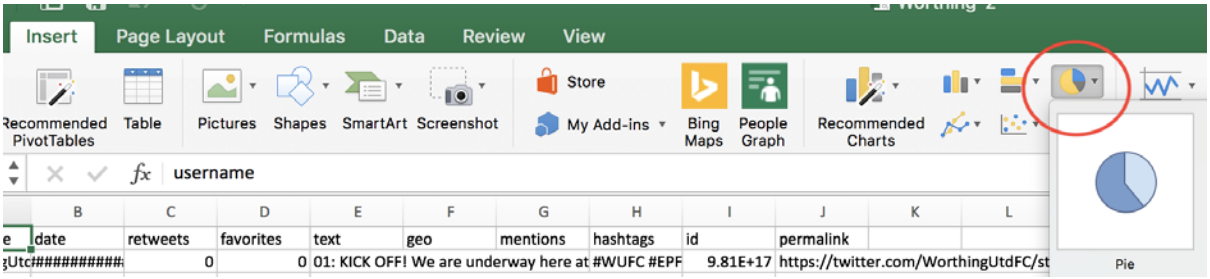


Figure 33: Create Pie Chart

WORD CLOUDS

Creating word clouds was a simple process. We used an online tool[6] that we able to import our data into to create word clouds. With Excel you can select all and copy the column containing the data needed, and then paste it into the online tool (see Figure 33).

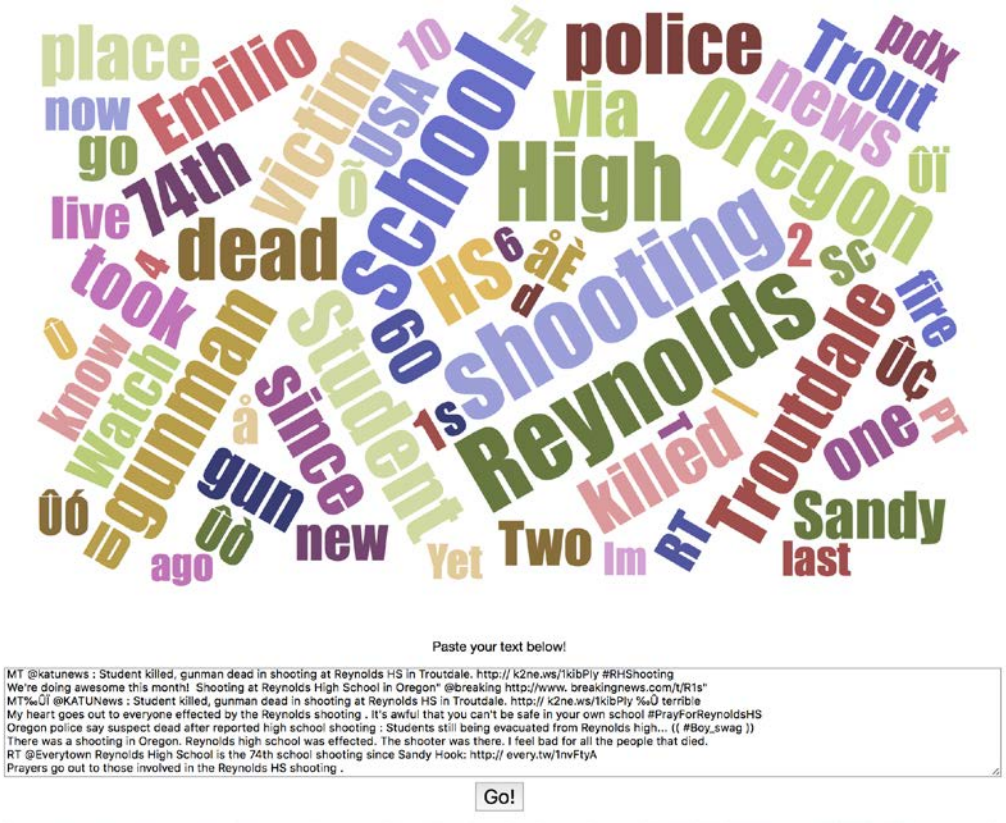


Figure 34: Create Word Cloud.

Once the data is in the text field you can click the go button and the word cloud will be generated. This process was used for tweets and webpages. The dataset for webpages that we

GETAR gave us included the URL and the text from the article. For the URL word clouds we selected the text from the article and placed that into the online tool[6].

LESSONS LEARNED

TIMELINE

Table 4 describes the key milestones accomplished over the course of the project.

Week	Project Milestone
January 29	Project assigned
February 5	First meeting with client
February 19	Establish visual types and themes
March 12	Identify possible data collection sources, including get-old-tweets, hydrator and NLTK Sentiment Analysis
April 9	Create visuals based on newly gathered data
April 23	Provide suite of visuals to client for approval or edition
May 1	Provide finished suite of visuals to client

Table 4: Schedule of project milestones

PROBLEMS

Several problems we encountered during this project have been outlined in our Evaluation/Assessment section. Some of the lessons we have learned from these problems are to contact any liaisons as early as possible and to always verify your data, especially if it comes from an outside source. Our team feels that if we had noticed that the datasets were skewed and unclean earlier in the project, then we would have better adjusted to incorporate our new tasks into our timeline.

Another major problem we encountered centered around the Dunbar High School shooting in 2009. As it turns out, we were unable to collect enough tweet data for this shooting for our visuals to be meaningful. After research, we found only 14 relevant tweets from our original GETAR datasets and we were not able to produce significantly more than that through our own data collection.

The problems we have outlined present a unique issue for our group, as none of us have ever worked on a data analytics team. We were initially unsure about how to approach the problem of data collection and analysis.

SOLUTION

Our options for solving some of the problems we had encountered seemed to range between spending a significant amount of time cleaning the data we were given or d data and starting over with a fresh data-collection task.

The first option we considered was cleaning our current datasets. Several members discarding of our team used two different tools to try to accomplish this task: Tableau and

Microsoft Excel. Tableau is useful because it is not only a data visualization tool, but it also allows users to manipulate their data within the desktop application. However, we found that Tableau was limited when dealing with missing or incorrectly formatted data, which most of our issues stemmed from. Microsoft Excel was slightly more flexible for what we needed but we found that a significant portion of the data editing, like merging cells where data was missing, involved too much manual work from the user. Since some of our datasets contain more than 7 million data points, we concluded that these two tools were not suitable for cleaning our datasets.

After deciding that we needed completely new datasets, we approached our client and our GETAR liaisons for the project and voiced our concerns. Some concerns involved the fact that Twitter's published API requires payment for any queries that go back more than 5 years. This was important because we needed data from 2007. After expressing the need to collect new data and our lack of tools to do so, one of our liaisons pointed out a very useful tool for collecting tweets and their associated information for free. We noticed that these new datasets were still partially incomplete. To combat this, we utilized another free tool called Hydrator. This tool allowed us to collect more complete data about a tweet based solely on the tweet ID[1]. With these tools, we were able to create accurate and complete datasets.

With regards to the Dunbar HS shooting, our team has decided that not enough tweet data exists for this particular shooting. We believe that creating visuals for it would not accurately portray the general public's emotional and statistical response from the shooting. Therefore, there are no visuals pertaining to the Dunbar shooting dealing with tweets in our Supplemental Material included with this report. There are several pertaining to URLs from websites referring to the shooting.

FUTURE WORK

The creation of a host of visuals was our main goal for this project and since we've been able to achieve that, there are no set goals for future work. It is possible that our client may wish to expand the set of visuals to include other kinds of visuals that portray other kinds of information. If that is the case, the data provided in the Supplemental Materials section of this report, as well as the Developer's Manual, will aid in the creation of more visuals.

In the future, in place of the Dunbar HS shooting, our client may choose to analyze the response from a different school shooting. If that is the case, data on the new subject could be collected and analyzed as we have done with the other nine cases.

ACKNOWLEDGEMENTS

Dr. Donald Shoemaker, client
Jason Callahan, project liaison
Ziqian Song, GETAR liaison
Liquing Li, GETAR liaison
The Data Visualization Studio in Newman Library, Virginia Tech
Dr. Fox, CS4624 Capstone Professor
NSF for funding GETAR through grant IIS-1619028

REFERENCES

1. Summers, Ed., Hydrator, (2017) GitHub repository, https://github.com/DocNow/hydrator, accessed March 15, 2018
2. Steven Bird, Ewan Klein, and Edward Loper (2009). Natural Language Processing with Python. O'Reilly Media Inc. http://nltk.org/book, accessed March 17, 2018
3. Jefferson-Henrique, (2018), GitHub repository, https://github.com/Jefferson-Henrique/GetOldTweets-python, accessed March 20, 2018
4. GETAR, (2016), eventsarchive.org, Virginia Tech, accessed April 2018
5. Tableau, www.tableau.com, Tableau Software, INC., accessed March 2018
6. Jason Davies, https://www.jasondavies.com/wordcloud, accessed March 18, 2018