CSCI E-64 Visualization
Project II Process Book

Bradley Taylor
Theodore Natoli

Spring 2013

Contents

Project Summary

Project Development Blog

A chronological account of the design decisions and implementation notes that went in to creating this project. Includes the initial and final project proposals. This is the detailed record of our process.

Project Summary

Overview and motivation

The subject of our project is the use of obscene and violent language on twitter. Our primary research aim was to discover whether any regional or temporal trends in such language could be identified.

In the wake of incidents such as the shootings in Aurora, CO and Newtowne, CT, we are concerned with the level of violence in our society. We are interested in the ways in which language interacts with people's broader experiences. Social media services such as twitter allow users to participate in a large online culture. The massive volume of data available from these services is a new and potentially valuable resource for those interested in conducting social research.

Our original goal was to attempt to connect the use of violent or obscene language to other social or environmental factors, such as crime rate or weather. Due to technical and time constraints we were unable to gather the orthogonal data sources necessary to explore these questions. However, we were able to collect around 30,000 tweets, each containing a date stamp and geolocation. This allowed us to look for patterns in the spatial and temporal distribution in the use of violent language by a significant number of individuals.

Description of the data: Source, scraping method, cleanup, etc.

We used the Pattern.web python package to access Twitter data. Pattern provides useful functions to query Twitter's API and parse the responses. We queried the API for a pre-specified set of vulgar, derogatory, and violent terms. A more thoroughly-conducted research study might select these terms based on some social-scientific criteria informed by prior research. In the interest of time, we contented ourselves with manually creating our own list.

We were interested in the spatial distribution of tweets. Twitter allows users to enable geo-location of their tweets as an opt-in feature, used by many third party applications. Location information for geocoded tweets is returned as a set of latitude and longitude coordinates in the API response. Pattern does not parse these fields as part of its standard response handling. To that end, we downloaded a copy of the Pattern source code and modified the Twitter class to extract geo information from tweets. We created a script to collect tweets by calling our modified Pattern Twitter class, up to the maximum daily query limit. By running this script for several days, we were able to collect approximately 30,000 tweets.

These data were used directly in our visualization (see the raw tweets map description below). We also used several other scripts to aggregate tweet data according to a factor of interest. An R script was used to generate plots of tweets against time as a form of exploratory data analysis. This script was also used to aggregate all tweets for a given date, for export to a

JSON file. This file was loaded by our visualization as the source of our time series data. This same script also aggregated tweets by search term, for use in our per-day bar charts.

A python script was used to aggregate tweets geographically according to US state. As Twitter only provides latitude/longitude coordinates, a reverse-geocode operation was necessary to get the state. This was accomplished via a call to the GoogleMaps web API (see link below). The query limit on this API was lower than some documentation led us to believe. As a result, only about 2,500 tweets could be reverse geocoded and aggregated by state. Given the time constraints on this project, the reverse geocoder was unfortunately not built to run over multiple days. It requires a GoogleMaps API key to use. Reverse-geocoded data was stored in an intermediary csv file, which served as input to a second python script which performed the by-state aggregation.
https://developers.google.com/maps/documentation/geocoding/

State population data was pulled from the table contained in the following wikipedia entry: http://en.wikipedia.org/wiki/US_states_by_population

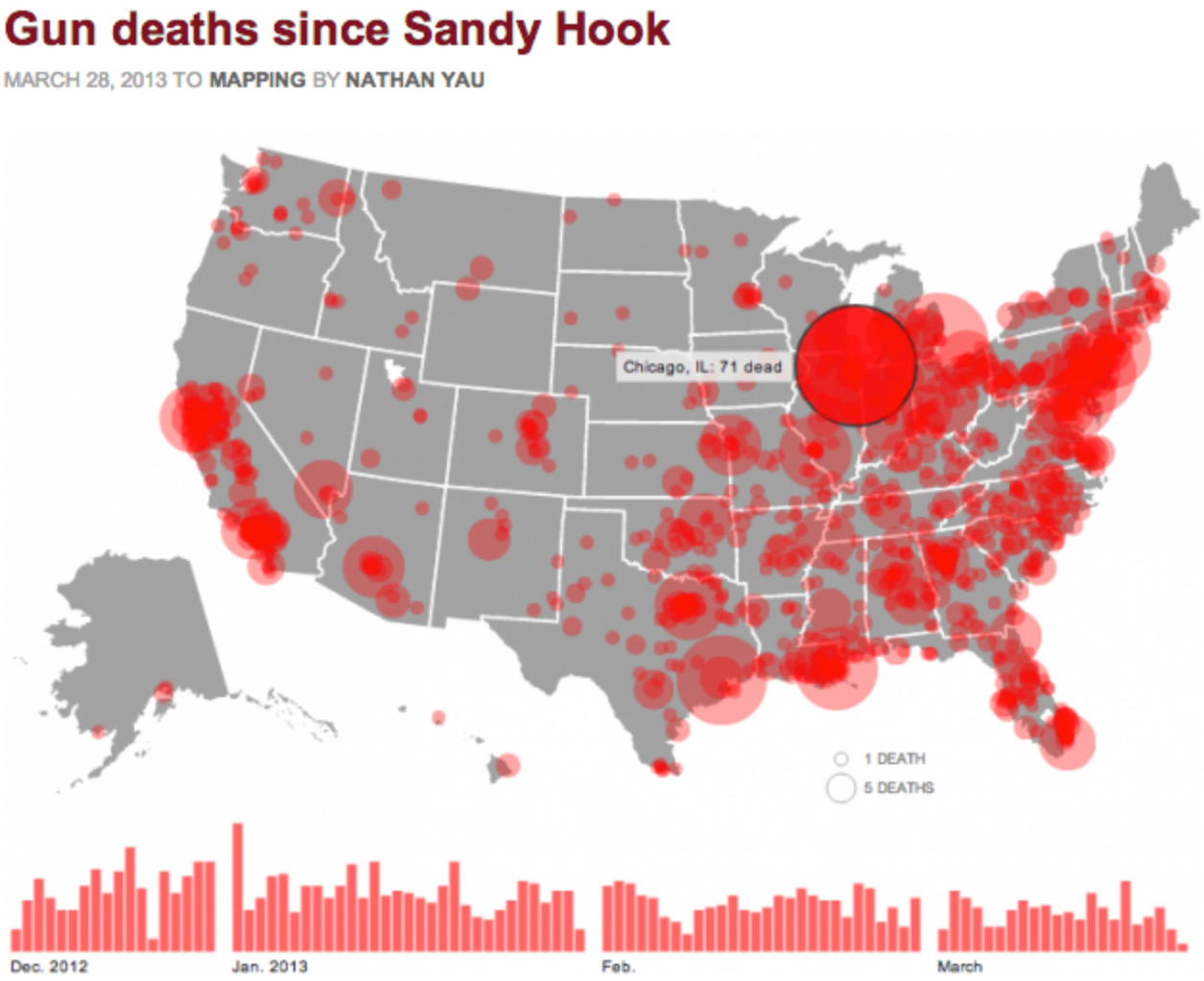
This data was manually cleaned for number formatting in MS Excel and saved as a csv. It was used by the state aggregation script to population-correct our raw by-state tweet counts, producing a dataset of obscene-tweets-per-capita for each state, stored as JSON.

Raw geocoded tweets, state-aggregated per-capita tweet counts, and date-aggregated tweet counts were the final data files fed into our visualizations.

Related work: Anything that inspired you, such as a paper, a web site, visualizations we discussed in class, etc.

The tweet maps seen here were major sources of influence for our design. They inspired us to implement a map of raw tweets encoded as dots:
http://ny.spatial.ly/
http://www.dailymail.co.uk/sciencetech/article-2222959/Twitter-map-London-shows-linguistic-diversity-truly-international-city.html

Another inspiring visualization was this map, which combines data and location information in a manner very similar to our visualization:



Design evolution: What are the different visualizations you considered? Justify the design decisions you made, and show any major changes to your ideas. How did you reach these conclusions? Provide clear and well-referenced images demonstrating the design evolution.

Please see the project blog for a full and detailed account of our design decisions, technical process, and the evolution of our project.

The principal change we introduced into our project was a reduction in scope. We decided to abandon efforts to collect additional data sources to relate against our tweet data, in favor of implementing visualizations exploring the time and geolocation information we had at hand. This was necessary for project achievability, and was undertaken based on the advice of our teaching fellow, who counseled us to restrict our ambition in her response to our project proposal.

For or geographic visualizations, the major decision was whether to aggregate tweets over some unit of geographic area, or to encode marks for each tweet. Aggregated data is less granular, but has the advantage that it can be population collected. By contrast, raw data provides excellent absolute detail, but is very clearly population-biased. We chose to display aggregated data initially, and then allow users to access a raw tweets map as a details-on-demand interaction. For our aggregated view, a chloropleth was chosen fairly early on. Our teaching fellow approved of this approach and we therefore maintained it throughout. US states were chosen as the unit of aggregation. Despite their widely varying sizes, this was the most technically feasible choice given the built-in support provided by datamaps.

For temporal information, we considered animating our map, or displaying an animated line chart. However, these were beyond the scope of what we could accomplish. Instead, we decided to use a line chart, as its use of positional encoding for quantitative values ensures high graphical integrity, and because it is an established visual paradigm for the display of temporal data.

Finally, we noted in our initial exploratory analysis that tweets were most frequently observed at midnight and noon, and were more frequent on weekends than weekdays. The ties to the working day were fairly clear. We considered the possibility that these observations reflected total tweet volume rather than periods of increased relative obscenity. To this end, midway through the project we began to collect tweets for the control terms “tweet” and “twitter”. We chose these based on an article that listed them among the most popularly tweeted words (circa 2009), and because they were some of the highest on this list that were not common sentence words such as ‘the’ and ‘a’ (which we feared would overwhelm our daily query limits and crowd out our obscenity terms).

We normalized our obscene terms against these control terms, and presented total, vulgar, and normalized data together as part of a multi-series line plot.

Visualizations: Describe the intent and functionality of your visualizations.

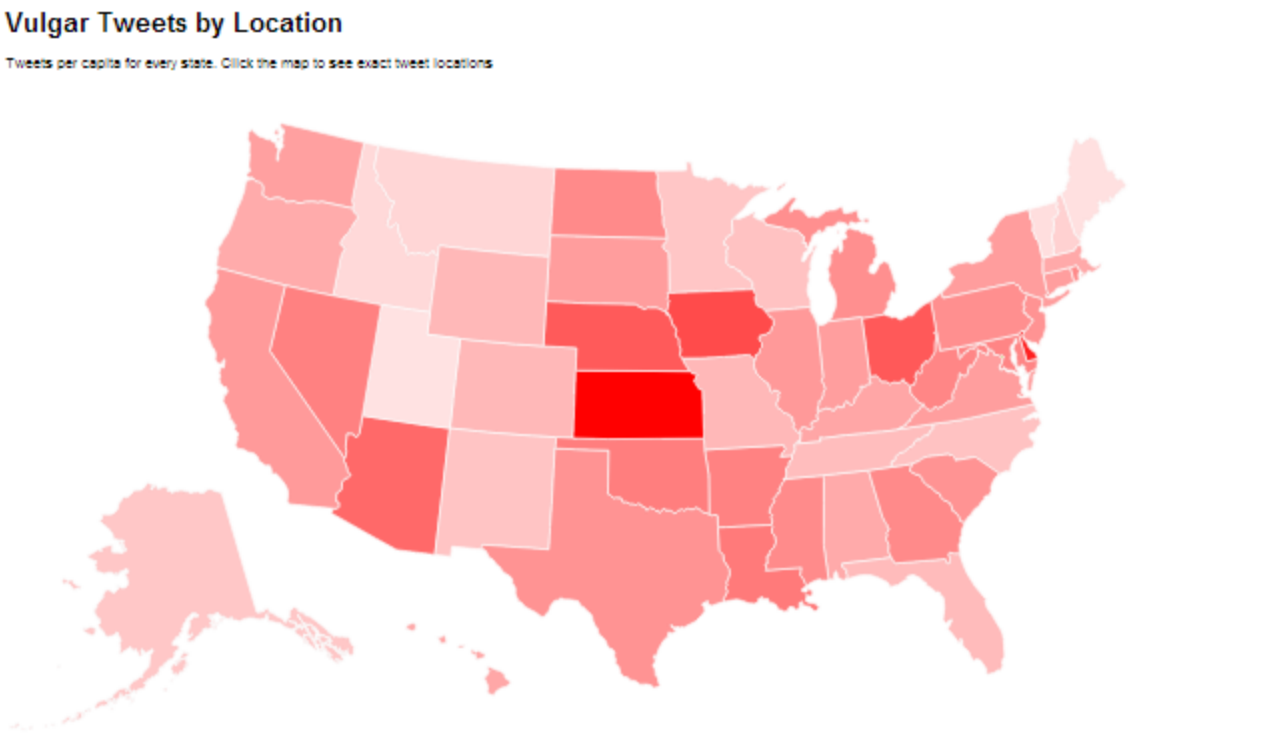
Our visualization initially presents the users with two overview-level views of spatial and temporal variation in vulgar tweets. The first is a chloropleth displaying the number of vulgar tweets collected per capita for each state. These are mapped to a monochrome color scale ranging from white (no tweets) to red (the maximum number of tweets/person observed, which corresponds to Kansas in the final view). The color scale was chosen based on course readings, which argue that people judge changes in value (color intensity) more accurately than they judge differences in hue. This accuracy is desirable, as our visualization seeks to present a continuous (rather than ordinal) scale for the purposes of accurate and nuanced quantitation. The choice of US states as the unit of aggregation was driven primarily by technical considerations, as this was most achievable using datamaps techniques previously introduced in homeworks. Further, US states are a convenient and available aggregation for population data,

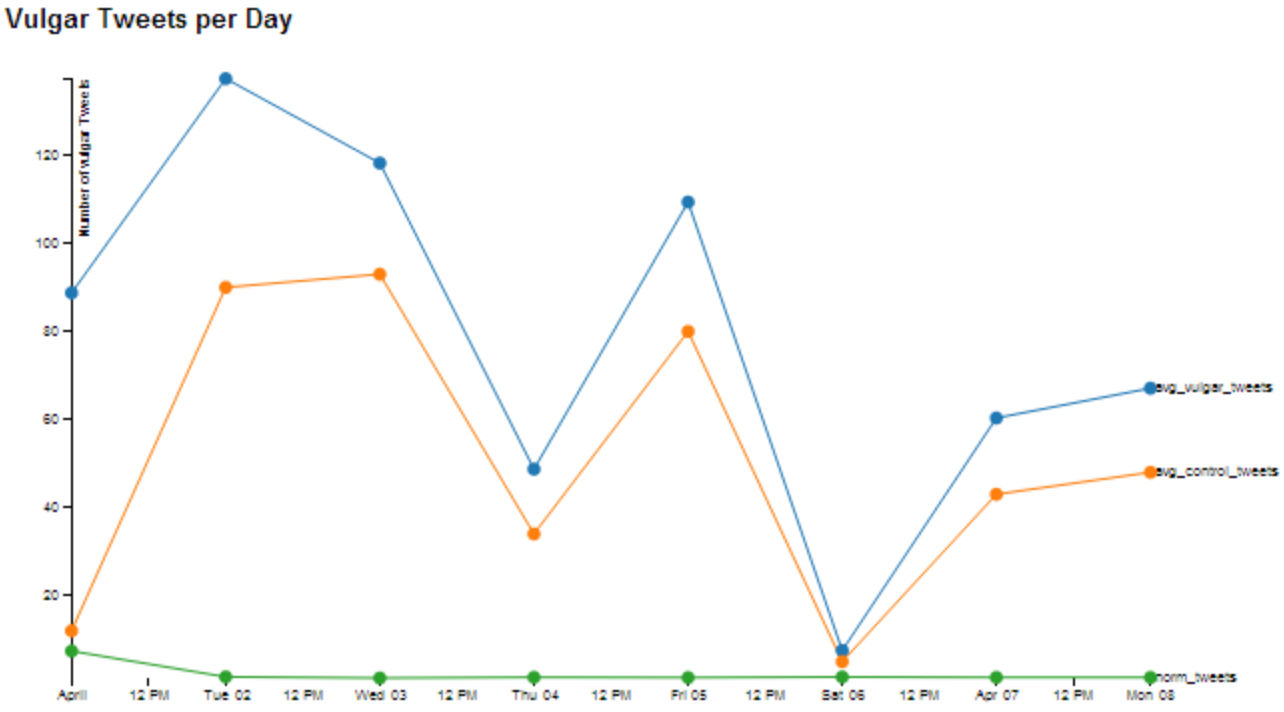
allowing us to correct our heavily population-biased raw tweet tallies. Ideally, a small unit of uniform shape and area would be chosen for such aggregations, but this simply was not possible.

The state name, tweets counted, and tweets-per-person values for a given state are presented in a tooltip when one mouses over a state.

Below the chloropleth is presented the line chart displaying the number of tweets over time. As discussed previously, we collected tweets for control search-terms in order to correct for total tweet volume in assessing the relative degree of vulgarity on twitter over time. Vulgar, control, and normalized values are presented as multiple series. Ideally, we would have been able to give the user control over which series they saw, allowing the chart to rescale (especially so as to observe detail variation in the normalized values). However, we were not able to achieve this in the time allotted.

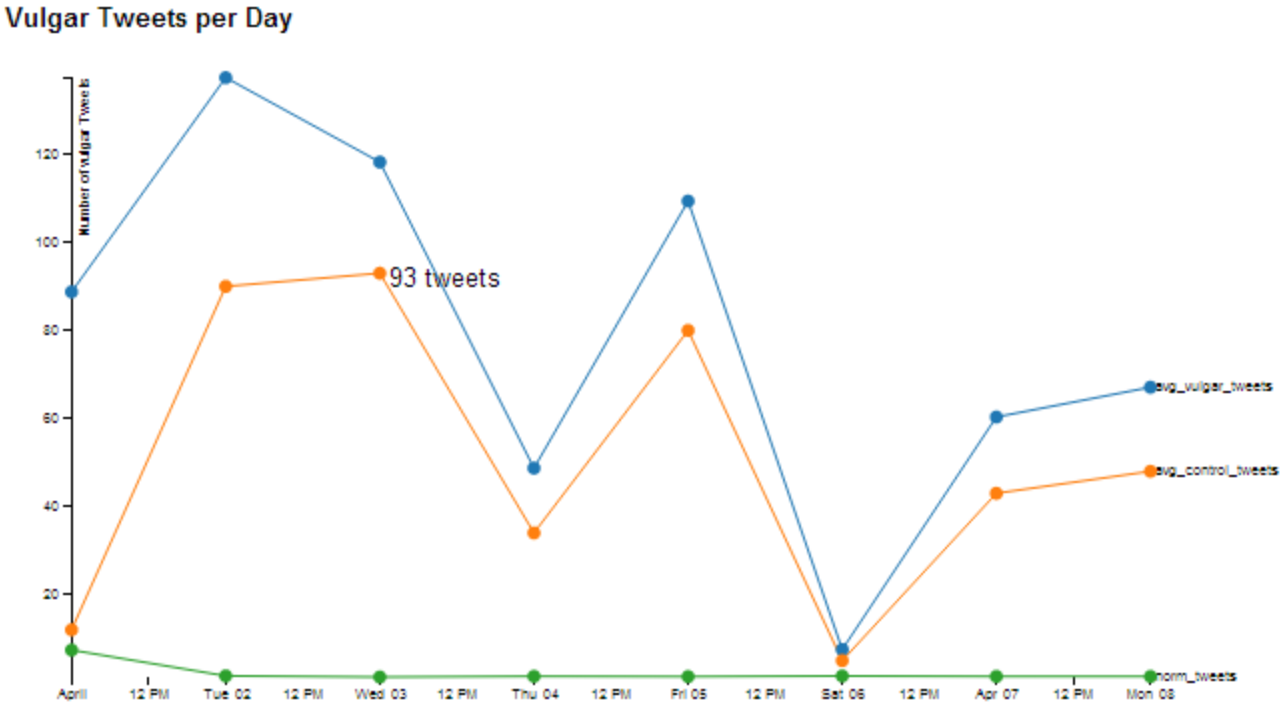
We chose to aggregate tweets over days and present these as individual dates. During the course of development, we discussed multiple levels of aggregation, including days of the week and hours of the day. In fact, several of these aggregations were produced during exploratory data analysis in R (see process blog). In the end, we decided to go with specific dates, as this conveyed day-of-week information while providing additional detail and context by allowing the user to see variation over weeks. Again, it would be ideal to present multiple options for aggregation and allow the user to select between them (such as switching between days and hours-of-the-day). However, time prevented this.

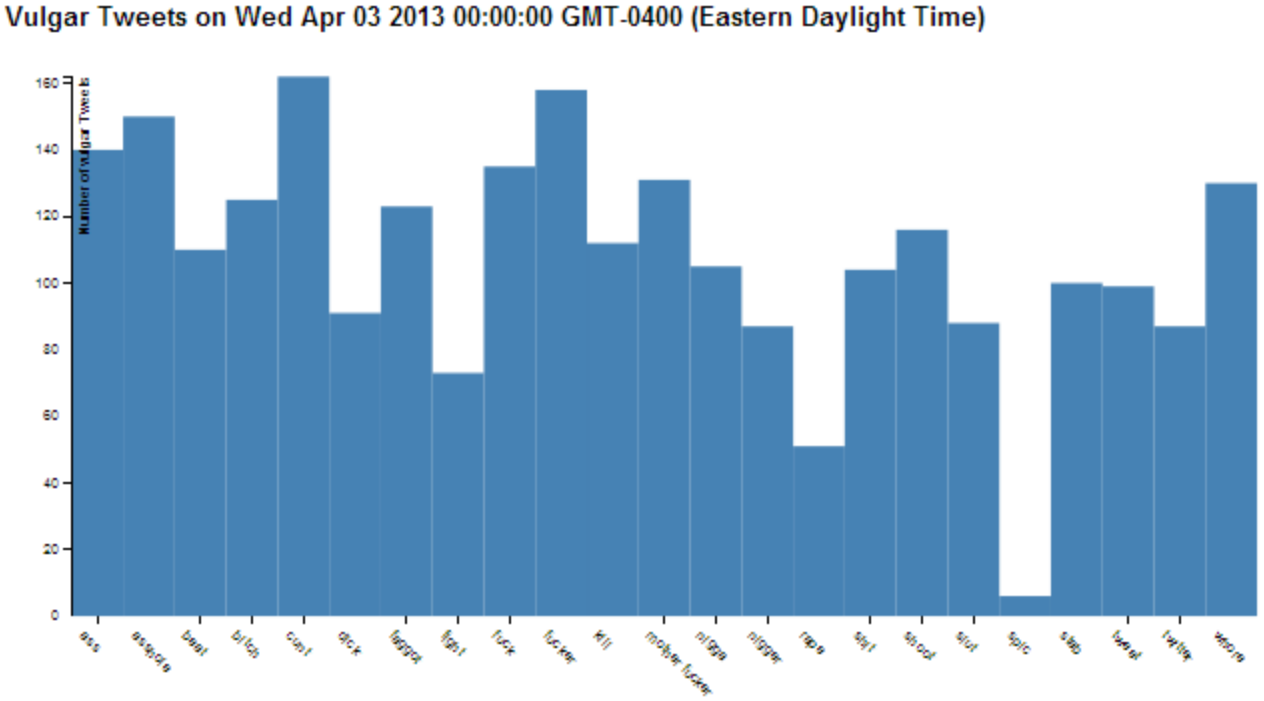




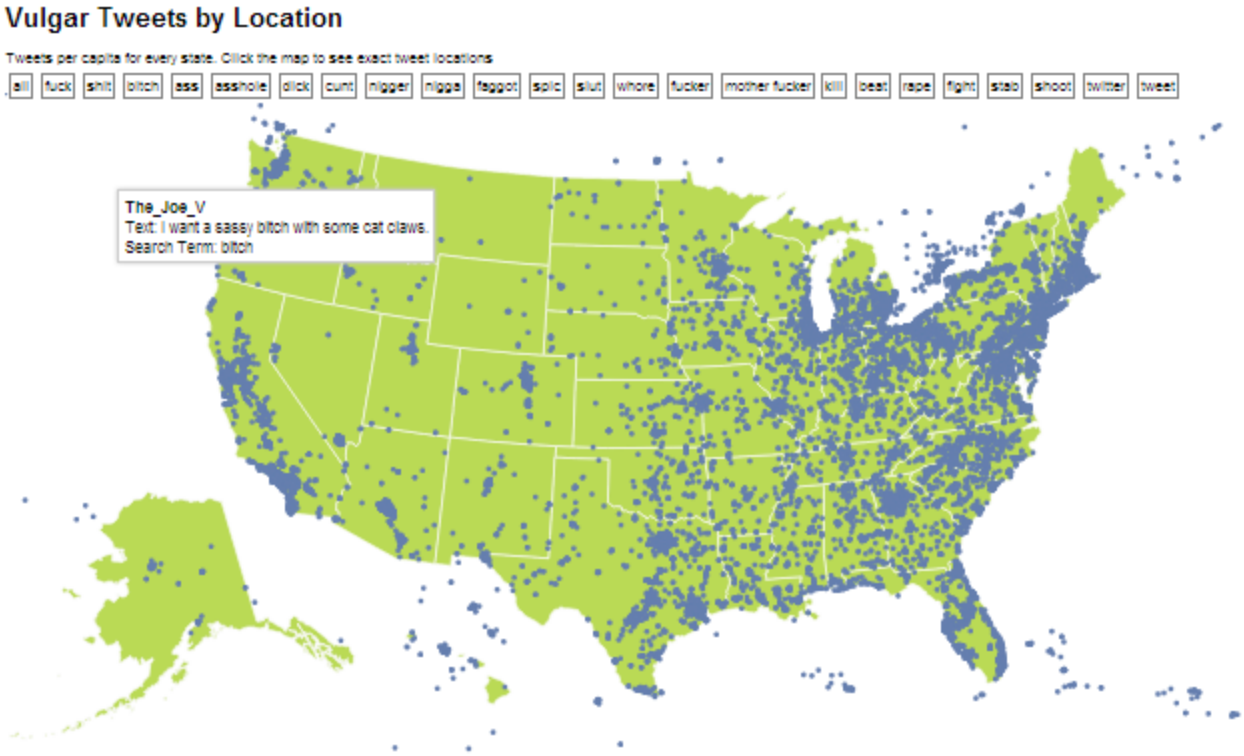
When one mouses over a particular circle in the line chart, a tooltip displays the tweets counted on that day for the highlighted series. Circles were chosen in the line chart to better highlight individual data points, and to provide a visual handle for users to interact with.

When one clicks on a day in any series, a bar chart appears providing additional details. This chart breaks down the total tweets observed on the selected day by the vulgar term they employed.





When one clicks on the data map, the view switches from a chloropleth aggregated view to a map displaying the locations of every single raw tweet. This additional detail is much richer, but is heavily population biased.



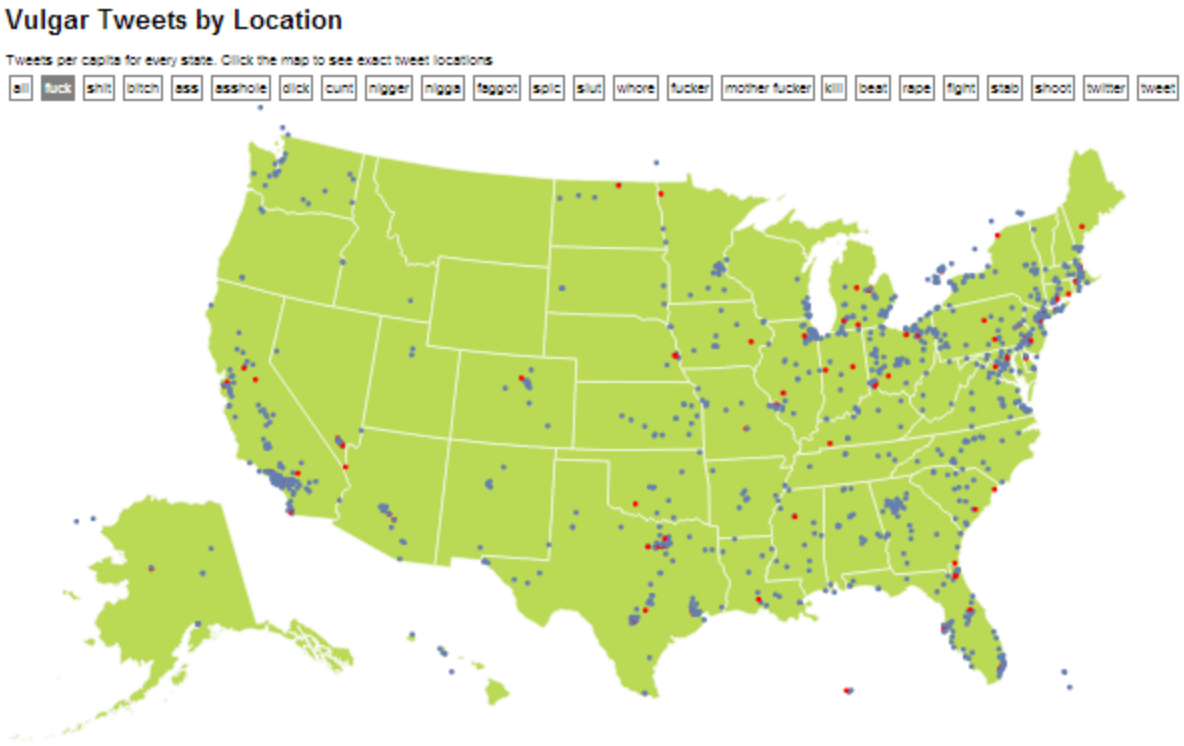
A tooltip displays the author and text of each tweet on hoverover, as well as the search term that caused it to be collected.

This view is linked to the line chart below. Hovering over a date in any series causes all of the tweets corresponding to that date in the map to be highlighted red.

Finally, note the list of buttons that now appear at the top of the screen. One is present for each search term. Clicking one of these buttons filters the tweets displayed in the map. Only those tweets corresponding to the selected search term remain. Using a combination of search-term filtering and date-highlighting, the user is thus given full control to see the exact set of tweets corresponding to a given search term on a given date, according to location, at single-tweet precision. We wished to empower the user with this level of detail without overwhelming them with it up front. We therefore designed our interactions to facilitate this level of detail when demanded by the user, after they have gotten comfortable with the visualization and once they desire to drill down.

Ideally, the filter buttons would always be present, and would filter data displayed in all of our views, including the chloropleth and raw tweet maps, the line chart, and the bar charts.

Unfortunately, it was not possible to implement this as part of the present project. Data aggregation for the chloropleth is not broken out by search term at present, and occurs via an offline script. Adjusting our other data manipulation procedures would be similarly cumbersome, and this was not achievable in the available time.

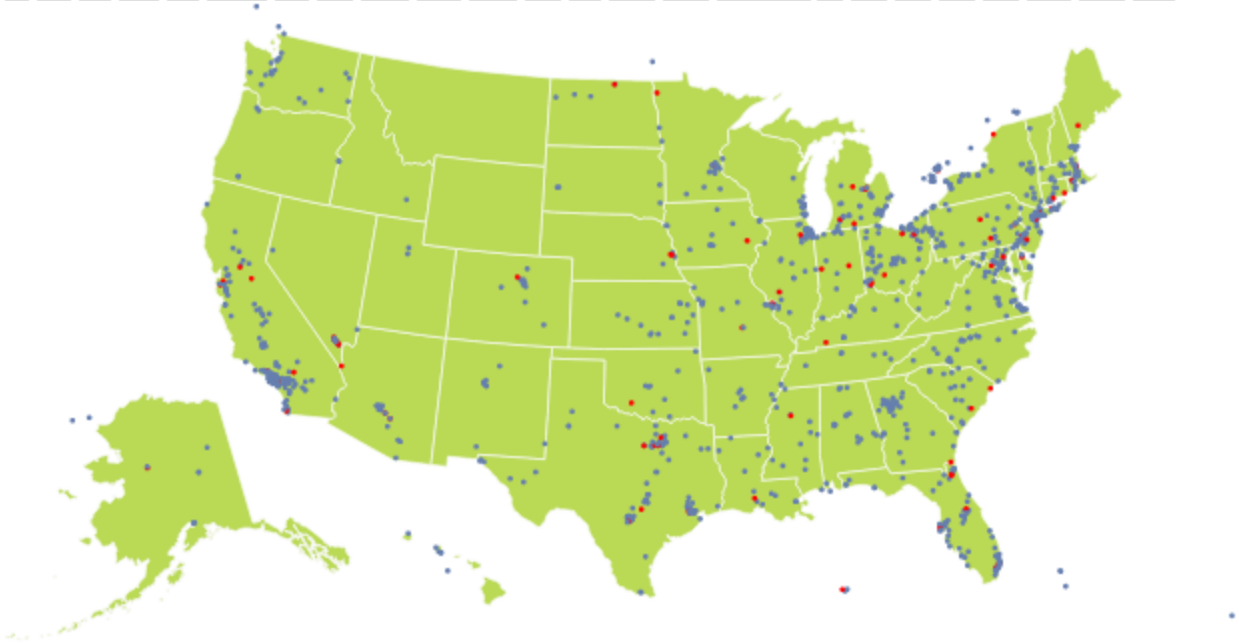


In summary, we were able to display spatial, temporal, and word-choice variation in social media vulgarity at various levels of detail, in a coordinated multi-view visualization that allows users to simultaneously interact with each of these dimensions.

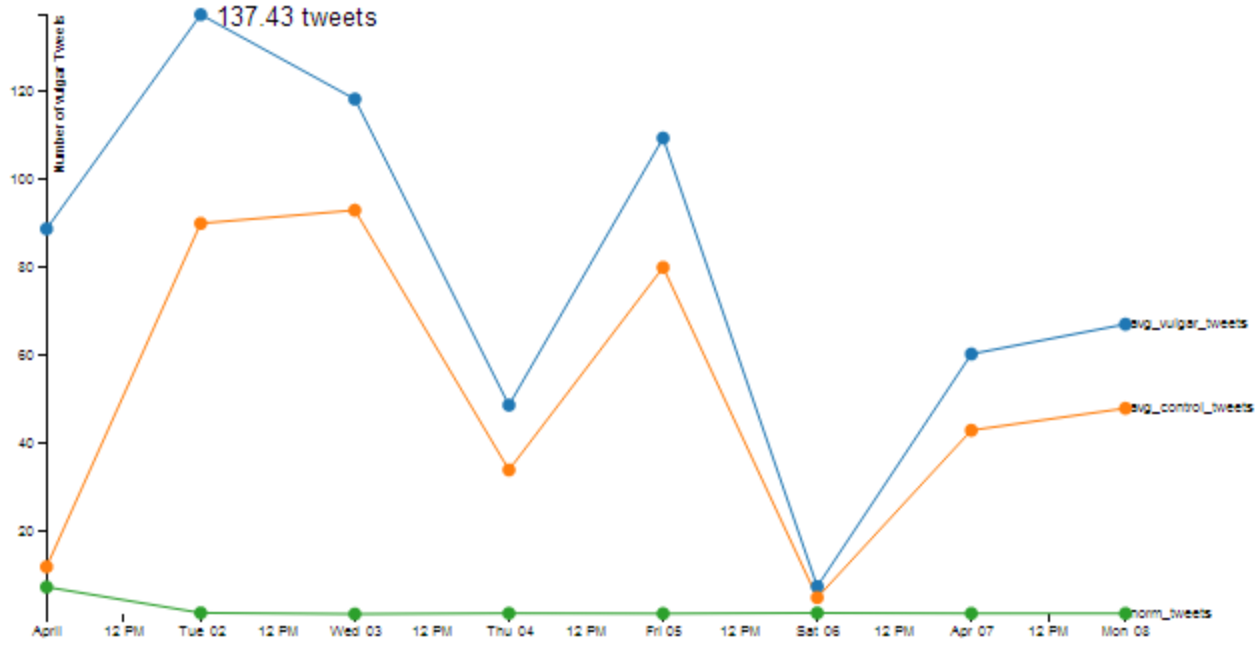
Vulgar Tweets by Location

Tweets per capita for every state. Click the map to see exact tweet locations

all | fuck | shit | bitch | ass | asshole | dick | cunt | nigger | nigga | faggot | spic | slut | whore | fucker | mother fucker | kill | beat | rape | fight | stab | shoot | twitter | tweet



Vulgar Tweets per Day



Analysis of data: What did you learn about the data by using your visualizations?

We were able to glean several insights from our visualizations. First, Twitter uses tend to be generally be more vulgar than not, at least in the time window we sampled. If we examine the line graph, we can see that the average number of vulgar tweets is always higher than the average number innocuous, or “control” tweets, and the ratio of the two is hovers around 1.3. This is consistent across weekdays and weekends, suggesting that vulgar tweet volume is just a function of overall tweet volume.

Additionally, we found that there does not seem to be a strong regional trend in tweets per capita. In a very general sense, the northern states tweet a bit less frequently than the southern states, but this is not a hard and fast rule. Somewhat surprisingly, Kansas residents, and not those of a more “urban” state, are the most frequent tweeters per capita.

If we examine the the bar charts over multiple days, we see that vulgar tweeters tend to prefer using “f-word” associated swears and derogatory female terms over other vulgarities, but not by a huge margin. Additionally, the one Hispanic racial slur we searched for generally tended to be one of the least frequently used vulgarities, and it is used less frequently than the two African American racial slurs.

Given the small sample sizes of these search terms, it is difficult to draw any concrete conclusions, but the data suggest that vulgar tweeters are more likely to target females and blacks than Hispanics. This is not simply a reflection of US population, as Hispanics make up 16.4% of the population, versus 12.6% for blacks as of 2012 (http://en.wikipedia.org/wiki/Demographics_of_the_United_States#Race_and_ethnicity). The reclamation of black racial slurs in hip hop culture and lyrics may contribute some bias in the use of these terms. An important caveat in considering vulgarity frequencies is that we are not analyzing the context in which terms are used. For example, we capture “shoot the breeze” and “I’d like to shoot you” equally. Future research might attempt a more detailed analysis of semantic linguistics, perhaps using Pattern’s natural language processing features.

Project Development Blog

3/3/13 Topic Selection Meeting
We discussed various ideas and decided on a subject. Initial subject: idea of finding a corollary for real-world violence in media or similar. Decided to look for violent speech on social media. Data accessible via pattern library. Discussed geographical and temporal research Q's.

Ted Initial Brainstorm 3/4/13

Tentative project title

Regional and temporal trends in Twitter vocabulary?

Name, email, programming comfortability of each member of your group

Can fill this in on the form:
https://docs.google.com/forms/d/1D9Ac_WF-HuPu5CB5IdqBcW9I3GmesSKzKZl8X8B9ajI/viewform

Research questions and hypotheses

Our primary question is "Are there any regional trends in Twitter vocabulary and do these trends correlate with other social patterns (ie. violence, crime) and/or with seasonal changes?" Our initial hypothesis is that the use of obscene, profane, or violent language will correlate with geographic regions of increased violence and crime and with typically unfavorable changes in weather (ie. heat wave, cold spell).

Motivation

The massive volume of Twitter and other social media users and the availability of social media data has enabled researchers to ask and answer questions in ways that were previously impossible. Recently, violent crime has come to the forefront of our collective social awareness, due to incidents such as the shootings in Aurora, CO and Newtowne, CT. Is there a way (possibly via social media cues) we might be able to forecast and prevent these incidents?

Data

We will use Python's pattern module to access Twitter (and Facebook?) data via its API. We will also use pattern's natural language processing capabilities to perform sophisticated analyses of the acquired data.

www.twitter.com
www.facebook.com

Visualization

Tier I

A heatmap with intensity encoding frequency of violent language usage?
possibly add a slider bar (or animate) to show changes over time (and/or with seasons)

Tier II

Brad Brainstorm 3/5/13

Tentative Title
"Fighting Words; Regional and temporal trends in Twitter vocabulary"
Just a thought. Punchy, if maybe a bit glib.

Research questions and hypotheses
High-level research question
"Are there any regional trends in [coarse vocabulary on social media] and do these trends correlate with other social patterns (ie. violence, crime) and/or with seasonal changes?"

I thought this was great. I will attempt to add to it by translating it down into some lower-level individual questions that might be solved by individual views:

- Does [violent,profane,others?] language vary geographically?
    - Is geographical variation in [violent,profane,others?] associated with similar variation in violent-crime rate?
    - Is geographical variation in [violent,profane,others?] associated with similar variation in climate (avg. temperature, precipitation, etc)?
    - Other social factors (see below)?
- Does [violent,profane,others?] language vary temporally?
    - What is the pattern of variation by
        - Season
        - Month
        - Day of the week (are we more profane on weekends?)
        - Time of day (Hypothesis: we can see road rage during the morning commute)

    - Is temporal variation in [violent,profane,others?] associated with similar variation in violent-crime rate?
    - Is temporal variation in [violent,profane,others?] associated with similar variation in weather?
- Does more [violent,profane,others?] language occur on Facebook or Twitter? (an additional question enabled by gathering data from both sites)

For our geographical variation in coarse language, the following are ideas for other social factors:

- type of municipality (city, town, village, etc)
- population density (does per-capita cussing increase when people are crowded together)
- avg income
- % college educated
- zoning (% residential, commercial, industrial, agricultural)

Each social factor would have to have its own data source (unless one source contains multiple).

We could also interrogate different language sets:

- profanity
- violence
- positive emotions (“happy”, “excited”) vs negative emotions
- desire (hope, wish, want, etc) vs fear

In the interest of an achievable scope, we should limit which social factors / language sets we are interested in. We can totally stick to violent-crime statistics and weather/climate for social factors and profanity/violence for language, in which case we have the above questions; I was just spitballing.

Regarding temporal trends, do we want to segregate by geography? We could show different regions as small multiples. Alternately, we could pick a location for simplicity and visualize temporal data just for that one.

Another possibility would be to have geography as the main view, and then display temporal information as the “detail-on-demand” feature required by the problem set spec.

Motivation

Thought this was great

Data

Thought: usage frequencies and violent crime stats should be adjusted by population to be on a per-capita basis.

Search for crime data

The FBI publishes Unified Crime Reports with a lot of statistics tables

Violent crime data by county (but for 2011 only)
http://www.fbi.gov/about-us/cjis/ucr/crime-in-the-u.s/2011/crime-in-the-u.s.-2011/offenses-known-to-law-enforcement/standard-links/county-agency

More tables:
http://www.fbi.gov/about-us/cjis/ucr/crime-in-the-u.s/2011/crime-in-the-u.s.-2011/violent-crime/violent-crime

And here is the preliminary report for the Jan-Jun 2012
http://www.fbi.gov/about-us/cjis/ucr/crime-in-the-u.s/2012/preliminary-semiannual-uniform-crime-report-january-june-2012

Might also be something at at the Bureau of Justice Stats
http://bjs.ojp.usdoj.gov/index.cfm?ty=tp&tid=3

Some wiki summaries:
http://en.wikipedia.org/wiki/Crime_in_the_United_States
http://en.wikipedia.org/wiki/Index_crime (FBI uniform crime reports)
http://en.wikipedia.org/wiki/National_Incident_Based_Reporting_System

Weather data:
Here is NOAA data (haven't looked through site yet)
http://www.ncdc.noaa.gov/
http://www.ncdc.noaa.gov/land-based-station-data/land-based-datasets

http://feedback.weather.com/knowledgebase/articles/27831-can-i-request-archived-weather-data-do-you-have-a

One question that remains is where we will find data displaying violent crime or weather over time. Perhaps we could focus on a single municipality for these views, if that data proves easier to find? In addition to source, we will also have to consider how to aggregate and represent these data.

Visualization
Tier I

- Heatmap

I agree that a geographic heatmap encoding language rate should be our primary goal.

We could allow "drill-down/filter capability" by giving the user an option to switch between a state-level view and a county-level view (easier to see cities, etc). If we only were to pick one, I would vote for a county-level view.

Can we use a map view to show co-localization of language / social factors?

We could encode language in red and encode a social factor in blue. The user would be allowed to select one of a variety of social factors at a time (crime rate, avg winter temprature, etc). Areas of white would be low for both language and factor. Areas of red would be high for language, blue would be high for factor, and purple would be high for both.
For inspiration see the county-level political map here:
http://politicalmaps.org/red-states-blue-states-purple-nation/

I worry about this color coding though (telling purple apart from blue, etc). Does it have integrity for quantitation?

Another option would be to color code language on the map as a factor, and allow the user to select it like others to view the geographic distribution (ie, language OR a factor, not language and 1 other factor). This way we are only color coding one thing at a time.

We could show comparisons by putting a scatter plot of language frequency vs [whatever factor the users selected] as a separate view, with each municipality as a point (could color code points by region). That's a nice way to visualize correlation. It would also give us a 'details on demand' feature.

Tier II

- Time series

Perhaps instead of an animation, we could offer a separate view that appears when a user clicks a state, as a details-on-demand feature. The view for that state would be a line chart of language over time. Users could select a factor to co-plot (weather, for example). Users would also be given a filter to change the period (hourly, monthly, seasonally).

Users could find trends by identifying patterns on the chart, or by scrolling through it, or maybe we automatically aggregate and present averages/deviations over the selected period, as here:
http://www.climate-charts.com/USA-Stations/VA/VA447925.php

Information for HW4 Initial Proposal Form

This is a condensed restatement of the brainstorms above, cleaned for submission as part of W4. Ted, please edit as you wish prior to submission

Tentative Title

"Fighting Words; Regional and temporal trends in Twitter vocabulary"

Group Member Info

Names

Ted Natoli, Brad Taylor
Emails
ted.e.natoli@gmail.com
brad.taylor3@gmail.com
Programming Experience
Ted: 3
Brad: 3

Research questions and hypotheses
High-level research question (domain)
Are there any regional trends in coarse vocabulary on social media and do these trends correlate with other social patterns (ie. violence, crime) and/or with seasonal changes? Our initial hypothesis is that the use of obscene, profane, or violent language will correlate with geographic regions of increased violence and crime and with typically unfavorable changes in weather (ie. heat wave, cold spell).

Lower-level questions (operational)
> Does violent or profane language vary geographically? Is geographical variation in violent or profane language associated with similar variation in violent-crime rate? Variation in regional climate?

> Does violent or profane language vary temporally? What is the pattern of variation by season, month, time of day? Is this variation associated with similar variation in violent-crime rate? Variation in weather?

> Does more violent or profane language occur on Facebook or Twitter?

Motivation
The massive volume of Twitter and other social media users and the availability of social media data has enabled researchers to ask and answer questions in ways that were previously impossible. Recently, violent crime has come to the forefront of our collective social awareness, due to incidents such as the shootings in Aurora, CO and Newtowne, CT. Is there a way (possibly via social media cues) we might be able to forecast and prevent these incidents?

Data
For linguistic data, we will use Python's pattern module to access Twitter (and Facebook?) data via its API. We will also use pattern's natural language processing capabilities to perform sophisticated analyses of the acquired data.
www.twitter.com
www.facebook.com

Municipal violent crime data is aggregated by the Federal Bureau of Investigation into annual Unified Crime Reports. Data may be downloaded in excel format from the FBI website. Many

tables are available, including violent crime data by county.
http://www.fbi.gov/about-us/cjis/ucr/crime-in-the-u.s/2011/crime-in-the-u.s.-2011/violent-crime/violent-crime

We are still looking for an appropriate source of weather/climate data. One possible source is the NOAA National Climate Data Center:
http://www.ncdc.noaa.gov/

Visualization
Tier I
A geographic heatmap with intensity encoding frequency of violent language usage? We may allow users to select alternate factors to encode on the map (violent crime rate, climate info). We may consider dual encoding langauge with these factors, as a 2-color map.

We could show language/social factor relationships by including a scatter plot adjacent to the map. This chart would display language frequency against a chosen social factor on separate axes, with each municipality as a point. We could color code points by region. This would also give us a 'details on demand' feature.

If we are unable to achieve additional tierII time-series charts, we may possibly add a slider bar (or animate) to show changes over time (and/or with seasons)

Tier II
To address questions related to temporal variation, we could offer a separate view that appears when a user clicks a state (a details-on-demand feature). This view could be a line chart of language over time for that state. Users could select a factor to co-plot (weather, for example). Users could also be given a filter to change the period (hourly, monthly, seasonally). We are still considering how to answer time-related questions.

Feedback from TF Megan Quintero 3/12/13:

Hello Ted and Brad,

I am going to be the TF grading your psets for the duration of the semester, and more importantly, project 2. From reading your proposal, I came up with the following feedback:

Very interesting project idea, however, I worry about the feasibility. I would suggest concentrating on the language used and how this changes and differs in different regions or during different seasons of the year. In order to draw conclusions based on language and its correlation with violence and crime, you would not only have to find a way to accurately collection that information, but have a hard time finding the best way to represent a "crime" score of an area and find a scale to determine the overall crime rate based on the both the frequency and

severity of crime.

I would stay away from Facebook at all costs as sites like OpenBook are rather questionable and not many people post publicly from their person facebook account. Additionally, I would strongly advise against using information from your own friends lists as this infringes upon privacy rights and will become overall very messy.

If you stick to answering questions such as the variance and frequency of language you can collect weather data, etc, to answer some just as interesting questions.

I believe your visualization idea of creating a heatmap sound great and is definitely the best approach. The line chart of language also sounds interesting. IN order to answer time-related questions, I would recommend also using a slider.

Please let me know if you have any questions or concerns regarding this feedback or in general. Do reach out to me with any further pitfalls you come across with project two.

Brainstorming session 3/12/13:

To do:

1. **start collecting tweets soon (Ted):** we need to get a feel for the scale of the data we'll be working with and whether we need to do any filtering up front to make the data set manageable.
    a. how many unique tweets do we see per some unit time?
    b. how many contain violent / profane language?
2. **make a violent / profane word list (Brad)**: how do we identify violent / profane tweets?
    a. conveying mood / intent?
3. **how did "wefeelfine" get their weather data?** can we use a similar method to collect weather data?
4. **initial viz sketches (both)**: include potential sub-visualizations and/or filtering options for user
5. **respond to Megan's comments as part of updated proposal (Ted)**

Final Project II Proposal 3/12/13:

Project title and teammate

Title: *"Fighting Words; Regional and temporal trends in Twitter vocabulary"*
Teammates:

1. Ted Natoli
2. Bradley Taylor

Research questions and hypotheses:

We have a number of questions we'd like to address in this project, but we have decided to triage and organize them by how realistic and attainable they are given the time frame of the project. In light of Megan's comments, we have decided to focus only on data from Twitter instead of Facebook.

Tier I research questions (less ambitious, more attainable)

1. **Does violent or profane vocabulary on social media vary geographically?** We hypothesize that it will be more prevalent in urban versus suburban areas.
2. **Does it vary with time of day?** We hypothesize that its use will spike during typical high-stress periods such as rush hour and will level out at other times.
3. **Does it vary with regional climate?** We hypothesize that it will be more prevalent in regions of extreme climate and less prevalent in temperate regions.

Tier II research question (most ambitious)

1. **Do trends in violent/profane language correlate with other social patterns (ie. violence, crime)?** Our initial hypothesis is that the use of obscene, profane, or violent language will correlate with geographic regions of increased violence and crime.

Motivation:

The massive volume of Twitter and other social media users and the availability of social media data has enabled researchers to ask and answer questions in ways that were previously impossible. Recently, violent crime has come to the forefront of our collective social awareness, due to incidents such as the shootings in Aurora, CO and Newtowne, CT. Is there a way (possibly via social media cues) we might be able to forecast and prevent these incidents?

Data source(s) and technical process:

Tier I Question Data Sources

For linguistic data, we will use Python's pattern module to access Twitter data via its API. We will also use pattern's natural language processing capabilities to perform sophisticated analyses of the acquired data. (Addresses Tier I Questions 1, 2)

www.twitter.com

We are still looking for an appropriate source of weather/climate data. One possible source is the NOAA National Climate Data Center:
http://www.ncdc.noaa.gov/

Another is Weather Underground: http://www.wunderground.com/.

These data sources will provide aggregate climate information. One or more climate-summary variables can be assigned to a given region as markers to indicate regional climatic variation (ex. Avg. January Temperature; Avg. June Temperature). This would allow us to answer Question 3, and determine if regional variation in language corresponds with regional variation in climate.

Tier II Question Data Sources
Municipal violent crime data is aggregated by the Federal Bureau of Investigation into annual Unified Crime Reports. Data may be downloaded in excel format from the FBI website. Many tables are available, including violent crime data by county or state. A simple sum of violent crimes would provide a rough aggregate measure of regional violence.
http://www.fbi.gov/about-us/cjis/ucr/crime-in-the-u.s/2011/crime-in-the-u.s.-2011/violent-crime/violent-crime

Regarding temporal variation for weather: If we are able to collect language information on a seasonal scale (uncertain, will depend on our scraper), these sources would provide data (Avg. temperature by month for a given region).

Potential visualizations and technical process:

Tier I
Our main visualization will be a geographic heatmap with intensity encoding frequency of violent and/or profane language usage. We will also provide a drop down menu to allow users to select alternate factors to encode on the map (violent crime rate, climate info) if we are able to obtain those additional data. Megan's comments on this aspect of the visualization were:

"I believe your visualization idea of creating a heatmap sound great and is definitely the best approach. "

These comments solidified our choice to use the heatmap to display geographic trends in violent and/or profane Tweets. We hope to achieve something like the visualization below, found at

ny.spatial.ly:

Tier II

To address questions related to temporal variation, we could offer a separate view that appears when a user clicks a state (a details-on-demand feature). This view could be a line chart of language over time for that state. Users could select a factor to co-plot (weather, for example). Users could also be given a filter to change the period (hourly, monthly, seasonally). We also plan to add a slider bar (or animate) to show changes over time (and/or with seasons). Megan's comments regarding this idea were:

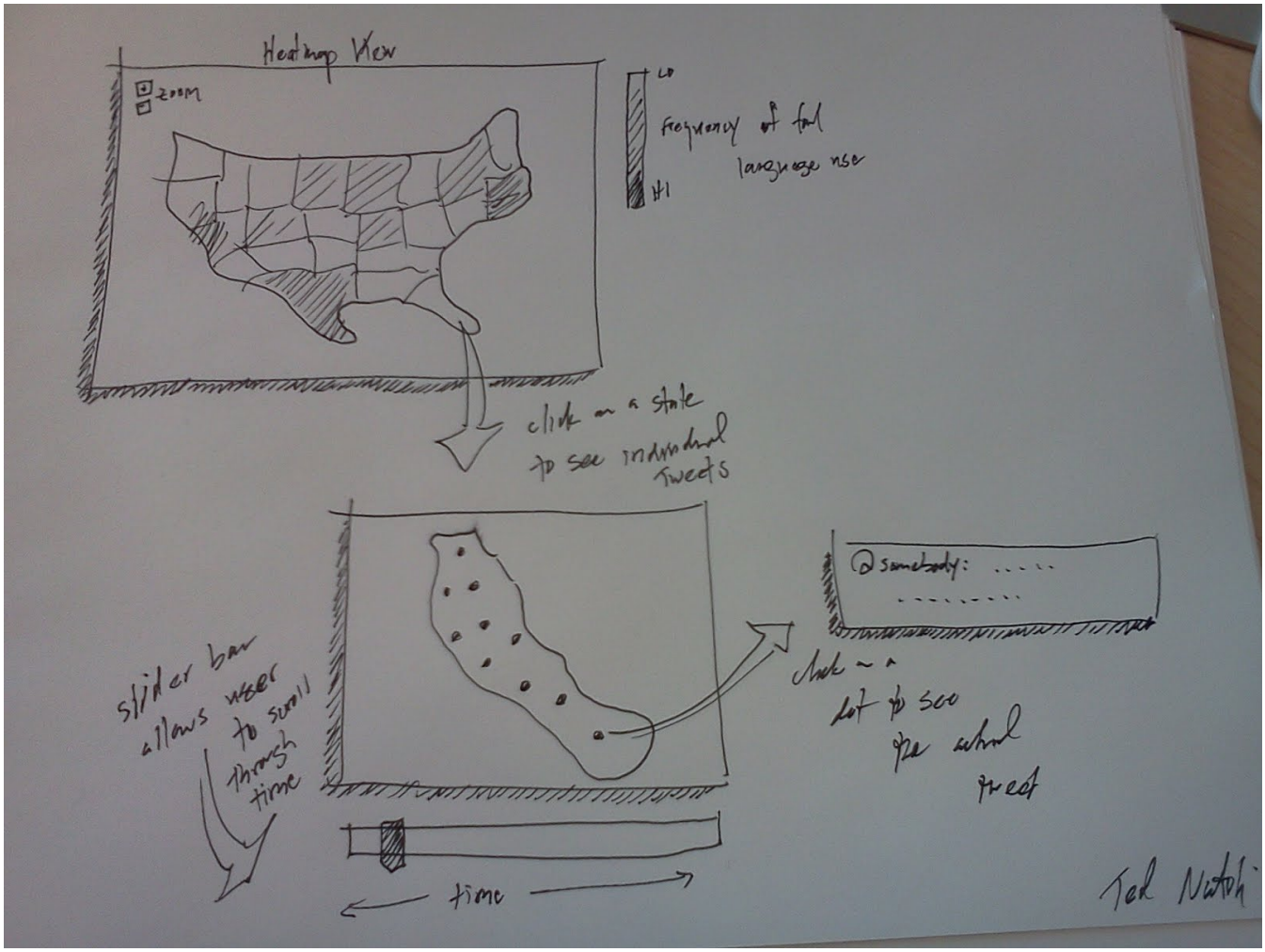
"The line chart of language also sounds interesting. IN order to answer time-related questions, I would recommend also using a slider."

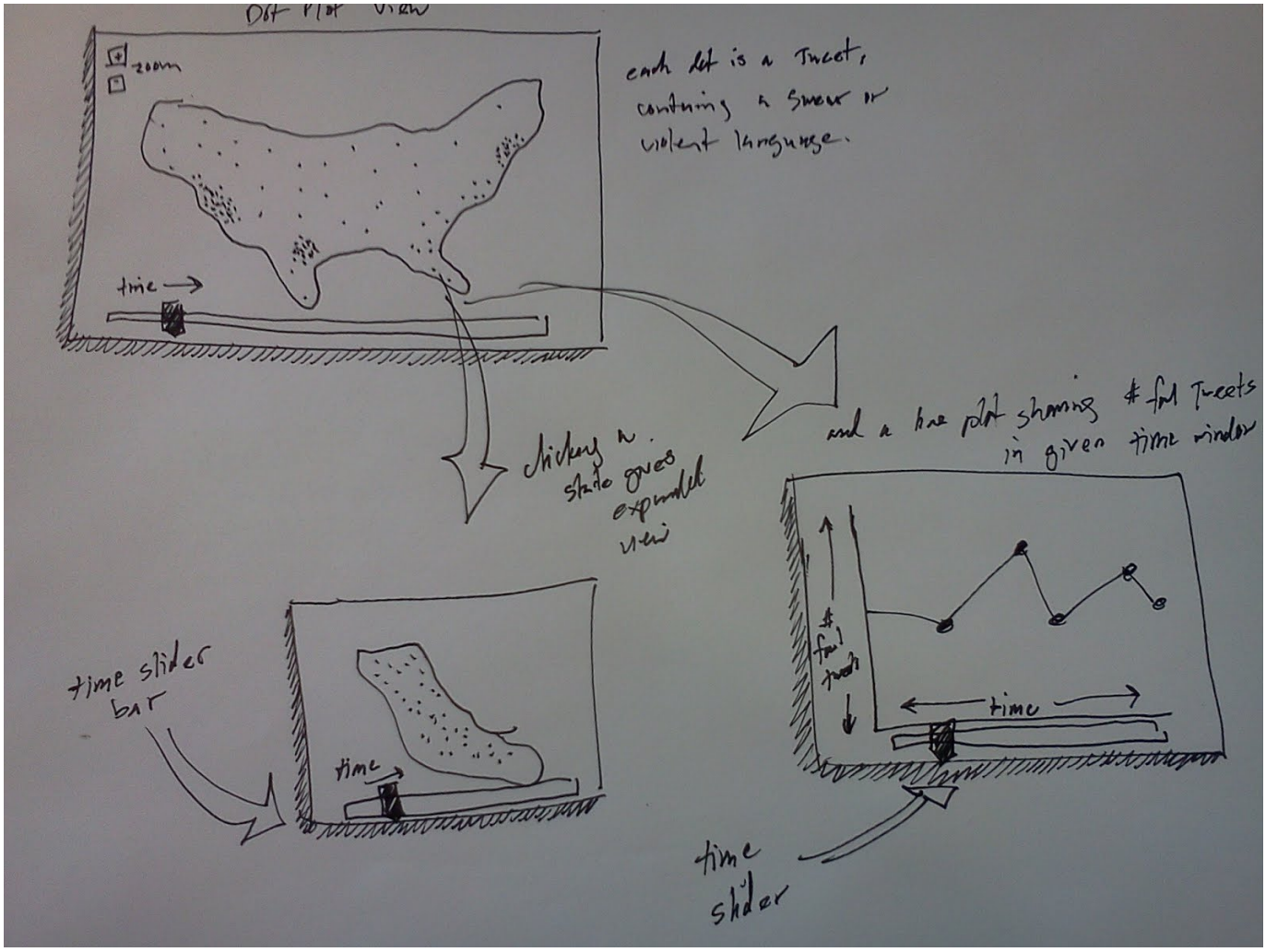
If we have time, we could show language/social factor relationships by including a scatter plot adjacent to the map. This chart would display language frequency against a chosen social factor on separate axes, with each municipality as a point. We could color code points by region. This would also give us a 'details on demand' feature.

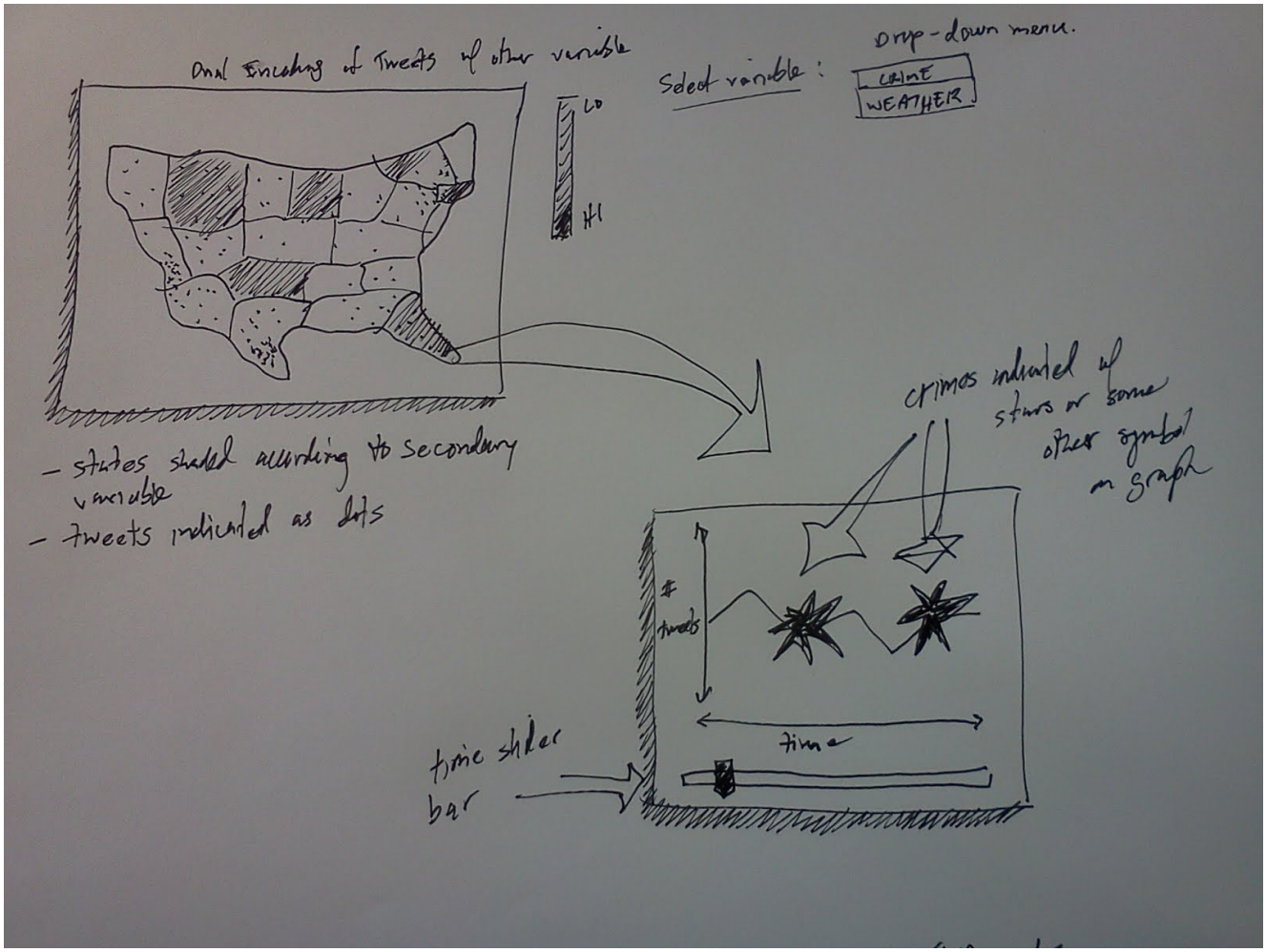
We plan to use a combination of HTML5, CSS and D3 to construct our visualization. We will create and style our web-based visualization with HTML5 and CSS. The heat map and other interactive components will be implemented in D3, but we have yet to work out the specifics of our implementation.

Three sketches:

Ted's sketches:

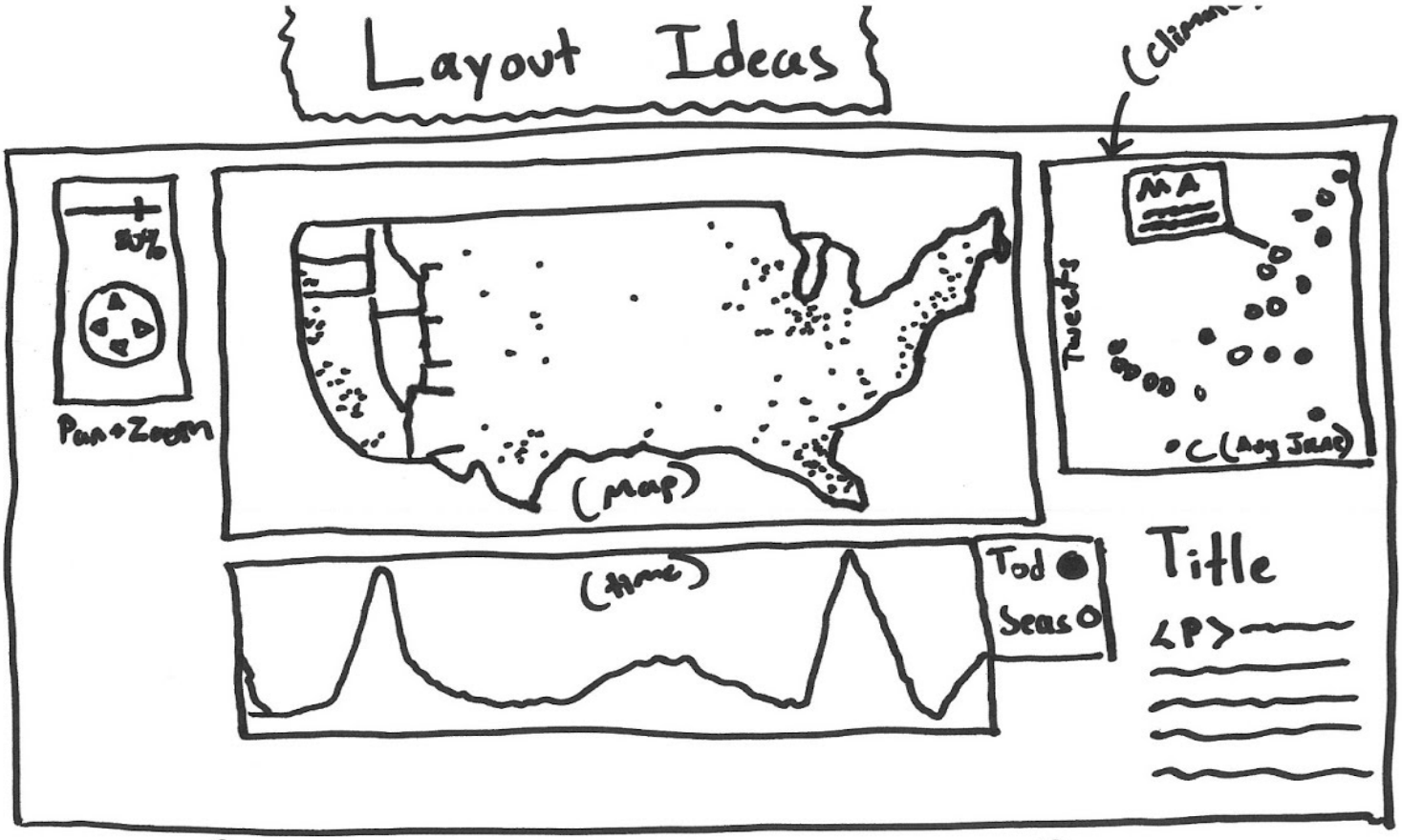

Heatmap View
zoom
Lo
Frequency of foul language use
Hi
click on a state to see individual tweets
@somebody: .....
slider bar allows user to scroll through time
time
Tod Nutch


Dot Plot View
zoom
time
each dot is a Tweet, containing a swear or violent language.
clicking on shade gives expanded view
and a line plot showing # foul Tweets in given time window
foul Tweets
time
time slider bar
time
time slider



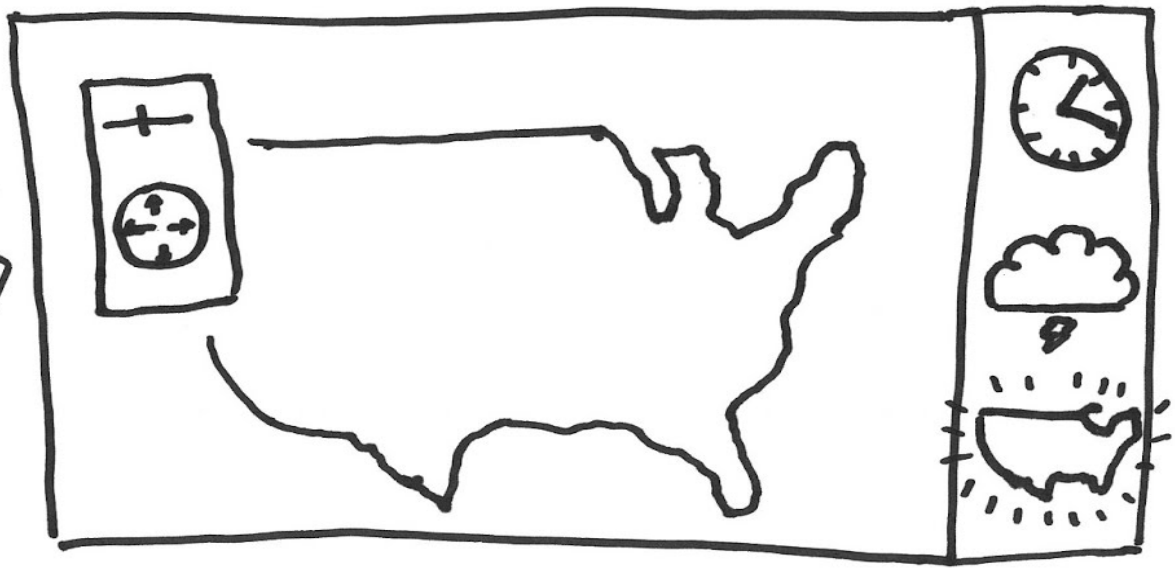
Brad’s sketches

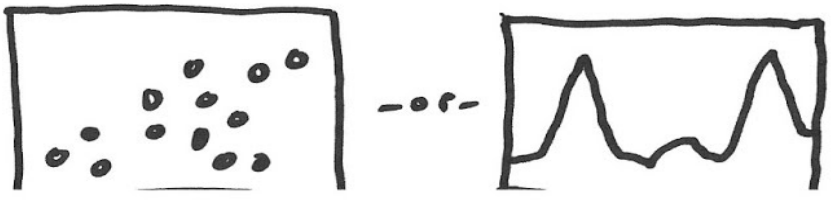
Layout Ideas



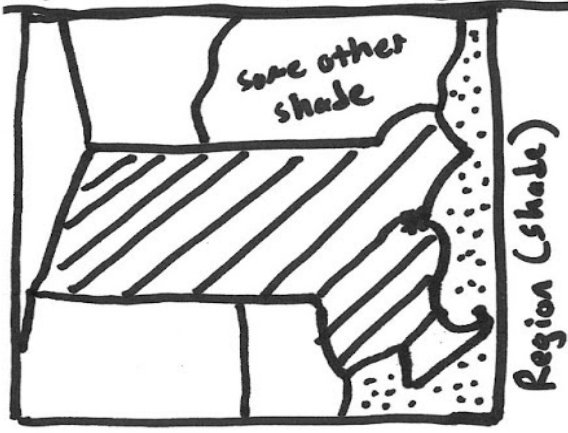
"The all in one"

"The Context Switch"

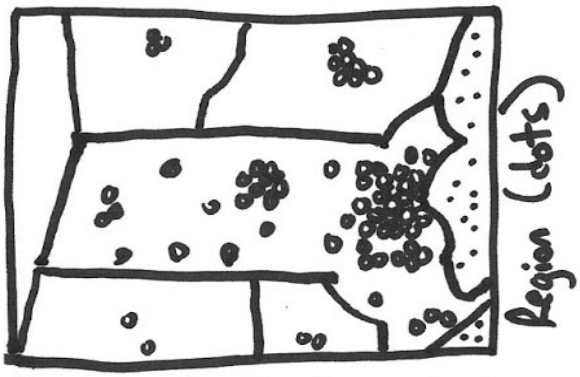




ENCODINGS



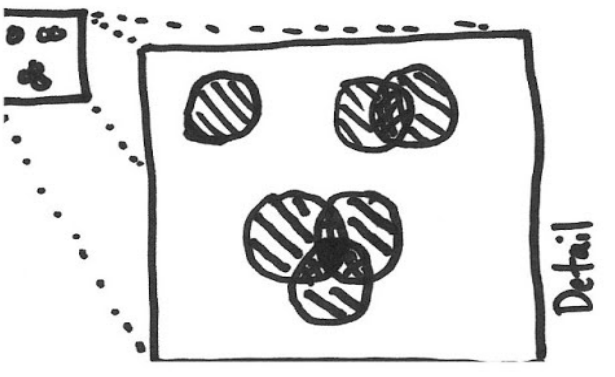
Aggregate by state + shade (color value)



Display single Tweets as Circles

- More Granular (each tweet a dot)
- Issue - too sparse outside pop. areas?
- Issue - pile-up + overlap in pop areas

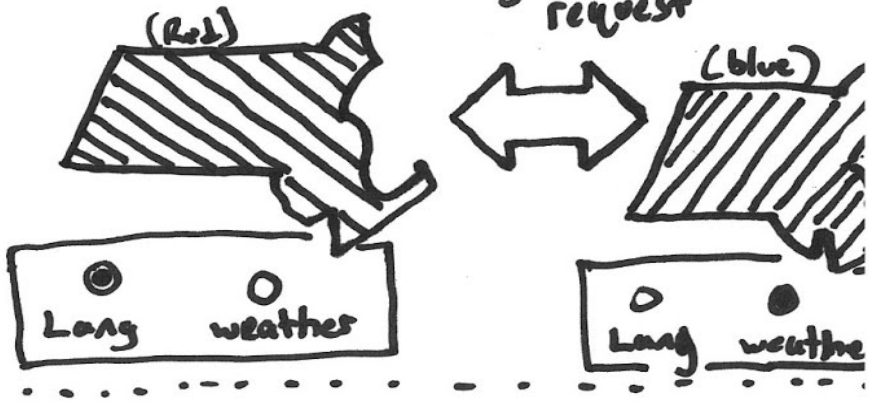
↳ show density
- add opacity transparency
- more dots = darker



'weather' = Avg. January °C

weather

Change shade on request

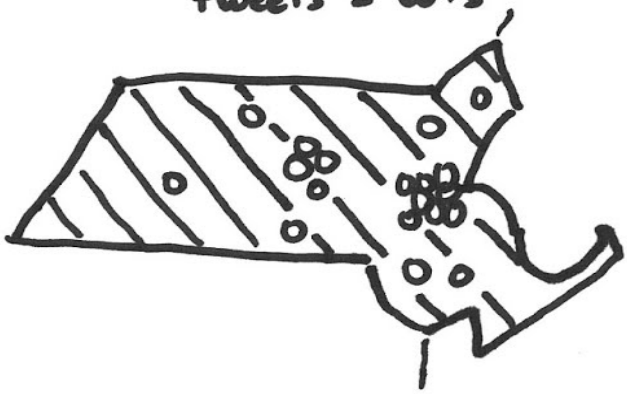


- OR -

Combination encoding

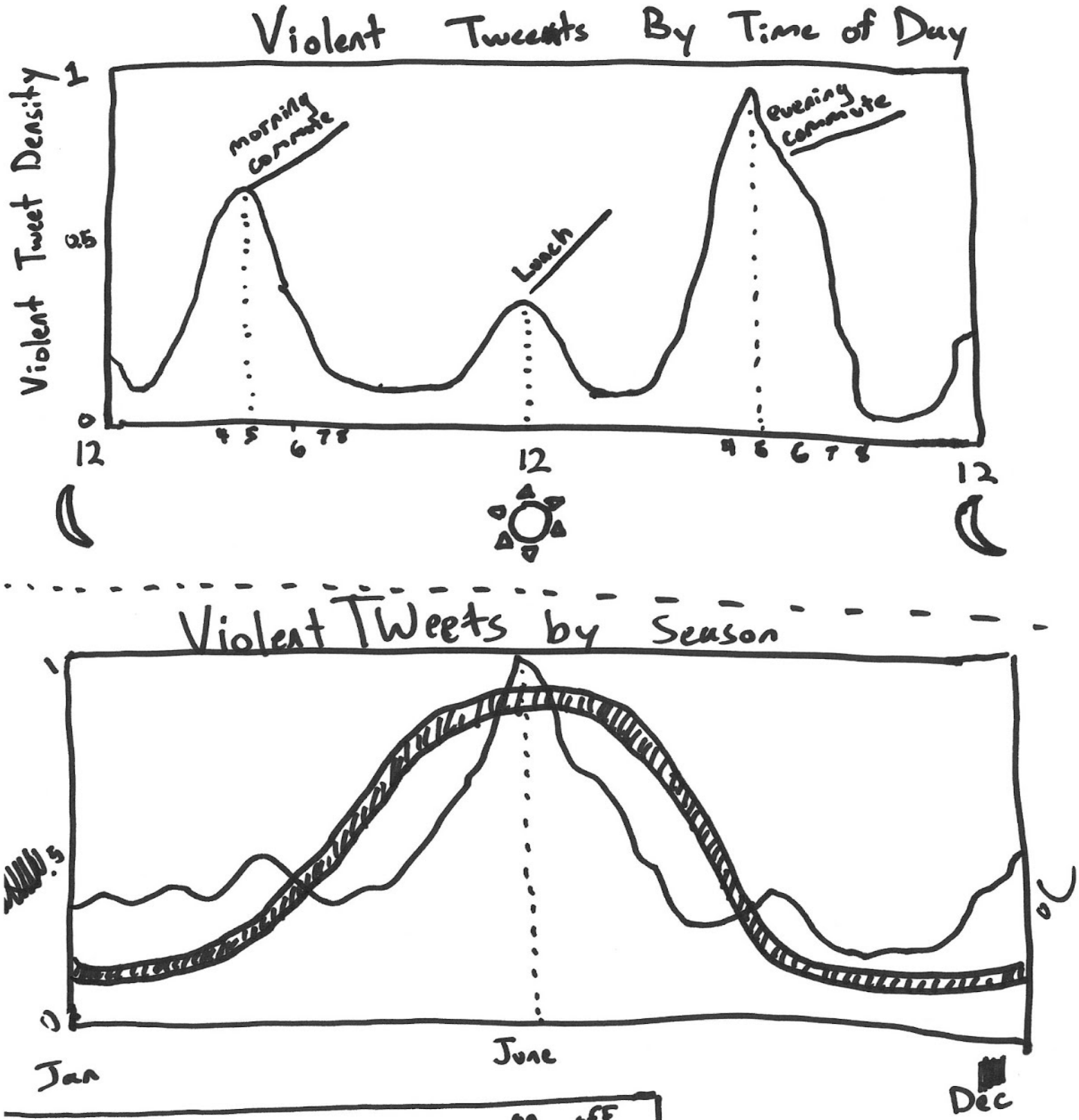
weather = shade

tweets = dots



Tweet + Weather (state Climate)

encodings in Regional Language Map


Violent Tweets By Time of Day
Violent Tweet Density
1
.5
0
morning commute
lunch
evening commute
12
4 5 6 7 8
12
4 5 6 7 8
12
Violent TWeets by Season
1
.5
0
Jan
June
Dec
°C
on
off
Avg Temp (Boston, MA)
Weekly Avg Violent Tweet Dens.

Brad Proposal Edits 3/13/13

Ted, thank you for refactoring and enhancing our initial proposal. I think your changes/additions perfectly encapsulate our discussion yesterday and do a great job responding to Megan's comments.

I made the following edits directly to the proposal above. Ted: We can revert or change them if you disagree with anything. Most are minor and for clarity. I added some thoughts on weather data to give more detail on what we propose to collect/store.

Research Q's

"Lower level research questions (...)" >> "Tier I research questions (...)"
"Higher level research questions (...)" >> "Tier II research questions (...)"
I had originally split questions into levels based on degree of detail (over-arching theme vs specific tasks). I think all of the ones we list are operational level (task specific), as I feel we could address each of them in a single view (in particular, crime correlation is addressed by the proposed scatter plot). This is a good thing, as we have refined our questions into specific goals to implement. The division is now based on achievability; immediate vs reach goals. Just wanted to make that explicit, and normalize with the jargon in the viz section.

Data Sources

"use Python's pattern module to access Twitter (and Facebook) data" *(removed to address feedback)*

"Are there any regional trends in coarse vocabulary on social media and do these trends correlate with other social patterns (ie. violence, crime)? "
"Do regional trends in violent/profane language correlate with other social patterns (ie. violence, crime)? "
We address the existence of regional variation as part of Tier I Question 1

"Tier I Question Data Sources"
"Tier II Question Data Sources"
Added divisions into tiers for our data sources section, to line sources up with questions and to help tell a unified story. I moved the crime data source into tier II

(Addresses Tier I Questions 1, 2)
Help to tie data to questions

These data sources will provide aggregate climate information. One or more climate-summary variables can be assigned to a given region as markers to indicate regional climatic variation (ex. Avg. January Temperature; Avg. June Temperature). This would allow us to answer question 3, and determine if regional variation in language corresponds with regional variation in climate.
Your restatement/refining of our questions helped clarify how we would address the issue of regional climate. I added a section in the data sources with an idea for this.

A simple sum of violent crimes would provide a rough aggregate measure of regional violence.
Addresses Megan's comment:
"You would...have a hard time finding the best way to represent a "crime" score of an area and find a scale to determine the overall crime rate based on the both the frequency and severity of crime."

Regarding temporal variation for weather: If we are able to collect language information on a seasonal scale (uncertain, will depend on our scraper), these sources would provide data (Avg. temperature by month for a given region).
Data source clarification for our idea of language variation by seasonal climate. This is definitely a tier II (or III) feature, since as we discussed it is questionable whether we could gather seasonal-scale language information

Finally, I went and looked through "We Feel Fine". Unfortunately, they do not expose their source code (just an API). Nor do they give much indication as to where they get their weather data, just that they pull it based on the location of the blog entry. We could similarly use location to scrape weather at the time of tweet collection, but I don't know where we'd get it from (pattern.search() using google as the engine??). I'm not putting any of this in the process book for now, because it seems like it would be pretty hard (much harder than crime, which we have a nice CSV data source for). I would work on this more, but...

Pattern.web does not return location results for twitter searches!!!! 3/13/13 (BT)

While thinking about how to link up weather and tweet data by location, I went back to the pattern.web documentation. It turns out pattern does not normally return a location parameter for its results! This is a major blocker, because location is the basis for our whole project.

I did a brief look into whether we could get location data for tweets, and whether pattern could be altered to deal with this. But I cant go very deep because I am at work

<u>The result object contains only:</u>
profile (picture)
language
author
text
date

A quick google indicates that twitter DOES expose location data through it's API.

http://support.twitter.com/articles/78525-faqs-about-tweet-location#

https://dev.twitter.com/discussions/3755
https://dev.twitter.com/discussions/9267
https://dev.twitter.com/
(havent read these fully)

Below is the source code for the Twitter API search engine class in pattern. From the pattern source code download (pattern-2.5/pattern/web/__init__.py)

Evidently, you can filter your results based on geo location. Perhaps we could create our own class that mirrors this but can also pack geo location into the result??? (need to take a look at the URL class which this makes use of).

```
#--- TWITTER
--------------------------------------------------------------------------------
# http://apiwiki.twitter.com/

TWITTER        = "http://search.twitter.com/"
TWITTER_STREAM  = "https://stream.twitter.com/1/statuses/filter.json"
TWITTER_STATUS  = "https://twitter.com/%s/status/%s"
TWITTER_LICENSE = api.license["Twitter"]
TWITTER_HASHTAG = re.compile(r"(\s|^)(#[a-z0-9_\-]+)", re.I)      # Word starts with "#".
TWITTER_RETWEET = re.compile(r"(\s|^RT )(@[a-z0-9_\-]+)", re.I) # Word starts with "RT
@".

class Twitter(SearchEngine):

        def __init__(self, license=None, throttle=0.5, language=None):
        SearchEngine.__init__(self, license or TWITTER_LICENSE, throttle, language)

        def search(self, query, type=SEARCH, start=1, count=10, sort=RELEVANCY, size=None,
cached=False, **kwargs):
        """ Returns a list of results from Twitter for the given query.
                - type : SEARCH or TRENDS,
                - start: maximum 1500 results (10 for trends) => start 1-15 with count=100,
1500/count,
                - count: maximum 100, or 10 for trends.
                There is an hourly limit of 150+ queries (actual amount undisclosed).
        """
        if type != SEARCH:
                raise SearchEngineTypeError
        if not query or count < 1 or start < 1 or start > 1500 / count:
                return Results(TWITTER, query, type)
        # 1) Construct request URL.
        url = URL(TWITTER + "search.json?", method=GET)
```

```
        url.query = {
                "q": query,
                "page": start,
                "rpp": min(count, 100)
        }
        if "geo" in kwargs:
                # Filter by location with geo=(latitude, longitude, radius).
                # It can also be a (latitude, longitude)-tuple with default radius "10km".
                url.query["geocode"] = ",".join((map(str, kwargs.pop("geo")) +
["10km"])[:3])
        # 2) Restrict language.
        url.query["lang"] = self.language or ""
        # 3) Parse JSON response.
        kwargs.setdefault("unicode", True)
        kwargs.setdefault("throttle", self.throttle)
        try:
                data = URL(url).download(cached=cached, **kwargs)
        except HTTP420Error:
                raise SearchEngineLimitError
        data = json.loads(data)
        results = Results(TWITTER, query, type)
        results.total = None
        for x in data.get("results", data.get("trends", [])):
                r = Result(url=None)
                r.url          = self.format(TWITTER_STATUS % (x.get("from_user"),
x.get("id_str")))
                r.text         = self.format(x.get("text"))
                r.date         = self.format(x.get("created_at", data.get("as_of")))
                r.author   = self.format(x.get("from_user"))
                r.profile  = self.format(x.get("profile_image_url")) # Profile picture URL.
                r.language = self.format(x.get("iso_language_code"))
                results.append(r)
        return results
```

Alternately, we'd have to find some other way to access twitter's API or pull tweets with geo location.

Alternately, we could significantly alter our project to ignore location. We could use time only, and find patterns/correlates there. Or we could look for relationships between our key words and other aspects of the sentence (using pattern.en?)

TN 3/17/13

Today am I looking into collecting Tweets that include some type of geo-location. It doesn't seem like there is a way to collect the exact geo-location of a Tweet directly from Twitter's API.

https://dev.twitter.com/docs/api/1.1/get/search/tweets
https://dev.twitter.com/terms/geo-developer-guidelines

As Brad mentioned, it is possible to specify a geo-location and search radius and return only results within that region, but that doesn't give us the precision we would ideally like.

After some additional research, I don't see a way to return the user's exact location along with his or her Tweet. I think that means we'll need to specify a list of locations a-priori, and then search for tweets within those locations and map the search results to the specified location.

A few lists containing some major cities' latitude and longitude locations.

http://en.wikipedia.org/wiki/List_of_cities_by_latitude
http://www.factmonster.com/ipka/A0001796.html
http://dev.maxmind.com/geoip/geolite

I downloaded the database from MaxMind and subset it down to just US cities. This resulted in over 58k entries, with granularity at the level of zip code. If we just use pattern's default 10km search radius, it is likely that search regions will overlap. I don't think that's a huge issue, as long as we hash tweets and make sure we only retrieve unique ones, and that we just attribute tweets to a general metropolitan area. For example, tweets within these locations would all be attributed to Boston:

GeoLiteCity-Location.csv:180160,"US","MA","Boston","02204",42.3389,-70.9196,506,617
GeoLiteCity-Location.csv:180468,"US","MA","Boston","02266",42.3389,-70.9196,506,617
GeoLiteCity-Location.csv:180483,"US","MA","Boston","02123",42.3584,-71.0598,506,617
GeoLiteCity-Location.csv:180939,"US","MA","Boston","02295",42.3389,-70.9196,506,617
GeoLiteCity-Location.csv:180944,"US","MA","Boston","02298",42.3584,-71.0598,506,703
GeoLiteCity-Location.csv:181072,"US","MA","Boston","02217",42.3389,-70.9196,506,617
GeoLiteCity-Location.csv:181133,"US","MA","Boston","02203",42.3607,-71.0591,506,617
GeoLiteCity-Location.csv:181698,"US","MA","Boston","02209",42.3584,-71.0598,506,617
GeoLiteCity-Location.csv:186943,"US","MA","Boston","02222",42.3663,-71.0628,506,617
GeoLiteCity-Location.csv:187520,"US","MA","Boston","02283",42.3389,-70.9196,506,617
GeoLiteCity-Location.csv:193889,"US","MA","Boston","02297",42.3389,-70.9196,506,617
GeoLiteCity-Location.csv:197237,"US","MA","Boston","02284",42.3389,-70.9196,506,617

Code testing:

My implementation seems to be working to some degree, but one thing I didn't account for is that some tweets will be doubly counted if they contain multiple curse words. How do we want to handle this? Also, how should we handle re-tweets?

I tried running one iteration of my algorithm. My parameters were:

- 3 curse words
- all ~58k US zip codes
- return up to 100 tweets per zip code

swear word list
http://www.noswearing.com/dictionary

Twitter API enforces 150 queries per hour limit. Subset our query space to match this?

Brad 3/18/13
Tweets are returned from API as a JSON object containing various fields. Including:
{geo: type: point, coordinates: [lat, long]}
https://dev.twitter.com/docs/platform-objects/tweets

I think pattern is packing info into a URL query sent to the API and then parsing the JSON string it returns.
from patter.web __init__ class “Twitter” method “search”
r.date = self.format(x.get("created_at", data.get("as_of")))

Perhaps we could cut our own custom class which mirror’s pattern’s twitter class but adds a line to pull out the geo coordinates from the JSON, similar to the above

Meeting 3/18/13
We tried this out live, together. We copied the whole pattern directory to pattern_copy
then we edited web/__init.py__

We added a single line to the Twitter class:
r.geo = self.format(x.get(“geo”))

We then imported our modified module in a test script and ran the typical pattern twitter search loop.
We pulled 10 tweet results, and checked the geo field for each. 9 were ‘null’ and 1 had coordinates: [lat, long]

Success. Now the question is, what %age of tweets have geo information (it is an opt-in feature people enable when their twitter client or a third party twitter plugin app asks them too)

If we can grab 1500 results / hour, a 10% return should net 150 geocoded tweets / hour.
This gives us tweet, location, time of day, enough information for those research questions.

We could increase our “catch rate” by using the geo kwarg in Twitter.search() to restrict range to just North America (stick a lat-long pin in the center of the US, define some radius covering entire continental US, pass this as part of query). This should filter out tweets outside of North

America, and tweets without geo codes.

We can execute (multiple) individual queries for a list of buzz words; up to our 150 query limit for a given hour. We will manually generate the list of buzz words for simplicity. We will capture all tweets containing a buzzword. Later on, we could filter by lexical context, perhaps using pattern.en. But capturing more now gives us flexibility later.

regarding RTs
We decided that RT's should not be filtered out. We are interested in quantifying acts of vulgarity, not in investigating how discussions or memes are created. Each RT is an independent act of vulgarity that a user chooses to commit. For our purposes users choosing to associate themselves with a vulgar or violent phrase via RT are not essentially different than users choosing to do so by composing a new tweet.

Regarding weather
we can get climate data fairly easily. Live weather is harder. We could use climate as a way of categorizing place (do average March temperature for every state or county). Color code this on map. It's a rough estimate (Michigan colder than Texas)

Live weather data can perhaps be obtained by putting a query to weatherunderground.com (or other). Query sends in lat/long.

If we could parse the result this query returns, we can extract lat/long from tweet and query weather at time of tweet acquisition.

Regarding crime
FBI unified crime report has an xls file listing the number of violent crimes for every county in the US in 2011. Again, it's not live info. But it does allow us to categorize locations into high-crime vs low-crime. We would need a way to map tweets to their county.

TODOs:
Ted- update tweet scraper and start gathering some tweets, see what our yield is

Brad- Get climate data. Try to create a live-weather query system which could then be plugged into tweet scraper.

We'll do crime if we get to it.

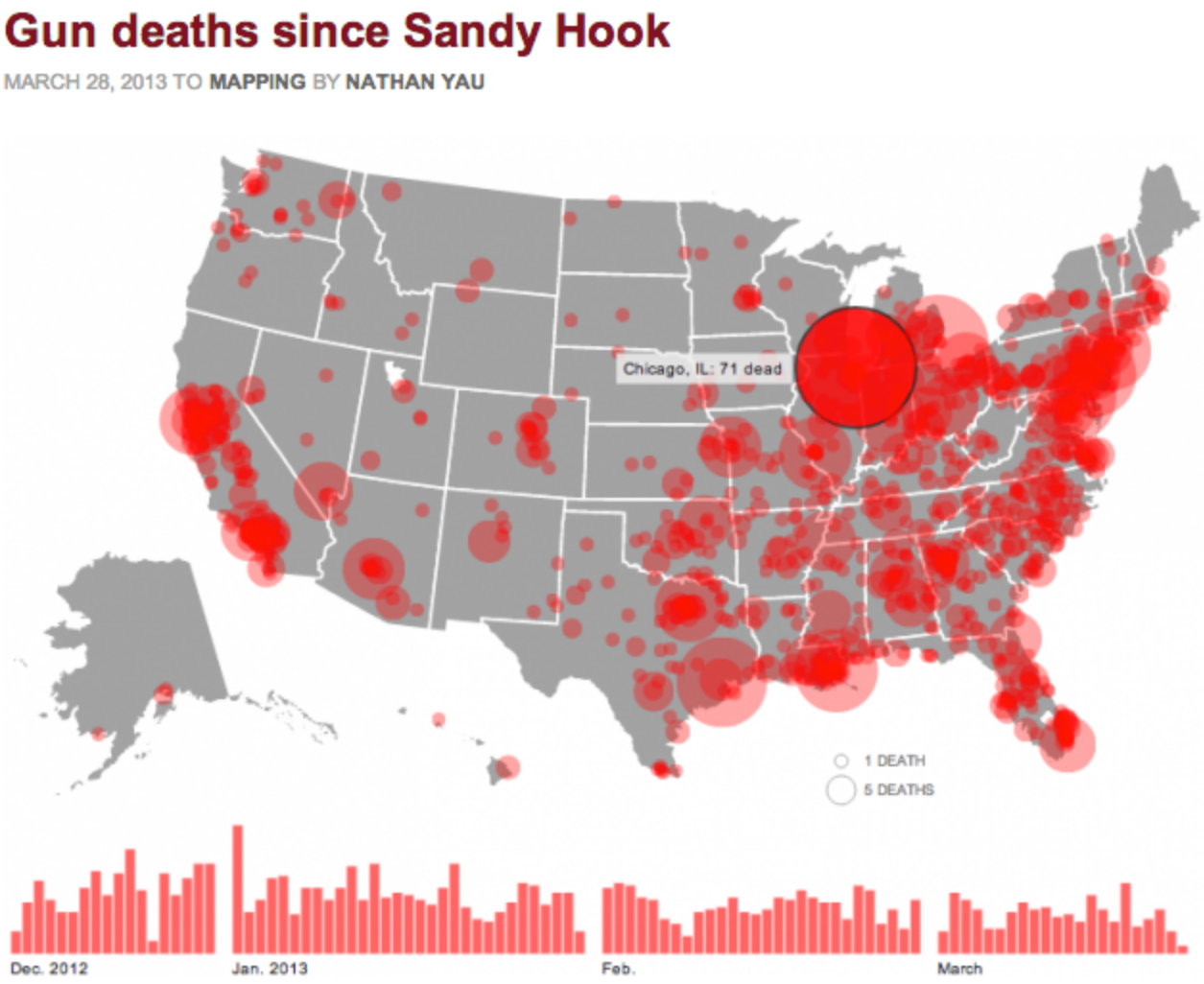
TN 3/21/13

I updated the scraper to use our modified version of pattern. It now searches for a list of

buzzwords (also updated) and returns only those tweets that have geo-location information. I've set it to run every 15 minutes, so we'll see how much data that generates over the next day or so.

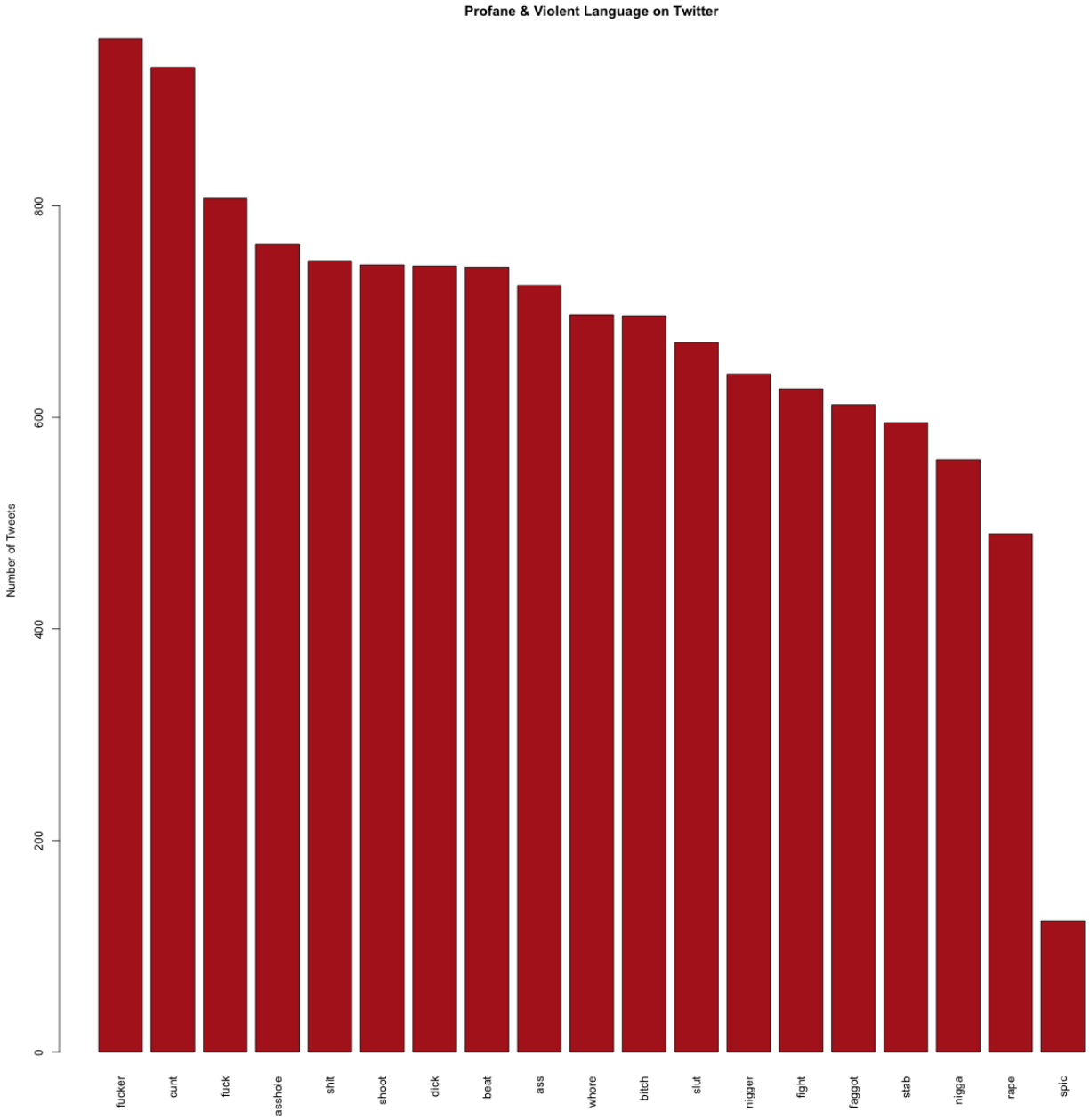
TN 3/28/13

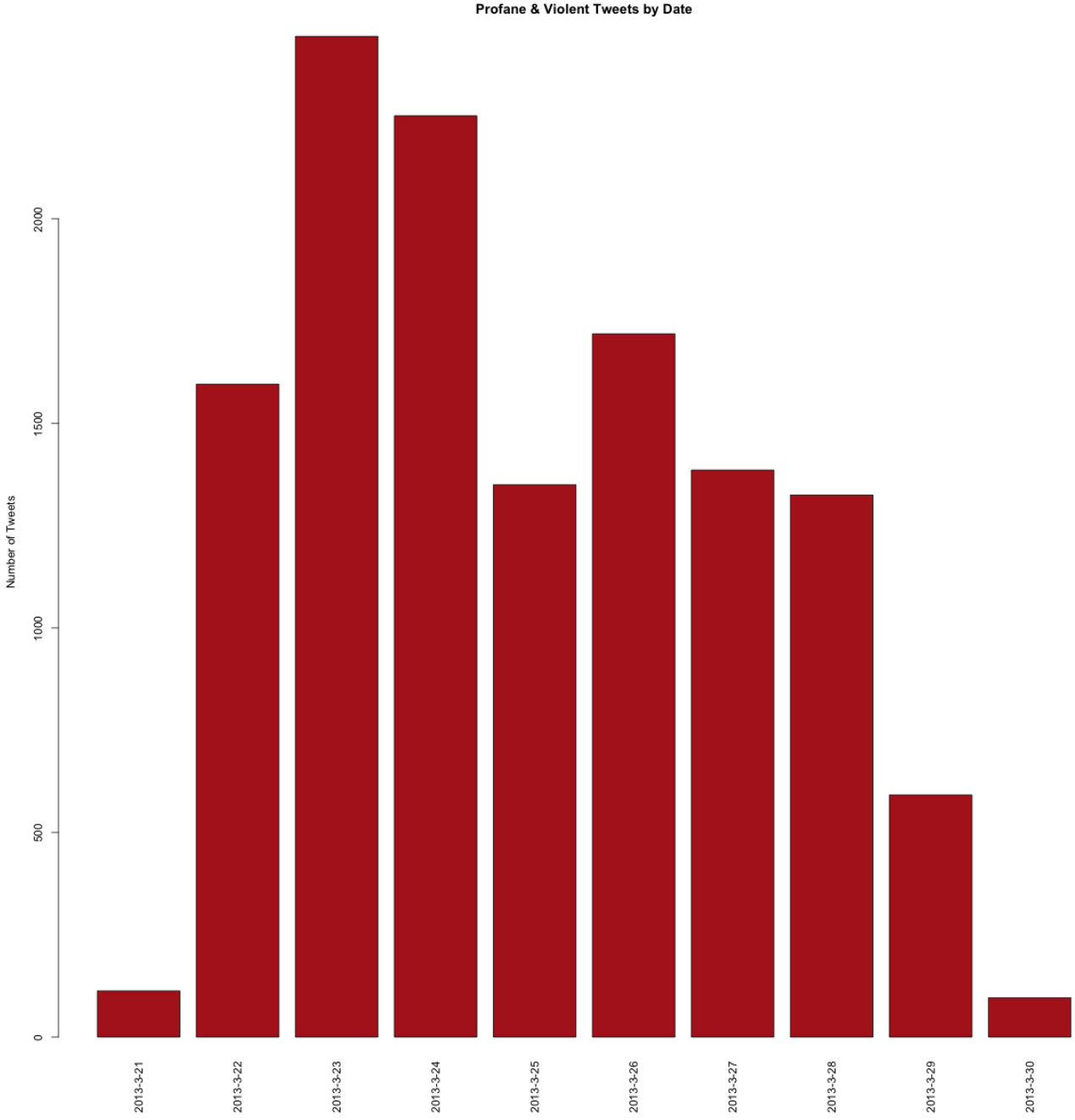
A viz I found inspiring

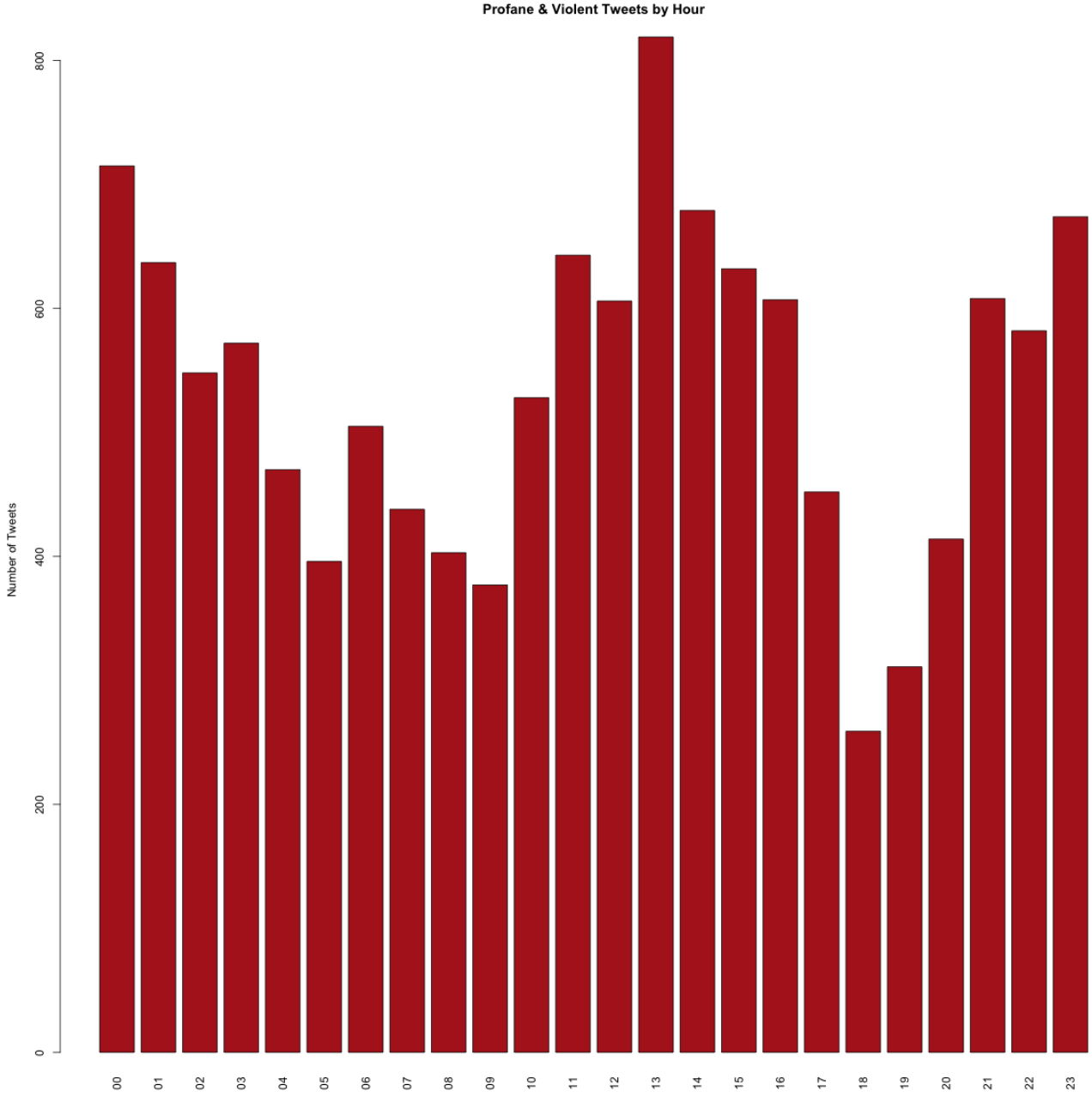


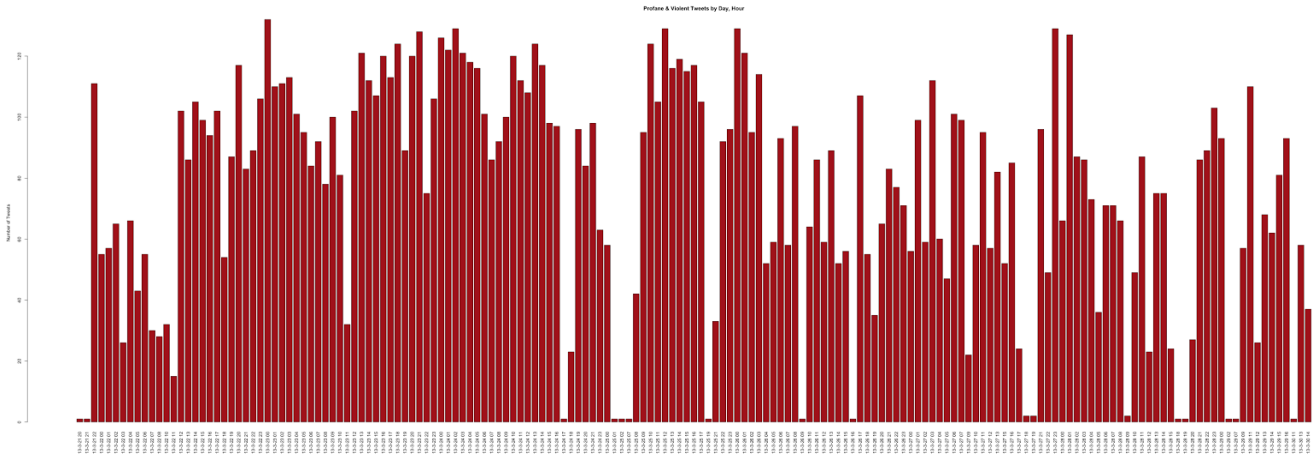
TN 3/30/13

I have been acquiring geo-tagged tweets for the past week and wanted to have an exploratory look at the data. Here are a few plots I made using R.




Profane & Violent Tweets by Date
Number of Tweets
0
500
1000
1500
2000
2013-3-21
2013-3-22
2013-3-23
2013-3-24
2013-3-25
2013-3-26
2013-3-27
2013-3-28
2013-3-29
2013-3-30


Profane & Violent Tweets by Hour
Number of Tweets
0
200
400
600
800
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23



BT 3/31/13

Ted, this is great work. The temporal stuff is really nice. We can already see that we got significantly more tweets on the weekend. Also spikes in tweeting in an apparent 12 hour cycle (noon and midnight). What great deliverables!

I wonder how the volume of profane tweeting compares to the volume of tweets in general.

Great work!

BT 3/31/13

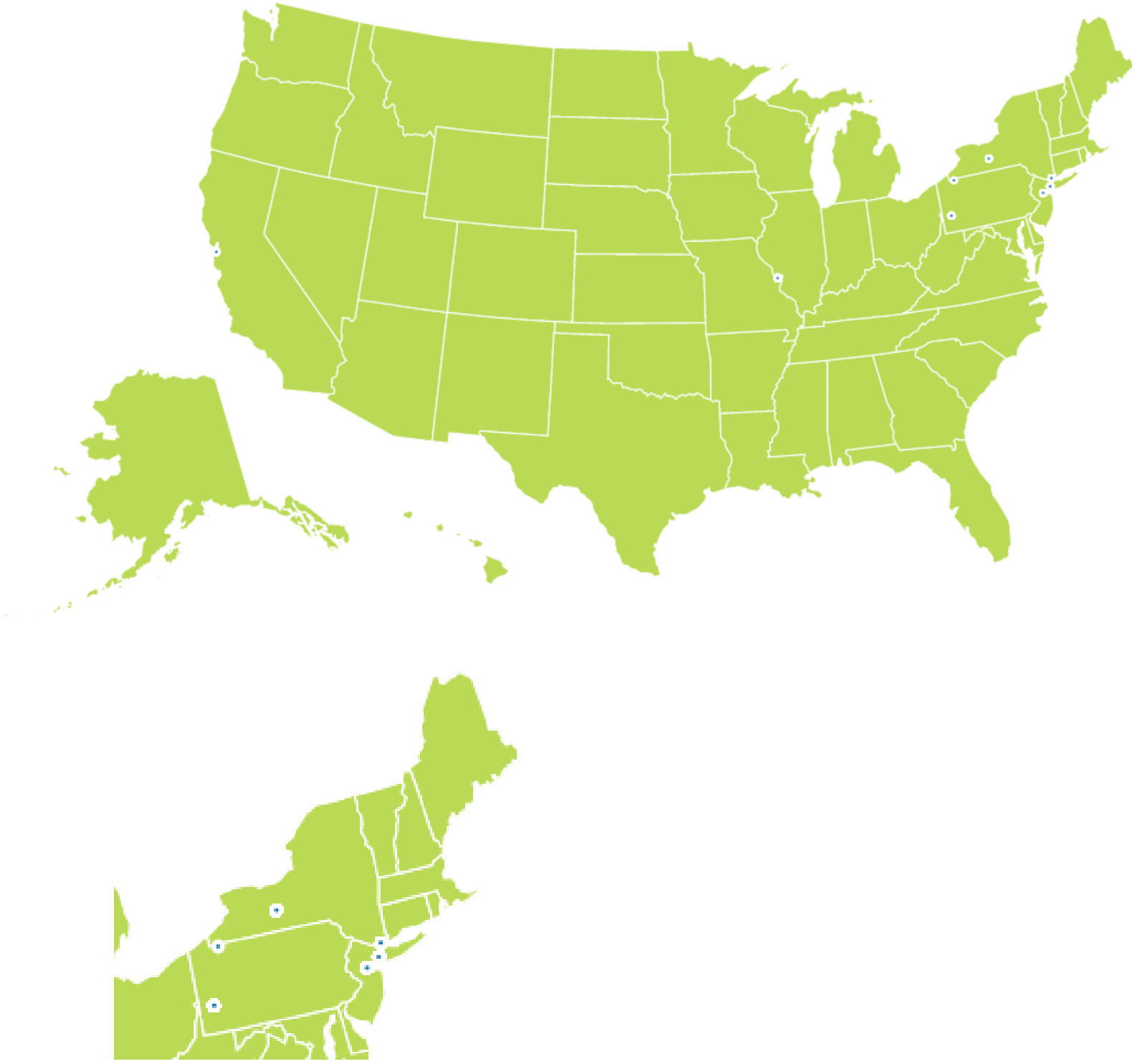
I did some playing around creating datamaps, to explore some of our geographic design ideas from our sketches, in rough rough draft form

Basically, this was me messing around a bit using some of the techniques we learned in Homework6, adapted to use our tweet data. I followed the practice from the homeworks of translating CSV data into formatted JSON (using a python script) and embedding the data in the webpage in a hidden text area. I then parsed that data and fed it into datamaps, using the "bombs" bubbles and "chloropleth" color map examples from the web.

I used the CSV file 'tweets2.csv' as my data source. There were only about 3500 tweets there, so I assume this was a draft dataset based on a test of Ted's scraper.

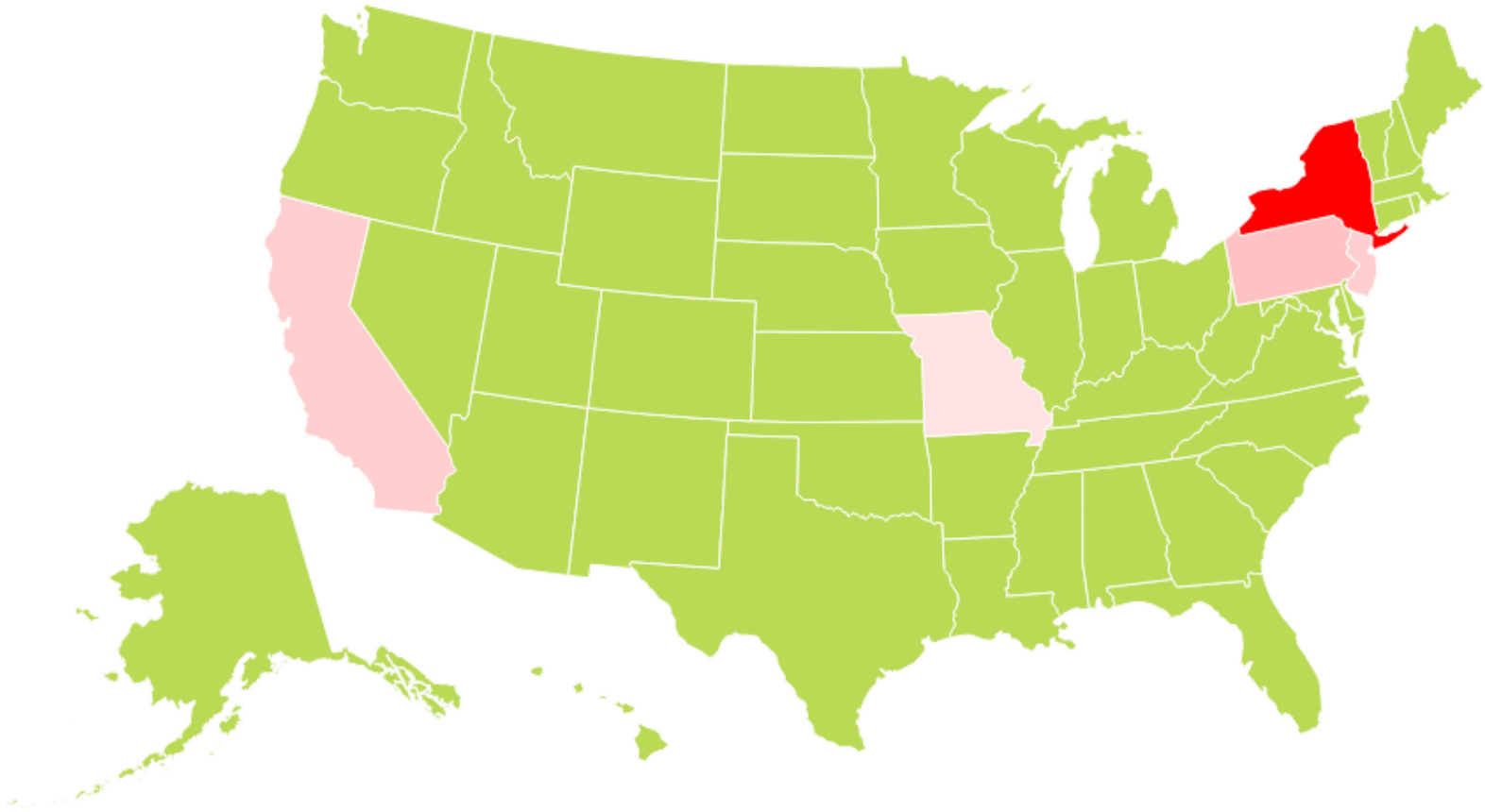
I made a 'bubbles' map where each individual tweet is a dot (image below). In this case, most of the tweets came from the same set of locations (NY, CA, MO, PA, etc). So the 3500 tweets I had were piling up in just a few areas. I think this might have to do with the scraping conditions used with this dataset. It's hard to see the bubbles since there are so few locations, but you can get a sense of what we might see in a more complete vis.

Test Implementation of a Tweet Datamap



Next, I made a chloropleth. Here, I aggregated tweets into state totals. I applied a linear color scale to the state totals, and colored the states based on total tweets (only a few states had tweets). I did this because it might be easier to correct state totals by state population. If we plot individual tweets without aggregation, we may end up simply highlighting high-population areas. I'm not sure what population-correction transformations we could apply if we want to go with the 'dots' view- something for think about.

Test Implementation of a Tweet Datamap



BT 4/2/13

Ted implemented a line chart (hooray Ted!). Unfortunately, I've been unable to see it because it uses d3.csv and I haven't been able to get the python web server to work (see piazza q621 for my comment from today https://piazza.com/class#spring2013/cs171/621). Ted has a workaround that he used to create his chart using a cloud-based IDE (cool!).

Took a look at the full dataset now in the repo. It seems we just have Lat/Longs rather than city,state,etc. Not a problem for our "dots" view, though this isnt really feasible to work on further until we get our python server

up.
If we still wanted to do a chloropleth view, the function that aggregates would have to convert lat/long to state. Google apparently offers an API for this.
http://stackoverflow.com/questions/13428284/get-city-state-from-latitude-longitude-via-jquery-ajax-with-google-maps-geocode
https://developers.google.com/maps/documentation/geocoding/#ReverseGeocoding
Should be fairly easy to implement a call/response-parse in python.
It might be extremely slow if we have to do 16K of those, though. And all the network traffic might cause stuff to break? I'll give it a go tomorrow but we might not be able to get data for this view.

Meeting 4/3/13

Met up. Ted showed me how to Brad how to deal with the localhost issue. Discussed visual layout. Discussed strategies for implementing details on demand and brushing/linking.

Details on demand ideas:
Tooltips on map, line charts (already have for line chart). Add a filter so users can select which word to see. Possibly add a bar chart aggregating all tweets by word- filter this by the day when the user clicks on a given day in the line chart.

Multiple views:
Have both map and time views on same vis. For time views- add a button to switch between days aggregation and hours-of-day aggregation view

Brushing / linking
If we use the 'dots' view for our datamap, we do this: if a user mouses over a day in the line chart, all the tweets from that day on the datamap light up. Dependent on getting the map executed. If this is not possible, we can make the line chart a stacked bar chart (individual bar sections = individual curse), and then co-highlight between this and the aggregated bar chart displaying differences between curse words aggregated over all days

Maps
Some potential issues foreseable with our maps. 'Dots' view- possibility that 16K datapoint map may take too much memory, load time. Possibly- if d3/datamaps prove unworkable, could we use the googlemaps api? Also, this has the analytical problem that tweeting problems will be significantly biased by population, with no clear way to correct for this (since we are displaying individual tweets). However, we may have to go with it if chloropleth not achievable. The cholopleth offers the option to correct state tweet counts by control tweet count (or state population). Chloropleth execution depends on being able to reverse geocode tweet data. This involves a callout to the googlemaps api, probably from a python script, and parsing the response. Will this consume too many network or hardware resources to be feasible? Ideally, the chloropleth would be the main view, and we could offer an option to switch to the dots view as a details on demand feature so users can see absolute tweet data at tweet resolution.

Plan going forward:
Ted is going to work on adding interactivity features to his tweets-vs-time views according to the ideas above, in order to satisfy the project specs. Brad is going to work on maps, now that he can load pages properly.

Folder organization / code organization
(discussed this so that we can both work without running into each other)

1. website
    a. index.html
    b. index.css
    c. js
        i. brad.js
        ii. ted.js
        iii. pageload.js - listener to call our separate js scripts and draw things in proper order
    d. data
        i. raw_tweets.csv
        ii. processed_version(s) of tweets

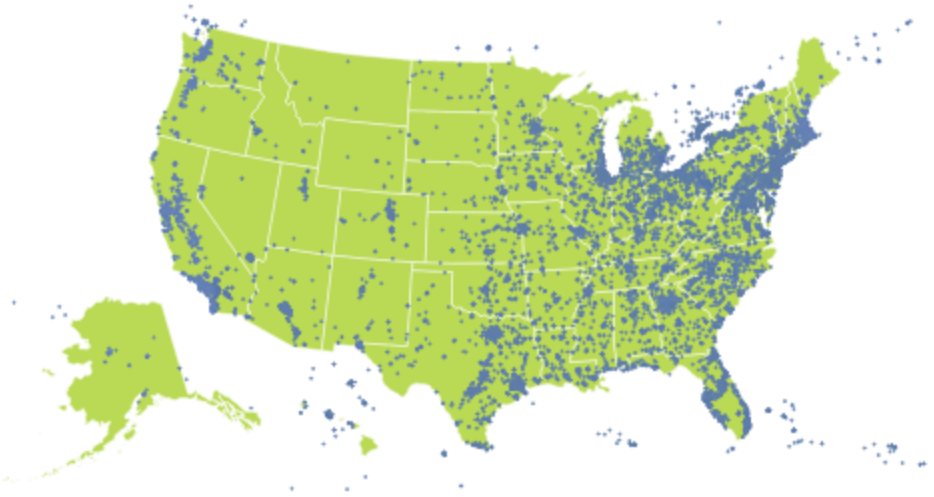
Brad 4/4/13

Ted- I reorganized our folders according to the plan above. I moved your code into a script (but did not delete anything in index.html- just commented out and made some additions. I also added a starter script to me that draws a map (empty for now). I have not added a CSS sheet, or moved any of the date. Hopefully none of these changes screw anything up for you!
Update:
Added code to brad.js to draw the dots view. Loads somewhat slowly but it works!

Ted 4/5/13

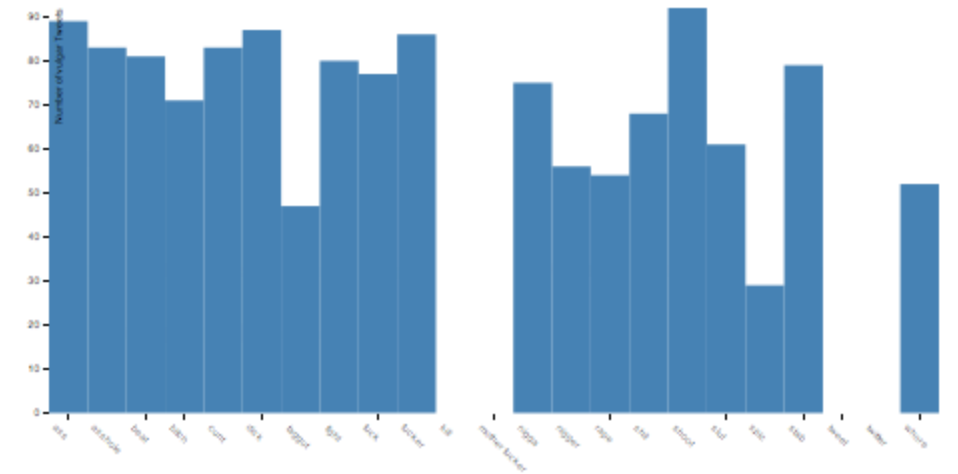
Hey Brad, nice work on the map. It looks good and I'm finding that it actually loads fairly quickly, especially given the size of the dataset. I've implemented a bar chart that can be accessed by clicking on the points in the line graph, so I think this should satisfy our details on demand requirement. Right now it only displays the raw Tweets, but I'm going to update so it displays the normalized data and then (hopefully) add a filtering mechanism so the user can switch between those views. I also reorganized the code a bit further, but made the appropriate path changes so that everything still works. I got rid of some of my scratch files, but let me know if I deleted anything you still need.

Vulgar Tweets by Location



Vulgar Tweets per Day

Vulgar Tweets on 2013-03-25



Brad 4/5/13 (udated 4/5, I did push the code to github)

Excellent work on the details-on-demand!

Some things I've been working on:

Reverse Geocoding:

I wrote a little script while I was at work to reverse geocode our data and append 'state' to every tweet. This works, but is currently throwing a unicode error (which I can fix). However, the query limit for the api turns out to be lower than I'd thought at first from API docs (only 2,500 per day!). If I can get something working quickly, I can knock out a cholorpleth (on a subset of our data) based on my scratch work from earlier. I should be able to implement a button switching between dots and chloropleth view (call map.redraw() in my script and bingo bango).

Details on demand features for maps:

1) I added a tooltip to my map (pretty easy).

2) I also have been playing around with D3 trying to get a series of check boxes for each search term, with the hope that I could change the dot colors for the mapped tweets corresponding to the selected term. Started night of 4/4. I didn't originally push these changes because my checkboxes currently show up in a random place on the site (near your line graph). My script code also got a little messy. 4/5- cleaned up and pushed.

Actually, the first thing I'm going to do is add a class attribute onto my dots containing the date. That way we can brush/link between your line chart and the map. If they highlight on a day, one can select all elements with class .bubble and then class .day and set their style properties. I'm not clear on d3.brush, but I can read up.

I'm not sure who actually should implement a brush (don't want to mess around with your code), but I figure having the handle there wouldn't hurt.

Brad 4/6/13

I was having significant trouble getting the correct behavior out of my input elements. d3 would not display text correctly for buttons without putting them inside a span. Radio buttons don't work correctly when isolated in this way. I tried to use checkboxes, but I was having trouble getting my selections right to uncheck the boxes. Instead, I made the spans themselves into little buttons. I could then implement radio button like functionality on them by dynamically styling. I could also get my recolor to work correctly (though it is a bit slow).

However, I am recoloring based on the class of each dot (which I set to "/search term/ bubble"). This might make things difficult if we need the class to include the date as a selection handle for a brush. Instead, I might try to filter the data and re-render the map to display ONLY the tweets for the term a user selects. Then set the class based on the date. However, the render might be too slow. Have to see. I still have to learn about brushing, so I'll try to do that.

Update

I am having significant trouble getting a reduced dataset and re-drawing the map. I think I'm going to leave the class of each dot as the search term. Instead, in the interest of feature completion, I am going to think about how to implement the recolor as a brush, hopefully including your bar chart (I won't alter your code). First, I'm going to spend some time with this week's lab. D3 is giving me significant headaches.

I'd like to get a brush into the visualization by the time I sleep tonight. Once we have all the project requirements fulfilled, I can use tomorrow to work on the write-up, as we've discussed.

Update
I got the redraw to where it wasn't crashing my browser or drawing a million maps. However, I could not get it to filter the tweet data either. I tried creating a new data array, populating it with a subset of tweets, and calling setting map.options prior to calling map.render(), as per Piazza question:
https://piazza.com/class#spring2013/cs171/701

But it is currently just drawing the map again, but with all the tweets still there. The map redraw also takes a really long time, making it less user friendly as an interaction feature. I commented out, and re-enabled the dot recolor.

Chloropleth (see image below)
In the meantime, I got my reverse_geocoder to work. I reverse geocoded the first ~2300 tweets (until I hit my daily query max). I then aggregated tweets by state to make a chloropleth. I pulled state population data from Wikipedia:
http://en.wikipedia.org/wiki/US_states_by_population

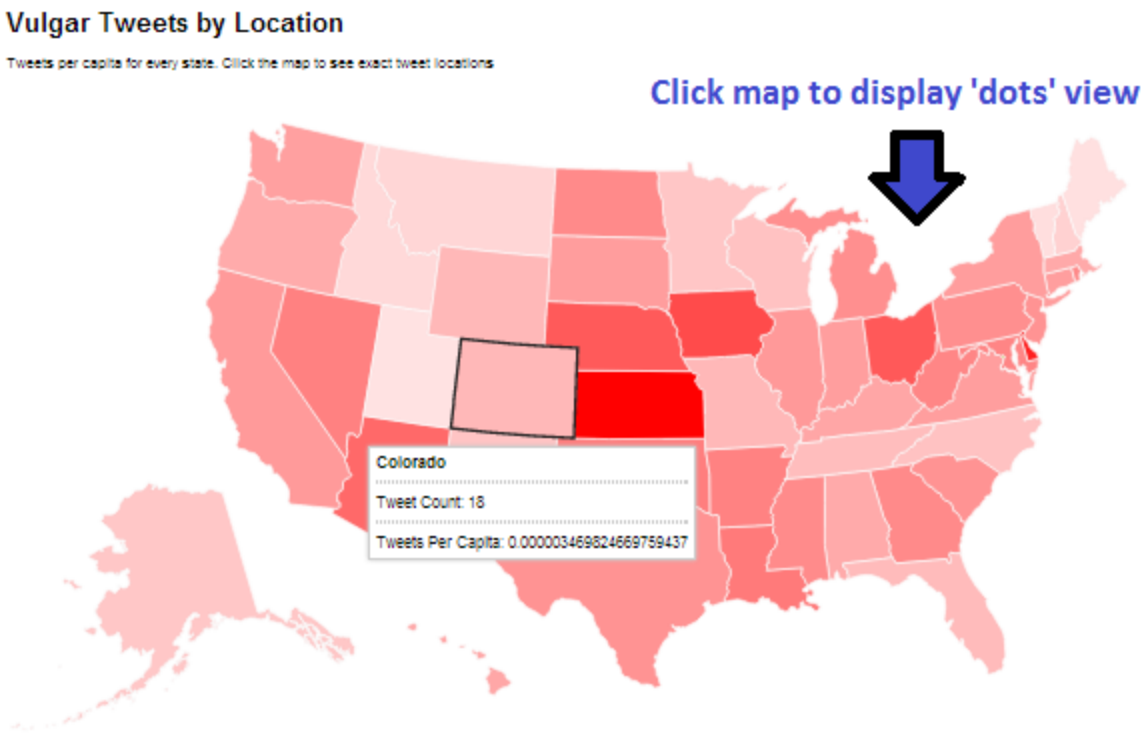
I used the population estimate for July 2012 and divided my tweets for each state by state population to get a tweets/capita number for every state. I'm very happy about this, because the plot of tweets by location was pretty clearly biased by population.

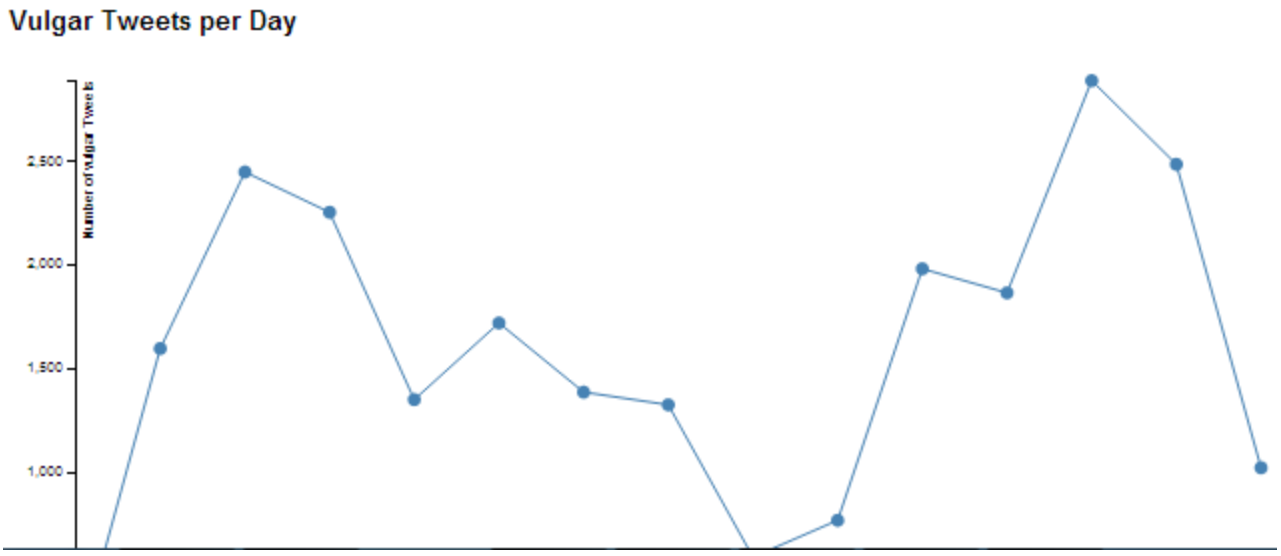
I threw the data into index.html in a hidden textarea (sorry, I know it's ugly but messing with d3's loadings would have delayed). I then used the data to display a chloropleth.

I made the chloropleth the default view, and set it up so that the user could click the map to switch to the 'dots' view, as a details on demand feature. I still have to fix the dots view so you can go in the reverse (get the chloropleth back). So that's one todo.

It also occurs to me, as I write this, that I left our control terms in there when I counted tweets. So really my current chloropleth is displaying which states tweet more frequently than average, instead of which ones tweet more violently than average. That's a quick fix (one if condition). I'll see to that tonight at some point.

I'm going to make a good effort to get the dots map redraw to work because having a filter feature is called for in the pset. Still not entirely sure what to do about brushing. I'll have to think about this.





TN 4/7/13

Awesome work with the chloropleth map. I think that loading the page with that view and then allowing the user to switch to the raw bubble view is a good idea. I also like the tooltip you added to the bubbles. I'm working on adding a control-normalized series to the line graph and updating the corresponding bar graph so that it displays normalized data. As for brushing and linking, I was thinking when the users mouses over a tweet on

your map, we could highlight the corresponding date on the line graph. I'll work on implementing that after I get the normalized data views working.

BT
Hmm, I was thinking the reverse- when a user mouses over a date on your line graph, all the tweets in the dots map would be highlighted. I suppose either direction works (tweets -> line point or line point -> tweets). I was thinking linepoints->tweets because then you highlight an attribute (date) and see all the data said attribute applies to. In the reverse case, you highlight a datum (tweet) and then see an attribute that applies to it, but which also applies to many other tweets. Again, I think either works and I'm totally open to reasons for going in the other direct. One such reason: a technical issue with linepoint->tweets was how to select the tweet dots. I am currently using the search term as the class for each tweet dot, not the date (to implement my search term recolor). This is why I was trying to add a filter to the data by search term, instead of recoloring- so I could reserve the class for packing in the date, giving you a handle to the dots. We need a filter feature in our project anyway, according to the spec, so I'll continue trying to get that to work.

I will be on Broad gchat whenever I am working, if you want to discuss anything. Feel free to text anytime as well (978-877-2725). I just want to make sure nothing I do messes you up.

Kudos on getting the normalized data implemented. Big big upgrade in giving our viz analytical accuracy.

TN

Ah, right. I think your suggestion makes sense. It does seem more reasonable to have the dots on the map highlighted when someone mouses over a data point on the line graph.

BT
I got my search-term-selection reimplemented as a filter/redraw instead of a recolor. It actually doesn't take that long to redraw the map! The project spec requires a filter, so this takes care of that. I think it's also better than a recolor since previously, some tweets were getting recolored but they were *behind* other tweets, so you couldn't see them. This is clearer, and still has good performance. I need to implement an "all" button, so you can get the full map back. First, though, I have to grab the date for each tweet in my csv-parse code and then pack it into the class for each svg circle, so we have a handle for linking.

With the filter-by-term and the proposed highlight-by-date, users can carve up the geo-plotted tweets by both our variables now, which offers nice interaction.

It would be really nice if the search terms were a global filter, that not only hit the dots map but altered the chloropleth, line chart, and bar chart as well. However, I don't think that is in any way achievable at this point (for ex., all the chloropleth counting math is being done offline in a python script- it would be time consuming to migrate this into our page js). But it might be nice to mention in the project writeup. From the spec, it sounds like they grade the thing holistically, and design intent is weighted higher than implementation. Just a thought.

Also: Interesting observation. People use swears WAY more on twitter than they use "twitter" or "tweet". I

thought it might just be because we started collecting these words mid-week, but if you look at the bar charts for the most recent days, these are always among the least-used words. Given that we chose them because of that article saying “twitter” and “tweet” were within the top 50 most used words, that’s pretty nuts. People love cussin’!

Update- implemented change in class of tweet dots from ‘bubble /search term’ to ‘bubble /date/.’

Update- I created a new feature branch in github. I implemented a link between Ted’s line chart and my tweet dots. Did this in a separate branch because it involved altering Ted.js
Currently does not work if somebody filters the tweets by search term first. Looking into this.

TN -

Nice, the filtering stuff is awesome, well done. I pulled your feature branch and checked out the linking between the line graph and the dot view, looks good. I’ll keep working on this from my end to see if I can get it to work post-filtering.

BT-
Just figured out the filtering thing (haven’t pushed yet). Since I removed/redrew my whole map, the date classes didn’t get put onto the new bubbles. My click interaction to change back to chloropleth also got clobbered by the redraw. I reset both. The chloropleth part works, but I’m still not getting the selection right for highlighting. Figure I can get this soon. I think the problem is the svg is getting put in a different location in the dom, by d3 append. I think a better selection would fix? I’ll push my partial solution first, then work on that. I’ll push to the feature branch and issue a pull request.

Then, I’m going to add an “all” button to my filter so the user can get the full tweets back.

After that, I have to fix one thing in the chloropleth (exclude control tweets). Then I think I’m done? I can work on writeup stuff at work tomorrow. Don’t tell my boss.

Update- got the linking to work even after the filter (my classes were being set from the wrong copy of the dataset). Added my all button. Pull request issued. Ted, if it looks good to you, can you merge at your convenience? Signing off for the night.

TN - Awesome, the highlighting works great. Nice work. I pulled and updated with a little bit of new tweet data. I think we’re in good shape. What do you think about meeting up today to nail down the final details and possibly record the video demo?

BT - I think meeting up would be great. I have to do one last bit on the chloropleth (see above), and then I’m going to start summarizing things for the writeup. No need to wait on that. We can meet up at any time today, except 1130 - 12. The stuff on my gcal in the afternoon is flexible.