«Under the pine trees of Mt. Etna», how does a volcano influence the life of people around it?

Devastating eruption of 1669 that destroyed part of Catania.

by Gabriel Hase and Sebastian Spiegl

Table of Contents

Initial Project Proposal (part of homework 4)

Tentative project title: «Under the pine trees of Mt. Etna»

Team Members:

Name	*E-Mail*	*Skills*
Gabriel Hase	gabriel.hase@gmail.com	Very comfortable with Web-Development, JavaScript (CoffeeScript), CSS (Less and Sass), HTML5, Ruby on Rails, Deployment (Github, Heroku, Amazon Cloud Services). I am doing this professionally.
Sebastian Spiegl	s.spiegl@me.com	Very comfortable with python, regex relatively comfortable with HTML5 and jscript.

Research Question(s):

Picking up on the volcano eruptions history that I (Gabriel) did in project 1 this project will be about the influence that a volcano has on the people that live around it. We will look at the Volcano Etna in Italy which is one of the Decade Volcanoes. The concrete questions we will answer in order of importance are:

1. How many people and towns are inflicted by volcano eruptions of Etna (deaths, damage, etc.)? Now and historic.
2. How do people feel about Etna (sentiment)? Now.
3. Where are lava flows going through? How close (or too close) do they get to towns?
4. How does Etna influence the real-estate prices in the region? Recent changes.

Motivation:

The first project about volcanoes studied eruption histories and recurrence patterns that might be useful for stochastic prediction of eruptions. It became clear that volcano eruptions are very complex and interrelated with other geological processes. Although this seems certainly an interesting topic, it becomes extremely involved with geological sciences and also it leaves out a very important aspect of volcanoes: their symbolic nature and the way people live around them

and try to make sense of them.
This project is motivated by the work in the first project and will be about how people live with volcanoes. It should be much more tangible and give ample room to turn the interactive visualization of project 2 into a coherent data story in project 3 including rich imagery and (possibly live) video.

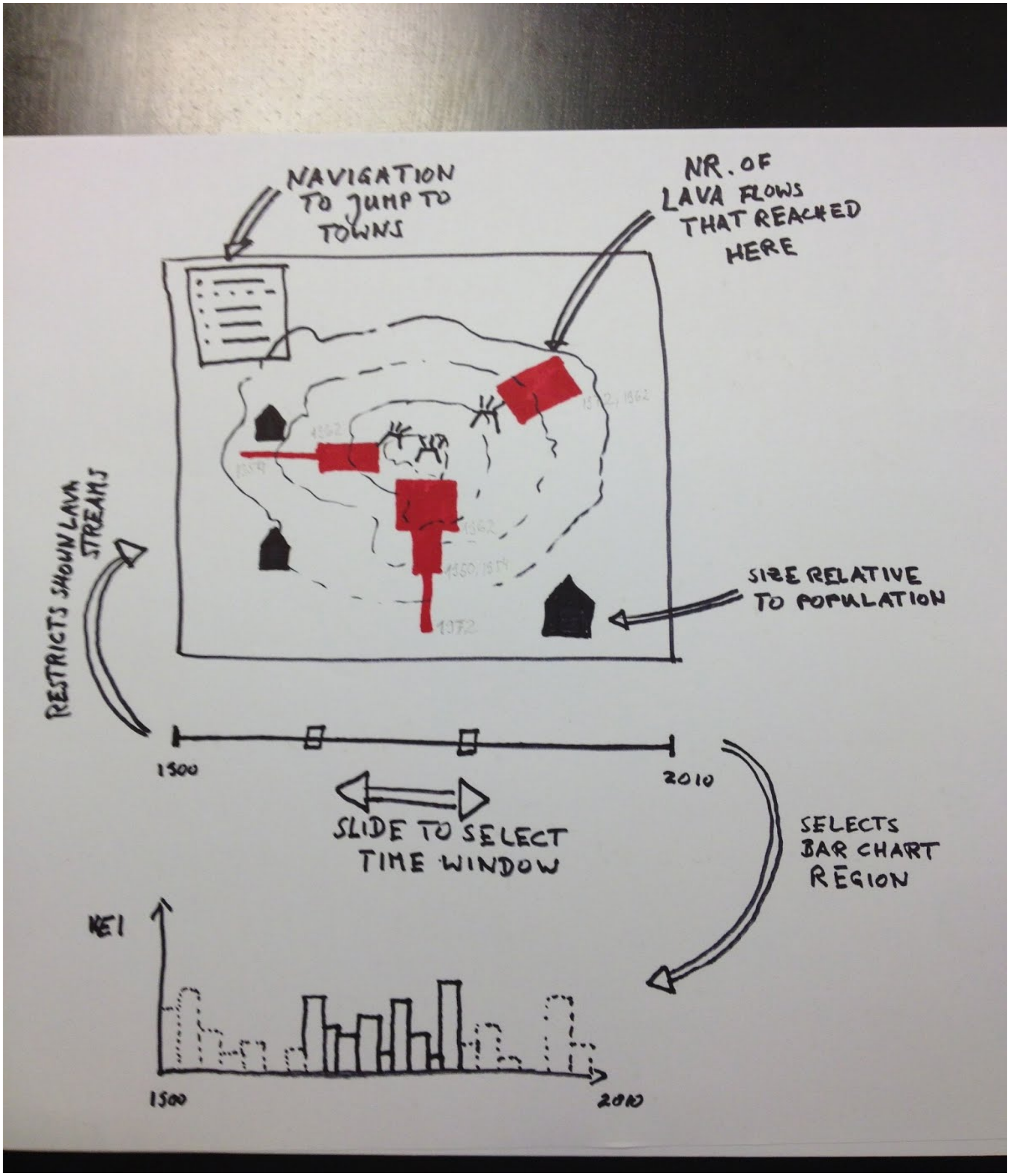
Data:

- Eruptions history with VEI and geolocation from project 1
- Geolocation Services (http://www.geoplugin.com/webservices/extras) and Google Maps for towns around volcanoes
- Data about towns around a volcano (probably have to be gathered by hand)
- Twitter API

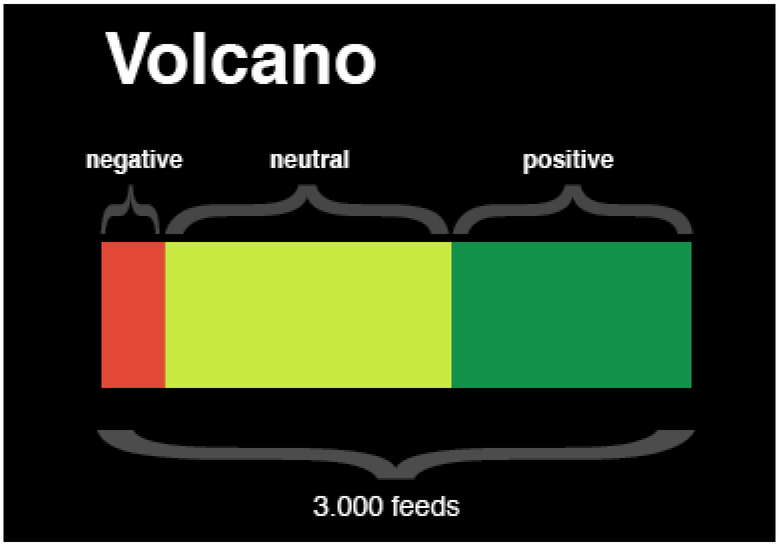
Visualizations:

Map with towns around the volcano Etna

The sketch on the next page shows a visualization that uses brushing and linking on a map and a bar chart. The central interaction point is the timeline from 1900 to 2010 where the user has two handles to see a certain timeframe.
The map shows a volcano with its craters the surrounding towns and where lava streams have reached. The thickness of the lava streams (red) indicates how many times (in bins of years) a lava stream reached a certain location, e.g. 3 miles north of the volcano (this is inspired by Charles Minard's visualization of Napoléon's march). The towns are represented with icons and the size of the icons is proportional to the population. A user can use the in-map navigation (top-left) to jump to the towns. It would be very nice to show a little profile of a town when a user jumps to a town (but due to time limits this might be more something for project 3).
The bar chart shows the eruptions for the volcano in the given years by the summed VEI (strength) of the eruptions. Upon selecting a time interval the bars that do not fall in the selected time interval are slightly faded out (less contrast).
The data for the towns around Etna is probably most effectively gathered by hand. E.g. this site has some information http://www.citypopulation.de/Italy-Sicilia.html. The towns around Etna are most effectively inspected with Google Maps.
The data for the eruptions is already present from project 1.
The data for the lava flows after 1968 can be gathered roughly by hand using the following page http://www.volcano.si.edu/world/volcano.cfm?vnum=0101-06=&volpage=var. It might make sense though to write to the Etna observatory if they have the data in a slightly friendlier format. As an alternative one could simply leave the lava flows and draw a circle around the crater proportional in circumfer to the strength of the eruption (VEI).


NAVIGATION TO JUMP TO TOWNS
NR. OF LAVA FLOWS THAT REACHED HERE
SIZE RELATIVE TO POPULATION
RESTRICTS SHOWN LAVA STREAMS
1500
2010
SLIDE TO SELECT TIME WINDOW
SELECTS BAR CHART REGION
VEI
1500
2010

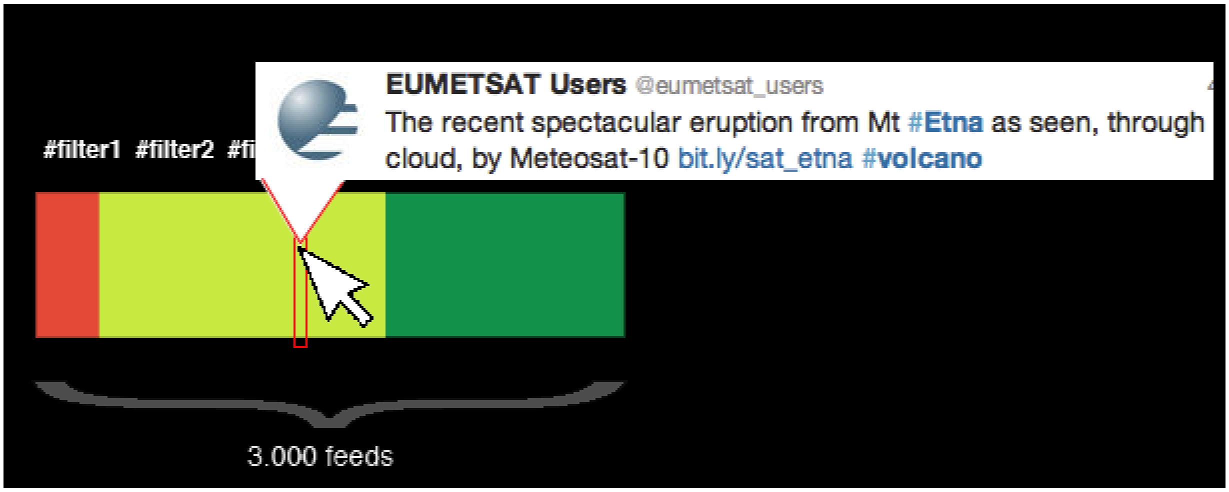
Twitter sentiment



The sentiment analysis graph displays a defined number of tweets containing #volcano AND #etna and a third freely definable filter passed through sentiment analysis as a stacked bar chart. Sentiment analysis returns a value between -1 and +1 to show its polarity. As sentiment analysis is not 100% accurate a threshold of -0.7 for negative and +0.7 for positive is used. Everything in between will be considered as “neutral”. Red will be negative tweets (below -0.7), yellow shows neutral tweets (above -0.7 but below +0.7) and green are positive tweets (above +0.7). To overcome color blindness issues there will be selectable color schemes.

We will provide an interface to enable additional filtering. All tweets will be available through an array and grep will be used (http://api.jquery.com/jQuery.grep/) to return tweets that match a certain filter. This filtered elements will then be used to show sentiment again.

Additionally, a “tooltip” will display tweets of a certain sentiment in respect to mouse position interpretet as sentiment valu



Research:

Decade Volcanoes: http://en.wikipedia.org/wiki/Decade_Volcano
Great Video Footage: http://www.wired.com/wiredscience/eruptions/
Might be worth asking him for data: http://www.wired.com/wiredscience/author/erikvolc/page/1
Maps: http://mapbox.com/tilemill/
Twitter Colcano Alert: https://twitter.com/VolcanoAlert
Effects of volcanic eruptions on env. and health http://www.ncbi.nlm.nih.gov/pubmed/18063533
Etna Observatorium: http://www.ct.ingv.it/en/
ETH Zurich with some links: http://www.swisseduc.ch/stromboli/index-de.html

Reflections on Initial Proposal

Getting geolocated data on the eruptions (e.g. the Lava flows or explosions) is very difficult. There is no public database for this. Maybe we can get the data from an observatory, but we certainly should try to make the visualization work without this data. (It might also be a good addition for Project 3).

Bringing together the map and the Twitter sentiment analysis seems also very difficult. In one direction Twitter is too young to give any reasonable data for the eruptions of the last 100 years. In the other direction the Tweet geolocation data is probably too fuzzy to be useful on the relatively narrow region around Mt. Etna.

This leads to the conclusion that it is probably a good idea to split the requirements for Project 2 into 2 visualizations:

- The map that allows **brushing and linking** and can give **details-on-demand** with a **tooltip** to give additional information about the towns around Mt. Etna.
- The Twitter sentiment chart that allows **drill-down/filter capability** for certain keywords.

These two visualizations fulfill together all the requirements for Project 2 and give a viewer a nice way to get a sense of what people think about Etna (Tweets) and what the eruptive history is (Map and Barchart).
Also the visualizations provide a good framework on which we can build in Project 3 to come up with a data story. So the goal in Project 2 would be to get these two visualizations running with the basic functionality and then in Project 3 having the possibility to refine them and bring them together in a data story.

Project II Final Proposal (part of homework 5)

Project title and teammate: «Under the pine trees of Mt. Etna» (Team: Gabriel Hase, Sebastian Spiegl)

Research Question(s):
The primary research question is: "How do people feel about the volcano Etna?". This question is directly connected with the secondary research questions:

- How (and how many) do people live around the volcano Etna?
- How do eruptions affect the people living around the volcano Etna?

The research questions take a different direction than the ones that were developed in the first project where the research was about patterns in the eruptive history (and possible stochastic prediction parameters). The direction change is due to the fact that we want to explore the more subjective influence that a volcano has on people as compared to the strictly scientific and geological effects.

Motivation:
Volcanoes are curious beasts: on the one hand they bring death and destruction through their unpredictable eruptions, on the other hand the soil around them is fertile and they both attract and fascinate people. We want to find out what people think about volcanoes today at the concrete example of the very active volcano of Mt. Etna. The work we do on Project 2 should provide us with insights on this question and an implementation of interactive web visualizations. In Project 3 we can take up this topic again and bring together the subjective findings from Project 2 with the scientific findings from Project 1 to tell a coherent data story about the volcano Mt. Etna. Our motivation is eventually also to find out ways how to tell an engaging story with data visualizations, similar to what the New York Times did with the Snowfall project (of course not in size only in inspiration). Volcanoes provide an excellent topic for this, in their nature and in their media coverage.

Data sources and technical Process:
To show the eruptive history of Mt. Etna we can use a subset of the data collected for Project 1. The data is in csv format.
To show the towns around Mt. Etna we will use the following Geoservice for the geolocation data:
www.geoplugin.net/extras/nearby.gp?lat=37.734&long=15.004&limit=50&radius=100
We will then look up the towns by hand (Google) to get additional information on them like the population, area, and a picture. The data will be saved in either csv or json.

Twitter Scraping:
To get a large amount of tweets we used a commercial twitter archive api provided by topsy.com. The request to gather tweets that contain etna or vulcano is sent to http://otter.topsy.com/search.json? with the following parameters concatenated to one URL.

```
?q=vulcano%20OR%20etna  #hardcoded searchterm
&page                   #inline concatenated with variabel page as +str(page)+
&nohidden=0             #hardcoded 0 for no simmilar tweets
&perpage=100            #hardcoded 100 results per page
&allow_lang=en          #hardcoded en for english tweets only
&apikey=4KS6YFR3VSP6HJUXM4KQAAAAAAIY6ICNMNIQAAAAAAAFQGYA   #apikey
```

As the amount of results per page can not exceed 100 tweets, additional tweets can be accessed by a page parameter (&page=*int*). A for loop takes care of this

```
for page in range(1,10):
   url = "http://otter.topsy.com/se....+str(page)+....AAAAAAAFQGYA"
```

As the resulting JSON holds results of different social media platforms a filter is introduced to only process twitter related results. The result holds information to the platform inside the "topsy_author_url" key as a link to f. the twitter'ers account. We simply filter for results that contain "twitter.com" in "topsy_author_url".

```
for i, tweet in enumerate(json.loads(data.read())["response"]["list"]):
      if "twitter.com" in tweet["topsy_author_url"]:
        # begin processing
```

As we will also provide a filter list based on the most used words we decided to use stemming to generate a comparable list. Stemming is used to reduce inflected words to their stem, base or root form. To use a text for stemming effectively all chars beside A to Z or a-z or .!() should be eliminated. Directly after the filter the stemming sequence is generated.

```
s= "".join(re.findall(r"[A-Za-z0-9.!() ]*", ss))
t = parse(s, lemmata=True)
```

The stem is made up of " " separated descriptor blocks(separated by ","). It gets split into descrpting blocks. As some "words" contain "http" they are filtered, as well as empty stemms. The 5th element of each descriptor is the stem, it is appended to an interims list which is joined to a string after processing.

```
 for e in str(t).split(' '):
            if not "http" in e:
               if e.split('/')[0]!="":
                  stem_tweet_.append(e.split('/')[4])
stem_tweet = " ".join(stem_tweet_)
```

the rest of the tweet data is relatively straightforward processable

```
# date conversion from epoch to a more parsing friendliy form
tweet_date=datetime.datetime.fromtimestamp(int(tweet["firstpost_date"])).strftime('%Y...
                    ….-%m-%d:%H:%M:%S')

# as some apostrophes come in as non unicode chars they are replaced automatically
tweet_text=tweet["content"].replace("&#39;", "'")

# sentiment analysis of the hole text
tweet_sentiment=str(sentiment(tweet["content"].replace("&#39;", "'"))[0])
```

finally the data that was appended to our interims list gets converted to a dictionary first, then it is dumped into a json format and saved as a file.

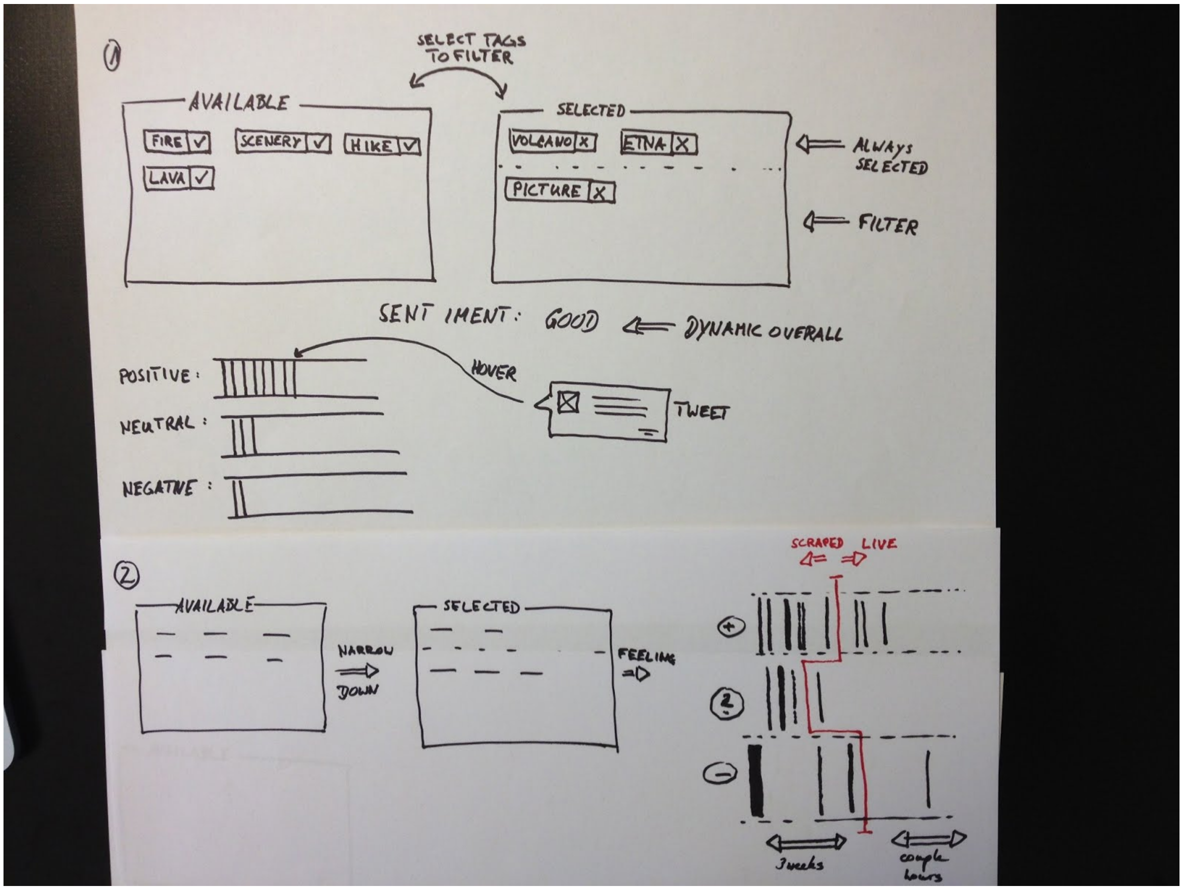
```
          data_cont_interrims.append([tweet_date, tweet_text, tweet_sentiment,...
                                        ….stem_tweet])

data_cont["tweets"]=[ row for row in data_cont_interrims ]

f = open("twitter_feeds.json", 'a')
f.write(json.dumps(data_cont) + "\n")
f.close()
```

Potential visualizations and technical process (including sketches):

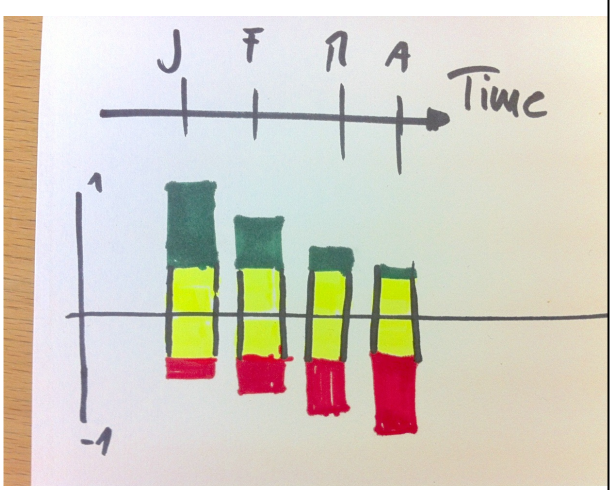
Twitter Sentiment

To visualize the primary research questions we use a Twitter sentiment analysis. The python scraper will gather Tweets matching the keywords "Etna" and "Volcano" over the next couple of weeks. We will do a simple word count analysis over the Tweets and find out which are other frequent keywords in the collection. The user will get the chance to use these keywords to further narrow down the selection ("Etna" and "Volcano" can of course not be de-selected). The filter will update the Sentiment element which is basically three rows, one for each of positive, neutral, and negative. The bars are colored and the color encoding will be allowed to change for color-blind people. In the second sketch we added time and algorithm confidence to the data represented by the Tweets. The spacing between the bars is given by the time of the tweet, the thickness of the bar by the confidence of the sentiment analysis. A nice addition would be to separate the Tweet bars into an area for the scraped Tweets and an area for live tweets that come in during the running application (live API). If this is too time-consuming for now it would also be a nice addition in project 3.
The technology for this visualization will be JQuery, D3.js, and possibly angular.js for the live

updates.



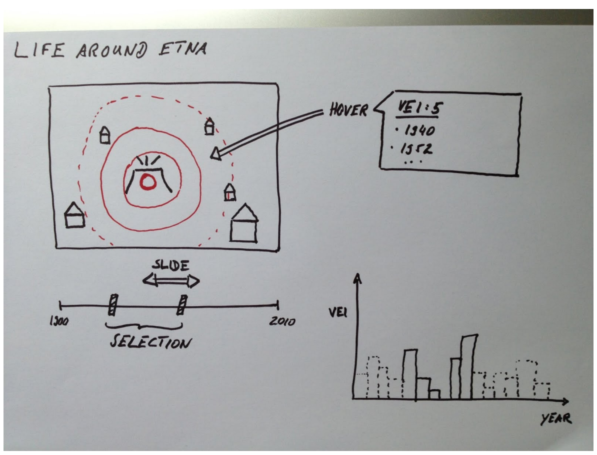
Sketch 1 (top) and Sketch 2 (bottom): Selection of Tags and according sentiment in matching Tweets, archived and live.



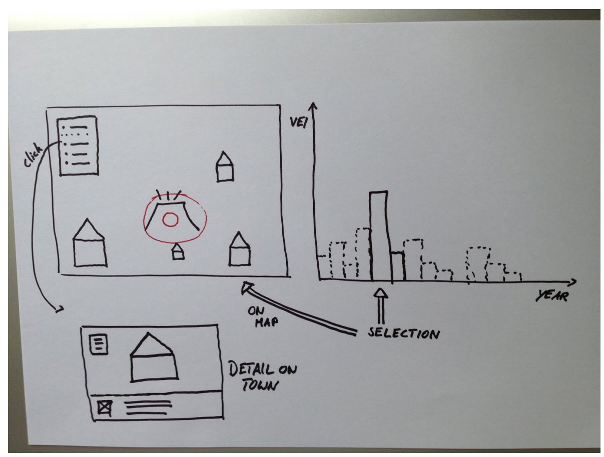
A difference in time after an eruption for a filte combination can be shown with a timeline of stacked bars. The forme positive tweets, mostly by vulcano enthusiast, make way for the negative tweets by affected people.

Map

To visualize the secondary research questions we will implement a map visualization together with a bar chart that shows the eruptive history since 1900. The map supports showing the towns around Etna as well as detail-on-demand data like the population of a specific town. The bar chart shows the eruptive history since 1900 and the VEI of the eruptions. While sketching the visualization I found that sketch 2 is actually better since the brushing (dragging over the bar chart) is implicit rather than introducing yet another control (time slider). The bar chart is linked to the map so that on selecting a range of eruptions on the bar chart there will be circles on the map corresponding to the VEI (strength) of the eruption giving an impression of the area of influence that an eruption had (not exact though, the correct shape would be much more complex than a circle, the data is not publicly available).
The map should be implemented with mapbox (www.mapbox.com). Mapbox has very nice options to do beautiful maps (much better than GoogleMaps) and at the same time there is good documentation on how to use D3.js to draw on a mapbox map.
The bar chart should be implemented with D3.js. There is plenty documentation how to do bar charts with D3.



Sketch 1: Time slider for brushing and linking of VEI data on bar chart and map.



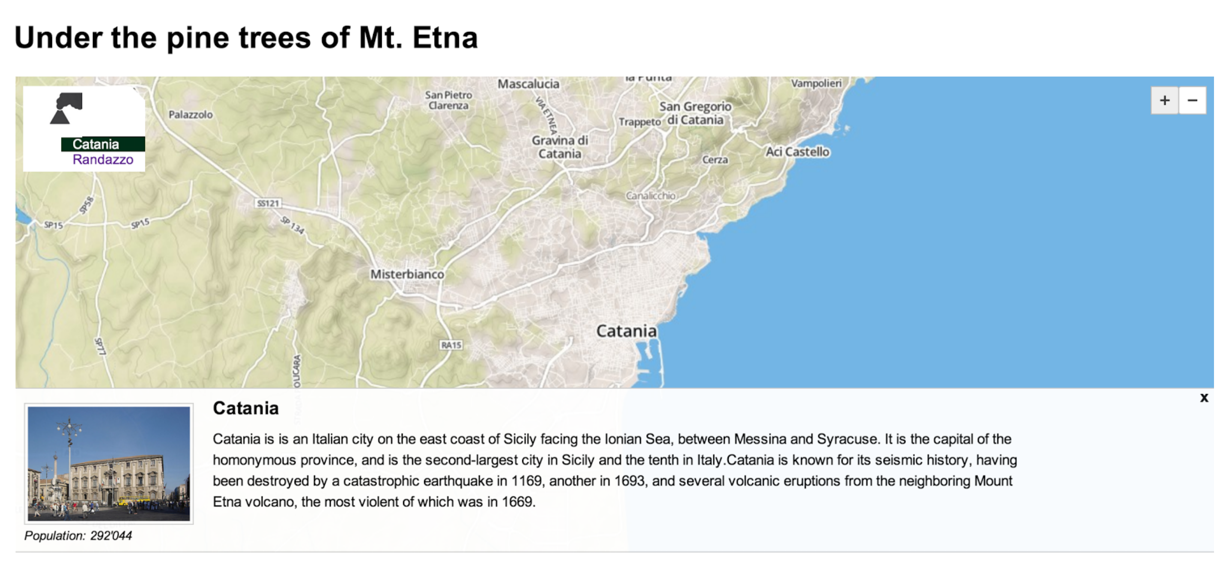
Sketch 2: Using bar chart directly for the brushing. Detail-on-demand for towns.

Map Prototype

To quickly get a sense of what the map will feel like, I set up a mapbox account and downloaded the mapbox.js script to control the common map features like zooming, panning, etc.
The first few steps were actually done in the mapbox web interface. I placed some pins over the towns around Etna. Yet this didn't quite work conceptually. I want to show areas where people live. The pins feel more like hotspots and are actually redundant since the towns are anyway labelled on a map. A pin doesn't add any more information.
So what I actually want is to give the user a navigation that gives the ability to drill-down into regions around Etna. I do this with 2 elements:

1. A navigation in the top-left of the map that shows the cities which have additional details.
2. An info-panel that is laid over the map and tells the user what one can do on this map (this panel is also used to show the detailed information on towns).

For the prototype I got data for 2 towns. In total there are about 10 towns of interest so we should be easily able to get this data by hand.

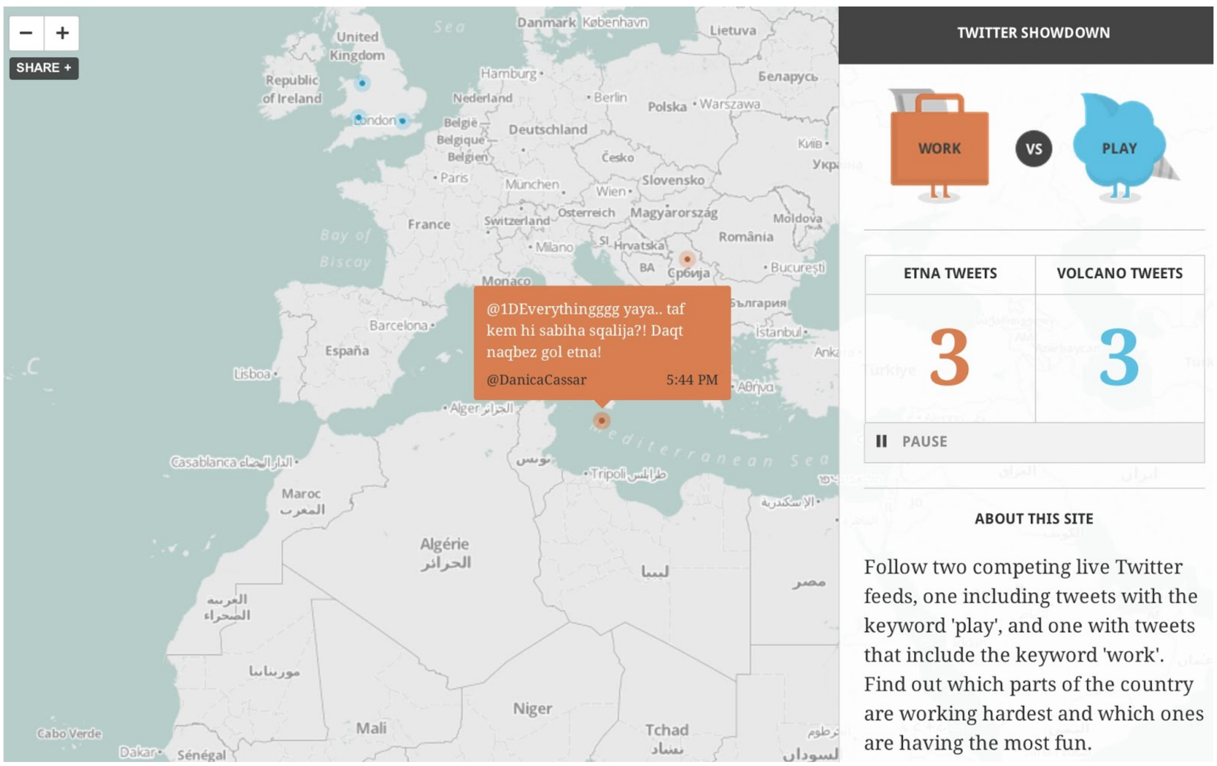


Detail-on-demand view of the town Catania

The technical setup for the map was actually very quick with mapbox. What took a lot of time was styling of the navigational elements and implementing a reasonable behaviour of the navigational elements with respect to map interactions (zooming, etc.).
The next step will be to find out how to use D3 to draw on the map and bring in the eruption data in the form of circle areas proportional to the eruption strength (see project proposal). Before doing this I want to try out if there is a way to bring in the Twitter geodata into the map. So far we figured that the geolocation is too imprecise to be useful. I found a good mapbox template (http://mapbox.com/map-sites/live-tweets/) that uses the Twitter live API, so we can use this to test if this is really true.

Mapbox Template for Twitter live API

As stated before we want to find out if the live Twitter API provides any useful data to show on the map. To do this we quickly hacked a mapbox template (http://mapbox.com/map-sites/live-tweets/) to work with our keywords. The below graphic shows a sad result: There are only 3 Tweets (in a span of around 3 hours) and 2 of them do not really have anything to do with Etna (they just have the keyword in there).

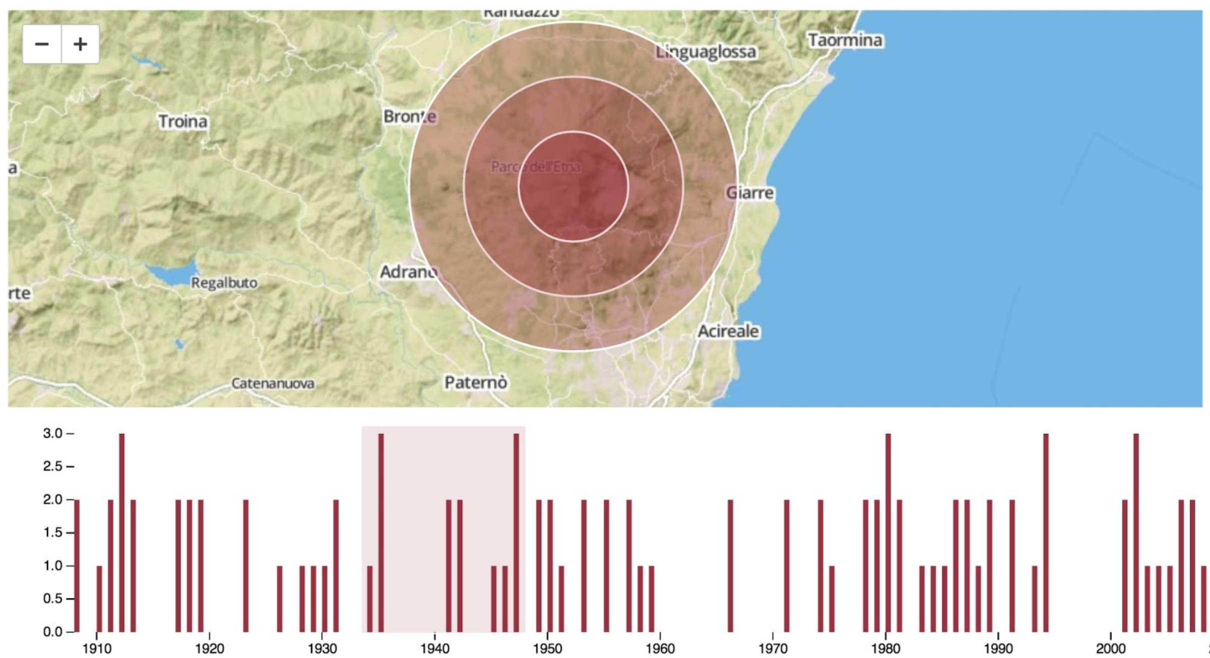


Hacked mapbox Template: Instead of “Work” and “Play” looks for the keywords “Etna” and “Volcano”.

So this little experiment shows that we cannot use the Twitter live API to get any interesting results for Etna. We will have to rely on the topsy service described in the proposal.

Drawing on the mapbox map (part of homework 6)

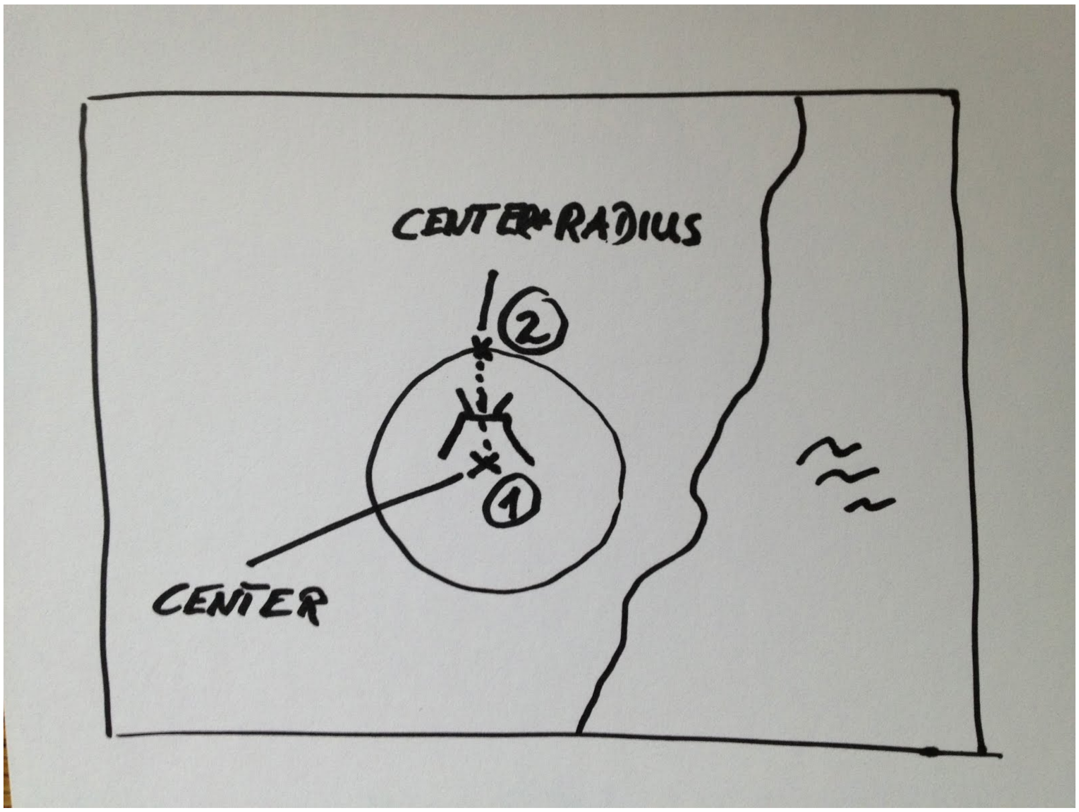
In homework 6 I did a prototype for the brushing and linking functionality described in the proposal. The scope was to draw a bar chart representing the eruptive history of Etna since 1900 and allow brushing on the bar chart. Once some bars are selected in a brush the values in the bars should be drawn to the map in circles with a radius that represents the area of influence of an eruption. The result can be seen below:



Brushed area between approx. 1935 and 1948 is selected in the bar chart and drawn on the map (brushing and linking).

The bar chart was done with d3. The most difficult part was to get the brushing to work and get the needed variables to the callback (I used currying to have nicer separation of concerns, http://www.dustindiaz.com/javascript-curry/).
The difficult part was though to get the circles on the map. The difference to an SVG map like we did in class where the map is defined as an SVG path in pixel-space is that with mapbox we need to calculate the correct pixel positions to draw at at runtime and for each of a pair of geolocation values and changing on user interactions like zooming (geographical position stays of course the same but the pixel-position, i.e. radius, changes when a user zooms). I started out with the canonical tutorial for drawing on a mapbox map with d3:
http://mapbox.com/mapbox.js/example/vectors-with-d3/
Understanding the code wasn't easy, especially since it was written in a very hackish manner (look e.g. at the f.draw function in the original code). So the first step was to refactor the code to be more usable and understandable. I opted for a module pattern (singleton) solution for the svg

layer that contains the circles. The barchart code is the main code in this prototype. For the integration into the project code I will need to modularize this part too in some sensible manner. The most difficult part to realize coming from the example code above was that svg circles and paths are not the same. While the example on the mapbox page has a collection of points for which it can then calculate the projection to pixel-space a sphere is not a collection of points, but one point (center) and a radius. Unfortunately, the mapbox javascript API does not provide any means to project a radius from geographical coordinates into pixel-space (there is only projection for lon/lat points). To overcome this problem I had to implement the circle differently:



Calculation of radius with the center and one virtual point on the circumfer to the north of the center.

The sketch above shows two points (1) and (2).
(1) is simply the center of the circle which is the location of the crater of Mt. Etna. This point is available in lat/lon coordinates and I could use the projection method from mapbox to translate this geolocation into pixel-space for each possible pan/zoom combination (the code uses the draw method as callback on zooming and panning events).
(2) is a virtual point on the circle which is x kilometers to the north in lat points from the center. The amount of kilometers is calculated from the VEI (volcanic eruption index) which indicates the

strength of an eruption. In this simple prototype I gave each VEI point an additional radius of influence of 5 kilometers, i.e. VEI 1 is 5 kilometers and VEI 2 is 10 kilometers. To translate the kilometers into lat points I used the approximate method that 111 kilometers correspond to 1 lat point (this is very inexact, but works if the involved distance is small which is the case here). With the calculations above I get the lat/lon values for point (2). These can be projected into pixel-space using the mapbox API.
Once I have points (1) and (2) in pixel-space I can simply subtract their y-values to get the radius of the circle and draw the circle using svg.
The fill of the circles was on purpose done with a semi-transparent color so that overlying circles make the red color darker. Thus in the initial screenshot the center is the darkest region and we can see that this area has been most affected by the eruptions.
I also implemented the callbacks in a way that zooming and panning will not de-select any active brushes, i.e. the circles stay on the map until a user changes the brush in the bar chart. This was done since I think a user should be given the possibility to use the map to explore the areas of influence of different eruptions, e.g. really zooming in closely and seeing which towns or streets are affected.

As a side note that came up while working with d3, I wonder why most of the d3 examples don't use any code architecture at all. Everything I've seen so far is just one big file with instructions in a top to bottom manner. I took a quick glimpse into the d3 source and it seems to be written extremely functional. Maybe the lack of modularization in the samples drives from this kind of thinking? I would be very interested where the coding style of d3 took its roots.

Reflections on brushing and linking prototype

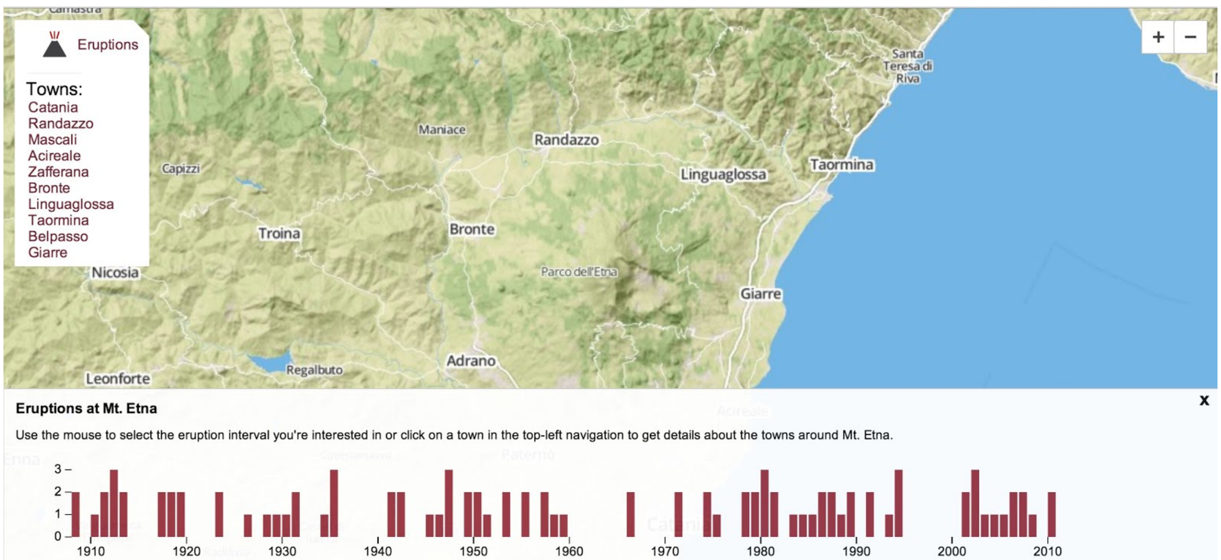
The interaction pattern of the brushing and linking works really well: it is immediately clear to the user what effect the brushing has. To make it clearer what the circles actually mean I will implement tooltips.
One difficult question for the integration into the main visualization will be where to place the brushing interface. The placement of the brushing interface, i.e. barchart, is not only styling but also a big part of the usability, i.e. conveying to the user what this element is all about. One option would be to make it part of the map interface and navigable through the top-left navigation similar to the towns. I will probably try out different options with css directly. Since the code is already there I think this is better than taking a pre-step on the sketching block.

A big problem that I found during inspecting the data is that the VEI index is a very bad indicator for actual area of infliction. The most devastating eruption in the 20th century was in 1928, destroying the town of Mascali to the east of Mt. Etna. The VEI of this eruption is though 1 thus very low. This is due to the calculation of the VEI: it calculates the explosiveness of an eruption and the amount of material emitted (http://en.wikipedia.org/wiki/Volcanic_Explosivity_Index). The destruction of town around Mt. Etna is though mostly given by Lava streams flowing down the flanks and not the explosions.
From my initial data source (http://www.volcano.si.edu/world/volcano.cfm?vnum=0101-06=&volpage=erupt) it seems that the lava volume is a better measurement. Unfortunately, it is not present for all eruptions. I could implement my scraper to get this measurement instead (or in addition) to the VEI.
Even better is the data from this source:
http://www.volcanolive.com/etna2.html
This page contains descriptive texts about all eruptions. It is not possible to automatically scrape this page for data and one would have to read it top to bottom and extract the data. This sounds like a lot of work, but it has one advantage:
Despite what one initially thinks, telling the area of infliction of an eruption is no easy task and there probably is not one objective and correct solution to it! From the map we can see that a circle from the center to the town of Mascali would include a lot of other towns as well. These towns were also endangered by the eruption since no one can really tell where the lava flow will go through exactly (could just as well have hit the town of Giarre). What then is the area of infliction of the 1928 eruption? Towns that are affected (Mascali) or towns that are endangered and would have probably evacuated by todays standards (a lot of them in 1928)? The answer to this question is not obvious although very very important for the visualization. Going through the texts to the different eruptions then does not seem to be such a bad idea if only to get a sense for what kind of beast we are dealing with. Maybe the kind of data gathered for area of infliction will look very different after reading more about the topic.
Since this is a lot of domain-specific work I am not sure whether I will have time to do this for project 2. But I definitely have to do it for project 3 since to tell a coherent data story it is crucial to know exactly what one is talking about.

Integrating the brushing and linking prototype

For the integration of the code from homework 6 I had to re-write the code in CoffeeScript (I like better to write CoffeeScript, but for the homeworks this is not allowed) and think of sensible modules for it. I went with 2 new modules (singletons), one for the barchart (eruptions) of the eruptions and one for the svg layer on top of the map where the circles are drawn (d3layer).

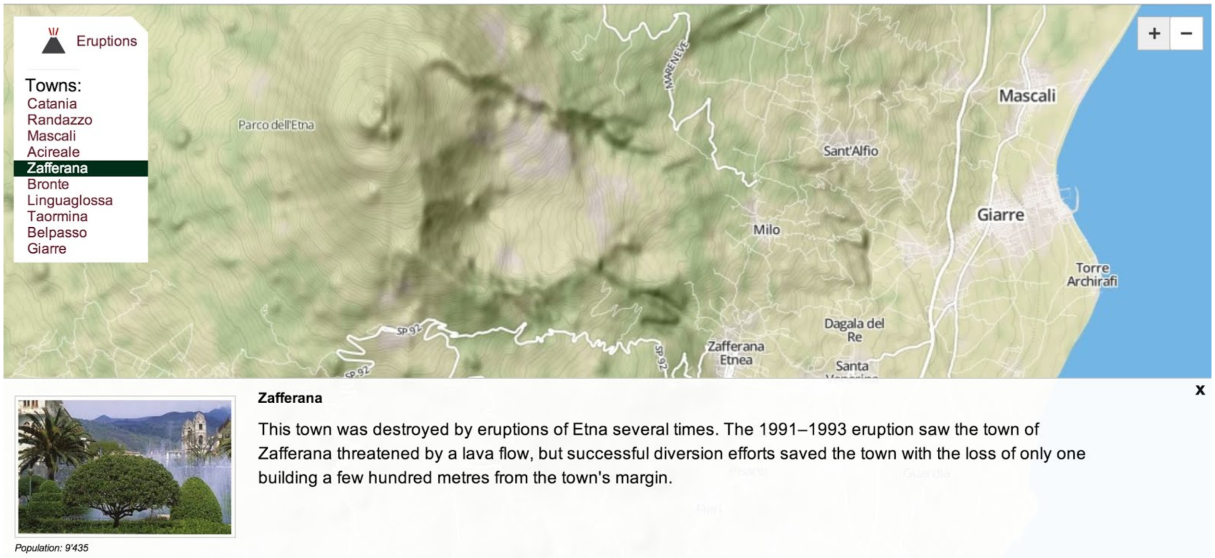
One challenge was to place the barchart on the page in a way that intuitively allows a user to interact with it and explore the linked areas on the map. I found that the best user experience was given when the barchart was actually placed on the map as shown below:



Barchart with all eruptions in the 20th century on a semi-transparent background over the map.

The barchart is immediately visible to the user when loading the page. It is situated on a semi-transparent overlay (the same that is also used for the descriptions of the towns). Upon navigating the page the overlay will toggle down. The user can though always bring it up again by either clicking a link in the bottom left or clicking on the volcano icon in the top-left navigation.

The image on the next page shows the selection of a single town. The overlays content is replaced with the detailed information on the selected town and the map zooms to the location of the town. If a user already selected a brush on the barchart the corresponding circles representing the area of influence will stay active. This makes sense to allow a user to see the eruptions and contrast them with the towns around Etna.



Selection of the town Zafferana with a description of the town.

I also chose to make the map the heading element of the page (there is nothing on the page above the map). The aim of this was to catch the attention of the user immediately and not distract him or her with additional content. To be able to give the visualization a title nonetheless I used JQuery to blend a title over the map on page load and fade it out over the first 6 seconds (more like a movie than a page).

Building a dataset

Different from the first project the data for this project is not present in a nice form. The idea for the visualization is of course based on data, but there is no public dataset in the exact form that would be needed for this visualization. So the project actually involves the steps of thinking which data to show and select to make the visualization clearer and then gathering this data and bringing it into a sensible form.

For the towns around Etna most of the data is hand-selected. From Wikipedia I got a sense which towns are affected by actual or historic eruptions of Mt. Etna. I then used Google and Wikipedia to get information and a nice image about these towns as well as a Lat-Lon-Finder to get the location values of the towns.

The real challenge though is the data for the area of influence of a volcano. The index used by volcanology to measure the size of an eruption is the VEI. As discussed before in this process book this measure is very weak. An article by the Guardian came to the same conclusion. There is a novel index that solves some of the issues with the VEI. There even is a novel database gathering eruptions in this new measure. A quick search on the database for Etna shows a sad picture though:
(1) The dataset is extremely sparse with only one eruption AD.
(2) The one eruption AD is NOT a devastating eruption, the ones in 1669 and 1928 were much more destructive.

This leads me to the conclusion that volcanology only has a measurement for science, but no measurement for humans. There is no index that measures how dangerous to human live an eruption was or how much damage it has done. This is both good and bad: it's bad because I can not rely on any publicly available and scientifically objective data source and good because the work I am doing is actually novel.
To come up with useful data I read through several texts describing eruptions at Etna (http://www.volcanolive.com/etna2.html, http://en.wikipedia.org/wiki/Mount_Etna) and tried to express the information in json. This gives me a sense of what the important data for the effect of an eruption on the people living there is. The structure of the json is approx. as follows:

```
"year": {
  "vei": int,
  "craters": [listOfStrings],
  "sideeffects": [listOfStrings],
  "lavaflows": [listOfObjects],
  "ash": [listOfStrings],
  "destroyed": [listOfStrings]
}
```

The VEI is still in there since it is the canonical index for volcano eruptions. There are 4 craters at Mt. Etna and the “craters” fields lists which craters had an explosion in an eruption. The sideeffects contain things like airport shutdowns. The lavaflows is an array of objects with each object giving the direction of the flow and the length in kilometers. The ash field is an array of strings with the names of cities that had falling ash after an eruption. The destroyed field finally is an array of towns that were destroyed by an eruption (luckily this field is mostly empty).

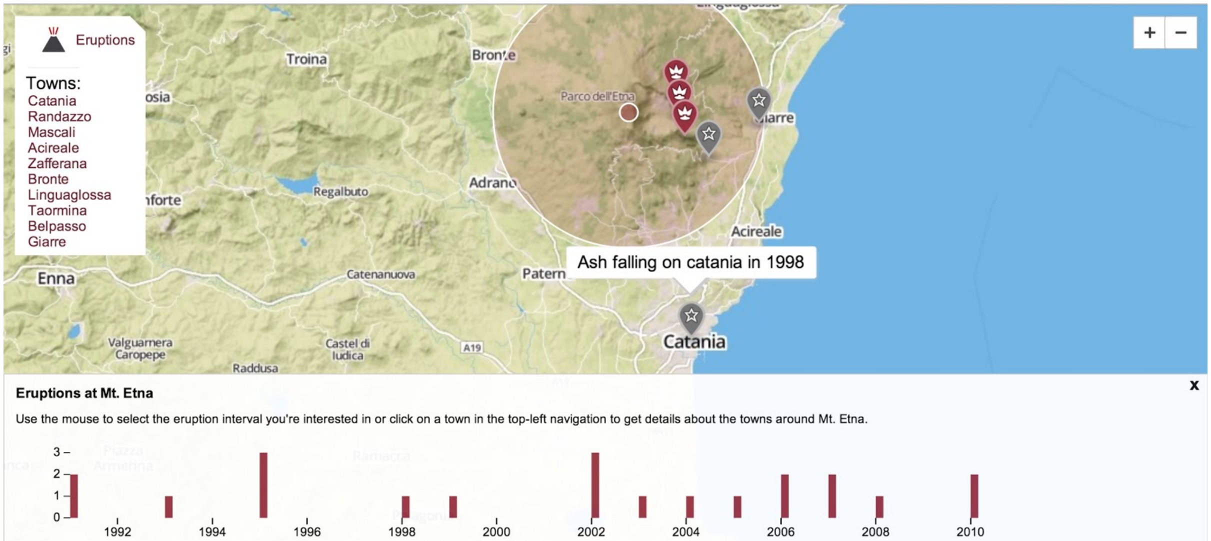
Building all of these values into the visualization would be too much for this project so I decided to do 3 things for now:
(1) Find a better measure for the circles drawn with d3
(2) Bring in the explosions at the craters
(3) Bring in the ash falling on different towns
Bringing in the rest of the data like lava flows (probably also drawn on the map with d3) or the sideeffects will be a nice addition for project 3.

For (1) I chose to go with the official plume value that is assigned to a given explosivity index (VEI). This numbers come directly from [Wikipedia](). So the circles drawn on the map correspond to the area of the plume for a given eruption. Of course these values are again just scientific approximations and there is no guarantee that the area corresponds to the actually affected area in a given eruption.

For (2) I chose to set markers at the locations of the different craters that had an explosion. Since there are possibly several explosions at the same crater in different selected years, I brought in this data into the tooltip of an explosion (on hovering over a marker one gets the years that this crater had explosions). Since mapbox has no data-binding for its markers as d3 has I chose to draw the markers in the brushend callback. An alternative might be to draw the markers in the brush method and device a method (maybe a hash value) that only redraws markers if the selection for them actually changed. Redrawing the markers in the brush callback crashes the browser.

For (3) I chose again to set markers at the towns locations with the same method of tooltips as for (2). I only take in ash falling on locations that are in the towns json. In the future I might choose to extend this to also take in other locations.

The final result with markers on the map looks like on the next screen:



Map with markers for crater explosions and ash fall on towns.

Since the data is hard to collect I only have a range of data back to 1990 for now. I will try to gather more data until the deadline. Since the data is in an external json file and does not affect the rest of the project this work can reasonably be done until the last second.

Deployment

I first tried to use GDrive for the hosting of the page. Unfortunately, mapbox uses http for getting the tiles of the map and GDrive expects only https calls (and blocks others). So GDrive can not be used for this case.

I then wrote a simple static server wrapper with Node.js (all code in app.js) and configured a heroku setup (I've done this a couple of times before). The app is in a local git repository and pushed to the heroku server with 'git push heroku'.

The final url of the project is: http://eye-on-etna.herokuapp.com/

NOTE: The node.js server can simply be ignored for local execution, i.e. the grading. You can just start the simple python server as per the project spec and view the page with this, there is no dependency on node.js except for the deployment to heroku.

Usability Test

I showed the application to three of my friends having each of them use the app for 10 minutes and then giving feedback. The results from the usability test are below.

- The interaction of using the barcharts brushing with the towns navigation is not immediately obvious. Ideas to leverage a better understanding:
  - Do a pre-selection of eruption on the barchart and map
  - Show some kind of pins for the towns on the map (have to be different pins than the pins for the eruptions events though, so I would suggest making my own pins with svg for this).
- The eruptions entry in the top-left navigation is not prominent enough (not clear that one can get back to the eruptions with this).
- The map needs a legend that explains what the area circles mean (plume) and what the different markers mean.
- Etna needs to be shown somehow.
- There are 2 bugs:
  - the selection on the barchart is lost after navigating to a town and back.
  - the pins overlay the barchart in the z-layer
- The population count should not be below the image (it has nothing to do with the image).

Reflection

In contrast to the first project this project was much more about getting the details of a visualization right, both conceptually and with respect to the design and interaction design. The data to visualize was also much more of a running target: In the beginning we started out with some data about eruptions and a rough idea of what we wanted to visualize. The rest of the data was gathered in the process, either by scrapers (Twitter) or by hand (towns and additional eruption data). Design and data were dependent on each other, for example gathering the additional data like crater explosions derived directly from the visual need that the map would be too weak with only the plume area (too little information for true exploration).

The project still has a lot of details that can be enhanced and should be in project 3 (to start with, not all the points of the usability test are considered yet). More additional data for the map, e.g. the lava flows, will make the explorative component more interesting. By telling a story (similar to the Snowfall project) we will have much better opportunities to bring the map and Twitter analysis into a coherent context (this project showed that this can not be done with geolocations since the Twitter locations are simply too bad). And of course there is ample media context like images and live webcams that can be integrated to tell a good interactive story.

So what? I think this question (from A. Cairo’s guest lecture) can be answered satisfactory: in order to provide the user a story with interactive elements to explore life around and with Etna. There is no one answer to this and an explorative tool (hopefully enriched with many viewpoints as actual stories) is ideal to let a user come to her own conclusions.