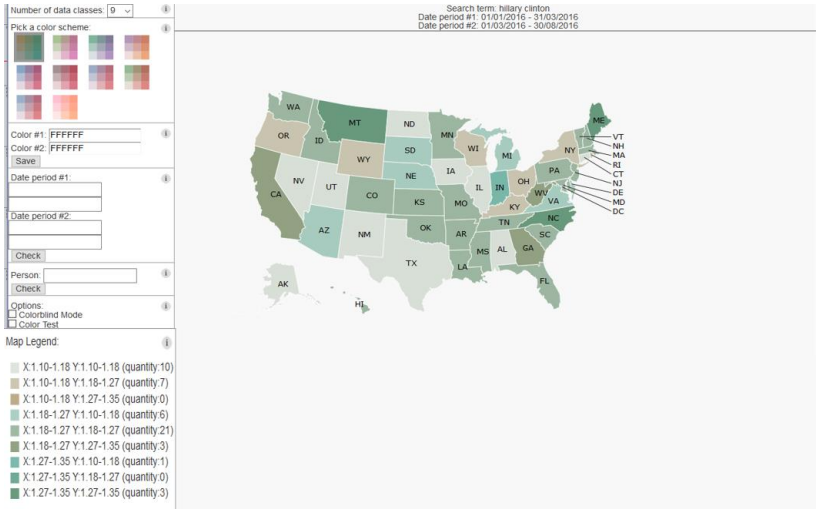
Data Visualization Project



Sharon Aboud 036689990

Shani Bernstein 302954540

Pre processing

Due to db size and our need to have a dynamic and responsive environment we performed the following preprocessing steps:

1. We download with the mongoexport command each item in the newsItems_Events table as a single object.
2. We wrote a script that preforms the following
    a. We loaded the ReverseGeoCode package with appropriate coordinates (allCountries.txt ,source http://www.geonames.org/)
    b. For each object from the newsItems_Events we ran the ReverseGeoCode package on the first georss:point.
        i. If it had a georss:point , we added the following key pairs:
            Country (e.g. “country:russia”,” country:”japan”)
            State Code (e.g. “stateCode:AZ”-Arizona ,”stateCode:AL”-Alaska)
        ii. If it didn’t had at least 1 georss:point we ignored the object.
    c. If the country code for the object is the US, then we consider this a valid object.
    d. Each group of 2000 valid objects (i.e. with a georss:point field and in the us) we group into a json array.
    e. Each array we save in a json file named part+ # of array .json (e.g. the third array is kept in part3.json file located in the data/newsItemsparts folder).
    f. The scripts splits the files because both Git and Firefox had problem handling very large json files (Git does not allow more then 1.g file and Firefox kept crashing).

All code for preprocessing is located in geo.zip, our preprocessing program is in the mainGeo file.

Remark: Due to Firefox slow down we use only the first 30 files (60000 records) but it is programmatically possible to use all files.

Data set

newsItems_Event :

- $oid (key): 56c469e45adbab1a826c062f—string (hex)
- emm:entity list: FBI, Global Positioning System, Tim Cook.--string
- PERSON list: Cook","Tashfeen Malik","Farook","Tim Cook","Syed Farook--string
- Updated: 2016-02-17T12:04:00Z –date
- Polarity: 1.340000033378601 –double
- Country(calculated): US--string
- state Code(calculated): CA--string

state table(constant,size = 51):

- id: 0 -int (1-50)
- ShortName: AL --string
- FullName: Alabama --string

User Task

The project is a tool that can be used to preform crises management and manage press coverage.

We see it as a good tool for publicist to help them view concentrated data about their client press coverage in each state, and specifically the polarity value and amount of coverage.

Being able to view to different time periods simultaneously allows the user to see if and what changes occur due to actions or events that happened.

With the coloring scheme after a learning period it should be easy for a user to see trends in the data and unusual patterns, this is especially important for publicist and crises management because their press budget and time is limited and focusing on specific locations might be more worthwhile.

We specifically took into account the coming U.S presidential election, because press coverage can sink or swim a candidate, and being able to analyze the data quickly is paramount, for this reason we added the electoral vote parameter on hover, but depending on different use cases the info can be changed.

How we do it

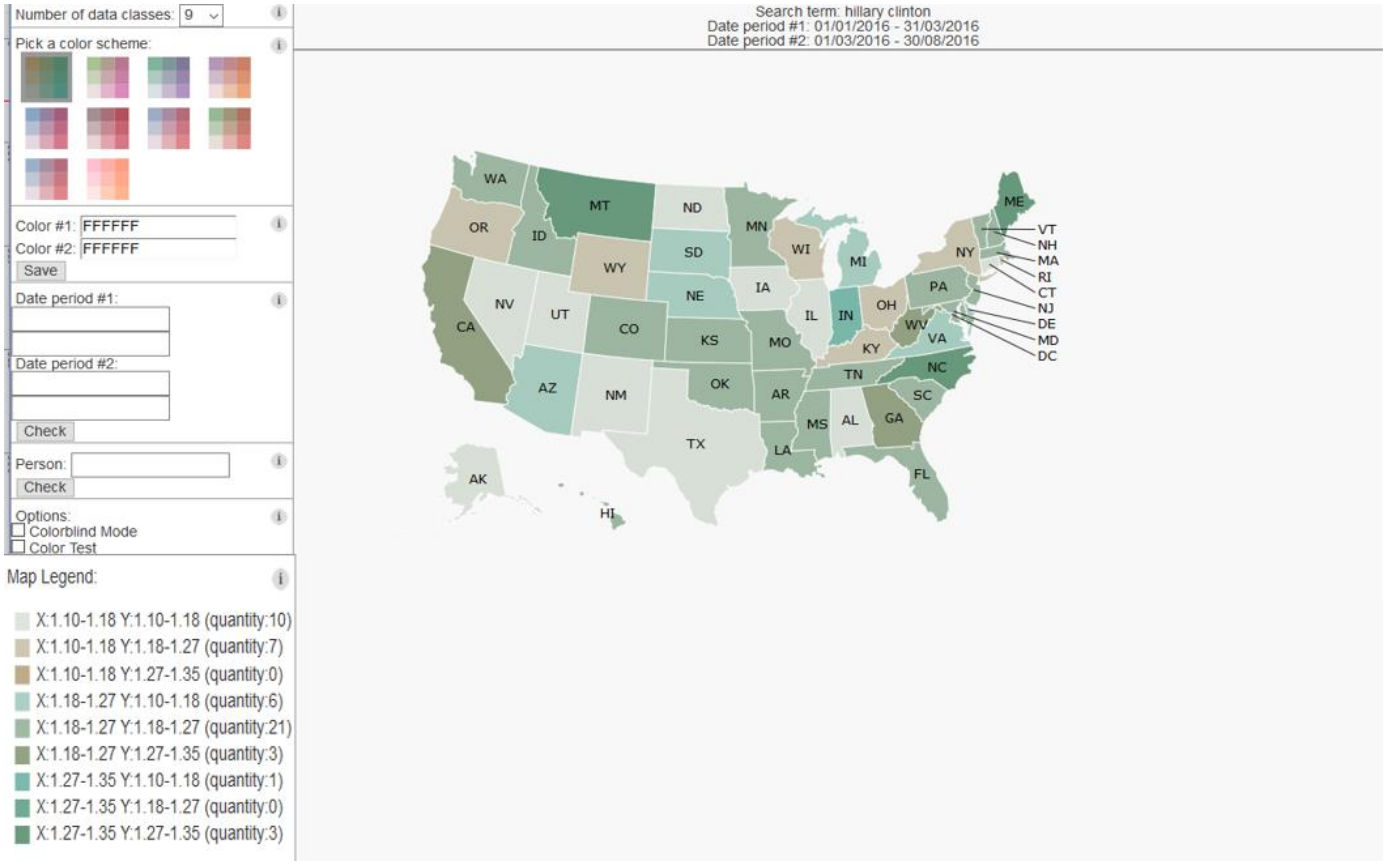
The user enters two time period and entity name (e.g. Barack Obama, F.B.I, Tim cook, apple and so on)

The system calculates the average polarity for each sate on each time period and colors the map according to the coloring algorithm.

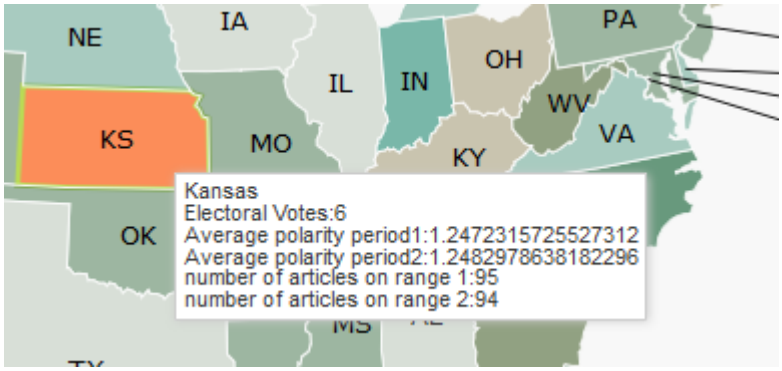
Due to the coloring scheme after the learning period the user should notice some interesting things:

- All colors in the main diagonal (location (1,1),(2,2),(3,3),(4,4)….) represent little to no change, so for example in the query bellow we see that there are 21 states that the polarity value remained at the same range as period 1 compared to period 2. This might be an area of interest for the user.
- All colors outside the main diagonal represent a change in polarity between the 2 time periods.
- If map color hue is more close to color 1, we can say generally that period 2 average polarity was higher the period 1. And vice versa.
- After a while the user should be able to recognize extreme fluctuation in the average polarity value and the color representing it(will always be top left corner and top bottom right corner), so for example bellow we see there was 1 state (Indiana) that polarity changed radically from highest range in period 1 to lowest range(in a square of 3 by 3 the color representing biggest change will be (3,1) and (1,3))
- There are many other insights the user may reach for himself.

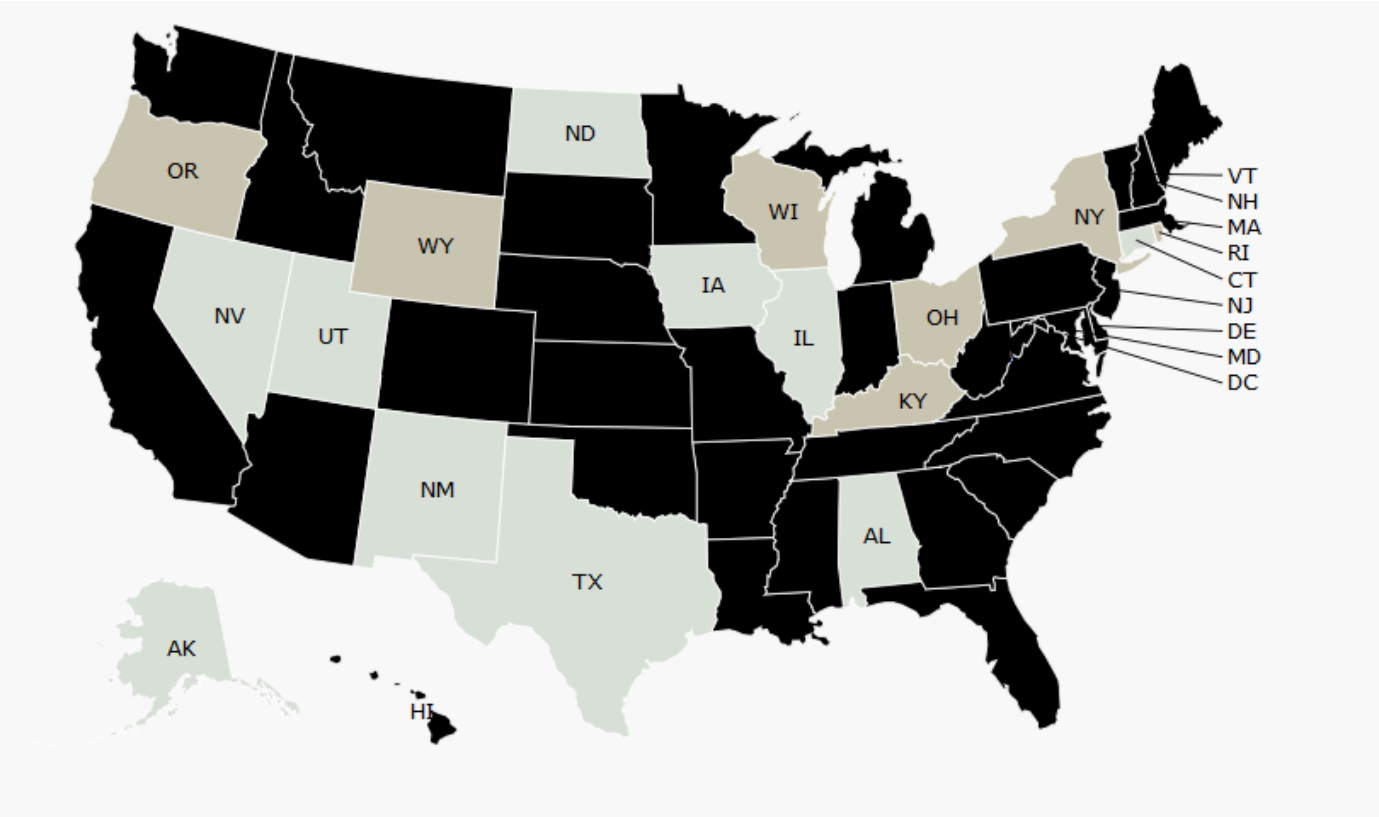
Example query



Example hover



Example filtering



Visual Mapping

- Map type: Choropleth map
- geographic area mapped (background)
- average polarity of state for period 1 mapped to color 1 (RGBA) –see algorithm explanation
- average polarity of state for period 2 mapped to color 2 (RGBA) –see algorithm explanation

Color mapping algorithm

- Input: time period 1, time period 2, keyword, color scheme (2 colors), number of data classes, min and max opacity value.
- The algorithm receives two fixed colors (the user may chose his own)
  Color1 and Color2.
- The algorithm calculates the average polarity of each state with the given keyword for each time period.
- The algorithm finds the max and min average polarity for all states.
- The algorithm calculates according to the number of data classes, ranges for each value (number of ranges will be equal to the root of number of data classes)

Example:

Min average polarity = 0

Max average polarity = 1.5248863629319451

Number of data classes = 9, root of number of data classes = 3

Size of each range = (Max average polarity - Min average polarity) / root of number of data classes = 0.5082954543106484

Range 1: 0 - 0.5082954543106484

Range 2: 0.5082954543106484 - 1.0165909086212968

Range 3: 1.0165909086212968 - 1.5248863629319451

Remark: This is simplified example the algorithm can also handle negative average polarity values.

- The algorithm calculates the opacity of each range

Example:

Min opacity = 0.10

Max opacity = 0.40

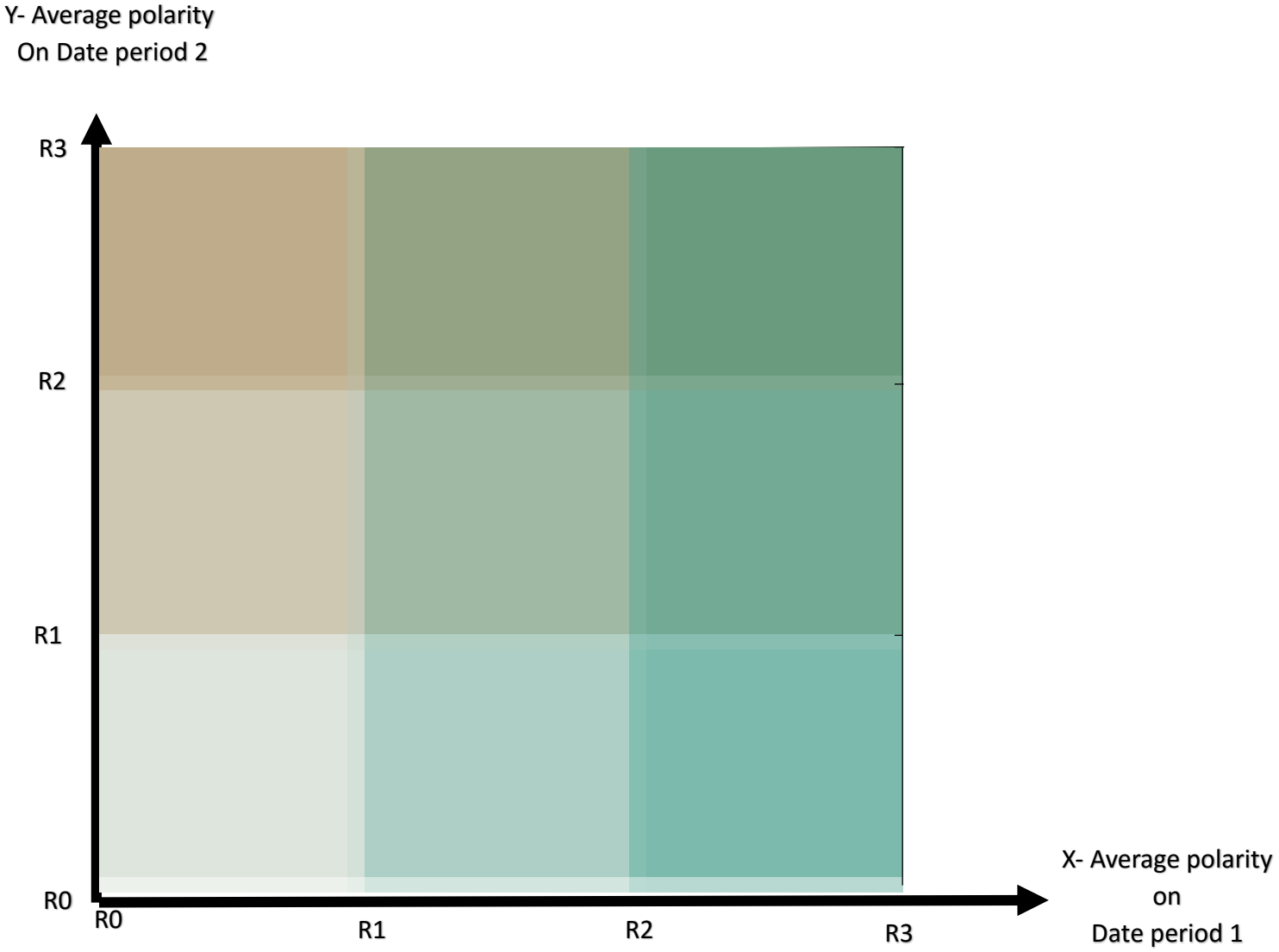
Number of data classes = 9, root of number of data classes = 3

Opacity for range 1 = 0.10

Opacity for range 2 = 0.25

Opacity for range 3 = 0.40

- The algorithm calculates the colors that will be used (number of colors is equal to Number of data classes) , where each color is superimposed from the two colors with the calculated opacity value(see diagram) (i.e. the algorithm calculates an RGBA color)



Example:

Min opacity = 0.10

Max opacity = 0.40

Min average polarity = 0

Max average polarity = 1.5248863629319451

Number of data classes = 9

Color 1(RGB): (1, 133, 113)

Color 2(RGB): (166, 97, 26)

Opacity for range 1 = 0.10

Opacity for range 2 = 0.25

Opacity for range 3 = 0.40

X = 0.6

Y = 1.4

New opacity = Opacity for range 2 + Opacity for range 3 * (1 - Opacity for range 2)

New opacity = 0.25 + 0.40 * (1 - 0.4) = 0.49

r = round((color1(r) * Opacity for range 2) + color2(r) * Opacity for range 3 * (1 - Opacity for range 3)) / new opacity)

r = (1 * 0.25) + 166 * 0.40 * (1 - 0.4) / 0.49 = 82

```
if (r > 255) {
        r = 255;
}
```

New RGBA color (82,G,B,0.49)

Features

- Color-blind Mode - Basing our work on color brewer, we added a color-blind mode that uses the same colors.
- Legend - We added a map legend where you can see what each color means, the legend also contains help, which explains how the values were reached.
- Highlighting - In the legend a user may choose a color or colors which he wishes to highlight. When pressed this will cause all other colors to black out. This was added due to request by user testers who reported the map was “too busy” and it was difficult to identify trends.
- Color Testing - This option was added in order to test the ability of users to identify all colors in a single map. It randomizes color distribution and uses all color hues in a single map.
- Selecting the number of data classes.
- Hover Info - When hovering over a state the following data is shown:
    - Full name of state
    - Number of electoral votes
    - Average polarity on period 1
    - Average polarity on period 2
    - Number of articles on period 1
    - Number of articles on period 2
- Person Selection - You may choose which person to look his polarity values (or enter an empty string, blank spaces to search for all keywords and people).
- Date Filtering.
- Color scheme choosing and personalization.
- All query and personalization info is saved in local DB, and as such you can keep your query and color personalization even after closing the site.

The Value Of Visualization (V = T + I + E + C)

T - While it takes a little bit of time to know how to work with the site, once you do know how it's quick to recognize trends, find the max/min average polarity, and perform user tasks. There's also the ability to compare average polarities quickly on a broad scale.

I - Being able to recognize trends, abnormal values, group data by location and recognize the most common class allows the user to gain a lot if insight on the subject of interest (keyword).

E - Being able to recognize trends.

C - The project doesn't generate confidence with the user.

Tasks

A1: Select states that belong to a specific set (have the same polarity color).
A5: Filter out states based on their set membership (color).
B1: Find the number of states in a set (average polarity in the same range = same color).
B13: Highlight states with specific color polarity.
C1: Find out the polarity values of both ranges on each state with a mouse-over.
C3: Compare the average polarity values between different states by coloring them (based on a color scale).

Scalability

Scalable by number of items (bigger maps) but not for data classes (using too many data classes confuses the user and makes it hard to distinguish different colors).

Review of design decision

- We decide to add as much in-site help as possible due to large learning time.
- We added Non-volatile query and color parameters due to extended setup time, and allowing the user to keep the color scheme he likes while also allowing him to present his query without the need to reenter it after quitting.
- We added the default query in the hopes that this example will show the user how to use the tool ,also because the DB is limited we placed the default end and start date of the DB so the user can search in that range by default.
- We added the number of data classes selection box because most of our trial user reported that they have problems recognizing the different colors, less color will give less information but make it easier for new users. This dose limit scalability of classes or sets, but due to color scheme to many classes become indistinguishable.
- We added the prefixed colors so a user may customize the map for his use, and certain colors seem to look better.
- We added the color customization for two reasons :
    - Some users disliked the prefixed colors.
    - Allowing for better customization for task (for example if comparing democratic candidates a red hue might be more significant).
- We separated the query to two steps (date entering and person choosing) because of two reasons:
    - Make it as simple as possible (don't force the user to fill out all fields before he gets a result).
    - Allowing the user to change only 1 of the parameters if he wishes might be a more common task by the user (e.g. a user may decide after getting the first result to change only the date so he can view another date but with the same person) .
- We added the colorblind mode and support in order to reach the biggest possible crowd, but without further testing we are not positive on usefulness.
- We added the color test mode for two reasons:
    - For our own testing.
    - For users to check that they are able to recognize all colors.
- We added the map legend, so a user may understand how the color was calculated.
- We added the map legend query to allow a user to filter quickly and find the range of his choice.
- We added the quantity info in order to be able to complete another common user task (checking set size) and allowing the user to get surmised data quickly.
- We added the hover info to allow the user to get more info on the subject while also allowing him to decide if the values are relevant (e.g. a user may decide the if only 3 articles were found for the state for the given query he shall ignore the data)

Recourse and Bibliography

- DiMarco, Mark. "DataMaps." *DataMaps*. N.p., n.d. Web. 31 Aug. 2016. <http://datamaps.github.io/>.
- Brewer, Cynthia, and Mark Harrower. "Colorbrewer." N.p., n.d. Web. 31 Aug. 2016. <http://colorbrewer2.org>.