**Pre processing**

Due to db size and our need to have a dynamic and responsive environment we preformed 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
   1. We loaded the ReverseGeoCode package with appropriate coordinates (allCountries.txt ,source http://www.geonames.org/)
   2. For each object from the newsItems\_Events we ran the ReverseGeoCode package on the first georss:point .
      1. If it had a georss:point , we added the following key pairs

Country (e.g. “contry:russia”,” contry:”japan”)

State Code (e.g. “stateCode:AZ”-Arizona ,”stateCode:AL”-Alaska)

* + 1. If it didn’t had at least 1 georss:point we ignored the object.
  1. If the country code for the object is the US, then we consider this a valid object
  2. Each group of 2000 valid objects (i.e. with a georss:point field and in the us)we group into a json array
  3. Each array we save in a json file named part+ number of array .json (e.g. the third array is kept in part3.json file located in the data/newsItemsparts folder)
  4. 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.

**Data set**

**newsItems\_Event :**

* $oid (key): 56c469e45adbab1a826c062f
* **Title:** **Apple rejects court order to unlock San Bernardino shooter Syed Farook's iPhone**
* **emm:entity list:** **FBI,** **Global Positioning System,** **Tim Cook.**
* **PERSON list:** **Cook","Tashfeen Malik","Farook","Tim Cook","Syed Farook**
* **Updated:** **2016-02-17T12:04:00Z**
* **Polarity:** **1.340000033378601**
* **Country(calculated):US**
* **state Code(calculated):CA**

**state table(constant,size=52):**

* **id: 0**
* **ShortName: AL**
* **FullName: Alabama**

**Visual Mapping**

Project presentation

Features

* Color blind mode- basing our work on color brewer , we added a color blind mode, that uses the same colors ,with only access to one color-blind person (color blinds type bla bla),we can report that color recognition for the colors in the color-blind mode was complete, but due to small test sample this is still not a good indicator.
* 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 category) 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 (testing result may be seen below). It randomizes color distribution and uses all color hues in a single map.
* Display both values
* Word selection
* Word exclusion
* Help
* Range mimaztion
* Max and min opacityseeting
* Random test

Development process and changes

Color testing

Color radmizer

Worth(formala)

Human feedback

Colorblind mode

Abitltes (is it scalable)

Mission types completion