Concept Version

Douglas Garcia Torres (1036847)
M. Bilal Zahid (1035291)
Pradyumna Majumder (1036151)

14-03-2017

Contents

1 Introduction

This open data project is focused on creating a tools to analyze and get insights from traffic violation information. The result of this project could be a useful visualization application for a smart city solution that aims to prevent traffic problems, improve road security and optimize the urban flow of people. The main goal of this project is to incorporate the traffic data into a space-time effective user friendly visualization.

In order to develop the project, we selected the Traffic Violations dataset of the Montgomery County of Maryland, USA, to use it as a model for the structure of the data. This dataset can be obtained in the U.S. Government's Open Data Portal at catalog.data.gov. It contains traffic violation information from all electronic traffic violations issued in the specified County. The information in this dataset includes:

- Date and time of stop
- Location information (state, district, latitude, longitude)
- Description of the violation
- Arrest type
- Vehicle information (type, make, model, color)
- Driver information (race, gender, city and state of origin)
- Other boolean fields (accident, belt use, injuries involved, alcohol consumption, fatalities involved, etc.)

Our resulting software will help to analyze large amounts of data through easy-to read charts and graphs. To develop this project we have decided to use JavaScript as the programming language with the popular D3.js libraries, which allow us to bind the data to a Document Object Model (DOM), and then apply data-driven transformations to this document. In addition, to include appealing geographical visualizations, we are also using the very powerful JavaScript based library, CESIUM, an open source code for world class 3D globes and maps.

The software will be published online in the next days through the GitHub Pages website hosting services pages.github.com (free HTML/JavaScript website hosting).

As part of the first submission, this report is intended to be a draft of the final report. Most of the information is still subject to change in the next weeks. Moreover, some of the sections are still incomplete or empty (results and limitations section).

2 Requirements

The main requirement for this project is to pick a non-trivial data set and create a tool for visually browsing that dataset for interesting information. It is also important (although not mandatory) that our problem involved a dataset with spatial-temporal data.

In addition, after defining our problem and selecting our working data set, another important requirement arose: because we are trying to build a data visualization tool for traffic information, it is natural to expect an integration of this tool with technologies like OpenStreetMap (OSM) or Google Street View with the intention to show the user real images of the geolocation data.

Another natural requirement for a visualization project like this, is that the tool should be highly interactive with the user. Therefore, we are considering the following features as requirements for our minimum viable product:

Visualization selection: the user can customize the type of visualization. Thanks to the CESIUM libraries, the user is table to select the layer of the map between multiple options like a satellite view, streets classic view, OpenStreetMap, National Geographic view, natural earth or black marble view (night view).

Hovering: the user can pop-up brief tooltip information about any part of the visualization by hovering over with the mouse. In addition, hovering a part of a specific visualization panel will **highlight** the related information in the main panel.

Drill-down: into detailed information by clicking on any part of the visual components with additional details available. After a left mouse click, a new panel will populate with the details of the traffic violations.

Filtering: providing the visualization tool with a panel with filtering controls by any of the available attributes for allowing the user to perform data analysis at any possible level. The filtering can be done by the following criteria:

- Time of stop (clustered by hours)
- Accidents involved
- Belt (use of)
- Personal injuries involved
- Property damage
- Fatalities involved
- Hazardous materials involved
- Alcohol consumption involved
- Vehicle type, make, color of vehicle, year
- Driver gender and race
- Charge and arrest type

Zoom in/out: the user is able to zoom in and out of the location of a specific incident or even on any random point on the main map graph.

Street View: the user is able to select the OpenStreetView as the visualization layer in the main graph with the possibility to zoom in and see nitid images of the locations of the incidents. In addition, if the time allow us, we are willing to include an integration with the Google StreetView services to provide a closer look.

Temporal analysis: because our problem and the dataset we use includes time dependent data, the tool should provide the means to analyze the evolution of data through the temporal dimension.

3 Analysis and Solution

3.1 Analysis

Considering the requirements stated in the previous section, the problem of analyzing traffic violation data requires spatial and temporal visualization tools. In addition, for any data analysis task it is mandatory to count with filtering and drill down capabilities without forgetting to provide ease of use to any kind of user.

The person who is performing a traffic violation analysis would be willing to visualize the evolution of the incidents over time, preferably mapped into a geographical layout view. In addition, this person would need to see the aggregated data of traffic violation records by any possible criteria. For example, analyze the evolution of the number of traffic violations of male drivers where alcohol consumption is involved.

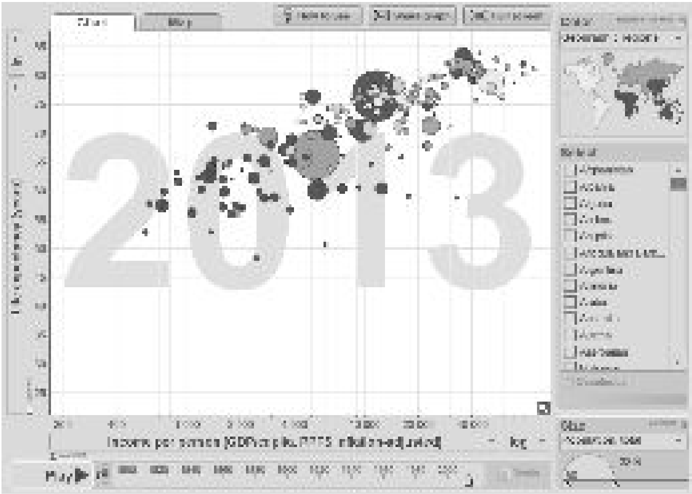
Our idea is to cover all the stated requirements to provide the means for performing such kind of analysis by developing a straightforward visualization tool with the technologies that are more appealing to us as Data Scientist.

3.2 Solution

The solution proposed is a visual web application developed using JavaScript language, the D3.js library and the CESIUM library to map the data into a spatial dimension. The first version of this application aims to be connected to a local data source in JSON format.

The web application provides a straightforward single perspective visualization with a main geographical view along with three additional panels: one panel to visualize the evolution of data over time, another panel to view histogram-like aggregated information, and a third panel to verify the detailed information of the traffic violations. This layout can be reviewed in more detail on the next section.

The single perspective is complemented with 2 widgets: a filtering widget to allow the user to analyze the data by any possible specific criteria, and a play-time widget to provide an effective solution to visualize time-dependent data. This play-time widget is inspired on the famous Gapminder charts which makes possible to visualize the behaviour of 4 different dimensions/measures over the dimension of time (see figure below).

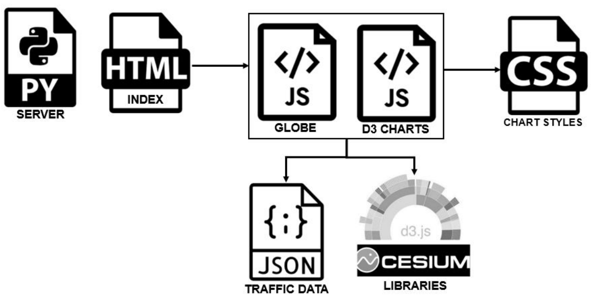


The Gapminder chart shows 5 dimensions of data. The life expectancy in years (Y axis), the income per person as GDP per capita (X axis), the countries (each country is a bubble), the population of a country (bubble size), the region of the country (bubble colour) and the time in years changing when the play-time widget is triggered.

With this tool any user could easily analyze the behaviour of traffic violations over space and time, while selecting specific criteria of available attributes to filter data. We have the intention to make this tool available online through the GitHub Pages website hosting services pages.github.com.

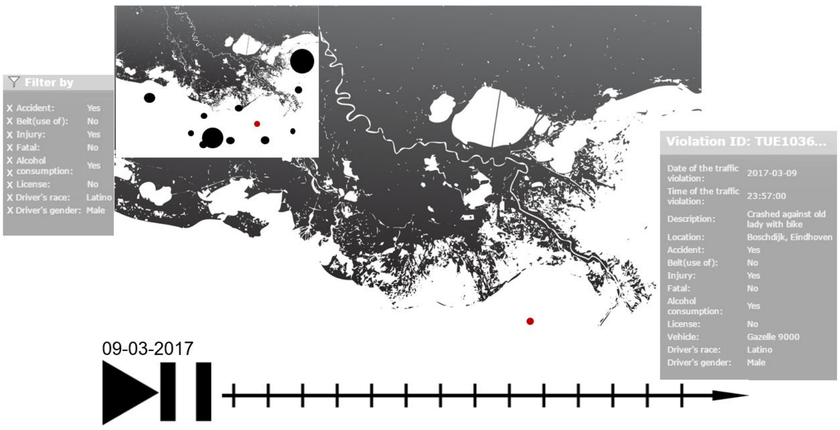
3.3 System Design

The solution we are building is mainly composed by seven components between HTML, CSS, Javascript, Python and JSON files along with the already mentioned JavaScript third party libraries. The scheme of the backend of our application is illustrated below:



- The main file is Index.html, and makes reference to the two main JavaScript files: Globe.js and the D3 chart file.
- The Globe.js file loads and maps the traffic data (from a JSON file) into the main visualization provided by CESIUM libraries. In addition, defines the functions for the user interaction with the main graph.
- The D3 Chart file loads the data into the D3 developed visualizations, also defining the interactive functions and the interoperability with the main visualization. It makes use of the CSS styles file for the graphs.
- The CSS styles file provides the D3 visualization with proper style formatting.
- All the traffic data is included into one only JSON file.
- The Server.py file is used to generate local web services in order to run the application.
- The rest of the files are included in two folders corresponding to the third party libraries used: D3.js and CESIUM.

On the other hand, the visualization is basically composed by five components: a main geographical visualization, the D3 developed visualizations with aggregated information, a detailed information panel, the filtering panel and the play-time widget. The design of this layout is shown below:



- The main geographical visualization show the location points of the traffic violations. It is synchronized with the other four elements:
  - If the user interacts (clicks on a specific part) with the D3 visualization of the top left side of the screen, the related traffic violations will be highlighted.
  - When the user executes a filtering function, just the traffic violations that meet the filtering criteria will appear.
  - When the user clicks on a specific point (traffic violation), the detailed information panel will pop-up in the right side of the screen.
  - The points shown will correspond only for the traffic violations occurring in the specific period of time (day) that is running on the play-time widget. We will consider to

include a parameter to configure this period of time as a day, a month, a quartile or a year.

- The D3 developed visualization on the left side of the screen shows 3 dimensions of data: latitude (Y axis), longitude (Y axis) and number of incidents (color). As the time passes (by the play-time widget), the color of the bubbles will go through a yellow-red scale as the number of incidents increases in a specific sector. This way, the user can detect the hot points where most of the traffic violations occur.

- The detailed information panel on the bottom right side of the screen show all the available information of the traffic violation selected on the main visualization.

- With the filtering panel, the user will be able to decide what data to visualize according to the constraints selected. The filtering options will be populated dynamically according to the loaded data values.

- The play-time widget is provided by the CESIUM library and it features forward and backwards linear execution of time. The speed can be tuned in the application code.

4 Motivation of Choices

4.1 Data: Traffic Violations

To select this traffic violation data, our motivation was to find a large and complex enough dataset which can fulfill all the requirements for this project. After reviewing different options, we selected this traffic violations dataset because it contains a large number of interesting visualization aspects, it is complex enough and also gives us various ideas to develop. Moreover, it also contains space-temporal information of a particular incident, which is required for this open data project.

4.2 Technologies: JavaScript, D3 and Cesium

The main reason to choose JavaScript as the programming language for developing this application was that it is the base language of the famous D3.js libraries. The use of this technologies was recommended to us by various colleagues, the professor of this course and the professor of the Visual Computing course, Michel Westenberg.

Further, the D3.js and Cesium libraries together, helps us to easily generate contents while providing a variety of appealing visualization aspects. A considerable number of the search results for visualization tool options, recommended to use these libraries. In particular, with the CESIUM libraries we can fulfill the OpenStreetMaps integration requirement while providing multiple additional features:

- Visualize high-resolution global terrain.
- A 3D globe, 2D map view.
- Layer imagery from multiple sources.
- Timeline and animation widgets for controlling simulation time.
- Base layer picker widget for selecting imagery and terrain.
- Selection and info box widgets for highlighting objects and displaying information.
- Geocoder widget for flying to addresses and landmarks.
- Home view widget to fly to the default camera view.
- Scene mode picker widget to morph between 3D, 2D, and Columbus view.
- Full-screen widget for toggling fullscreen mode.

4.3 Functionality and Interactive features

The choices we made to show specific visualization aspects based on our past experience (Visualization courses) and interests. As all group members are data scientists, we have developed understanding that what information will be helpful and interesting to the end user. Moreover, we were influenced by the flexibility and appeal of visualizations like Hans Rosling's Gapminder among others examples from the D3 homepage.

5 Results and Limitations

5.1 Results

5.2 Limitations

There are various limitations on this application as a consequence of what we consider out of scope for this project. For the moment, we can mention the following:

- The geographical scope of this application lies on the limits of the Montgomery County of Maryland, USA. For the moment we have not found a more complete traffic violations dataset on another location. Therefore, this dataset's structure is taken as a model. To include the data for any other region, it has to be pre-processed to fulfill the structure and format requirements.

- The application runs over a local dataset. In order to perform analysis on updated data, it has to be manually downloaded, cleaned and preprocessed (if necessary). Although the development of a web crawler API would be of great value for this project, it is considered out of the scope.

6 Conclusions and Future Work

At this moment, one of the conclusions we can mention results from our surprise regarding ease to create highly powerful visualizations with available open source tools like the D3.js and CESIUM libraries. We have discovered that most of the difficulties of creating visualizations come from the data processing tasks.

The future work related to this project is going to be driven by the interests of the members of this team. At this moment we can think of three possibilities: enhancing the features of the traffic violations visualization, combining and complementing traffic violations data with other traffic related dataset (e.g. violations without police intervention), or applying this visualization layout and code to another unrelated topic.

With the first possibility, we could enhance this visualization with drag and drop capabilities, more panels showing different graphs and perspectives of traffic violations aggregated information, or by giving the user the possibility to personalize the already developed features (e.g. adding the possibility to change the dimensions involved in the spatial bubble graph of the traffic violation count). We also think of the possibility to enhance the tool by developing a web scraper, i.e. a web API to provide pre-processed and cleaned traffic violation data.

With the second possibility we could enrich the current visualization with other traffic related data to give more value to this open data project. One possibility could be to find (or define) a unpunished traffic violations dataset and develop a combined visualization of punished against unpunished records. Such visualization could be very useful to traffic authorities.

The third possibility in mind is to adapt the developed visualizations for this project to World Development Indicators data. Having in mind that the popular World Bank dataset, does not count with an appealing geographical data visualization. In addition, such project can include the use of the World Bank data catalog API for a real-time visualization of data and adapt the play-time widget used for this project, to simulate the famous Hans Rosling's Gapminder bubble charts with any of the World Development Indicators.