CS686 Data Visualization Final Project Report

Seimei Matsusaki
Pan Wang

December 12, 2014

Abstract

We visualized "Shelter" database given by our community partner, One World. Our approach is to merge different data visualizations into a single unit. We experienced failures, but our final visualization finally becomes a powerful and convinced proof of One World's contributions to the improvement of children's shelter conditions. Although it is not clear how real audience (potential donors) react to our visualization, we believe that it will help One World to attract more external donors.

1 Introduction

1.1 Goal

From our perspective, the goal is to create a visualization to show how much our community partner has contributed to the improvement of shelter conditions for children around the world. From our client's perspective, the goal is to attract more external donors, using our visualization.

1.2 What we tried to do

For the alpha release, we created visulizations (e.g., barcharts, geomap) in Tableau, and were going to merge them into one later. However, we found that it was not exactly what the community partner expected us to make; what it really needed was a web-based and interactive visualization, not static one. Therefore, we entirely recreated our visulization in D3. As a result, our early visualizations are no longer used, yet their idea (i.e., integration of different visualizations) is reflected in the new visualization.

2 Approach

2.1 What approach(es) we used

As briefly described above, we tried to combine geomap with other charts. This means that we use the map as the user-interface, and other charts (i.e., barchart and stacked bar chart) are shown as a result of the user interaction. Simply put, when user clicks a country, the map will zoom-in and show the corresponding data to the country.

2.2 Under what circumstances we think it should work well

Basically there are two circumstances that we think that our visualization work well: when enough data is available to show in the geopmap, and when our visualization is planned to be updated.

2.3 Why we think it should work well under those circumstances

Firstly, if enough data is given, the geo-visualiztion can become better than other visualizations: The user easily can understand with which country our community partner works and see its influence around the globe at a glace. Secondly, if the visualization will be updated, data charts can be a matter. In the future, our community partner may want to show a different type of charts instead of the current charts. In such a case, since our geomap UI is independent from chart types, it is easy to connect a new chart with the UI instead. Besides, even though the number of countries increases, it is not required to make a new navigation interface. In the geomap UI, what is necessary then is just to color new countries and connect them to the corresponding data.

3 Methodology

3.1 Map(to show partners in different countries)

- What we implemented: Tooltip to show an overview of the country data; Zoom-In and Zoom-Out functionality.

- What we did not implement: In early stages, we planned to place barcharts inside the boundary of a country, not the right side. Yet we realized that some countries such as East Timor were too small to put charts inside.

- Possible alternatives: It is possible to use a list as an alternative. It is easy to implement since it only requires text. Still, it may need a page navigator when the amount of data increases. As a result, the user may need additional steps to see details. That can lead a bad user experience.

3.2 Bar charts

- What we implemented: Vertical barcharts.

- What we did not implement: Horizontal Bar Charts in order to show the geomap(UI) bigger for better user experience

- Possible alternatives: if values had been ordered by the time, we would have used another chart such as line chart. Yet, the actual values are not ordered by specific values, so we wonder if there are barchart alternatives here.

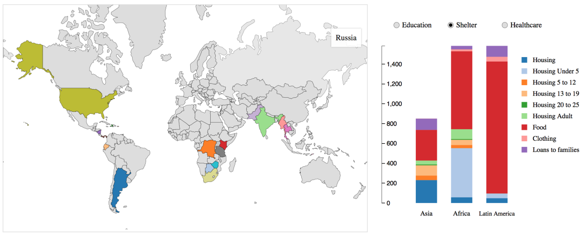


Figure 1: Geomap and Stacked Bar chart

3.3 Stacked Bar Charts

- What we implemented: Vertical stacked barcharts.

- What we did not implemente: Horizontal stacked barcharts for the same reason as the barchart.

- Possible alternatives: Grouped bar charts or multiple bar charts could be used for this part. As compared to the stacked barchart, their values are easy to be compared. Still, it becomes difficult to fit in a limited space instead. Since we wanted to show the UI bigger, this aspect was not ignorable. Therefore, we chose the stacked barchart.

4 Demo and Results

4.1 demo

Figure 1 shows the first screen tha the user will see. Only countries that have the corresponding data are colored. The stacked bar chart at the right side shows world data, rather than data associated with specific countries. There are three buttons on the upper right corner. These allow the use to click and switch different datasets: Shelter, Education and Healthcare. The user also can click any contries they want to see. If the corresponding data is available, it is shown as a barchart at the right side (See, Figure 2). If it is not, ”NO DATA” notice is shown instead. In either case, when a country is clicked, the map zooms in. If clicked one more time, the map zooms out.

4.2 How we measure success

We think that our success from a short-term perspective equals how much the community partner is satisfied. With this respect, we got a good feedback from the community partner: ”This is wonderful to see. I love the map and the bar chart at the side!.” This was a much better result than we expected. Still, the real success should not be measured only by the feedback: if our visualization increases real donors, then we can say our visualization is successful in a real sense.

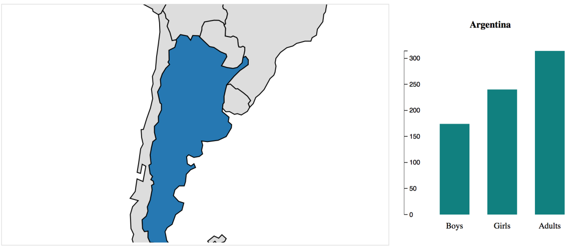
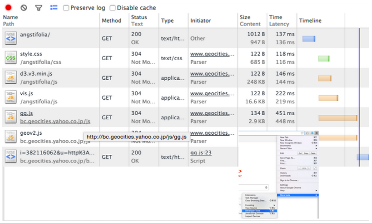


Figure 2: Country and its data in the barchart



Name Path	Method	Status Text	Type	Initiator	Size Content	Time Latency	Timeline
angstifolia/	GET	200 OK	text/ht...	Other	1012 B 947 B	137 ms 136 ms	
style.css /angstifolia/css	GET	304 Not Mo...	text/css	www.geocities.... Parser	122 B 685 B	118 ms 116 ms	
d3.v3.min.js /angstifolia/js	GET	304 Not Mo...	applica...	www.geocities.... Parser	122 B 248 KB	146 ms 144 ms	
vis.js /angstifolia/js	GET	304 Not Mo...	applica...	www.geocities.... Parser	122 B 16.6 KB	222 ms 219 ms	
gg.js bc.geocities.yahoo.co.jp/js	GET	304 Not Mo...	applica...	www.geocities.... Parser	134 B 2.9 KB	451 ms 448 ms	
geov2.js bc.geocities.yahoo.co.jp/js				www.geocities.... Parser			
i=382116062&u=http%3A... bc.geocities.yahoo.co.jp/b...	GET	200 OK	text/ht...	gg.js:23 Script			

http://bc.geocities.yahoo.co.jp/js/gg.js

Figure 3: Chrome Developer Tool

4.3 Experiment we conducted

The performance of the visualization is probably one of the most important concerns since it totally can change the user experience and impression on the viusalization. Nobody will like visualizations that take more than, say, 30 seconds to show. We recognized that ours was not such a case, yet were interested in the actual performance. For this experiment section, we conducted to time the performance of our visualization, using the latest Chrome and Chrome Developer Tool (See, Figure 3 and 4). For every execution, internet cache was cleared. The measurement was carried out five times each on OS-X Yosemite and Windows 7. The results are visualized in Figure 5.

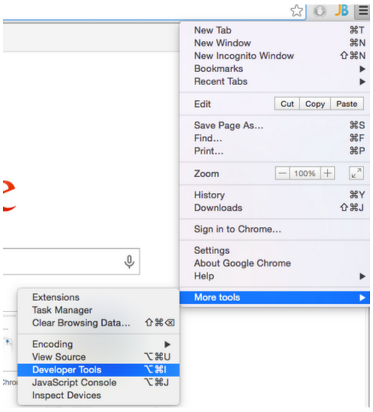


Figure 4: How to run the tool

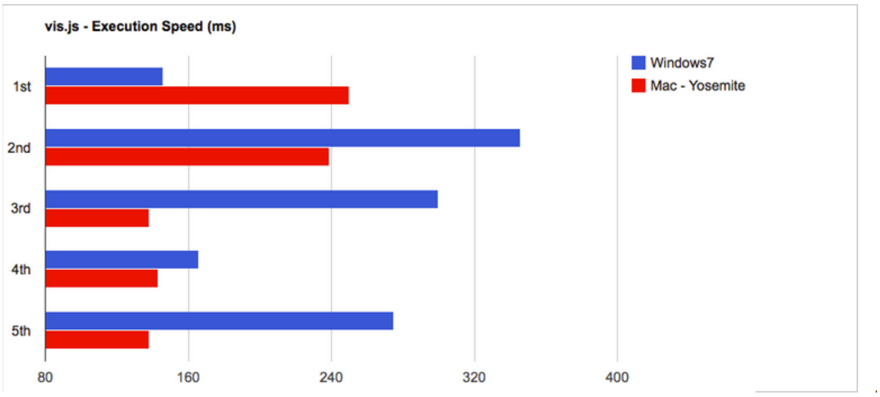


Figure 5: Experimental Result

4.4 What our results indicate

Regardless of the OS, the execution speed is all less than 0.4ms. Since our implementation relies on vis.js to add and manipulate DOM elements and show visualizations, these results mean that ours are fast enough. That is, they do not harm the user experience. Perhaps, the size of the data contributed to this good performance: small data does not take much time to be processed.

5 Discussion

5.1 Is our approach promising?

We hope so, but it is not clear unless real users evaluate our visualization and give feedbacks to us.

5.2 Possible different or better approaches

One possible approach is to use a bigger map, and merge charts totally into it. This is what we discussed in an early section. If this approach is possible, the design consistency of the visualization can increase, and we may be able to get a better responce from the user, comparing to just putting the map to the left side and charts to the right side. Another possible approach is to use multiples charts. This is a straightforward way. Although this approach has a weakness that the user does not necessairly check each chart, it is still easy to implement and maintain.

5.3 What we learned from this project

First, we should not assume that our community partner has the same knowledge with us: the partner is not necessarily familiar with the computer and programming. It will probably be the best to avoid using of technical terms (e.g., debugging) when to ask the partner about the project or report the progress. Second, it is import to consider how our product will be used. In class assignments, we basically do not need to care about some aspects of programs such as maintenance, deployment or future updates. Yet, in this project, the situation is totally different. Considering those aspects is essential to be successful.

5.4 What we would do if we could start the project again

If we could start the project again, we would show some samples such as design sketches on paper to the community partner first. Fornatunately, this time, we could get a good feedback from the partner without showing such a sample, yet if we had shown, we may be able to delineate what the partner really wanted.

6 Conclusion

We have worked with our community partner, One World in this project so far. We experienced a lot of trial and errors, yet finaly could go through creating a visualization, which we hope to achieve the goal of our partner. During this project, we realized that communication with the client can be more important than coding. As mentioned in an early section, we needed to recreate our visualization from scratch because once we failed to find out the exact needs of the partner. We both did and learned a lot, but miscommunication still can happen. it can not only lose our time but also make the project difficult to be successful. Fortunately, this project seems to end without more troubles. We hope we can apply what we learned to future projects and visualization projects.