Final Project Technical Writeup

Alfred Lam

Contents

Abstract

As technology rapidly becomes ubiquitous and easy to access for more and more people across the world, we find ourselves with an increasing number of tools to visualize data - for instance, the D3 API. Through a simple and easy to use interface, D3 presents beautiful, data-driven visualizations with minimal code. The question then becomes what kinds of data can we visualize to take advantage of the incredible features that such API's offer? In this project, I use it to analyze the results of the midterm elections of last year, something that, while seemingly insignificant, may present itself as an important topic given the contentious and volatile nature of America's political state today.

1 - Introduction

Project Description

Our project focuses on one particular issue on the massive list of voter interests: California's Proposition 6, a bill that would repeal an earlier bill which increased gas taxes over time. Prop 6 failed to pass in California, with massive opposition. We hypothesize that this may be because areas that utilize less gas and resources wouldn't mind the gas tax, and thus would vote against repealing a bill that would only help them. However, our problem is that there is no easy way to see if this is true - so we constructed a visualization of all of California's counties, how they voted regarding Prop 6, their relative gas consumption, and a few other factors, to see if our hypothesis was true.

As our final project, we constructed an extension of our visualization in Program 2 by implementing brushing and linking, creating a secondary bar graph that allows users to much more easily visualize the gas consumption of each county, and thereby see whether or not there exists a correlation between gas consumption and voting patterns on Prop 6. Our

implementation of linking and brushing allows the user to select certain parts of the data and see these highlighted in both visualizations for easy comparison.

2 - Technical Implementation

2.1 - Data Collection and Parsing

Gathering data was perhaps one of the most difficult parts of this project - each site I retrieved data from put it in its own format, with its own nuances and sometimes missing county data as well. First, I found data containing the gas consumption of each county, from the California Energy Commission website. This allowed me to choose between Residential, Non-Residential, or Total gas consumption in each county, for all years up to 2017. I chose to utilize only the residential area data to begin with, as the majority of voters would not care or have any choice over the non-residential uses of gas.

Next, in order to determine political affiliation, I used the midterm election data from the California Secretary of State's site. Here I decided to use the governor's election results as the determining factor in how a county's political affiliation leaned, as these elections tend to have the highest voter turnout amongst other choices. Within the program, political affiliation is calculated percentage-wise, taking into account the drastic difference in population size across different counties.

I then collected population data of the different counties, and finally the ballot results from the midterm elections. Population data, as well as the map coordinates used to construct the visualization, were retrieved from the US Census sites. As with political party, the ballot count was calculated percentage-wise, since each county had a varying number of population, as well as varying voter turnout rates.

Here I had 2 choices: 1) I could manually copy and paste only the data I needed from 5 different CSV files into 1, and load only that one into the program. Or 2) I could simply load all 5 files into the program and deal with the complexities of that in the code. I chose the 2nd option, as I believed it would be faster, but I unfortunately think I made the wrong choice here. The result of this choice was a ridiculous series of nested asynchronous file-loading functions, along with a penta-nested for loop to search through the data and join the appropriate parts together. In future applications, I will need to find better ways to do such data retrieval and parsing. Once the data was loaded, the values were stored along with the coordinate information of the respective counties, allowing for easy access afterwards.

2.2 - Technology

The main technologies used in the project were CSV, JSON, and geoJSON files for data, and D3 and Javascript for the code. D3's incredibly useful and optimized API made it easy to make things like transitions, interactivity, and data joins, and is effectively what the entire project is built on.

To utilize the program, the user need not download anything, as the required files and libraries are imported automatically within the code.

The code structure I used is extremely simple, with only 2 files, an HTML file representing the webpage, and a Javascript file that parses the data and constructs the interactive visualization. The HTML file is fairly self explanatory, with a lot of CSS organizing the appearance of the page to make it look pleasing and polished, and a couple placeholder elements in HTML for the data to fill up with visualizations. In the Javascript, the code is mostly sequential, with the data first being loaded in, then parsed, and then joined together with the map information. The controls are then linked with the buttons on the HTML page, and determine which data is shown using D3 when a button is clicked.

2.3 - Limitations and Constraints

After using D3 in the last project, I was much more comfortable working with its API, thus leading to a lot more polish in the visualization, as well as more confidence in experimenting with new ideas. That being said, in my attempt to make the visualization simple and clear, it was not able to hold as much data as I would have liked, as I could not utilize any 3D effects, and more 2D choices only ended up making the visualization more cluttered and confusing. I believe that with linking and brushing, a second, side-by-side visualization will solve this problem effectively, as I will have more options to display data with. At the moment, it is difficult to compare multiple factors at once, since there is only 1 map, and only so many colors can be shown before it becomes distorted and hard to understand.

Another limitation was with the circles that I used to represent voting results and population size - it was difficult to find a good scale for the area of circles to convey relative populations sizes, without the entire map being filled with yellow and green circles, since the counties were often small and within close proximity to each other. The color choices were also a difficult challenge, as I originally used green and red for the circles (representing yes and no votes), but it would be very hard to see red on red, when the user swapped to the gas consumption visualization. On the other hand, green and yellow worked alright relative to the background, but consequently did not represent the voting results of yes and no quite as well as I would've liked.

3 - Visualization Choices

3.1 - Representation

As with Program 2, I used only D3 and Javascript to build the map for this program, utilizing geoJSON data and D3's convenient API to construct a representation of California. All the data was placed onto the map in some form of color, size, shape, and opacity to represent multiple factors that may have contributed to the results of the election. In addition, my original intention was to aim at making the visualization simple, intuitive, and easy to use, which was much more difficult than I had thought.

As the main extension, there was a lot of consideration put into the choice of link/brush to implement, and I eventually settled on the final bar graph, as this was the most clear and straightforward way to rerpesent data that I could fine, which was the thing that was the most lacking from the initial visualization - while the map was effective at conveying multiple

layers of data, it was confusing and could be difficult to process - with linking and brushing implemented in a simple bar graph right next to the main visualization, this solved the problem. I chose to also represent multiple layers of data in the 2nd visualization as well, using both color and bar height to convey different data, allowing the user to compare the counties both through gas consumption, votes, and location.

While at first seemingly obvious, it turned out to be very difficult to choose how to represent the data I wanted to show - I ended up not finding a way to represent other contributing factors, such as quality of public transportation, number of vehicle licenses, etc, partly because of lack of data, but mostly because there wasn't a clear place to put that data.

Color

As one of the most effective means of communicating data, I utilized mostly color to display my results - choosing colors that complemented each other, without sticking out or being too hard to distinguish, was a difficult process, and I ended up relying on premade color schemes (such as the one used for the gas consumption visualization) rather than creating my own color palettes.

Opacity

This is a feature I may have relied too heavily on, as it turns out that transparent colors on top of dark colors can be very difficult to see. However, I chose to use this because *relative* transparencies were very easy to see, and one of the points of this visualization was to compare the counties results to each other.

Size

While used to convey population, I would ideally like to show this in a different form, for reasons discussed above. It just isn't very clear to the human eye when there are differences in area or size of a location.

Height

With the bar graph, height was a very clear and understandable visualization technique to convey information - it is much easier to see which counties use up the most gas per population, and then select them to investigate further, using linking and brushing, then it was before, looking at relative shades of colors on the map.

3.2 - User Interface

The user interface is straightforward, with the user being able to select 3 buttons. They may choose to toggle the voting results, which makes a number of varying opacity circles show up on the map, corresponding to the ballots of each county and how one-sided the vote was. They may choose the gas consumption visualization, which maps each county to a different shade of red/orange, with darker colors indicating higher consumption rate (this also takes into account population size). Finally, they may choose the political affiliation visualization, which displays how Democratic or Republican voters in each county voted, based on the election for governor.

In addition to this, the user may hover over any county, with any of the visualizations above

selected, to display a tooltip on the side, with detailed data about the county hovered, should they wish to know specific numbers rather than simply a general, relative visualization. This was my attempt at a simple version drill down effect for the project.

As the extension for linking and brushing, the user may now also select any county on the map visualization, or any bar on the bar graph, making the visualizations highlight the corresponding data in the other visualization. This allows the user to select specific parts of the data that they would like to investigate in further detail, while still keeping in mind the bigger picture and relative differences among the entire dataset.

4 - Results and Conclusions

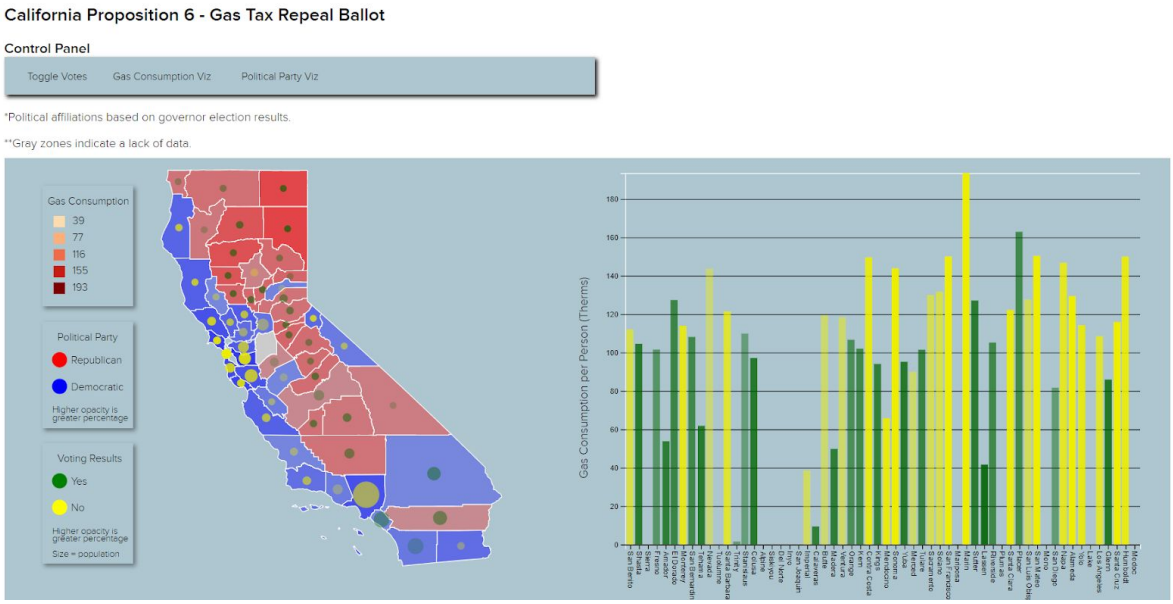
Interestingly enough, our visualization actually seems to prove our hypothesis wrong - the areas with the highest gas consumption ended up being the counties that most strongly opposed the gas tax repeal. On the other hand, there happened to be a strong correlation between political affiliation, and the way that a county voted. These seemingly counterintuitive results make me wonder if there are less obvious contributors to these voting results, for instance, environmental education, economic standing, and other things that are unrelated to gas itself. These may warrant further analysis in a future project.

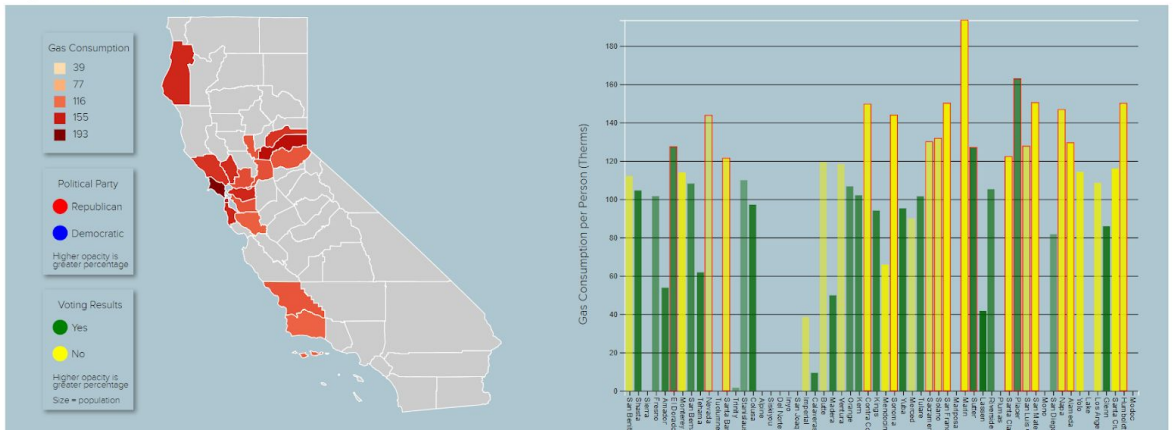
Regarding visualization, I've found that one of the most challenging problems for me remains the issue of displaying a large amount of data in a single visualization. The original intent for Program 2 was to display many more different types of data on the map, but this quickly proved infeasible and hard to understand/process as a user. Linking and brushing made it a little better, with more data/specific details being able to be displayed in the same space, while still allowing for clarity, but only to an extent. Since linking and brushing itself requires a bit of understanding and interactivity, I cannot just add more data on top of it, without thinking carefully about how to best represent that data.

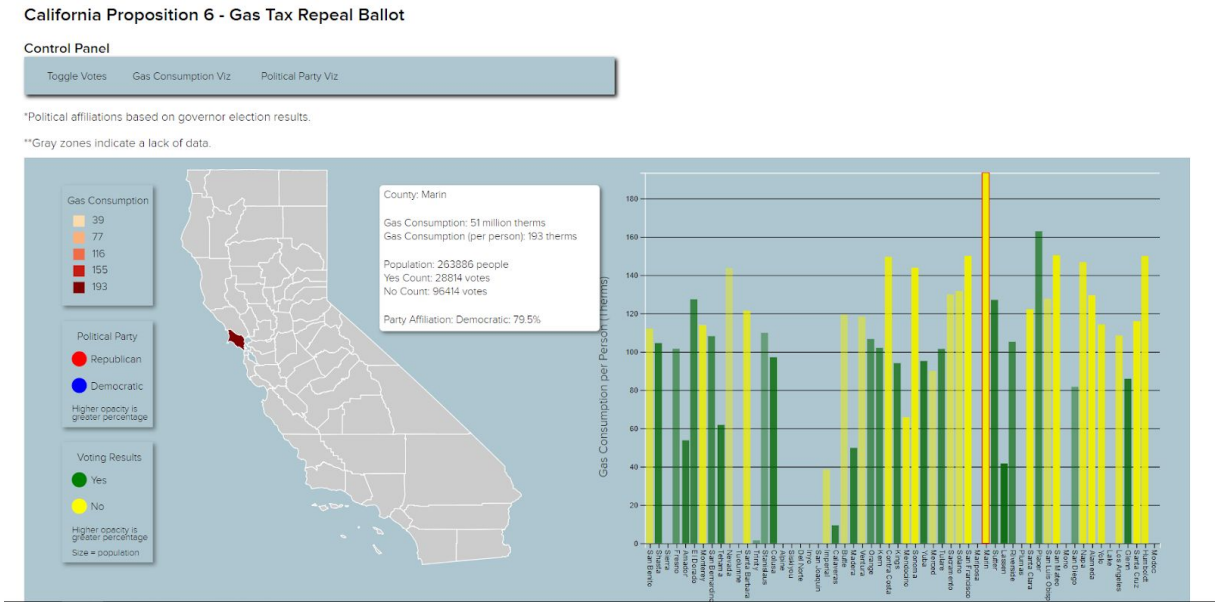
Overall, I've learned that I need to plan out my visualizations more thoroughly, and that I cannot hope to add too much data to a single visualization all the time - it would've been better to have separate toggles for separate visualizations, like the gas consumption vs political party ones, rather than try to combine them all. While this makes it harder to compare different datasets, it makes it much easier to understand individual visualizations.

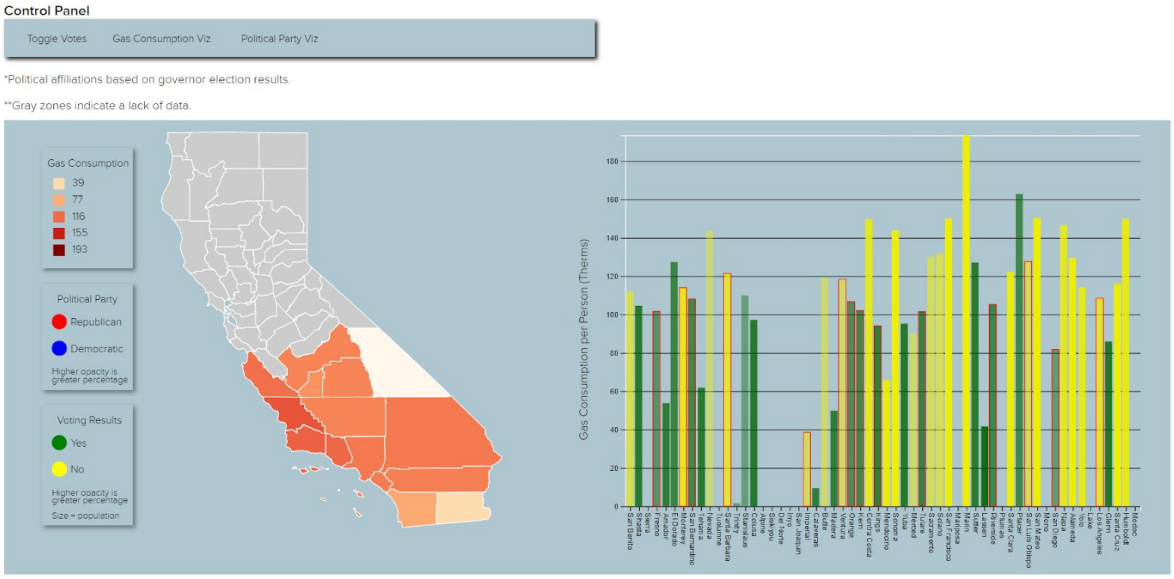
I would consider the project a success because, though it disproved our hypothesis of the vote being related to gas consumption, it showed a great correlation between political party and vote choice, which, though a worrying trend, is fairly understandable. We can conclude that perhaps politics in today's day and age is too divided among party and idealogical lines, rather than on practical and actual policy and how it affects us. This is valuable information, and directs us on how we should address the issue of idealogical politics taking over in America - at the very least, it tells us that there *is* a problem to be fixed.

5 - Image Gallery









6 - Reference

[1] Linking and Brushing InfoVis Wiki - https://infovis-wiki.net/wiki/Linking_and_Brushing

[2] D3 API Reference - https://github.com/d3/d3/blob/master/API.md

[3] Mike Bostock's D3 Examples - https://bl.ocks.org/mbostock

[4] Gas Consumption Data from California Energy Commission - http://www.ecdms.energy.ca.gov/gasbycounty.aspx

[5] Midterm Election Results from CA Secretary of State - https://www.sos.ca.gov/elections/prior-elections/statewide-election-results/