Members

Julian White - jwhit139
Andrew Zheng - azheng4

Modifications

For Phase A, we eliminated some of the questions that we felt that were not the most relevant to how we wanted to present the data. Our frontend shows bar graphs and pie charts showing the census values for each congressional district and state. Therefore we decided to eliminate questions like “What is the difference in mean income between the average of all Democratic districts and all Republican districts?” and “What is the average demographic makeup of districts with close elections; where vote count differs by less than 10%?”, as it did not align with our end goal. We thus stuck with stats like percentages and rates that could be easily compared in a bar graph and stats like state demographics that could be easily represented in a pie chart.

In Phase B of our project, we did not include a candidates table as we believed that just listing the party winner for each election would provide us with enough meaningful information. However, after cleaning up our design we reflected and decided that having a separate candidates table would be more beneficial. This would allow us to query information about the difference in votes for each party for elections. We could also see how different demographics affect polling results as certain candidates run in county, state-wise or even national level elections.

Process

States Census Data:
All state census data was taken from the US Census, specifically from the website: https://www.census.gov/mycd/. All of the csv files that come from this site are separated by state and include a plethora of information on each of the different congressional districts of every state. We used a python script in order to combine information from these csvs into different relations. The script only extracts the attributes that we found most interesting (all of which are modeled by our diagram in Phase B). There was a lot of data transformation that had to be done; since the csvs were per state, and each state had each of its districts as column attributes whereas districts are tuples in our schemas, we had to do more than simply delete columns. Instead of describing the whole transformation process, I have linked our GitHub repository that contains the script called *stateDataParser.py* that creates the state-related tables. The data used to populate those tables can be found in the director *states/* found on the same repository.

https://github.com/jawhite1612/database_project

Election Data:

Similar to the states data, election data was parsed and relation files were made using a python script. However, the data that we received from this source required much less processing. It basically required removing a few columns that would not add to our project. The election data was taken from an MIT database (https://electionlab.mit.edu/data). We decided to focus on presidential, senate, and US house races from 2016-2018. These years were chosen since our census data is up to date as of 2018, and we wanted to include a presidential election as well. We then combined all of the election data into two relations (Election and Candidate) using a python script. The python script is included in the GitHub as *electionDataParser.py*, and uses data from the *elections/* directory.

Successes

One particularly challenging aspect of writing our database was mapping the political party designation to each congressional district and state. This was accomplished by finding the max number of votes for an election and mapping that back to the winning candidates political party. The stored procedure would then find the ratio of democratic party wins over the total number of elections for the district. We were then able to color the bar graphs accordingly by this ratio, where bars colored more blue would have leaned more democratic (ratio closer to 1.00) and bars colored more red would have leaned more republican (ratio closer to 0.00) over the past three years. Bars that are yellow represent an even split between democratic and republican party victories (ratio equal to 0.50).

Another technical portion of our project that we were proud of was the interactive map that is on our frontend. For each different census statistic, the map changes as a heatmap, where a darker shade of green represents a higher value for the given value type and a lighter shade representing a lower value. Users are also able to click on individual state outlines on the map to obtain the relevant information for that state. We thought that the heatmap was an appealing and relevant feature to add for users.

Known Issues

One known issue with our project is that there lies potential discrepancies with our data. Since there are megabytes of data from many elections, it is hard to go through and test that there are no issues residing with the information from our data. For example, one wrong entry for a stat for an individual congressional district would skew the average for the state.

Extensions

Predictive Model:
If we had more time to work on the project, we would have incorporated a prediction component to our frontend. Our database has election data dating from 2016-2018 and we could have developed a classification model to predict an outcome (1 for Democratic or 0 for Republican) for an election based on various features that the user specifies. The features for the model would come from the state census data. Potential classification models we could have used in this extension include naïve bayes classifier, k-nearest neighbors and random forest classifier.

Feature Selection:
Another extension using machine learning would be to use dimensionality reduction techniques to identify potentially the most important features that contribute to an election leaning left or right. This could be done in many ways, including but not limited to greedy feature selection, principal component analysis and correlation analysis. We could then present the user on the web interface with a list of the top 5 most and least important demographics that contribute to an election's result on a state or nationwide scale.

Time Series Data:
One final extension would be to use additional historical census and election data to provide time-series graphics and representations. For our project, our scope was limited to current census data, but it would be interesting to see how political results change over time for individual congressional districts and states. This could also open up other interesting areas of analysis such as analyzing candidates that participated in multiple types of elections (house, senate, state, nation).