Travis Li

CSE 416 Gerrymandering Project – Preprocessing

The work I have done for preprocessing is hard to visualize with just three scripts. Thus, this document is here for me to tell what I have done in order to get my team the data in a workable condition.

The first step I did was to get the precinct level census data of our states, New York, Pennsylvania, and Maryland. There were many different sites, but the one I used was nhgis.org. This site provides all sorts of census and survey data from the ethnicity and race of people living in an entire state to the party people voted for at a precinct level. I chose data sets for the race and ethnicity of the whole population, and the voting age population, which is those over 18 years old, at a precinct level. This returned a very large csv file, which I broke into separate files for each state. The data I needed from each row were the GEOID, state abbreviation, precinct ID, precinct name, county name, and the population of each race. The one difficulty I encountered with this data is that if there were people that were multi-race, then they would not be counted within the total count for a single race. My first solution was to create a new column where they would all be counted as a population of two or more races. However, this became a problem as in the final design of the project, we would need to select specific minorities to analyze. If two or more races were its own section, then nobody in that section would be analyzed if someone were to choose analyze Asian voting age population percent across districts. To solve this problem, I created a method to split each multi-race population equally across the races it included. When there is only the remainder left, I randomly assigned one person from the remainder to a race that was in the label until there were no more people left over. This repeated for every entry in the csv. In this script, I also normalized the GEOIDs of each precinct. Across the three states we chose, they all had different ways of writing the GEOID. Thus, I had to normalize the GEOID in order to make them easier to distinguish. The format I ended with was stateID(2)countyID(3)precinctID(4). An example would be 240430008, where 24 represents the state ID of Maryland, 043 represents the county ID the precinct is in, and 0008 represents the precinct ID within the county. Using these, I created a new csv that would be used in the script to insert them into our MySQL database.

The next step was to get the shapefiles of the states we chose at a precinct level. Both Harvard and MGGG had repositories that contained these shapefiles, so I downloaded them and used QGIS to view them. Using QGIS, I joined the precinct level census data with the shapefiles in order to give the precincts some data to be associated with. A problem I had with the data is that I needed to know which congressional district each precinct was in, but neither the shapefiles nor census data had that information. Thus, I found the geojson files for the congressional districting plans enacted during 2010, the year our project is analyzing. To find which precincts are within a district, I created a map of centroids for each precinct in QGIS. I then intersected the centroids with the congressional district geojsons and found which district

each precinct belonged to. Taking this data, I joined it with the original shapefiles to create a new column that indicated the district ID.

A specification of the project was to find which precincts neighbor each other. This was done using the library libpysal and its spatial weight functions to take in a shapefile or a geojson file and find out which polygons are adjacent. I used the Rook function to make sure that precincts are only considered adjacent if they have a common edge. The Queen function would have included precincts touching by a corner, which would not fit within our definition for adjacency. This script outputs a csv of the GEOID and the neighboring GEOIDs, which I then joined into the original shapefiles.

Now that the shapefiles were done, I took them into mapshaper.org and simplified them, making sure that the polygons snapped to each other to prevent gaps, as well as keeping internal polygons to keep enclosed precincts. The average amount of simplification I used for each state was around 95%, which reduced file size to below 10mb for each state.

After downloading these simplified geojsons, I then sent them off to the team for them to display them in the GUI, as well to be worked with in the future, such as when we include the geojsons in the summary files created from each job.